Méthodes d'apprentissage statistique pour la détection de la signalisation routière à partir de véhicules traceurs

Yann Meneroux

UNIVERSITÉ PARIS-EST

IGN
INSTITUT NATIONAL DE L'INFORMATION GÉOGRAPHIQUE ET FORESTIÈRE

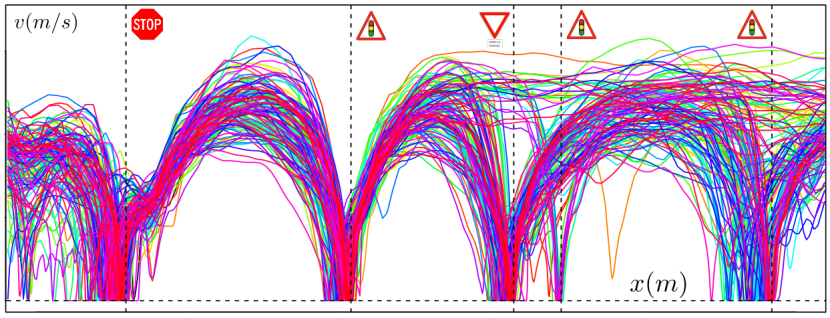
THÈSE

pour obtenir le grade de Docteur de l'Université Paris Est

préparée dans le cadre de l'École Doctorale Mathématiques et STIC

Spécialité : Signal, Image, Automatique

Yann MÉNEROUX



Méthodes d'apprentissage statistique pour la détection de la signalisation routière à partir de véhicules traceurs

Rapporteurs
- **Dr. Valérie Renaudin** — Directrice de recherche, IFSTTAR
- **Pr. Anne-Françoise Yao-Lafourcade** — Professeur, Université Clermont Auvergne

Examinateurs
- **Pr. Samia Bouchafa** — Professeur, Université d'Évry, Présidente du jury
- **Dr. Arnaud Le Guilcher** — Ingénieur des Ponts, des Eaux et des Forêts, IGN
- **Pr. Jean-Michel Loubes** — Professeur, Université de Toulouse
- **Dr. Olivier Orfila** — Chargé de recherche, IFSTTAR

Directeurs de thèse
- **Dr. Sébastien Mustière** — Ingénieur divisionnaire des TGCE (HDR), IGN
- **Dr. Guillaume Saint Pierre** — Chargé de recherche, HDR, Cerema

Résumé de thèse

Avec la démocratisation des appareils connectés équipés d'un récepteur GPS, de grandes quantités de trajectoires de véhicules deviennent disponibles, notamment via les flottes de véhicules professionnels et les applications mobiles collaboratives de navigation et d'assistance à la conduite. Récemment, les techniques dites de *map inference*, visant à dériver de l'information cartographique à partir de ces traces GPS, tendent à compléter, voire à remplacer les techniques traditionnelles. Initialement restreintes à la construction de la géométrie des routes, elles sont progressivement utilisées pour enrichir les réseaux existants, et en particulier pour construire une base de données numérique de la signalisation verticale. La connaissance fine et exhaustive de l'infrastructure routière est un prérequis indispensable dans de nombreux domaines : pour les gestionnaires de réseaux et les décideurs dans le cadre de travaux d'aménagement, pour les usagers avec le calcul précis des temps de parcours, mais aussi, plus récemment, dans le cadre du véhicule autonome.

Dans ce contexte, les méthodes d'apprentissage statistique apportent une perspective intéressante et garantissent l'adaptabilité de l'approche aux différents cas d'utilisation et à la grande variabilité des données rencontrées en pratique.

L'objectif de ce travail de thèse est d'étudier le potentiel de cette classe de méthodes, pour la détection automatique de la signalisation routière, en temps différé, à partir d'un ensemble de profils de vitesse GPS. Le premier cas d'application est celui de la détection des feux de circulation, étendu par la suite à d'autre types de signalisation comme les passages piétons.

En premier lieu, nous travaillons sur un jeu de données expérimental de haute qualité, à l'aide duquel nous étudions les performances de plusieurs classifieurs et nous comparons deux représentations mathématiques des données : une approche classique de reconnaissance d'image et une approche fonctionnelle consistant à agréger et à décomposer les signaux de profils de vitesses sur une base d'ondelettes de Haar. Les résultats obtenus montrent la pertinence de l'approche fonctionnelle, en particulier lorsqu'elle est combinée à l'algorithme des forêts aléatoires, en termes de fiabilité de détection et de temps de calcul. L'approche est alors appliquée sur d'autres types d'éléments de l'infrastructure.

Dans un second temps, nous tentons d'adapter la méthode proposée sur le cas de données observationnelles, *i.e.* acquises en environnement non-contrôlé, pour lesquelles nous cherchons également à estimer la position des feux de signalisation par régression statistique. Les résultats montrent la sensibilité de l'approche axé sur l'apprentissage face à des données fortement bruitées ainsi que la difficulté liée à la définition de l'emprise spatiale des instances individuelles sur un réseau routier complexe. Nous tentons de lever ce second verrou à l'aide d'approches globales fondées sur une segmentation d'image par réseau de neurones convolutionnel. Enfin, nous expérimentons une approche permettant d'exploiter

l'autocorrélation spatiale des variables cibles sur les instances individuelles à l'aide de la topologie du graphe routier et en modélisant la zone d'étude sous forme d'un champ de Markov conditionnel. Les résultats obtenus montrent une amélioration des performances de détection par rapport à l'apprentissage non-structuré.

Ces travaux de thèse ont également suscité le développement de méthodes originales de pré-traitement des trajectoires GPS (filtrage, interpolation, débiaisage et recalage sur un réseau routier de référence) ainsi que l'élaboration de critères objectifs d'évaluation de la qualité de ces pré-traitements.

Abstract

With the democratization of connected devices equipped with GPS receivers, large quantities of vehicle trajectories are becoming available, particularly via professional fleets of vehicles, mobile navigation and collaborative driving applications. Recently, map inference techniques, aiming at deriving mapping information from such GPS tracks, tend to complete or even replace traditional techniques. Initially restricted to the construction of road geometry, they are gradually being used to enrich existing networks, and in particular to build a digital database of road signs. Detailed and exhaustive knowledge of the infrastructure is a prerequisite in many fields (network management, driving time estimation, autonomous vehicle prior map...).

In this context, statistical learning methods (e.g. Bayesian methods, random forests, neural networks,...) provide an interesting perspective and guarantee the adaptability of the approach to different use cases and to the great variability of the data encountered in practice. In this thesis, we investigate the potential of this class of methods, for the automatic detection of road signs (mainly traffic signals), from a set of GPS speed profiles.

First, we use an experimental, high-quality dataset, on which we compare the performances of several classifiers on classical image recognition approach and on a functional approach, aggregating and decomposing speed profiles on a Haar wavelet basis whose coefficients are used as explanatory variables. The results obtained show the relevance of the functional approach, particularly when combined with the random forest algorithm, in terms of accuracy and computation time. The approach is then applied to other types of road signs.

In a second part, we try to adapt the proposed method to observational data for which we also try to estimate the position of the traffic signals by statistical regression. The results show the sensitivity of the learning approach to the data noise and the difficulty of defining the spatial extent of individual instances on a complex road network. We attempt to solve this second issue using global image approaches based on a segmentation by convolutional neural network, allowing us to avoid the explicit definition of individual instances. Finally, we experiment an approach leveraging spatial autocorrelation of individual instances using the graph topology with Conditional Random Field (CRF). The results obtained show an improvement compared to the performance obtained with non-structured learning.

These thesis works also led to the development of several original methods for pre-processing and quality assessment of GPS trajectories.

Remerciements

Je tiens à remercier en premier lieu Sébastien Mustière, qui, bien avant que l'aventure ne commence, m'a proposé un poste en sortie d'école, au service de la recherche de l'IGN. Il acceptera par la suite de diriger cette thèse, et m'aidera à trouver un sujet de recherche. Je le remercie pour sa bienveillance, ses conseils précieux, ses relectures minutieuses, et de m'avoir laissé une grande liberté dans mes travaux, tout en restant disponible, en particulier dans les moments les plus délicats.

Je souhaite exprimer également toute ma gratitude à Guillaume Saint Pierre, tout d'abord pour m'avoir fait confiance en acceptant de co-diriger cette thèse, pour son suivi attentif, y compris lorsque près de 10 000 km et 7 heures de décalage nous séparaient, pour sa gentillesse et son enthousiasme ainsi que pour les nombreuses discussions enrichissantes que nous avons eues. Je le remercie pour m'avoir initié à la rigueur de la méthodologie scientifique, et pour avoir réussi l'exploit (dont je ne parviens pas encore à comprendre le secret) d'avoir soigneusement balisé le chemin devant moi, tout en me laissant l'autonomie nécessaire pour explorer diverses pistes, y compris les plus incertaines.

Ce travail n'aurait bien évidemment pas pu voir le jour sans l'aide quotidienne d'Arnaud Le Guilcher, qui a accepté de m'encadrer, et grâce à qui j'ai appris beaucoup. Je garde un très bon souvenir des heures passées ensemble à réfléchir face au tableau blanc de l'un de nos deux bureaux. Outre les aspects scientifiques, je lui suis extrêmement reconnaissant pour sa bienveillance et ses encouragements, pour avoir été toujours très à l'écoute et pour avoir trouvé les mots justes pour me motiver dans les périodes les plus difficiles.

Je remercie également Olivier Orfila pour avoir accepté de co-encadrer ce travail, pour m'avoir aidé à mettre en place mes expérimentations à l'IFSTTAR, pour son regard critique bienveillant et pour ses précieux conseils méthodologiques.

De toutes les expériences vécues au cours d'une thèse, celle de la relation privilégiée avec directeurs et encadrants, est très certainement la plus enrichissante à mon sens.

Je remercie bien évidemment Anne-Françoise Yao-Lafourcade, Valérie Renaudin, Jean-Michel Loubes et Samia Bouchafa, pour m'avoir fait l'honneur de participer à mon jury de thèse, ainsi que pour les précieuses remarques et les discussions qui ont suivi la soutenance, et qui, j'en suis certain, éclaireront la suite de mes travaux de recherche.

Un peu plus en amont, je souhaite en particulier remercier François Bouillé, pour m'avoir aimablement conseillé dans ma quête de potentiels sujets de thèse, ainsi que Mickaël Brasebin, pour avoir guidé mes premiers pas dans le monde de la recherche.

Une partie conséquente de ce travail ayant été réalisée lors d'une mobilité à l'Université de Tokyo, je souhaite exprimer toute ma gratitude au Professeur Ryosuke Shibasaki, directeur du laboratoire CSIS, qui a permis cette expérience enrichissante. Je remercie aussi les

personnes qui m'ont encadré ou aidé sur place : Hiroshi Kanasugi, Karlvin Cuaresma, Teerayut Horanont, Ayumi Arai, Dinesh Manandhar, Wataru Ohira, Saurav Ranjit, Apichon Witayangkurn et Kentaro Itokawa. Je remercie tout particulièrement le Professeur Yukio Sadahiro qui a accepté de me recevoir pour discuter de mes travaux et de me prodiguer ses précieux conseils.

Je remercie Cindie Andrieu-Dupin et Baptiste Gregorutti, de la société Safety Line, dont les travaux respectifs ont été une grande source d'inspiration pour moi, pour leurs conseils avisés. Je suis également très reconnaissant envers Loïc Landrieu pour m'avoir guidé dans le domaine merveilleux de l'apprentissage structuré. Le dernier chapitre, en particulier, lui doit beaucoup. Je remercie aussi Xavier Collilieux et Ali Seba, pour avoir grâcieusement accepté de m'aider à traiter les données GNSS, ainsi que Lâmân Lelégard pour son expertise en traitement d'images. Je remercie également les membres de l'IFSTTAR de Satory pour leur accueil chaleureux, en particulier Benoît Lusetti, pour son aide indispensable à la mise en oeuvre des expérimentations, ainsi que Mohammad Ghasemi Hamed, pour m'avoir transmis une part de son savoir et de sa rigueur en validation statistique.

Une partie de la formation de thésard consistant à donner des cours, je souhaite témoigner ma gratitude envers Benoît Costes, à l'époque chef du département informatique de l'ENSG et Samia Bouchafa, directrice du Master 2 Systèmes Automatiques Mobiles de l'Université d'Évry, pour leur confiance et leur sympathie, ainsi qu'envers leurs étudiants respectifs, pour la bienveillance avec laquelle ils m'ont permis de faire mes premières armes dans le monde de l'enseignement.

Bien entendu, je n'oublie pas les membres du personnel administratif et technique de l'IGN et de l'École Doctorale, pour les nombreux dépannages, y compris (et bien souvent) dans l'urgence causée par ma forte capacité à procrastiner, et à qui je souhaite exprimer à la fois mes remerciements et mes excuses : Marie-Claude Foubert, Sylvie Cach, Alain Sombris, Cécile Aubert, Olivier Gueguen, David Correia et Léon Priam.

Je souhaite également remercier vivement tous mes collègues du laboratoire, pour la très bonne ambiance qui y règne. Je remercie en particulier, Marie-Dominique, avec qui j'ai pris grand plaisir à préparer des TP, à donner des cours, à encadrer des stages d'étudiants et à faire des premiers tests d'apprentissage. Je la remercie par ailleurs pour m'avoir aidé lors de mes débuts à l'IGN et m'avoir transmis son intérêt pour l'informatique ainsi qu'une part de ses connaissances dans le domaine, qui m'auront été grandement utiles, sinon indispensables, par la suite dans mes travaux.

Je remercie Ana-Maria, dont j'ai eu la chance de partager le bureau durant ces quatre années de thèse, pour ses conseils avisés et son regard critique extérieur, notamment pour la rédaction du manuscrit, mais aussi pour sa gentillesse, pour sa bonne humeur quotidienne ainsi que pour les nombreuses discussions enrichissantes.

Je remercie aussi Imran, Ibrahim, Paul et Laurent (avec qui la collaboration ne s'arrête d'ailleurs pas aux murs du labo, mais se prolonge parfois jusqu'au 5^{eme} étage du bâtiment B), pour leur amitié, pour les nombreuses discussions (souvent) enrichissantes en salle café, mais aussi pour leurs encouragements, leur aide et leurs conseils d'aînés.

Mes derniers remerciements vont bien entendu à Anthony, Thomas, Stéphane et André, pour m'avoir supporté et encouragé pendant ces années, ainsi qu'à ma famille, à Eric, Gaëlle et Nicolino pour leur soutien, à mes parents pour m'avoir transmis leur intérêt pour les sciences et à François, qui m'aura transmis son goût pour la cartographie.

Table des matières

Liste des abréviations

.

ABS : Anti-lock Braking System
ACC : Accuracy
ADAM : Adaptive Moment Estimation
ADF : Analyse de Données Fonctionnelles
AUC : Area Under Curve
CAN : Controller Area Network
CART : Classification And Regression Trees
CDMA : Code Division Multiple Access
CDR : Call Detail Record
CSIS : Center for Spatial Information Science
CNN : Convolutional Neural Network
DBSCAN : Density-Based Spatial Clustering of Applications with Noise
DSP : Densité Spectrale de Puissance
DSR : Délégation à la Sécurité Routière
EKF : Extended Kalman Filter
EM : Expectation-Maximization
EMQ : Erreur Moyenne Quadratique
ESP : Electronic stability control
F1M : Mesure F_1
FCD : Floating Car Data (ou données issues de véhicules traceurs)
FDA : Functional Data Analysis
FFT : Fast Fourier Transform
FMD : Floating Mobile Data
FPR : False Positive Rate
FWB : Fixed-Width Band
GNSS : Global Navigation Satellite System
GPRS : General Packet Radio Service
GPS : Global Positioning System
GSM : Global System for Mobile
HMM : Hidden Markov Model
I2V : Infrastructure To Vehicle
IFSTTAR : Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux
IGN : Institut National de l'Information Géographique et Forestière
IID : Indépendant et Identiquement Distribué
ITS : Intelligent Transportation Systems
KDE : Kernel Density Estimation

KL : Transformation de Karhunen-Loève
KNN : K Nearest Neighbors
LASSO : Least Absolute Shrinkage and Selection Operator
LDM : Local Dynamic Map
LIDAR : Light Detection and Ranging
LOESS : Locally Estimated Scatterplot Smoothing
NB : Naive Bayes
OFT : Offline computation Time
ONT : Online computation Time
OOB : Out Of Bag
OSM : Open Street Map
PPV : Positive Predicted Value ou Plus Proche Voisin
RELU : Rectifier Linear Unit
RF : Random Forest
RFID : Radio-Frequency IDentification
RMSE : Root Mean Square Error
ROC : Receiver Operating Caracteristics
SAE : Society of Automotive Engineers
SMOTE : Synthetic Minority Over-sampling Technique
SNB : Semi-Naive Bayes ou Random Ferns
SPC : Spécificité
STP : Stop confusion rate
STV : Sensibilité
SVM : Support Vector Machine
TF : Transformation de Fourier
TFD : Transformation de Fourier Discrète
TPR : True Positive Rate
UTM : Universal Transverse Mercator
V2V : Vehicle To Vehicle
VA : Véhicule Automatisé ou Variable Aléatoire
VAC : Véhicule Automatique et Connecté
VC : Véhicule Connecté
WGS84 : World Geodetic System 1984
XFCD : Extended Floating Car Data

Chapitre 1

Cadre général et enjeux de la thèse

Sommaire

Avec 725 milliards de kilomètres parcourus annuellement, la voiture reste le mode de déplacement intérieur privilégié des français, loin devant le transport ferré et l'autobus. Au total, près de 9 déplacements sur 10 se font sur la route, et le parc automobile français n'a jamais été aussi large, avec 39 millions de véhicules, tous types confondus, en 2017. Dans le même temps, on estime le coût total des embouteillages (en carburant et en temps perdu) à environ 700 euros par an et par foyer (Grenapin, 2013).

Pour s'adapter à ces évolutions, le réseau routier n'a pas cessé de s'étendre, pour une longueur totale dépassant le million de kilomètres, soit trois fois la distance Terre-Lune. Cependant, face à des enjeux environnementaux de plus en plus pressants, et avec un territoire d'emprise limitée, une utilisation plus optimale du réseau existant semble être la seule solution viable sur le long terme (Purson et al., 2015). Les nouvelles technologies peuvent permettre de répondre à ce besoin : on parle de transports intelligents (ou ITS pour *Intelligent Transportation Systems*), pour désigner l'application des technologies issues de l'information numérique et de la communication aux transports terrestres, avec l'objectif d'améliorer l'efficacité et la sécurité des déplacements, tout en réduisant leur impact environnemental, énergétique et sanitaire (Mallik, 2014).

Une connaissance fine et détaillée de l'infrastructure routière paraît être un prérequis indispensable à toute tentative d'optimisation de son usage. Cette connaissance se situe en

réalité à deux niveaux :

- Quel est l'état des lieux de l'existant ?
- Quel usage en est fait par les conducteurs ?

La première question, qui sera au centre de nos préoccupations dans cette thèse, consiste à disposer d'une cartographie exhaustive et à jour du réseau de routes, avec un inventaire des équipements associés. Un tel recensement n'existe pas à l'heure actuelle. Si les collectivités territoriales disposent en général bien d'une telle base cartographique, la décentralisation de certains états rend difficile la remontée, l'aggrégation et l'uniformisation des données. C'est le cas en particulier en France après plusieurs réformes constitutionnelles, notamment celles de 1982 et 2003 (Marcou, 2004).

La réponse à la seconde question n'a d'intérêt que si l'on est au préalable parvenu à apporter une réponse satisfaisante à la première. En effet, étant donnés par exemple la localisation des zones de congestion ainsi que l'estimation d'une matrice origine-destination sur un réseau routier, l'amélioration de la mobilité pourra se faire en considérant plusieurs hypothèses, *e.g.* : peut-on améliorer la fluidité du trafic en ajoutant des axes de communication ? Le paradoxe de Braess (Zverovich et Avineri, 2015) nous enseigne que ce n'est pas obligatoirement le cas et que l'ajout d'une route dans un réseau peut induire une baisse de l'efficacité globale. La formation de zones de congestion est-elle due à l'usure du revêtement d'un axe particulier que les vehicules tendraient à éviter ? S'agit-il d'un problème de calibration d'un feu tricolore ? Le gestionnaire de réseau devra retenir une ou plusieurs de ces hypothèses, nécessitant ainsi une connaissance aussi fine que possible de l'infrastructure routière et de son usage.

Ehrlich et al. (2014) indiquent trois axes majeurs motivant cette connaissance :

- L'**amélioration de la mobilité :** assurée à l'aide de capteurs temps réel, mesurant différentes grandeurs physiques propres aux flux de véhicules, telles que des vitesses, des temps de parcours et des débits, elle permet de réguler le trafic en conseillant les usagers.

- L'**amélioration de la sécurité routière**, avec un volet temps réel, dédié à la gestion réactive des incidents et des accidents, ainsi qu'un volet long-terme comprenant par exemple l'identification des comportements et des sections à risques.

- L'**amélioration de la gestion des réseaux**, permettant à chaque gestionnaires d'assurer la maintenance, de rénover et d'optimiser son infrastructure routière.

1.1 La collecte des données de trafic

1.1.1 Les capteurs fixes

Traditionnellement, la surveillance des réseaux routiers s'effectue à l'aide de différents capteurs fixes. On parle de collecte *intrusive* lorsque le capteur est installé de manière permanente sur la chaussée ou aux abords, avec une adaptation structurelle et non-réversible

de l'infrastructure routière sur la zone concernée. Par opposition, la collecte *non-intrusive* désigne les capteurs mobiles, destinés à être installé sur une période de temps allant de quelques heures à quelques mois.

Parmi les capteurs intrusifs, on pourra citer notamment :

- Les boucles inductives : enterrées à quelques centimètres de profondeur sous la chaussée, et composées d'une ou plusieurs bobines parcourues par un courant induit par la variation du champ magnétique ambiant lors du passage de la carcasse métallique d'un véhicule à la verticale du capteur.
- Les capteurs à effet piézo-électrique, composés d'un matériau en céramique possedant la propriété de générer une tension sous la contrainte mécanique exercée par le passage d'un véhicule.
- Les capteurs à tubes pneumatiques, composés de tuyaux souples partiellement enfouis transversalement dans la chaussée et terminés à leurs extrémités par des capteurs de pression.

L'observation d'un réseau complet nécessitant une densité importante de mesures, une solution s'appuyant exclusivement sur des capteurs intrusifs pourrait s'avérer rapidement très onéreuse. Il est donc en général plus rentable et plus efficace de compléter localement et périodiquement l'observation par des capteurs non-intrusifs, parmi lesquels on pourra citer en particulier :

- Les capteurs à effet Doppler : disposés en accotement de chaussée et permettant de mesurer la vitesse radiale d'un véhicule par comparaison des fréquences des signaux émis et reçus.
- Les capteurs infra-rouge : passifs (détectant le rayonnement thermique émis par le véhicule lors de son passage) ou actifs (détectant le signal réfléchi).
- Les capteurs acoustiques : passifs (microphones) ou actifs (émetteurs ultrasons).
- Les caméras à reconnaissance automatique de plaques d'immatriculation (LAPI).

Pour un panorama des systèmes de recueil de données de trafic routier, on pourra se référer à Leduc (2008).

Les données obtenues par ces deux types de capteurs sont par la suite filtrées et passées en entrée d'algorithmes relativement simples, permettant ainsi d'avoir accès à différentes grandeurs en un point donné : le débit, la longueur, la masse et la silhouette des véhicules, la vitesse, le taux d'occupation et la distance inter-véhiculaire.

En surveillance de réseau, la plupart des grandeurs intéressantes sont distribuées spatialement et temporellement. À titre d'exemple, le temps de parcours est défini pour un itinéraire reliant deux points particuliers, et peut potentiellement varier rapidement dans le temps. Aussi la solution consiste-t-elle en général à mesurer chaque grandeur en combinant les deux types de capteurs decrits ci-dessus. Le principe général est résumé sur la figure 1.1. On pourra trouver un exemple d'application avec l'estimation du trafic moyen journalier annuel dans (Islam, 2016).

Ces méthodes de collecte s'appuient en général sur des technologies matures, et permettent d'obtenir des estimateurs fiables des grandeurs mesurées (bien que présentant l'inconvénient d'être potentiellement sensibles aux conditions climatiques). En revanche, elles sont relativement coûteuses et lourdes à mettre en place et à maintenir. Par exemple, le coût d'installation d'un capteur à boucle inductive (qui figure parmi les types de capteurs les moins onéreux) s'élève à quelques milliers d'euros, ce à quoi il faut ajouter des frais de fonctionnement et de maintenance de l'ordre de quelques centaines d'euros sur la durée de service (Leduc, 2008). Récemment, grâce aux avancées technologiques réalisées dans les domaines de la localisation temps réel et de la communication, de nouvelles sources de données issues de capteurs embarqués deviennent disponibles, et tendent à remplacer, ou du moins à compléter, les méthodes de collecte traditionnelles.

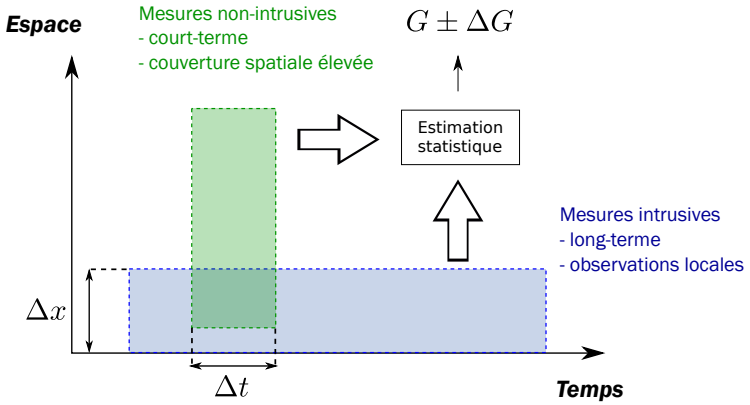


FIGURE 1.1 – Combinaison des données issues de capteurs intrusifs et non-intrusifs pour l'estimation statistique d'une grandeur G. Adapté de Leduc (2008).

1.1.2 Les sources de données flottantes

Avec l'intégration par les constructeurs automobiles d'instruments de navigation toujours plus sophistiqués dans les véhicules, ainsi qu'avec le prolifération de téléphones mobiles équipés d'un système de positionnement par satellites (GPS), de grandes quantités de trajectoires individuelles (généralement horodatées) deviennent disponibles. On parle de données issues de flottes de véhicules traceurs ou encore de *Floating Car Data* (FCD) en anglais. Les grandeurs qui auparavant ne pouvaient être estimées que par dérivation via l'observation d'autres grandeurs, deviennent à présent directement observables.

C'est le cas par exemple de la mesure des temps de parcours, que l'on pouvait estimer à partir d'un couple de caméras munies d'un système LAPI et positionnées aux extrémités de l'itinéraire. Cette méthode présente l'inconvénient d'être relativement peu précise [1], peu robustes (obstructions visuelles d'un véhicule par un autre, absence de garantie sur le fait que le véhicule ait parcouru l'itinéraire sans interruption) et inopérante dès lors que les conditions de visibilité sont mauvaises (brouillard, pluie, neige...). Or c'est précisement

1. Il faut noter ici que les systèmes de reconnaissance de plaques actuellement utilisés souffrent d'un taux d'erreur de l'ordre de 5 à 10%

dans de telles circonstances que le temps de parcours peut avoir le plus besoin d'être estimé de manière fiable. D'autre part, les caméras étant fixées, l'estimation des temps de parcours sur des itinéraires dont les extrémités ne coïncident pas exactement avec les positions des caméras nécessite l'emploi de méthodes d'interpolation et d'extrapolation avec leur lot d'incertitudes associées.

À l'inverse, avec les données FCD, le suivi individuel du véhicule est possible sur tout son itinéraire, et le temps de parcours peut ainsi être mesuré précisemment, moyennant l'utilisation de quelques algorithmes de filtrage des données, d'élimination des outliers et de détection des points d'arrêt (nous reviendrons sur ces points dans le chapitre suivant discutant le pré-traitement des données).

De manière similaire, le kilométrage annuel est aujourd'hui estimé en France en combinant les remontées des stations de carburant, des questionnaires adressés aux conducteurs et des mesures ponctuelles de trafic (Leduc, 2008). À présent, les FCD permettent d'obtenir une estimation fiable et directe du kilométrage annuel à partir d'une unique source de données.

Plus généralement, l'apparition des sources de données flottantes dans le paysage des méthodes de surveillance du réseau routier introduit un changement de paradigme. L'approche dite eulérienne repose sur l'estimation du flux en divers points de l'espace. Le référentiel de travail est fixé à l'infrastructure, et les grandeurs sont des aggrégés macroscopiques de véhicules. Cette approche convient parfaitement à une observation par capteurs fixes (boucles magnétiques, capteurs acoustiques, caméra...). Inversement, l'utilisation de capteurs embarqués permet l'adoption d'une description lagrangienne du problème : les paramètres observés sont attachés à une fraction des véhicules individuels (Gilliéron et Peyret, 2018; Després, 2010). Intuitivement, si la densité de capteurs fixes est infini et si tous les véhicules sont munis de capteurs embarqués, les deux formalisations sont équivalentes. En pratique, aucune de ces deux conditions n'est remplie, mais si la première semble physiquement impossible à satisfaire, la seconde paraît seulement limitée par des facteurs économiques. Une approche lagrangienne semble donc plus avantageuse pour une observation quasi-exhaustive du réseau routier. D'autre part, la description lagrangienne est souvent plus proche des grandeurs finales que l'on souhaite estimer en pratique (temps de parcours, matrice origine destination...).

Ce changement de paradigme n'est pas anodin dans la structure des données manquantes. On donne en figure 1.2 un extrait d'une étude menée par Bar-Gera (2007), visant à comparer deux méthodes d'estimation de la vitesse du trafic : la première en ayant recours aux données de boucles d'induction, la seconde à l'aide de données issues de téléphones mobiles. Dans le cas des capteurs fixes, les zones d'ombres se structurent en bandes horizontales, correspondant à des zones non-équipées. À l'inverse, avec les capteurs embarqués, l'observation du trafic est réduite aux heures creuses, et les données manquantes prennent la forme de bandes verticales. Cette différence met en évidence la nécessité de travailler en hybridation avec les réseaux de capteurs fixes et les sources de données flottantes, de manière analogue au procédé illustré sur la figure 1.1.

Sous l'acception la plus générale, les FCD désignent les données issues de capteurs embarqués, capables a minima d'enregistrer périodiquement la position du véhicule (les autres paramètres cinématiques tels que le cap, la vitesse ou l'accélération pouvant être

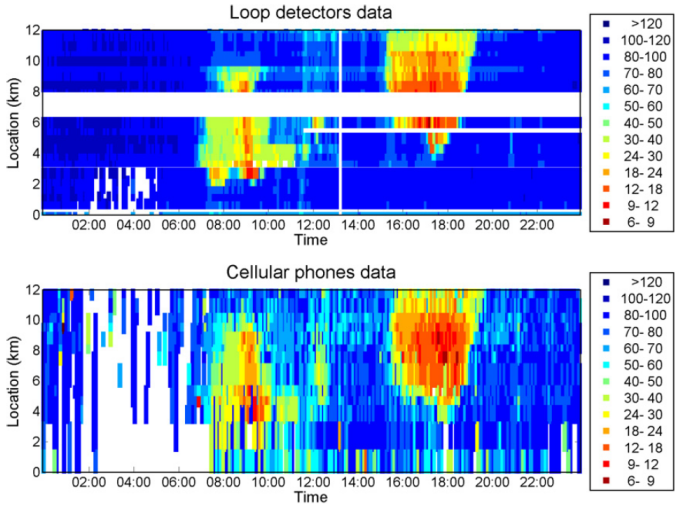


FIGURE 1.2 – Comparaison de données issues de capteurs à boucles inductifs et de téléphones mobiles Bar-Gera (2007).

calculés en post-traitement par dérivations numériques). Plus précisemment, on distingue les données collectées à bord sur une carte mémoire et récupérées lors du retour du véhicule traceur (c'est le mode le plus communément utilisé pour les expérimentations) des données transmises en temps réel (ou légèrement différé) sous forme de trames de navigation, généralement via les réseaux mobiles pré-existants (CDMA, GPRS, GSM...). Dans les deux cas, la localisation peut être déterminée de plusieurs manières :

- **Par téléphonie mobile :** avec ce mode de localisation, le capteur est dissocié du véhicule, et il devient possible d'étudier la mobilité des personnes empruntant une large gamme de moyens de transport. Le calcul de la position est effectué à l'aide du réseau de téléphonie. Dans sa version la plus simple, la position de l'utilisateur est déterminée par la cellule de couverture de l'antenne réceptrice. La précision du positionnement dépend alors de la taille de la cellule (de plusieurs km en campagne, à une centaine de mètres en zone urbaine). Il est possible d'affiner cette mesure en prenant en considération le temps de propagation des signaux, ainsi que le secteur de réception des antennes-relais (chaque antenne couvrant un angle de 120° en général). Une précision optimale de l'ordre d'une cinquantaine de mètres peut être atteinte en combinant les informations reçues par plusieurs antennes (Krishnamurthy et al., 2013). Les données issues de téléphones mobiles sont pafois désignés sous les noms de *Floating Cellular Data* (FCD), *Floating Mobile Data* (FMD) ou encore *Call Detail Record* (CDR). Notons que les appellations divergent suivant les sources : si les CDR renvoient exclusivement aux données localisées par le réseau de téléphonie mobile et disponibles uniquement lors des appels ou des échanges de messages texte, les FMD peuvent être localisées alternativement par un récepteur GNSS [2] embarqué dans le téléphone mobile ou par le réseau d'antennes-relais.

- **Par satellites :** le véhicule, ou le téléphone mobile du conducteur est équipé d'un récepteur GPS qui se géolocalise par trilatération à partir du signal reçu par un minimum de 4 satellites. La précision typique de ce type de positionnement est de 3

2. Dans ce manuscrit, nous utiliserons souvent le terme GPS par abus de langage (*cf* section 2.2.1).

m en conditions nominales (7 à 8 m en zone urbaine dense). Nous reviendrons plus en détail sur les problématiques soulevées par les erreurs GPS dans le chapitre 2. Un récepteur typique estime sa position une fois par seconde, mais le nombre de positions effectivement disponibles dépend des capacités du réseau de communication ou de la taille des cartes mémoires embarquées. On relève plusieurs types d'échantillonnage :

  - temporel : les positions GPS sont transmises (ou enregistrées en local) à intervalle régulier (typiquement entre 1 point par seconde et 1 point par minute). C'est le mode le plus couramment rencontré.
  - spatial : chaque nouvelle position est échantillonnée dès que l'utilisateur a parcouru un distance minimale depuis le dernière échantillon. Les positions intermédiaires n'étant pas conservées, la métrique utilisée est la distance à vol d'oiseau. Cette solution permet de limiter la bande passante (ou l'espace mémoire nécessaire au stockage des données) en éliminant naturellement les points peu informatifs.
  - hybride : ce type d'échantillonnage combine les deux modes précédents, *i.e.* une nouvelle position est échantillonnée dès qu'une certaine distance, ou un certain intervalle de temps, sépare l'utilisateur de la position précédente. Cette méthode présente l'intérêt de limiter la quantité d'information à transmettre ou sauvegarder, tout en assurant un résolution temporelle minimale.

- **Par tag électronique :** le véhicule, ou le téléphone mobile du conducteur est équipé d'une puce d'identification radio-fréquence (RFID) ou d'un émetteur Blutooth/Wifi, qui s'active au passage de bornes positionnées sur les accotements de la chaussée.

Notons que la classification des méthodes de collecte varie suivant les auteurs. À titre d'illustration, Lopes et al. (2010) proposent la classification suivante : *site data* (répertoriant l'ensemble des capteurs fixes, intrusifs et non-intrusifs), *Floating car data* et *wide-area* (pour les méthodes de télédétection et de photogrammétrie à base d'image et radar, satellite et aéroportés). D'autres auteurs (Antoniou et al., 2011) décomposent les méthodes en *point* (capteurs fixes), *point-to-point* (capteurs fixes en mode différentiel, tels que les caméras à reconnaissance de plaques) et *wide-area* (comportant la plupart des FCD et les méthodes de télédétection suscitées). Selon cette seconde classification, les méthodes FCD de localisation par tag électronique sont répertoriées dans la catégorie *point-to-point*.

D'après un rapport du Service d'études sur les transports, les routes et leurs aménagements (Sétra), la plupart des grandeurs d'intérêt pour la connaissance du trafic routier peuvent être obtenues par combinaisons de données flottantes FCD, de données de téléphonie mobile (FMD) et de capteurs fixes (Guichon et Piel, 2013).

On distingue quatre acteurs principaux en mesure de collecter et de distribuer des données flottantes :

- **Les constructeurs automobiles** intègrent directement un récepteur GPS dans certains modèles de véhicules et proposent par exemple à l'acheteur de souscrire à un programme de maintenance préventive, souvent gratuit, mais en échange d'une libre utilisation des informations collectées (dans le respect des lois relatives à la protection des données privées). Les données sont alors mises à profit par le constructeur, par exemple pour mettre à jour la cartographie routière ou pour mettre en place

un système de communications et d'alertes temps réel entre véhicules (V2V). C'est le cas du constructeur BMW par exemple, qui annonce une flotte de 10 millions de véhicules dans les années à venir (Doche, 2018). Les données peuvent également être revendues à des partenaires pour d'autres types d'utilisation. Sur le long terme, cette source de données semble être celle qui présente le plus de potentiel.

- **Les opérateurs de télécommunication** peuvent parfois mettre à disposition des partenaires des données flottantes pour des études ponctuelles. On pourra trouver des exemples d'utilisation de données CDR fournies par l'opérateur *Orange* (Bahoken et Olteanu-Raimond, 2013).

- Minoritaires dans le paysage des FCD et relativement localisées, les données issues de **flottes de véhicules professionels** peuvent permettre de caractériser précisément l'état du trafic sur une zone donnée. Cette catégorie regroupe notamment les bus, les taxis, les véhicules de service (ramassage des déchets, voirie, service postal, etc.) mais on peut également y inclure les flottes de véhicules spécialement équipés pour un besoin précis. C'est le cas par exemple d'une flotte de 200 véhicules, équipés par le Conseil Départemental de la Haute-Garonne, en partenariat avec Continental Automotive et Météo France, pour mettre en place un système local d'aide à la conduite (Dupont, 2018). C'est aussi le cas des données que nous utiliserons dans la première partie de cette thèse, qui ont été collectées dans le cadre d'un projet européen pour une étude expérimentale sur l'éco-conduite. Les constructeurs automobiles prennent en compte ce besoin en amont et proposent des flottes de véhicules pré-équipées avec parfois même l'accès aux services de traitement des données collectées. Par exemple, le groupe PSA (Peugeot Société Anonyme) en a livré environ 60 000 en France. La nature spécifique de ces flottes de véhicules limite le potentiel de ce type de source, en particulier en termes de représentativité.

- Enfin, **les applications de navigation et de conduite collaborative** constituent certainement à l'heure actuelle la source de données la plus vaste. Des applications mobiles, comme *Waze* par exemple, proposent aux conducteurs de collecter ses données de navigation, en échange d'une aide à la conduite, de recommendations d'itinéraires de substitution et d'alertes de dangers. On pourra citer également *Coyote* et *Autoroutes Trafic* (France), *TomTom* (Pays-Bas), *Inrix* (États-Unis) et *Navitime* (Japon).

Récemment, de nombreuses études ont contribué à montrer que la pluplart des grandeurs physiques caractérisant l'état du trafic routier pouvaient être estimées via les données FCD avec un degré de précision équivalent, sinon meilleur, qu'avec les méthodes traditionnelles. Par exemple, on pourra citer les travaux de Sohr et al. (2010) pour la comparaison des FCD et des boucles d'induction pour la quantification du trafic, ou encore Haghani et al. (2010), qui a comparé les estimations de vitesses moyennes par tronçon obtenues par les sources des données flottantes avec les estimations calculées par les balises Bluetooth. Ces travaux ont servi de point d'appui pour des publications ultérieures, visant à valider des modèles théoriques d'écoulement du trafic, à partir des données flottantes, seules ou en hybridation avec les capteurs fixes (Qiang et al., 2012; Anuar et al., 2015).

Outre le schéma d'échantillonnage et la précision du système de localisation utilisé, la proportion de véhicules équipés d'un système capable de transmettre des données sur une

zone donnée est un paramètre critique dans l'utilisation des données FCD, appelé *taux de pénétration* ou encore *taux d'équipement*. En pratique, ce taux est relativement difficile à estimer, d'autant plus que les gestionnaires de flottes ont une tendance à la rétention d'information, pour des raisons commerciales évidentes, mais aussi pour assurer la protection des données privées. Par exemple, en recevant un jeu de données, il peut arriver de ne connaître que le nombre de trajectoires, les identifiants des conducteurs étant aléatoirement modifiés toutes les 24 heures, rendant ainsi difficile l'estimation, même grossière, du taux de pénétration. D'après un document interne résultant d'études menées à l'IGN, le taux de pénétration de la flotte de véhicules du groupe PSA semble être compris entre 1‰ et 1%. Ce chiffre semble beaucoup plus conséquent pour les applications de conduite collaborative : Waze, par exemple, annonce avoir franchit la barre des 10 millions d'utilisateurs en France. Sur une base de 32 millions de véhicules particuliers, le taux de pénétration théorique maximal s'élève à près de 30%. Pour obtenir le taux effectif en pratique, il faudrait connaître la proportion d'inscrits utilisant l'application en moyenne à chaque instant. Entre ces deux extrêmes, on peut supposer que certaines flottes de véhicules permettent d'obtenir un taux de pénétration significatif sur des zones restreintes. C'est le cas en particulier des taxis, pour lesquels un rapide calcul à l'aide des statistiques publiées par l'INSEE (chiffres de 2016) montrent que ce taux pourrait atteindre les 2 à 3%, moyennant la mise en commun des données de tous les opérateurs. La flotte des bus de la RATP pourrait quant à elle atteindre un taux de 1% sur Paris dans les prochaines années. En l'abscence d'information plus précise, certains travaux sur la qualité des données FCD, tels que (Sohr et al., 2010), font arbitrairement varier ce taux sur une large gamme comprise entre 1‰ et 50%.

En pratique, pour de nombreuses études où les quantités recherchées sont des grandeurs globales (kilométrage parcouru, temps passé sur la route...), la connaissance du taux de pénétration ne revêt que peu d'intérêt. En effet, pour un nombre total de véhicules en circulation suffisamment élevé (typiquement quelques milliers), les variances associées aux quantités numériques estimées sont inversement proportionnelles au nombre de trajectoires observées, et ne dépendent donc pas du dénominateur du taux de pénétration (sous l'hypothèse de représentativité statistique de l'échantillon observé, *cf* 1.1.3). La connaissance du nombre de trajectoires observées dans le jeu de données est généralement suffisante. En revanche, pour une observation exhaustive du trafic, un bon taux de pénétration est la garantie de couvrir de manière satisfaisante le territoire (et la plage de temps d'intérêt). Cependant, le taux de pénétration peut être trompeur : par exemple un taux de 1% constitué de véhicules appartenant à une flotte de taxis en service 7 heures par jour et 5 jour par semaine, n'aura aucune commune mesure avec un taux similaire obtenu par des véhicules utilisés uniquement pour le trajet domicile-travail (Sohr et al., 2010). Dans cette thèse, lorsqu'il s'agira d'évaluer l'impact du taux de pénétration des FCD sur la détection de la signalisation, nous lui subtituerons le nombre de trajectoires disponibles sur la portion de route concernée.

Par ailleurs, pour s'adapter à l'évolution des normes en matière de sécurité et pour assurer le confort de l'automobiliste, les constructeurs tendent à automatiser le maximum de processus, en particulier ceux qui consistent à effectuer des tâches répétitives, fastidieuses ou qui exigent un temps de réaction minimal. Un véhicule moderne comporte une centaine d'unités de commande électronique, gérant un grand nombre de sous-systèmes tels que le niveau de carburant, la transmission, les airbags, le freinage ABS ou encore le correcteur électronique de trajectoire ESP (Saarikivi, 2011). Des informations numériques

sont échangées en grandes quantités via le bus de données CAN (Controller Area Network), permettant ainsi de gérer l'arrivée aléatoire des mesures relevées par les différents capteurs selon un protocole de priorité. On estime à environ 4000 le nombre de signaux différents échangés sur le bus, ce qui correspondrait à plusieurs gigaoctets de données par heure si toutes ces informations venaient à être sauvegardées en mémoire ou transmises à une station de base (Massaro et al., 2017). On appelle xFCD (pour *extended floating car data*) l'ensemble des données collectées au niveau du bus, qui peuvent être composées :

- de paramètres cinématiques directement observés tels que la vitesse par Doppler GPS (i.e. la vitesse quasi-exacte) ou odomètrique (qui correspond à la vitesse lue par le conducteur sur le tableau de bord), les accélérations longitudinale et latérale, le jerk, le cap, l'inclinaison... (mesures proprioceptives)

- de paramètres environnementaux (exteroceptifs) : luminosité, température, pression, humidité...

- de paramètres de commandes : essuie-glace, embrayage, levier de vitesse, freins, rayon de braquage du volant...

- de paramètres internes : ABS, ESP, système de refroidissement, niveau de carburant, régime moteur...

Ces données sont par exemple celles utilisées dans la première partie de cette thèse, comme illustré sur la figure 1.3.

Les données xFCD ont un champ étendu d'applications potentielles (Leduc, 2008), par exemple en prévisions météorologiques, avec l'utilisation des capteurs climatiques du véhicule ou des informations corrélées comme l'activation des feux de brouillard, des essuie-glace ou du système d'air conditionné (Bartos et al., 2018). Des cas d'utilisation peuvent également être trouvés en matière de risques (l'activation systématique du freinage ou de l'ESP sur une portion de route peut indiquer des zones de danger potentielles), de maintenance (les capteurs accéléromètres et jerks peuvent permettre d'inférer l'état d'usure du revêtement de la chaussée) ou encore d'éco-conduite (avec l'analyse des consommations de carburant en fonction des différentes pratiques des conducteurs). On pourra trouver une large gamme d'applications avec les niveaux de maturité associés dans Ehrlich et al. (2014).

Pour plus d'informations sur le contenu et l'utilisation des données xFCD, nous renvoyons le lecteur au travail de Huber et al. (1999).

1.1.3 Limites des données flottantes

À l'heure actuelle, on relève encore quelques points bloquants à déverouiller pour une utilisation optimale des données de véhicules traceurs, en particulier sur les plans technologique et légal. En premier lieu, on doit tenir compte d'un biais de sélection dans les données produites par des véhicules traceurs. De manière évidente, les données issues de flottes de véhicules professionels, telles que les flottes de bus ou de taxis, ne sont absolument pas

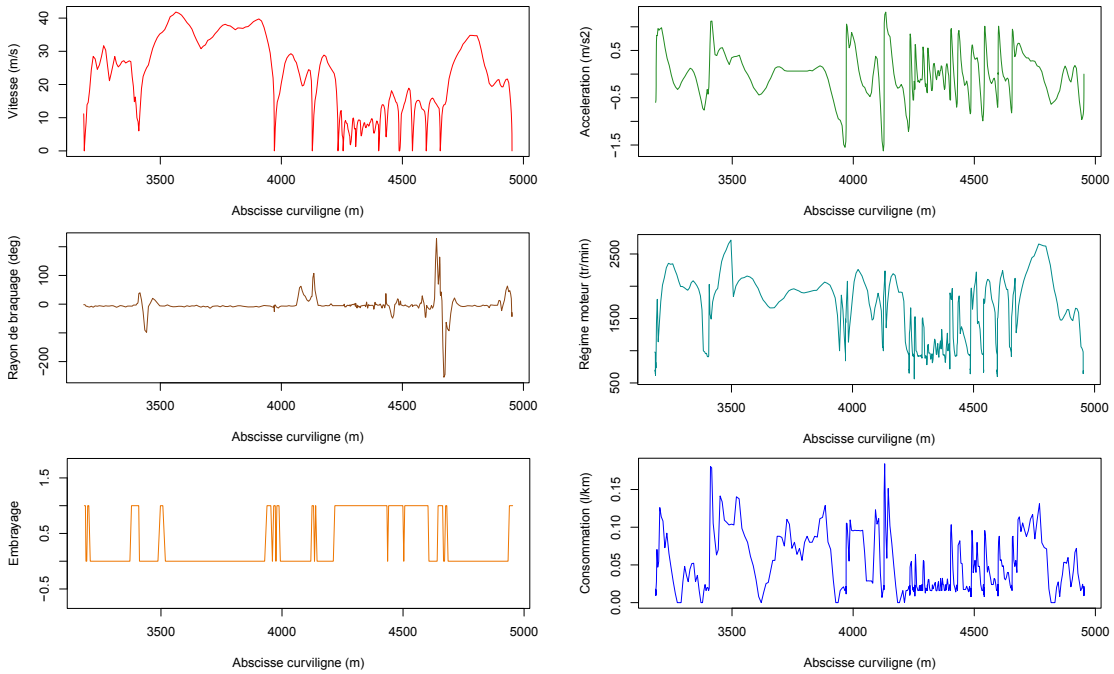


FIGURE 1.3 – Données xFCD échantillonées à 1 Hz collectées pour une expérimentation d'éco-conduite sur une portion de 1800 m dans la commune de Versailles (cf chap. 3). De gauche à droite et de haut en bas : la vitesse Doppler, l'accélération longitudinale lissée, le rayon de braquage, le régime moteur, l'embrayage (1 en position embrayée) et la consommation instantanée de carburant.

représentatives du comportement et des habitudes de l'ensemble des usagers de la route (Sohr et al., 2010), ce qui limite considérablement le potentiel de ces données, en particulier dans le cadre d'études théoriques. Mais c'est aussi le cas dans une moindre mesure pour tous les autres types de sources de données. Les propriétaires de véhicules particuliers équipés d'un système de transmission de FCD ne peuvent être considérés aujourd'hui comme statistiquement représentatifs de l'ensemble des conducteurs. L'utilisation de ces données nécessitera donc dans certains cas de débiaiser les estimateurs calculés, en particulier en tenant compte de facteurs liés à l'origine sociale de l'échantillon de population observé. Le même problème se pose avec l'utilisation des données mobiles FMD (Arai et al., 2015), bien que l'on puisse considérer que l'utilisation du téléphone cellulaire est beaucoup plus répandue et homogène dans la population des usagers de la route. On rencontre le même type de difficulté avec les données fournies par les applications de navigation et de conduite collaborative : s'il est cette fois possible de supposer l'échantillon des utilisateurs comme grossièrement représentatif de la population de conducteurs, on doit objecter que le recours à l'application aura tendance a être plus massif sur certains types de trajets. Il existe à l'heure actuelle peu d'études (à notre connaissance) quantifiant l'impact de ce biais d'utilisation sur les estimations calculées.

La seconde difficulté est plutôt de nature stratégique ou politique. Contrairement au cas des capteurs fixes, la mutualisation des investissements et des coûts d'utilisation des données FCD exige une synergie parfaite entre les différents partenaires et prestataires : constructeurs automobiles, gestionnaires de réseau, opérateurs de télécommunnication et

décideurs publics (Leduc, 2008).

En troisième lieu surviennent les difficultés liées aux aspects juridiques de la protection des données privées. Classiquement, les fournisseurs de données FCD attribuent un identifiant unique à chaque véhicule, aléatoirement et périodiquement modifié (par exemple toutes les 24 heures). Il est parfois également d'usage de bruiter ou d'agréger à un niveau de résolution supérieur les points sensibles de la trajectoire, en particulier les extrémités, qui sont susceptibles de fournir des informations sur les lieux de résidence et de travail de l'individu dont la trace est analysée (Giannotti et Pedreschi, 2008). La question consistant à savoir si ces précautions sont suffisantes est toujours une question de recherche active. Dans le domaine de l'anonymisation des bases de données, on parle de *quasi-identifiant* pour désigner un sous-ensemble d'attributs permettant d'identifier (avec une probabilité plus ou moins élevée) les individus de la base (Dalenius, 1986; Sweeney, 2000). Dans une base de trajectoires FCD long-terme suffisament résolu spatialement, toute position peut constituer ou faire partie d'un quasi-identifiant, permettant ainsi, par croisement avec des bases de données annexes, de retrouver les identités des conducteurs (Bettini et al., 2005). On parle de *de-anonymization attack*.

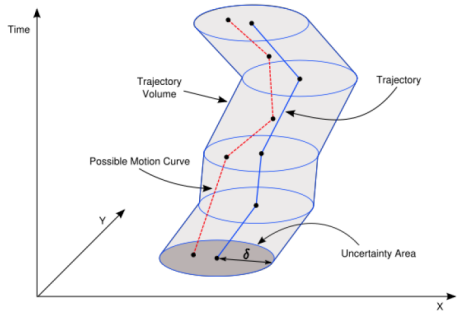


FIGURE 1.4 – Géo-anonymisation d'une trajectoire (Giannotti et Pedreschi, 2008).

En environnement contraint (comme par exemple sur le réseau routier ou ferré), certains travaux ont montré que les données de positions pouvaient même être remplacées par des données de type *dead-reckoning*. À titre d'illustration, Hua et al. (2017) ont démontré que les données accélérométriques pouvaient permettre d'inférer la position d'un usager dans le métro après seulement quelques minutes de trajet (4 à 6 stations) avec un taux d'erreur marginal. En pratique, l'anonymisation parfaite d'un jeu de données de localisation est utopique, et on se satisfait souvent d'une contrainte plus lâche : la k-anonymisation. Une base de données FCD est dite k-anonymisée si chacune de ses trajectoires est indiscernable d'au moins $k-1$ autres trajectoires. Pour obtenir des garanties suffisantes, on souhaite une valeur de k assez élevée, en considérant bien entendu que l'utilité d'une base de données diminue avec le degré d'anonymisation, comme illustré sur le graphe schématique de la figure 1.5, inspiré des travaux de Raafat et al. (2016).

Enfin, la dernière difficulté est d'ordre technologique : comme souligné précedemment, le nombre de véhicules capables de transmettre des FCD ne cesse d'augmenter, tout comme les capacités de transmission et d'enregistrement des données collectées (en particulier avec les données xFCD). À terme, la masse des informations à traiter risque de mettre à mal la

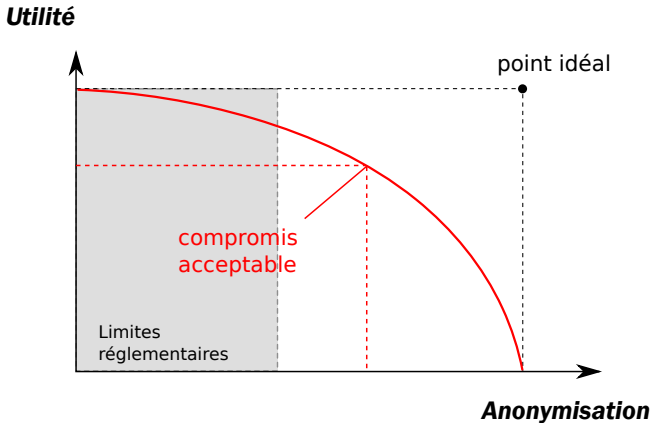


FIGURE 1.5 – Utilité d'une base de données en fonction de son degré d'anonymisation et points de fonctionnement idéal et acceptable en pratique.

plupart des serveurs de données et de calculs. Les technologies dites *Big Data* (Witayangkurn et al., 2013; Ranjit et al., 2014) peuvent permettre de lever ce verrou.

1.2 La cartographie de l'infrastructure routière

Dans la plupart des pays, le relevé cartographique de l'infrastructure routière incombe à un institut national dédié. En France, depuis 1940, c'est l'Institut National de l'Information Géographique et Forestière (IGN) qui est en charge collecter, de traiter et d'archiver les données topographiques. Les informations collectées sont organisées en couches thématiques, chaque couche contenant un certain type d'information : bâti, courbes de niveau, modèle numérique de terrain, hydrographie, végétation... Outre son intérêt pragmatique pour la navigation, le thème du réseau routier constitue en plus un canevas de référence visuel pour une lecture aisée des cartes. De plus, nombre d'informations présentes dans d'autres thèmes peuvent être référencées par rapport au réseau routier, comme par exemple les adresses des bâtiments, les parcelles de terrain, les repères de nivellement... En conséquence, les instituts cartographiques le considèrent comme un thème central, et investissent massivement dans l'acquisition, la mise à jour et les contrôles de qualité des routes (Bonin, 2002; Liu et al., 2012; Zhang et Couloigner, 2006).

On dénombre cinq composantes principales dans la qualité d'un réseau routier : la *géométrie* désigne la précision de positionnement des axes routiers dans le référentiel cartographique. La *topologie* dénote la bonne connectivité des nœuds et des arcs du graphe sous-jacent (on parle de navigabilité). L'exactitude *sémantique* désigne la précision avec laquelle la nature des éléments constituants le réseau a été correctement identifiée (route départementale, nationale, autoroute...). On peut l'associer à la précision *attributaire* qui évalue l'exactitude des caractéristiques de ces éléments (nombre de voies, séparateur central...). Enfin, la *toponymie* désigne ici le référencement des noms des impasses, des rues, des routes départementales, nationales... Notons que la distinction entre les composantes géométrique et attributaire n'est pas toujours nette : lorsqu'une grandeur géométrique est

commune à l'ensemble des objets du thème (*e.g.* largeur de la route, largeur des voies...) elle pourra être renseignée sous forme de variable catégorielle dans un attribut.

1.2.1 Les méthodes traditionnelles

La composante de base dans le processus de relevé du réseau routier repose sur les technique de stéréorestitution. Une campagne de prises de vues aériennes consiste à faire voler un avion équipé d'une caméra, selon un plan de vol similaire à celui illustré à droite de la figure 1.6. L'appareil prend une série de clichés à intervalle plus ou moins régulier, de sorte que la distance séparant deux sommets de prises de vues consécutives soit inférieure à la moitié de l'emprise au sol d'un cliché. Ce procédé permet de garantir que le recouvrement entre deux images voisines est supérieur à 50%, assurant ainsi que chaque point de la zone couverte est visible sur au moins deux clichés différents. Les techniques stéréoscopiques peuvent alors être employées pour reconstituer en trois dimensions les positions des éléments visibles sur les photographies.

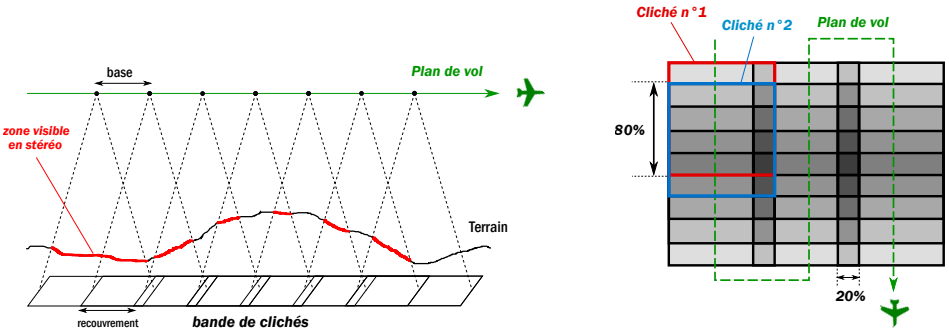


FIGURE 1.6 – Principe des prises de vues stéréoscopiques (illustration inspirée de Boureau (2008)) et plan de vol typique d'une campagne de prises de vues aériennes.

En pratique, les paramètres de la mission sont calculés pour assurer un recouvrement supérieur à 50% (typiquement 60 à 80%) suivant les zones. Pour assurer la continuité entre les bandes, on introduit également un recouvrement de sécurité (typiquement 20%) comme illustré sur la figure 1.6. À longeur focale et taille du capteur matriciel fixées, la résolution métrique de la prise de vue dépend de la hauteur de vol de l'avion. Plus la hauteur de vol est basse, plus la résolution est élevée, mais plus le champ couvert par un cliché est réduit et donc plus le nombre de clichés et le nombre de bandes augmentent. Si l'augmentation du nombre de prises de vue se traduit uniquement par un temps de traitement prolongé au sol, l'augmentation du nombre de bandes sur la zone allonge le temps passé en vol et donc le coût de la mission [3]. Les résolutions typiques des images aériennes varient de quelques cm à quelques dizaines de cm.

La restitution cartographique se fait à l'issue de la mission, et combine systématiquement trois processus. Tout d'abord, le *référencement* consiste à orienter et à positionner de manière relative les sommets et les axes des prises de vue au sein de chaque couple à l'aide des paramètres de vol de l'avion et de points appariés entre clichés, appelés points

3. Pour des clichés classiques de 23×23 cm, 540 photographies sont nécessaires pour couvrir un département français moyen à l'échelle 1/25 000^{e}. Ce nombre s'élève à 3600 au 1/10 000^{e} (Boureau, 2008).

de liaison, ou points homologues, puis à géoréférencer le bloc de clichés dans un système géodésique, en utilisant des points d'appui, dont les coordonnées ont été calculées sur le terrain au préalable lors d'une étape dite de stéréopréparation. La stéréorestitution se fait alors sur des clichés numérisés, à l'aide de micro-ordinateurs classiques équipés d'un écran 3D. La restitution des éléments sur la carte se fait directement à l'aide du pointeur de la souris. Alternativement, il est possible de générer en amont une image dite *orthorectifiée* (*i.e.* géoréférencée et parfaitement superposable à la carte), à partir de laquelle un opérateur va saisir les différents éléments du terrain, auquel cas l'étape de restitution et dissociée de l'étape photogrammétrique à proprement parler, et la saisie ne nécessite plus que des outils informatiques de base.

Cette méthode d'acquisition de données topographiques possède l'avantage de relever plusieurs couches thématiques sur la même zone : le routier, l'hydrographie, le bati, la végétation et de manière générale l'ensemble des éléments du terrain visibles depuis les photographies aériennes. D'autre part, les calculs menés lors des processus de géoréférencement des images permettent la livraison d'un ensemble de sous-produits tels que les modèles numériques de terrain (MNT), les orthophotographies aériennes (directement affichables en fond de carte) ou encore le réseau de base compensé des points d'appui.

En contre-partie, les étapes de stéréo-préparation (avec la collecte des points d'appui [4] référencés au sol), de planification de la mission aérienne et de stéréorestitution du réseau routier (quasiment exclusivement de manière manuelle par des opérateurs spécifiquement formés à cette tâche) sont très coûteuses et chronophages. D'autre part, l'intervalle de temps entre deux campagnes de prises de vues aériennes sur une même zone s'élève typiquement à quelques années (3 à 5 ans par exemple à l'IGN), rendant cette méthode de collecte plus propice à la construction du réseau routier *ex-nihilo*, mais relativement peu adaptées au suivi de l'évolution du territoire et aux mises-à-jour en temps réel, en particulier dans les zones urbaines des pays en voie de développement, où le rythme des changements peut être extrêmement rapide. Enfin, il est important de noter que s'intercale généralement entre deux phases de stéréorestitution, une phase de collecte sur le terrain des éléments invisibles sur l'imagerie aérienne, rendant ainsi l'ensemble du processus d'autant plus long et coûteux. Cette phase est complémentée par un recueil des remontées de mise à jour, issues principalement des actes administratifs des communes, ainsi que des signalements fournis par les services déconcentrés de l'État, par les conseils généraux, ou encore par les sociétés d'autoroutes.

Plus récemment, certaines méthodes ont vu le jour pour réaliser une saise automatique ou semi-automatique du linéaire routier à partir d'image aériennes et satellitaires, multi-spectrales et à haute résolution (Zhao et al., 2002; Bentabet et al., 2003; Zhang et Couloigner, 2006). La segmentation de l'image est en général effectuée à l'aide de méthodes d'apprentissage automatique orientées images, telles que les réseaux de neurones artificiels, en se fondant sur un ensemble de descripteurs de natures spectrale, géométrique, texturale et contextuelle. On distingue en général les méthodes orientées pixel (ou *pixelwise*), qui s'attachent à classifier chaque pixel de l'image comme faisant partie ou non d'un élément de route, des méthodes orientées objet, qui adoptent une description de plus haut niveau de la scène et permettent une segmentation plus fine et robuste de l'image (Maboudi et al., 2016). On pourra trouver un état de l'art assez exhaustif des méthodes utilisées en segmentation d'images aériennes pour l'extraction du réseau routier dans (Sirefelt, 2015). Si ces

4. Les points de liaison sont calculés automatiquement à partir des clichés.

nouvelles méthodes permettent d'économiser le temps des processus de stéréorestitution manuelle, elles ne sont pas encore pleinement opérationnelles, et peuvent présenter des faiblesses significatives, notamment dans les zones d'ombres, ou dans les zones urbaines denses où la présence d'occlusions peut aisément mettre à mal les algorithmes. De plus, cette alternative possède toujours l'inconvénient d'être limitée en actualité par la fréquence des prises de vues aériennes qui peut être relativement faible (en particulier pour le cas des images haute-résolution) ainsi que par les perturbations climatiques et d'ensoleillement (couverture généralement plus sporadique, voire complètement absente en hiver).

Enfin, la dernière méthode active de relevé du réseau routier repose sur les contributions citoyennes volontaires, avec notamment le projet *OpenStreetMap*, qui vise à fournir aux citoyens un procédé simple et gratuit pour collecter et échanger de l'information géographique. Cette méthode présente l'inconvénient de proposer une couverture et un niveau de précision relativement disparates sur le territoire (Girres et Touya, 2010; Liu et al., 2012), sans compter la présence potentielle de contributions fallacieuses (Truong et al., 2017) qui pourraient s'avérer problématiques pour les applications nécessitant un haut niveau de fidélité dans les données, telles que le véhicule autonome en particulier (*cf* 1.2.3).

Jusqu'au début des années 2000, l'IGN maintenait une base spécifique de la cartographie routière (Eltchaninoff, 1996), à destination des applications de navigation, par exemple pour la cartographie numérique embarquée des constructeurs automobiles BMW et Renault, via les sociétés américaines *Philips* et *Navigation Technologies*. Les données de la base étaient mises à jour en continu par une centaine de collecteurs de terrain. Le produit a par la suite été abandonné, et la plupart des informations relatives au réseau routier ont été intégrées dans la base générique BD CARTO®, qui contient actuellement un peu plus d'un million de kilomètres de routes. Deux bases annexes interopérables en sont dérivées, pour des utilisations respectivement à l'échelle régionale et nationale. Malgré l'abandon de la base spécifique, l'arrivée des problématiques liées au véhicule autonome, semble impulser un regain d'intérêt à l'IGN pour une connaissance plus précise du réseau routier, et plusieurs actions de développement visant à compléter les bases ont été recemment lancées, notamment en partenariat avec la Délégation à la Sécurité Routière (DSR) : points routier, points kilométriques, vitesses pratiquées et limitations de vitesse, sens de circulation, non-communication, élévation par rapport au sol, nombres de voies ou encore viabilité des tronçons.

1.2.2 Le map inference

L'arrivée des données FCD exposées au 1.1.2 peut constituer un atout majeur pour la complétion de la cartographie routière. Biagioni et Eriksson (2012a) définissent le *map inference* par opposition aux méthodes traditionnelles exposées ci-avant, comme l'ensemble des techniques visant à dériver de l'information cartographique à partir de trajectoires enregistrées par des véhicules traceurs. Parmi leurs nombreux avantages, les FCD sont collectées en continu et généralement accessibles en temps réel ou légèrement différé, ce qui permet une réactivité sans précédent dans le processus de mise à jour. À l'inverse, avec les méthodes classiques, l'espacement des campagnes aériennes, ainsi que les délais additionnels nécessaires au traitement des prises de vue et à la restitution cartographique, peuvent s'avérer être des facteurs extrêmement limitant dans certains cas d'utilisation où l'actualité des données est un paramètre critique, tels qu'en gestion des risques, notamment pour la planification des opérations d'évacuation, de secours et de résilience.

Avec les données FCD et les algorithmes de map inference, une modification du réseau routier est potentiellement détectable aussitôt qu'un nombre suffisant de traces a été collecté sur la zone pour confirmer la robustesse statistique du changement observé. De plus, les données sont collectées en tant que sous-produit d'une activité annexe (trajet domicile-travail, collecte des déchets, livraisons...), rendant cette approche très peu coûteuse comparativement aux campagnes de prise de vues aériennes et aux relevés topographiques sur le terrain, avec une prise en charge des coûts mutualisable entre les différents acteurs.

Par ailleurs, l'imagerie aérienne n'est pas toujours accessible dans les pays en voie de développement, ou alors à un coût prohibitif, tandis que l'utilisation du téléphone mobile semble assez étendue de par le monde. La collecte des données FCD peut constituer une solution de substitution économique permettant de dériver et mettre à jour une cartographie routière de haute qualité.

De manière plus anecdotique, on pourra noter que la plupart des algorithmes ont une tendance naturelle à converger vers une solution plus précise à mesure que le nombre de traces collectées augmente. Pour une répartition homogène des collecteurs FCD dans une population de conducteurs, on pourra remarquer que les techniques de map inference permettent un niveau de détails et de précision cartographique du territoire proportionnel à la fréquentation des citoyens.

La littérature de référence distingue en général trois grandes catégories d'algorithmes de map inference pour la construction d'une cartograhie routière numérique (Hillnertz, 2014; Qiu et Wang, 2016; Biagioni et Eriksson, 2012b; Dupuis et al., 2014) :

- **L'approche par estimation de densité**

  Une approche orientée image est adoptée. La première étape consiste à estimer un histogramme en deux dimensions et lissé de la densité des traces sur la zone à cartographier. Dans une version plus évoluée, la densité est directement estimée par des techniques adéquates, telles que la méthode des noyaux (*kernel density estimation*, ou KDE), produisant ainsi une carte raster de la fréquentation des traces, comme illustré sur la figure 1.7. L'image est ensuite seuillée, pour en extraire les zones sur lesquelles la densité estimée est supérieure à une certaine valeur, choisie après calibration. Plusieurs techniques alternatives de traitement des images peuvent alors être employées pour estimer les lignes centrales des axes routiers : diagramme de Voronoï (Davies et al., 2006), opérations de morphologie mathématique (Chen et Cheng, 2008) ou encore segmentation par watershed (Steiner et Leonhardt, 2011).

  Notons que du fait de la prévalence de l'utilisation de la méthode des noyaux, cette approche par estimation de densité est en général abusivement appelée *approche KDE*. On distingue en général les méthodes KDE fondées sur des points GPS individuels, des méthodes basées sur des segments de trajectoire.

  L'estimation de densité par noyaux permet une modélisation probabiliste de la répartition des points sur le réseau routier. Malheureusement, la force statistique de la méthode ne peut être conservée par les opérations de seuillage et les traitements géométriques subséquents, rendant ainsi l'approche peu robuste aux données aber-

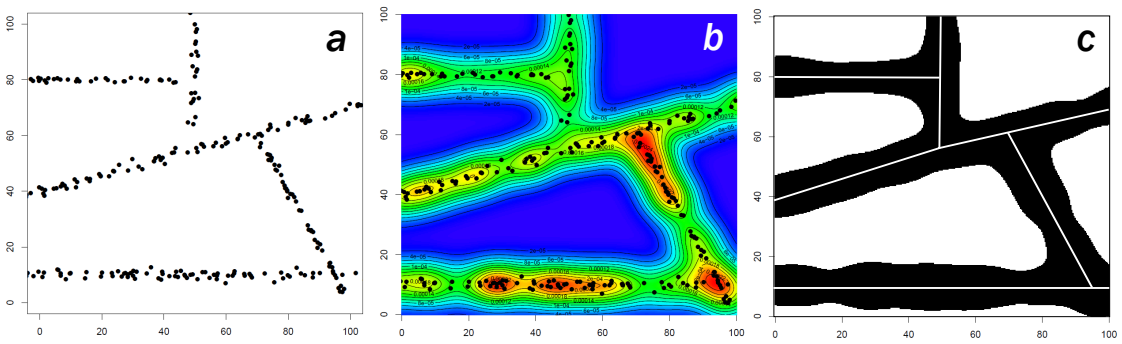


FIGURE 1.7 – Test de map inference du réseau routier sur les données de Mitaka (*cf* chapitre 4) : a. Positions GPS observées. b. Éstimation par noyaux de la densité des points GPS. c. Seuillage binaire et reconstruction du graphe par squelettisation.

rantes. Pour contourner cette faiblesse, Biagioni et Eriksson (2012b) proposent une construction en deux phases : dans un premier temps, l'approche KDE classique est employée pour former une approximation grossière du réseau, sur laquelle les trajectoires sont recalées dans un second temps dans le but d'affiner la topologie du réseau en ajoutant ou en supprimant des arcs dans le graphe routier au niveau des incohérences produites par le recalage. Cette méthode permet donc de combiner les avantages des approches KDE ponctuelles et KDE par segments.

- **L'approche par clusterisation**

  Dans cette approche, les positions GPS sont clusterisées en sous-groupes, en fonction de leurs paramètres (position, cap...). L'algorithme le plus classiquement employé est celui des *k-means* (Schroedl et al., 2004; Worrall et Nebot, 2007) mais on relève également l'emploi de méthodes de clusterisation plus spécifiquement adaptées aux traces GPS, telles que DBSCAN (Qiu et Wang, 2016). Les trajectoires de chaque cluster sont agrégées de sorte à fournir une estimation de la géométrie de la route.

  Après cette clusterisation, la topologie du réseau est construite en liant les différents clusters générés à l'aide des trajectoires complètes. Dans le cas où la quantité de traces disponibles est insuffisante, certains auteurs tels que Chen et al. (2010) ou encore Dupuis et al. (2014) proposent de compléter la construction avec un modèle a priori sur la géométrie de la route (généralement une courbe polynomiale par morceaux, ou tout autre type de fonction possédant de bonnes propriétés de régularité). L'estimation paramétrique de la route est alors calculée par un filtrage de Kalman en combinant les observations relevées par GPS avec un modèle dynamique simple pour le déplacement du véhicule. Cette approche est particulièrement adaptée lorsque le nombre de traces disponibles est faible mais que les récepteurs GPS sont de bonne qualité (e.g. en GPS différentiel). Nous adopterons une démarche similaire pour prétraiter les profils de vitesse à l'aide de lectures Doppler et odométriques, et d'un modèle de trajet de référence (cf chapitres 2 et 3).

- **L'approche par fusion de traces**

Dans cette dernière approche, les trajectoires sont considérées dans leur entiereté. L'algorithme procède à une construction incrémentale de la carte par intégrations successives des traces. Par exemple, Cao et Krumm (2009) proposent une approche reposant sur une modélisation énergétique avec un champ de potentiel attractif incitant les trajectoires à s'agréger autour de trajectoires moyennes lisses. Une approche similaire est adoptée par Xie et al. (2015), qui incorporent en plus dans la méthodologie, un alignement des différents points des trajectoires à l'aide d'un algorithme de type *curve registration*.

Cette approche gloutonne n'est pas toujours optimale, mais présente l'avantage de permettre une construction incrémentale de la carte, s'adaptant ainsi à l'arrivée progressive des données à mesure que les véhicules de la flotte FCD parcourent le réseau.

Il est possible de considérer que pour un nombre suffisant de trajectoires FCD, et une fréquence d'acquisition des données de l'ordre du Hz, le problème de l'inférence du réseau routier à partir des traces GPS est pratiquement résolu à l'heure actuelle (notamment au regard des quantités de données FCD arrivant sur le marché prochainement). Cependant, il reste encore beaucoup de progrès à faire dans le domaine de l'enrichissement de ce réseau.

- **Raffinement de la topologie :** les méthodes présentées ci-dessus ne permettent que la construction de la géométrie du réseau routier ainsi que d'une topologie sommaire. Dans de nombreux cadres d'applications, il peut être important de modéliser de manière plus fine la topologie, principalement au niveau des intersections complexes.

- **Attributs de routes** : limitations de vitesse (Van Winden, 2014), largeur, état d'usure du revêtement, accessibilité par type de véhicule... (Biljecki et al., 2013). Schroedl et al. (2004) ont proposé un algorithme permettant d'estimer conjointement le nombre de voies d'un axe routier, ainsi qu'une position précise de chaque ligne centrale, à partir de GPS différentiels embarqués dans des véhicules traceurs. La solution obtenue peut également permettre de contribuer au raffinement topologique des intersections à voies multiples. L'idée a été étendue aux récepteurs GPS bas de gamme par Chen et Krumm (2010) qui proposent une modélisation probabiliste de la répartition transversale des observations GPS sur une chaussée à voies multiples, à l'aide d'un mélange gaussien, que l'on peut considérer comme une version paramétrique de l'approche KDE pour la construction du réseau routier.

- **Informations sémantiques et toponymiques** : nature de la route (nationale, départementale...) et nom de rue principalement. Ces informations sont difficiles à collecter exclusivement à partir de données FCD. En général, les travaux de recherche dans ce domaine procèdent par fusion de données, à l'instar de Li et al. (2015) qui proposent un algorithme hybride pour inférer les noms de rues à partir de trajectoires GPS et de données textuelles issues de réseaux sociaux.

- **Signalisation routière** : feux tricolores, stops, cédez-le-passage... En pratique, la position exacte du panneau de signalisation n'a pas grand intérêt et il est en principe plus pertinent d'enregistrer uniquement les coordonnées longitudinales (on parlera d'abscisses curvilignes dans la suite) des extrémités de la zone d'application réglementaire du panneau. Dans cette optique, un élément de signalisation routière (tout

comme la plupart des éléments énumérés ci-dessous) peut être considérés comme un couple d'attributs (type, position).

- **Dispositifs d'apaisement de la circulation (traffic calming devices)** : cassis, radars, caméras, chicanes...

- **Marquages au sol** : lignes continues, chevrons, pistes cyclables, passages piétons...

- **Évènements éphémères** : accident, déviation, chantiers...

Parmi les pistes d'enrichissement proposées ci-dessus, la signalisation routière ne figure pas dans la plupart des listes de potentialités offertes par l'arrivée des données FCD. En effet, il peut être tentant de penser que chaque échelon local dispose d'une cartographie fine et détaillée de cette signalisation, qu'il serait aisé d'agréger au niveau de l'État pour fournir une base nationale, d'autant plus qu'une ordonnance impose aux collectivités de remonter les données et statistiques liées à la sécurité routière. En pratique, aucun décret d'application n'a été publié à ce jour pour accompagner cette ordonnance, la rendant ainsi inopérante. La collecte des différentes bases de données locales (quand elles existent) nécessite donc un important déploiement de moyens pour l'intégration et l'homogénisation.

Sur ce point, les plateformes d'échange et de mise à disposition de données géographiques ne peuvent se targuer de faire mieux. À titre d'exemple, pour une raison que nous ignorons, Google Maps semble avoir récemment retiré la position des carrefours contrôlés par des feux tricolores de ses fonds de cartes en ligne. En parallèle, les statistiques [5] de la plateforme *OpenStreetMap* font état de 6746 carrefours contrôlés par des feux tricolores sur l'ensemble du territoire national en 2018, ce qui correspond à seulement 5% du nombre total de carrefours de ce type qui pourraient être repertoriés. Les chiffres annoncés au niveau mondial sont encore plus déficitaires.

La signalisation fait foi d'obligation réglementaire, et nécessite de ce fait l'existence d'une infrastructure routière numérique souveraine, que l'on ne peut raisonnablement espérer créer et maintenir à l'aide de remontées individuelles.

1.2.3 Le véhicule autonome

Bien que particulièrement populaire actuellement, le concept de véhicule autonome n'est pas nouveau, et remonte à 1977, lorsqu'un laboratoire de robotique de Tsukuba (Japon) mena avec succès des expérimentations de véhicule à conduite automatisée sur un ciruit dédié muni de marquages au sol spécifiques. Ce véhicule pouvait atteindre des vitesses de l'ordre de 30 km/h. L'expérience a été réitérée quelques années plus tard par le constructeur automobile *Mercedes-Benz*, en partenariat avec l'Université de la Budeswher de Munich, avec un utilitaire équipé de caméras et d'un logiciel de reconnaissance d'images, sur un réseau routier standard sans trafic. Les expérimentations s'accélèrent alors en Europe à partir de 1987, avec le programme Prometheus, lancé à la demande de l'industrie automobile et financé à hauteur de 800M€. Il débouchera en particulier à la démonstration par *Daimler-Benz* du potentiel de cette technologie émergente, avec une expérimentation

5. `https://taginfo.openstreetmap.fr/`

de conduite autonome sur l'autoroute A1 au départ de Paris, en condition réelle de trafic et à une vitesse de 130 km/h. Malgré ces premiers succès très prometteurs, la route est encore longue avant l'industrialisation d'un véhicule parfaitement autonome, capable de transporter passagers et/ou marchandises en sécurité et en toute situation (Ehrlich, 2017).

On peut définir le véhicule autonome comme un terme parapluie désignant l'ensemble des technologies destinées à automatiser les processus de perception de l'environnement, de décision et de manœuvre d'un véhicule terrestre avec pour finalités la réduction des risques associés à la circulation routière et l'amélioration de la mobilité (Danish Ministry of Energy et Climate, 2017). Il regroupe en particulier deux sous-catégories de véhicules :

- Le **véhicule connecté** (VC) qui peut s'apparenter aux flottes FCD que nous avons passées en revue dans la section 1.1.2 et qui est aujourd'hui la norme sur le marché et est en passe de devenir majoritaire parmi les véhicules en circulation.

- Le **véhicule automatisé** (VA) qui permet de retirer certaines tâches au conducteur et désigne donc une large gamme de modèles, allant des véhicules munis de systèmes d'aide à la conduite jusqu'à des véhicules sans pilote.

La fusion de ces deux catégories constitue le **véhicule automatique et connecté** (VAC).

Dans une logique de partage des recherches et d'uniformisation des textes réglementaires, une définition consensuelle du véhicule autonome est rapidement devenue indispensable (Ehrlich, 2017). La définition actuellement majoritaire (dénommée recommendation SAE J3016) propose 6 niveaux d'automatisation, le niveau 0 correspondant au véhicule standard des années 1990, le niveau 5 correspondant à un véhicule parfaitement autonome, capable de conduite en toutes circonstances et dans tous les types d'environnement (urbain, autoroute, rural...) sans intervention humaine. Entre ces deux extrêmes, 4 niveaux viennent graduer l'échelle : les niveaux 1 et 2 référant à des systèmes d'aide à la conduite, pour une (niveau 1) ou les deux (niveau 2) commandes d'action longitudinale (contrôle de vitesse de croisière, freinage d'urgence...) ou latérale (changement de file...). Le niveau 3, qui correspond au stade le plus avancé des véhicules actuellement en circulation, désigne les modèles capables de remplacer le conducteur dans des conditions prédéfinies, avec une supervision de l'humain qui doit être capable de reprendre les commandes à tout instant. Ce dernier point le différencie du véhicule de niveau 4, dernière étape avant l'automatisation complète, qui désigne un système capable de conduire dans certaines conditions avec un système de mise en sécurité du véhicule en cas d'impossibilité à gérer une situation, ne nécessitant ainsi plus un contrôle humain permanent.

Les estimations divergent selon les sources, mais la plupart des études prédisent la commercialisation du véhicule de niveau 4 d'ici 2025, suivie du niveau 5 avant 2040. Ces estimations sont à prendre avec précaution, puisque sensibles à un grand nombre de facteurs extérieurs aux seules avancées technologiques : dispositions légales en faveur du véhicule autonome, confiance accordée par les potentiels consommateurs, évolution démographique, accessibilité aux moyens de transport en commun, développement du télétravail, baisse du prix des technologies développées au standard de coûts de l'industrie automobile... (Ministère de la transition écologique et solidaire, 2018)

D'un point de vue matériel, un véhicule autonome est équipé de plusieurs types de capteurs, lui permettant de cartographier son environnement à différentes échelles (comme illustré sur la figure 1.8) ainsi que d'un ordinateur de bord, lui permettant d'analyser les données collectées, de décider des manœuvres à entreprendre et de transmettre les commandes au bus de données qui à son tour les distribue sur les différents actionneurs.

La localisation grossière est assurée par un récepteur GPS bi-fréquence (généralement en mode multi-constellation pour augmenter les chances d'avoir un nombre suffisant de satellites en vue [6], en particulier en environnement urbain dense), parfois en hybridation avec une centrale inertielle. La position précise est par la suite calculée à l'aide de points d'amers visuels détectés à l'aide des autres capteurs : plusieurs caméras munies d'un système de reconnaissance d'images, un radar basse fréquence pour les alertes de collisions dans l'environnement proche (risque de collision avec des piétons, etc), un radar longue portée pour la détection d'obstacles lointains, un système LIDAR [7] pour les alertes de collisions et le repérage des marquages au sol et enfin un réseau d'émetteurs/récepteurs ultrasons pour la détection à très courte distance. En règle général, plus un capteur est capable de collecter des informations à longue portée, plus son faisceau est directionnel.

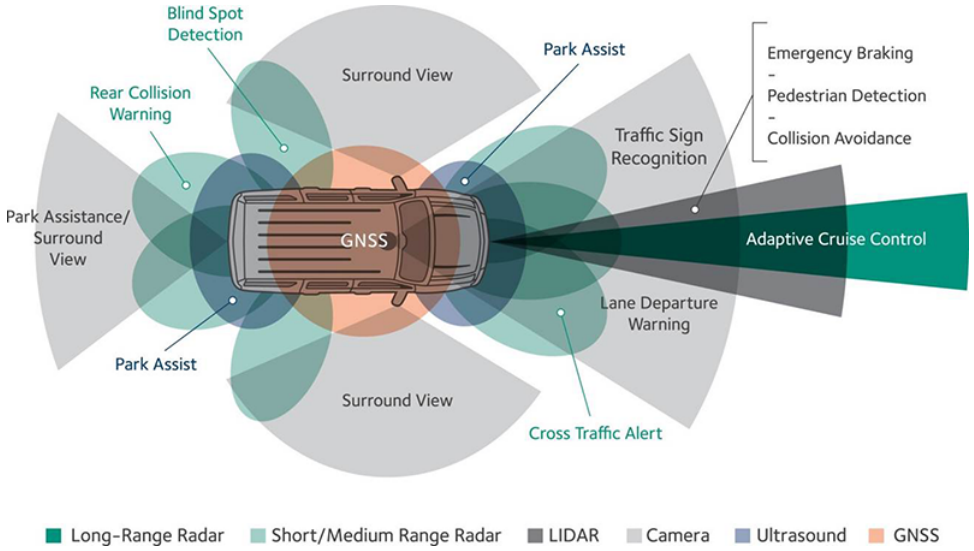


FIGURE 1.8 – Réseau de capteurs d'un véhicule autonome typique. Source : `https://www.novatel.com/industries/autonomous-vehicles/technology`

L'ordinateur de bord utilise les données collectées par les capteurs [8] pour calculer un faisceau de trajectoires possibles et déterminer en temps réel la trajectoire la plus optimale (Ehrlich, 2017). Ce mode de fonctionnement se décline en trois niveaux d'échelle spatio-temporelles :

- Le **niveau stratégique** : le véhicule utilise des informations cartographiques globales pour calculer son itinéraire.

6. *cf* 4 et annexe A
7. Light detection and ranging : système de détection d'obstacles par balayage d'un faisceau laser à très haute fréquence dans le champ de vision.
8. Typiquement environ 4 To de données par jour soit l'équivalent de 2500 utilisateurs internets ou encore un millier de fois la quantité de données produites par un véhicule traceur.

- Le **niveau tactique** désigne l'ensemble des manœuvres à réaliser à l'échelle de quelques secondes : changement de voie, dépassement... Il possède une contrainte d'interopérabilité très forte avec le niveau stratégique.

- Le **niveau opérationnel** comprend les actions urgentes à effectuer, en général en réaction à un évènement par nature imprévisible (piétons, accident...)

Les données nécessaires pour la plannification du niveau stratégique sont en général largement présentes dans les bases de données géographiques actuelles, et déjà intégrées, par exemple dans les navigateurs GPS. À l'inverse, le niveau opérationnel ne repose que sur des données locales, qui sont donc quasi-exclusivement acquises à l'aide des capteurs du véhicule. Entre ces deux niveaux, la planification tactique est en partie assurée par les capteurs, mais leur portée est malheureusement limitée [9]. En matière de sécurité routière et similairement au cas des conducteurs humains, il est bien connu que les risques peuvent être évités à l'aide d'une meilleure anticipation des actions à effectuer, ce qui nécessite d'élargir l'horizon électronique du véhicule. Deux sources de données peuvent venir compléter les informations manquantes au niveau tactique : les données externes reçues, notamment par d'autres véhicules (V2V) ou par l'infrastructure (I2V) ; les données cartographiques stockées en amont dans le véhicule. Cette cartographie doit être plus détaillée que ce que préconisent les standards actuels (principalement au niveau de la géométrie des routes), mais doit également contenir la signalisation routière, les marquages au sol, ainsi que l'ensemble des éléments de mobilier urbain ou du paysage qui seraient susceptible de servir de point d'amer visuel pour la localisation précise du véhicule.

En particulier, une connaissance fine de la signalisation routière revêt une importance capitale. Même en étant équipé d'un système robuste de reconnaissance de la signalisation, un véhicule autonome ne pourra jamais se prémunir du risque d'avoir un panneau obstrué par un autre véhicule, des conditions climatiques défavorables ou encore de devoir faire face à des attaques, en particulier avec le concept récent d'*adversarial machine learning* (Evtimov et al., 2017). La carte a priori (concept de *prior map*) permet de remédier à ce problème, en complétant ou en se substituant aux données acquises par les capteurs en cas de besoin.

Ce manque de données spatiales pour la phase de planification tactique a été relevé par plusieurs opérations pilotes, par exemple dans les conclusions préliminaires du projet MOOVE [10], qui visait à analyser des situations de conduite critiques (avec conducteur humain) afin d'identifier précisément les problèmes de sureté auxquels le véhicule autonome sera potentiellement confronté. Actuellement, plusieurs phases de réflexions sont en cours pour définir le rôle précis de l'IGN dans la constitution et la mise à disposition d'une telle cartographie (Ministère de la transition écologique et solidaire, 2018). Les actions incombant à l'IGN pourraient se structurer en quatre axes : identifier les besoins prioritaires en matière de cartographie haute définition (en particulier pour le repérage des points d'amer visuel), étudier des solutions alternatives en cas de perte de signal GPS, entretenir et diffuser une donnée géographique de référence et assurer un service d'homologation de la qualité des couches cartographiques nécessaires au fonctionnement nominal des véhicules

9. On estime à environ 100 à 200 m en moyenne la portée du réseau de capteurs d'un véhicule autonome
10. `http://www.vedecom.fr/presentation-du-projet-moove-a-lautonomous-vehicle-test-development-symposium-de-stutggart/`

autonomes.

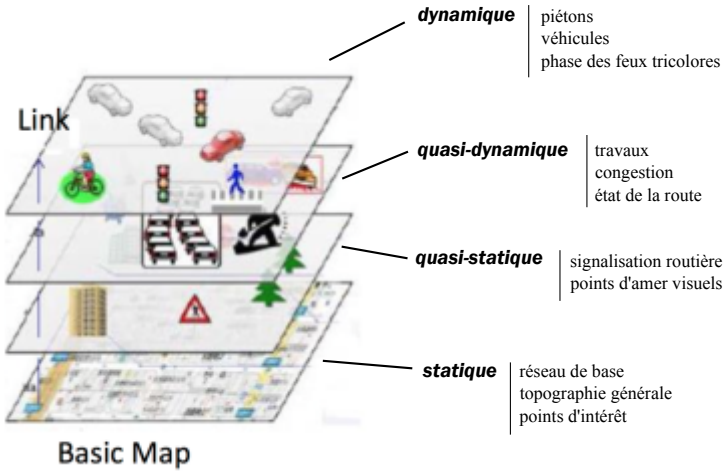


FIGURE 1.9 – Carte a priori de type *Local Dynamic Map* (LDM), nécessaire au bon fonctionnement d'une flotte de véhicules autonomes. Illustration adaptée de Shirato (2016).

Plus schématiquement, en représentant la carte à constituer selon le concept de *Local Dynamic Map* (LDM) à plusieurs niveaux (où chaque niveau représente des données plus temporaires que le niveau qui lui est immédiatement inférieur, comme illustré sur la figure 1.9) une question essentielle subsiste dans la définition du nombre de couches qui incomberont aux agences de cartographie.

1.2.4 Motivations annexes

La connaissance d'une telle cartographie pourrait en réalité servir des motivations qui dépasse le simple cadre du véhicule autonome. À un horizon plus proche, on peut penser aux applications suivantes :

- Le calcul précis d'itinéraire et des temps de trajet est une fonctionnalité embarquée dans tous les récepteurs GPS de navigation du marché. La détermination de l'itinéraire est en général effectuée par un algorithme de calcul de plus court chemin sur un graphe valué par des coefficients inversement proportionnels aux vitesses limites, ou aux vitesses pratiquées (Bonin, 2002). L'utilisation des données FCD permet une estimation plus fine en tenant compte de la fluidité moyenne du trafic sur le réseau (Wang et al., 2014). Une connaissance exhaustive des feux (voire de leurs schémas de cycles temporels) peut permettre d'améliorer encore cette estimtaion, résultant parfois en un choix plus judicieux de l'itinéraire à suivre.

- Pour les gestionnaires de réseau, cette cartographie est un prérequis indispensable pour des simulations d'écoulement de trafic, en vue d'optimiser la topologie du ré-

seau (Annunziata et al., 2007). De plus cette connaissance peut assister les décideurs publics dans les politiques d'aménagement de l'espace urbain.

- Enfin, si la LDM peut être utilisée comme source de données a priori par un véhicule autonome pour élargir son horizon électronique, et ainsi renforcer sa sécurité, elle peut aussi être employée pour transmettre des messages au conducteur, en amont d'un danger potentiel (Wilson et al., 1998; Chen et al., 2016a). D'un point de vue environnemental, la plupart des normes d'éco-conduite s'appuient également sur une anticipation des actions à effectuer 200 ou 300 m en aval (Andrieu et al., 2013b). Les couches intermédiaires de la LDM peuvent y trouver un cadre d'application immédiat.

1.3 Approche du travail de thèse

Historiquement, les méthodes statistiques étaient utilisées pour résoudre des problèmes liés à l'industrie, à l'agriculuture ou à la démographie, dans lesquels les quantités de données à traiter restaient modérées. Depuis l'avènement de l'ère informatique, ces quantités grandissent de manière exponentielle [11], mettant ainsi en échec la plupart des méthodes classiques et faisant ressentir le besoin d'algorithmes capables d'analyser de gros volumes de données pour en extraire l'information pertinente (Friedman et al., 2001). L'apprentissage automatique (ou apprentissage machine, traduction directe de l'anglais *machine learning*) a été impulsé dans ce sens au début des années 60, avec la conception de calculateurs électroniques s'inspirant du fonctionnement du cerveau humain, et qui seront les précurseurs des réseaux de neurones artificiels qui connaissent actuellement une popularité sans précédent. De manière paradoxale, cette classe de méthodes a disparu du champ de recherche dans les années 80, avant de ré-apparaître une vingtaine d'années tard, à la faveur de l'augmentation des capacités de calcul des machines informatiques modernes (en particulier avec le développement de processeurs graphiques toujours plus performants, permettant une parallélisation massive des calculs à distribuer sur le réseau de neurones). Entre temps, d'autres méthodes d'apprentissage ont vu le jour, permettant de traiter une vaste gamme de problèmes. À l'heure actuelle, l'apprentissage statistique est utilisé de manière opérationnelle dans de nombreux secteurs, par exemple, dans l'industrie automobile (Mitrović, 2004; Torkkola et al., 2004), en sécurité routière (Chen et al., 2016b), en cartographie automatique (Miyazaki et al., 2016), en agriculture (Kuwata et Shibasaki, 2015) ou encore en diagnostic médical (Nguyen et al., 2013) pour ne citer que quelques exemples.

L'apprentissage est considéré comme un champ de l'intelligence artificielle, dans lequel on cherche à imiter les processus de décision humains à l'aide de méthodes statistiques, en se basant sur un jeu de données d'entraînement. On distingue deux catégories d'algorithmes, l'apprentissage *supervisé*, lorsque les variables à inférer sont données à l'algorithme lors de la phase d'entraînement, et l'apprentissage *non-supervisé* dans lequel on cherche à trouver la structure sous-jacente des données en fonction des seuls attributs observés. Dans cette thèse, nous utiliserons quasi-exclusivement des méthodes supervisées.

11. La quantité totale de données stockées devrait être multipliée par 5.3 d'ici 2025.

1.3.1 L'apprentissage supervisé

Dans cette section, on considère l'espace $\mathcal{X} \times \mathcal{Y}$, où $\mathcal{X}$ désigne l'espace des descripteurs et $\mathcal{Y}$ représente l'espace des étiquettes (on parle également de labels, ou encore de variables cibles). En général $\mathcal{X} \subseteq \mathbb{R}^p$, où p est une dimension fixée, tandis que $\mathcal{Y}$ peut être de deux types différents en fonction de la nature du problème à résoudre. Dans les problèmes de classification, on cherche à déterminer à quelle classe y appartient un élément $\mathbf{x}$, dans ce cas $\mathcal{Y}$ est un espace catégoriel (par exemple $\{c_1, c_2 ... c_k\}$ pour un problème à k classes distinctes). Si $\mathcal{Y}$ ne contient que 2 élements, on parle de classification binaire, et il est d'usage de noter $\mathcal{Y} = \{0, 1\}$ ou encore $\mathcal{Y} = \{-1, 1\}$. Lorsque la variable à inférer est continue, on parle de problème de régression, et on a en général $\mathcal{Y} \subseteq \mathbb{R}$.

On appelle *jeu d'entraînement*, ou *base d'exemples* un ensemble $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}$ pour $i \in \{1, 2, ...n\}$, où les données $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}$ sont n réalisations indépendantes et identiquement distribuées (i.i.d) suivant une loi jointe inconnue $p(X, Y)$.

L'objectif du problème consiste à construire une *fonction de décision* $f_{\boldsymbol{\theta}}$ (où $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, un ensemble paramétrique de dimension quelconque), qui à un ensemble de descripteurs $\mathbf{x} = (x_1, x_2, ...x_p)$ associe une étiquette y :

$$f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Y}. \tag{1.1}$$

Considérons un exemple simple. Imaginons que nous souhaitions concevoir un classifieur capable de reconnaître des caractères écrits à la main [12]. Chaque caractère est une image en niveau de gris de 12 pixels de côté. Si on sait a priori que ces caractères ne contiennent que des chiffres, l'espace des variables cibles sera discret et contiendra 10 modalités $\mathcal{Y} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Si chaque pixel est codé sur un octet, l'espace des descripteurs sera un espace à 144 dimensions (12×12), chaque dimension pouvant prendre 256 valeurs différentes : $\mathcal{X} = \{0, 255\}^{144}$. Pour une image $\mathbf{x}$ donnée en entrée, la fonction de décision $f_{\boldsymbol{\theta}}$ devra retourner le chiffre représenté par la vignette.

Idéalement, on souhaite que $f_{\boldsymbol{\theta}}$ reproduise fidèlement le comportement observé dans les données du jeu d'entraînement. Pour s'en assurer, on se munit d'une fonction de perte $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$, qui à un couple d'étiquettes (y, y') associe le coût de pénalité $L(y, y')$ associé à l'erreur commise lorsque que la fonction de décision $f_{\boldsymbol{\theta}}$ attribue l'étiquette y' à une donnée étiquetée y. En général, la fonction de coût est nulle lorsque $y = y'$ (on ne pénalise pas les bonnes décisions), mais ne partage pas nécessairement toutes les propriétés des distances. Par exemple, dans le cas d'un diagnositic médical de routine (problème de classification binaire), il est intuitivement moins fâcheux de faire subir des examens complémentaires à une personne saine que d'échouer à détecter une maladie grave. La fonction de perte n'est donc pas nécessairement symétrique.

On définit alors la fonction de risque $R : \boldsymbol{\Theta} \to \mathbb{R}^+$ (dépendant de la paramétrisation $\boldsymbol{\theta}$ de la fonction de décision), comme l'espérance (sur la loi jointe des données) du coût de l'erreur de décision prise sur une donnée suivant la loi p :

12. Il s'agit d'un problème modèle classique en apprentissage premettant de comparer les performances de plusieurs algorithmes, et généralement désigné sous l'appellation MNIST (Deng, 2012).

$$R(\boldsymbol{\theta}) = \mathbb{E}[L(y, f_{\boldsymbol{\theta}}(\mathbf{x}))] = \int_{\mathcal{X}\times\mathcal{Y}} L(y, f_{\boldsymbol{\theta}}(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy. \tag{1.2}$$

La paramétrisation optimale $\hat{\boldsymbol{\theta}}$ est alors calculée en minimisant le risque R, et on en déduit une estimation de l'étiquette y_{new} attribuée à une nouvelle donnée $\mathbf{x}_{new}$:

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\ R(\boldsymbol{\theta}), \qquad\qquad \hat{y}_{new} = f_{\boldsymbol{\theta}^*}(\mathbf{x}_{new}). \tag{1.3}$$

Cette définition est en réalité inopérante, car la loi jointe $p(\mathbf{x}, y)$ est inconnue. Si ce n'était pas le cas, on pourrait aisément en déduire la loi conditionnelle $p(y|\mathbf{x})$ en la divisant par la loi marginale de $\mathbf{x}$ et on aurait toute l'information nécessaire à l'estimation de y_{new}. On contourne ce problème en substituant à R un risque empirique $\widetilde{R}$, estimé à partir du jeu de données d'entraînement :

$$\widetilde{R}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} L(y^{(i)}, f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})). \tag{1.4}$$

Cette substitution est possible puisque les données $(\mathbf{x}^{(i)}, y^{(i)})$ de la base d'entraînement sont *i.i.d.* par hypothèse et la loi forte des grands nombres (Suquet, 2003) nous garantit que la variable aléatoire $\widetilde{R}(\boldsymbol{\theta})$ converge presque sûrement vers $R(\boldsymbol{\theta})$.

Dans ce manuscrit, nous traiterons uniquement des problèmes de classification binaire (pour la détection d'un élément de signalisation routière) ou de régression (pour sa localisation). En régression, la fonction de perte classiquement utilisée est la perte quadratique :

$$L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2. \tag{1.5}$$

La fonction à estimer devient alors l'espérance conditionnelle $f^*(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$.

Pour un problème de classification binaire, il existe deux modes d'estimation différents. Dans un premier cas, on peut estimer une fonction $g_{\boldsymbol{\theta}}$ dont le signe donnerait la classe de la variable cible. Par exemple, pour $\mathcal{Y} = \{-1, +1\}$:

$$f(\mathbf{x}) = \begin{cases} +1 & \text{si } g_{\boldsymbol{\theta}}(\mathbf{x}) \geqslant 0 \\ -1 & \text{sinon.} \end{cases} \tag{1.6}$$

C'est en particulier l'approche suivie par les algorithmes de type Séparateurs à Vaste Marge (SVM), qui construisent un hyper-plan séparateur affine (paramétré par $\boldsymbol{\theta}$) dans l'espace des descripteurs. Une nouvelle donnée (*i.e.* un nouveau point de l'espace) est alors

projetée sur l'axe porté par le vecteur normal à l'hyper-plan, et le signe de la valeur résultante indique la prédiction du classifieur pour cette donnée. Les réseaux de neurones artificiels utilisent une approche similaire.

Dans un autre mode de fonctionnement, on peut chercher à estimer la valeur de probabilité $\mathbb{P}[Y = 1|\mathbf{X} = \mathbf{x}]$, la fonction de décision s'exprime alors à l'aide de la règle de décision de Bayes :

$$f(\mathbf{x}) = \begin{cases} +1 & \text{si } \mathbb{P}[Y = 1|\mathbf{X} = \mathbf{x}] \geqslant 0.5 \\ -1 & \text{sinon.} \end{cases} \tag{1.7}$$

Cette fonction minimise le risque R pour la perte indicatrice 0-1 : $L(y, y') = \mathbb{1}_{y \neq y'}$.

Dans ces deux modes, on voit que la classification binaire peut être traitée comme un cas particulier de régression. Cependant, la seconde approche, offre une modélisation pleinement probabiliste du problème, bien qu'il existe des méthodes pour exprimer la fonction $g_{\boldsymbol{\theta}}$ de la première approche à l'aide d'un indice de confiance, par exemple via la fonction *soft-max*. Dans notre cadre d'application, il est important de disposer d'une valeur de probabilité permettant de caractériser la confiance que l'on peut mettre dans la détection de la signalisation routière. Pour cette raison, à l'exception du chapitre 5, nous utiliserons exclusivement des méthodes de classification binaire fonctionnant suivant la seconde approche.

Il reste cependant deux points importants auxquels il faut prêter attention. En premier lieu, notons que si l'espace des paramètres $\boldsymbol{\Theta}$ n'est pas suffisament contraint (en termes de nombre de degrés de liberté), en particulier par rapport au nombre de données disponibles dans le jeu d'entraînement, il est toujours possible de trouver une paramétrisation $\boldsymbol{\theta}^*$ qui annule le risque empirique. On parle alors de sur-apprentissage (*overfitting*) pour désigner cette situation où l'algorithme a perdu en pouvoir de généralisation en capturant le bruit présent dans les données d'entraînement. Deux précautions permettent de se prémunir du sur-apprentissage :

- On régularise l'estimation, en ajoutant des contraintes sur l'espace $\boldsymbol{\Theta}$. En apprentissage, cette option correspond souvent à choisir les paramètres de l'algorithme de sorte à limiter le nombre de degrés de liberté du modèle.

- On teste le modèle sur une base de validation, de structure similaire à la base d'entraînement, mais contenant de nouvelles données. Tant que le résultat n'est pas satisfaisant, on retourne au point précédent pour modifier les contraintes de l'algorithme (on parle d'hyper-paramètres du modèle).

Enfin, on utilise une troisième base d'exemples (appelée base de test) permettant une évaluation finale des performances de l'algorithmes, à l'aide de données fraîches (*i.e.* des données qui n'ont pas été utilisées, ni pour la paramétrisation $\boldsymbol{\theta}$, ni pour l'hyper-paramétrisation de $\boldsymbol{\Theta}$).

D'autre part, on observe que l'évaluation du risque empirique dépend de deux aspects bien distincts : de la capacité de l'algorithme à reconstruire une étiquette $f_{\boldsymbol{\theta}}(\mathbf{x})$ qui soit aussi proche que possible (au sens de la fonction de perte) de l'étiquette y réellement associée à $\mathbf{x}$; mais aussi des probabilités d'apparition de données étiquetées y. Prenons un exemple pour un cas de classification binaire, avec un algorithme qui reconnaît parfaitement des données étiquetées y^0 mais qui échoue en moyenne une fois sur deux pour des données étiquetées y^1. La valeur du risque empirique dépend alors de la proportion de données de type y^0 dans les bases d'exemples. Pour une base ne contenant que des données y^0, le risque empirique sera nul (classifieur parfait). À l'inverse, si le jeu ne contient que des données de type y^1, le risque empirique vaudra 0.5 (classifieur purement aléatoire), ce qui représente une situation diamétralement opposée. Cet écueil sera pris en compte lors des phases d'entraînement (en rééquilibrant le jeu de données si besoin ou alors en définissant une fonction de perte ad hoc), mais aussi dans les phases de validation (en analysant séparément la capacité de l'algorithme à traiter des données de types y^0 et y^1).

1.3.2 L'apprentissage de données fonctionnelles

D'un point de vue informatique, où les quantités adressables sont nécessairement finies, il est toujours possible de considérer une fonction à valeurs réelles comme un vecteur $\mathbf{x}$ de $\mathbb{R}^p$, dont les composantes désignent des valeurs régulièrement échantillonnée et où la dimension p est choisie suffisamment grande pour permettre une modélisation fine de la fonction. En pratique, ce mode opératoire n'est pas satisfaisant, en particulier pour deux raisons principales :

- Décrire la fonction avec un degré de précision suffisant nécessite un nombre d'échantillons élevé, forçant ainsi les algorithmes d'apprentissage à travailler avec des données en grande dimensions. Malheureusement, à mesure que la dimension des données à traiter augmente, le nombre d'exemples nécessaires pour couvrir l'espace des descripteurs croît exponentiellement. À partir d'une certaine valeur de p, il est donc pratiquement impossible de collecter un jeu d'entraînement suffisamment vaste pour entraîner un modèle statistique (Giraud, 2014). On parle de *fléau de la dimension* (ou *curse of dimensionality*). Certains algorithmes, tels que les forêts d'arbres aléatoires, que nous utiliserons à partir du chapitre 3, permettent avec plus ou moins de succès de contourner ce problème, mais on obtient souvent de meilleures performances en réduisant le nombre de dimensions des données à traiter dès la phase de modélisation des instances.

- L'échantillonnage fin de la fonction pose en réalité un second problème. Pour des signaux suffisamment réguliers, la fonction d'auto-corrélation au voisinage de 0 est nécessairement significative, et deux échantillons successifs x_k et x_{k+1} vont être statistiquement semblables. Certains algorithmes d'apprentissage, comme par exemple le LASSO (Tibshirani, 1996) ou les forêts aléatoires (Gregorutti et al., 2017) permettent de ne sélectionner que les descripteurs les plus pertinents dans l'estimation. Malheureusement, les corrélations entre les descripteurs posent de sérieux problèmes à ces algorithmes de sélection. Par exemple, lorsque plusieurs valeurs sont corrélées, elles se partagent le contenu informatif et leurs importances individuelles (relativement au reste des variables) diminuent, les rendant ainsi susceptibles d'être éliminées.

Une solution à ce double problème peut consister à échantillonner plus finement les zones de fortes variations de la fonction qui sont également les zones les plus informatives. Malheureusement, rien ne garantit que ces zones sont co-localisées sur l'axe des abscisses des fonctions à traiter, ce qui exclut la possibilité d'établir systématiquement un schéma commun d'échantillonnage pour l'ensemble des données de la base d'entraînement.

L'analyse de données fonctionnelles (ou FDA pour *Functional Data Analysis*) est une branche des statistiques modernes, qui s'est développée notamment à partir des travaux de Deville (1974), puis de Besse (1979) et Saporta (1981), bien que l'utilisation du terme *Functional Data Analysis* n'ait pas été relevée avant les travaux de Ramsay et Dalzell (1991). Plus récemment, on pourra trouver de nombreux ouvrages de référence, en particulier ceux de Ferraty et Vieu (2006) et Ramsay et Silverman (2007). Pour une approche plus pratique, on pourra citer Ramsay et al. (2009). Dans ce manuscrit, nous utiliserons la définition de Ferraty et Vieu (2006) :

Une variable aléatoire est dite fonctionnelle si ses valeurs sont dans un espace de dimension infinie. Une observation d'une variable fonctionnelle est appelée donnée fonctionnelle.

Les techniques d'analyse de données fonctionnelles (ADF) sont rencontrées dans de nombreux problèmes pratiques, et sont fréquemment employées en amont d'algorithmes d'apprentissage pour préparer, débruiter et caractériser un signal dans lequel on souhaite rechercher des motifs, par exemple en médecine, pour la classification de signaux image d'électro-encéphalogramme (Flamary, 2011), en finance et marketing pour la prédiction de l'évolution du prix des billets d'avion (Wohlfarth, 2013), en sécurité routière avec l'analyse de trajectoires à risque pour des véhicules légers (Koita et al., 2013) ou des deux-roues motorisés (Attal, 2015), en prévision du trafic routier (Loubes et al., 2006) ou encore en matière de renseignement, avec la reconnaissance automatique de véhicules terrestres à partir de signaux acoustiques (Choe et al., 1996).

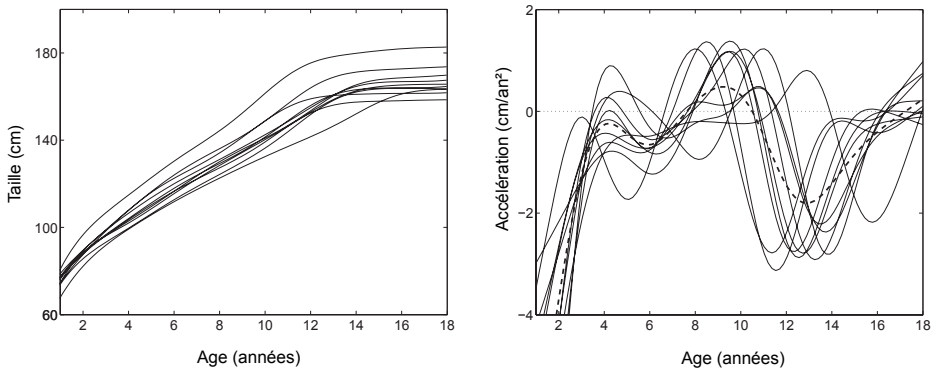


FIGURE 1.10 – Exemple classique d'analyse de données fonctionnelles tiré de Ramsay et Silverman (2007), avec l'étude de la croissance de 10 sujets féminins jusqu'à l'âge de 18 ans, en cm (à gauche) et en cm/an^2 (à droite). En ADF, la clé réside souvent dans une représentation adéquate du jeu de données.

Dans ce manuscrit, nous nous appuierons régulièrement sur les travaux de thèse de Gregorutti (2015), qui a utilisé l'analyse de données fonctionnelles pour prédire le risque d'atterrissage long à partir de données acquises par l'enregistreur de vol au cours de la phase d'approche des avions. La stratégie générique consiste à considérer les profils de vitesse GPS issus de véhicules traceurs, comme des éléments de l'espace de Hilbert des fonctions de carré intégrable $L^2([0,1])$ muni du produit scalaire usuel $\langle .,. \rangle$, puis à choisir une base de fonctions orthogonales $\{\varphi_i\}_{i\in\mathbb{N}}$ sur laquelle projeter les données fonctionnelles :

$$\forall\ x \in [0,1] \qquad v(x) = \sum_{i=1}^{m} \langle v, \varphi_i \rangle \varphi_i(x) + \varepsilon_m(x), \tag{1.8}$$

où v est un profil de vitesse, $\varepsilon_m(x)$ représente l'erreur commise par la troncature de la série à l'ordre m, et où les fonctions de base sont indicées dans l'ordre décroissant du niveau d'information porté par les coefficients. L'espace $L^2([0,1])$ est séparable (et de dimension infinie) donc isomorphe à l'espace des suites $l^2(\mathbb{N})$. En conséquence, on pourra identifier un profil v à la suite des coefficients de la décomposition 1.8, qui pourra être passée en entrée d'algorithmes d'apprentissage classiques (moyennant une erreur d'approximation, dépendant de l'ordre m de la troncature), permettant ainsi de se ramener au cadre de l'apprentissage en dimension finie.

Nous adapterons cette approche pour classifier des données composées d'un ensemble de courbes, en nous basant sur la modélisation fonctionnelle des profils de vitesse, proposée par Andrieu (2013).

1.3.3 Map inference par apprentissage

Si la plupart des tâches de map inference étaient originalement traitées à l'aide d'algorithmes déterministes, ou à l'aide d'approche probabilistes classiques, il y a quelques années Liu et al. (2012) ont proposé une batterie d'indicateurs numériques permettant de mesurer la qualité de reconstruction d'une cartographie routière à partir de traces GPS, ouvrant ainsi la voie à une résolution du problème par apprentissage automatique (Biagioni et Eriksson, 2012a). Par la suite, Van Winden et al. (2016) ont testé plusieurs algorithmes d'apprentissage pour inférer des attributs du réseau routier, et ont mis en évidence l'adéquation des Machines à Vecteur de Support (SVM) pour l'estimation des limitations de vitesse. Dans le même registre, Dabiri et Heaslip (2018) proposent d'utiliser un réseau de neurones convolutionnel (CNN) pour l'inférence des modes d'accessibilité par type de véhicule à partir de paramètres cinématiques extraits de trajectoires GPS.

Pour la détection et la localisation de la signalisation routière, on pourra citer deux travaux particuliers :

Wang et al. (2017) utilisent un jeu de données de trajectoires GPS pour détecter et localiser les lignes d'arrêt associées à des feux tricolores. Dans un premier temps, les carrefours sont extraits par analyse de l'entropie des caps de chaque trace passant sur une zone donnée. Dans un seconde temps, les points d'arrêt des véhicules sont extraits par seuillage sur la vitesse, puis modélisés comme un processus stochastique ponctuel, distribué suivant un mélange de gaussiennes, dont les composantes sont extraites à l'aide d'une

recherche exhaustive de l'espace des solutions. La i-eme gaussienne du processus, représente la distribution des arrêts d'un véhicule situé en i-eme position dans la file d'attente des véhicules en amont du feu. La position de la ligne d'arrêt est alors déterminée à partir des paramètres de la première gaussienne. Les résultats relevés par les auteurs affichent une précision sub-métrique. Cependant, la méthode proposée fait l'hypothèse que toutes les intersections sont contrôlées par des feux tricolores, ce qui est peut-être vérifiable dans les grandes agglomérations nord-américaine, mais est difficilement transposable aux villes européennes. À l'inverse, cette méthode ne permet pas de détecter les feux tricolores situés hors carrefour, comme par exemple les feux associés à des passages piétons.

Le modèle de mélange gaussien utilisé par Wang et al. (2017) peut être considéré comme une méthode d'apprentissage non-supervisée. Plus récemment Munoz-Organero et al. (2018) ont proposé d'utiliser des algorithmes d'apprentissage supervisé pour détecter en temps réel les différents éléments de la signalisation routière à partir de signaux de vitesse et d'accélération, estimés par différenciation à partir des positions GPS. Bien que retournant de très bons résultats, cette méthode montre ses limites sur la détection des feux tricolores, notamment en comparaison du cas des carrefours giratoires et des intersections. D'autre part, une extension naturelle de ce travail serait de relâcher la contrainte de temps réel pour combiner les informations issues de plusieurs véhicules ayant circulé sur le même axe. C'est ce qui est proposé dans la suite de ce manuscrit.

1.3.4 Organisation du manuscrit et contributions

Dans cette section, nous donnons un bref descriptif des problématiques abordées dans les chapitres suivants de cette thèse.

Chapitre 2 : Méthodes et algorithmes pour le pré-traitement des trajectoires GPS

Ce premier chapitre vise à présenter les techniques permettant de préparer les données, en particulier pour le recalage des trajectoires GPS sur le réseau routier (map-matching) ainsi que pour la construction, l'interpolation et le filtrage de profils de vitesse. Nous nous appuyons sur des méthodes classiques de la littérature, que nous particularisons, ou complétons pour gérer plus spécifiquement les problèmes rencontrés dans nos jeux de données. Nous y menons également une étude théorique sur l'impact de la qualité géométrique du réseau routier de référence sur les opérations de map-matching.

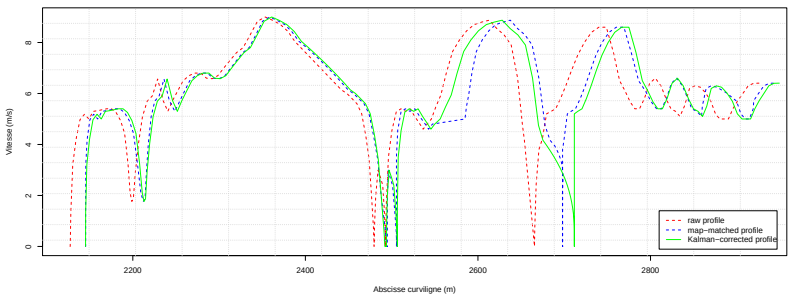


FIGURE 1.11 – Exemple de pré-traitements sur un profil de vitesse GPS : recalage sur un réseau routier de référence et filtrage statistique.

Chapitre 3 : Comparaison des approches image et fonctionnelle en conditions expérimentales

Dans ce chapitre, après une introduction aux méthodes d'apprentissage, on utilise des données collectées dans le cadre d'une expérimentation destinée à évaluer l'impact des consignes d'éco-conduite sur le profil de consommation et la sécurité. Les données ont été acquises avec des récepteurs GPS de bonne qualité. L'objectif de l'étude est de concevoir un algorithme d'apprentissage permettant de détecter les feux triocolores dans un sensemble de fenêtres glissantes. Deux approches principales sont testées pour représenter les données en amont des algorithmes d'apprentissage : une approche image, où l'ensemble des profils de vitesse est considérée comme un image raster dans laquelle chaque pixel représente la densité des profils en un lieu et à une vitesse donnés. Dans une seconde approche, nous adoptons une modélisation fonctionnelle des profils de vitesse, permettant ainsi de représenter finement les données, de manière parcimonieuse et expressive. Les résultats montrent que l'approche fonctionnelle semble plus performante, en termes de qualité de détection, mais également en temps de calcul.

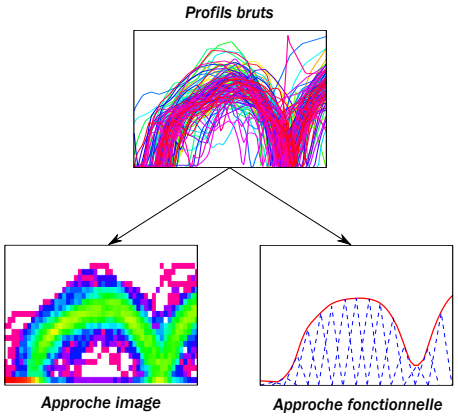


FIGURE 1.12 – Deux approches pour la modélisation des profils de vitesse sur des données expérimentales (chapitre 3).

Par la suite, nous testons cette approche sur d'autres éléments de la signalisation routière : les passages piétons et les stops, et nous menons quelques analyses de sensibilité des résultats obtenus, relativement au nombre de profils de vitesse disponibles, à la précision des récepteurs GPS (en position et en vitesse) ainsi qu'à la fréquence d'acquisition des données.

Chapitre 4 : Étude du potentiel des méthodes d'apprentissage sur un cas opérationnel

Les travaux de ce chapitre correspondent à une période de mobilité internationale de 4 mois, effectuée au sein du laboratoire CSIS de l'Université du Tokyo. L'objectif principal était des tester les algorithmes développés sur un cas de données réelles. Nous ajoutons également l'objectif de la régression de la position des feux détectés. Les résultats obtenus ont permis de mettre en évidence la difficulté des algorithmes d'apprentissage à extraire de l'information à partir de données bruitées. D'autre part, nous y avons appris que le passage de trajectoires aquises sur un circuit en boucle à des trajectoires circulant librement sur le réseau n'est pas trivial, et pose de nombreux problèmes notamment au niveau de la définition des instances individuelles.

Cette section contient également quelques tests préliminaires d'exploitation de l'auto-corrélation spatiale des variables à inférer sur le réseau routier, approche que nous explorerons plus en profondeur dans le chapitre 5.

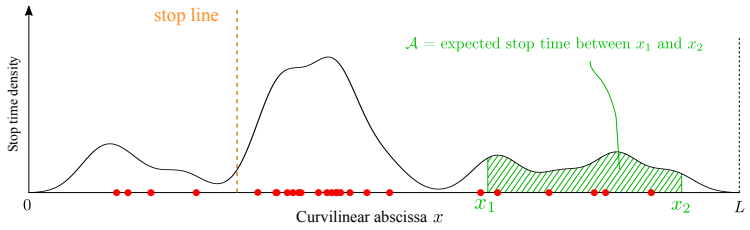


FIGURE 1.13 – Densité saptio-temporelle des arrêts de véhicules.

Chapitre 5 : Approches globales : réseaux de neurones artificiels et apprentissage structuré

L'objectif de ce dernier chapitre est de tester une approche image pour une détection globale de l'ensemble des feux tricolores sur une zone donnée, en utilisant un réseau de neurones convolutionnels. Les résultats principaux montrent que la méthode est intéressante, en particulier de par sa simplicité au niveau du pré-traitement des données, mais soulève des difficultés au niveau de l'extraction finale des positions des feux tricolores à partir des cartes de probabilités estimées.

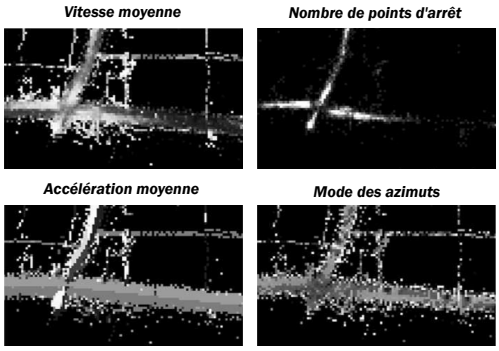


FIGURE 1.14 – Champs scalaires de descripteurs pour la détection de feux tricolores par réseau de neurones convolutionnels.

Enfin, dans une partie à vocation essentiellement exploratoire, nous tentons d'utiliser des modèles probabilistes graphiques pour exploiter l'auto-corrélation entre les différents axes voisins dans un graphe routier, dans le but d'améliorer la performance de détection des algorithmes développés dans les chapitres 3 et 4 notamment.

Chapitre 2

Méthodes et algorithmes pour le pré-traitement des trajectoires GPS

Sommaire

Dans ce chapitre, nous présentons l'ensemble des algorithmes développés tout au long de ce manuscrit, pour préparer les traces GPS en vue de les transformer en instances pouvant être traitées par des algorithmes d'apprentissage. La plupart des techniques décrites sont génériques et peuvent ainsi être utilisées pour le pré-traitement de trajectoires GPS dans de nombreux cadres d'application. De ce fait, ce chapitre constitue une partie technique du manuscrit, qui pourra être sautée en première lecture, mais qu'il nous a néanmoins paru indispensable d'intégrer pour faciliter la reproductibilité des travaux.

2.1 Les jeux de données

Puisque certains des processus listés dans les sections suivantes vont varier en fonction du type des données à traiter, nous commençons par une description succincte des deux jeux de données principaux que nous avons utilisés.

2.1.1 Données expérimentales : projet ecoDriver

Dans le cadre du projet européen *EcoDriver* visant à évaluer l'impact des consignes d'écoconduite sur le comportement des automobilistes (Cheng et al., 2013), 30 conducteurs différents ont parcouru plusieurs fois un même circuit à bord de véhicules munis de capteurs. Au total, 87 paramètres ont été mesurés et enregistrés, tels que la position géographique, la vitesse instantanée, le niveau de carburant restant ou encore le rayon de braquage du volant (suivant une logique de collecte de données xFCD).

L'expérimentation a été menée sur un circuit de 25 km dans la commune de Versailles et ses environs, contenant une partie urbaine et une partie plus rurale avec également un tronçon d'autoroute. Pour chaque conducteur, le circuit a été parcouru entre 4 et 6 fois (de manière non successive), si bien que le nombre total de traces disponibles s'élève à 170, correspondant à un total d'environ 150 heures de conduite, et 5.5 millions d'enregistrements (soit environ 2 Go de données). Les conducteurs ont utilisé le même modèle de véhicule, une Renault Clio III, équipée d'un data logger connecté au bus CAN du véhicule et à l'antenne d'un recepteur Garmin GPS 16x LVC.

Dans cette thèse, nous avons utilisé ces données principalement pour comparer différentes approches d'apprentissage sur un jeu de données expérimentales de bonne qualité (en particulier sur les mesures de vitesse GPS) et avec un protocole opératoire conduisant les véhicules à circuler sur un circuit en boucle prédéfini et soigneusement choisi de sorte à contenir une densité suffisamment élevée d'élements de l'infrastructure routière.

2.1.2 Données observationnelles fournies par Navitime

NAVITIME JAPAN [1] est une compagnie japonaise privée oeuvrant dans le développement de technologies liées à la navigation, dans le but de proposer aux conducteurs d'automobiles (et plus généralement à toute personne en déplacement) un certain nombre de services, tels que des conseils d'itinéraires et des indications de temps de parcours.

Les jeux de données qui ont été mis à notre disposition pour la durée de ces travaux de thèse couvrent une période d'un mois, en octobre 2015, sur deux villes de taille moyenne du Japon : Mitaka et Tsukuba. Chaque enregistrement (nominalement 1 par seconde) contient un identificateur unique (alétatoirement modifié toutes les 24 heures), un timestamp et les coordonnées géographiques (longitude et latitude) du récepteur GPS au moment de l'acquisition. Chacun des deux jeux de données contient de l'ordre de 25 000 trajectoires, pour un total approximatif de 13 000 heures de conduite et 30 millions d'enregistrements (soit 1 Go de données).

1. http ://corporate.navitime.co.jp/en

2.2 Généralités sur les capteurs

Cette section vise à donner quelques éléments sur les capteurs principaux des véhicules traceurs et à étudier l'ordre de précision typique que l'on peut en attendre. Nous nous restreindrons à trois paramètres principaux : la position, la vitesse instantanée (toutes deux mesurées par le GPS) et la distance cumulée depuis le départ (mesurée par l'odomètre du véhicule).

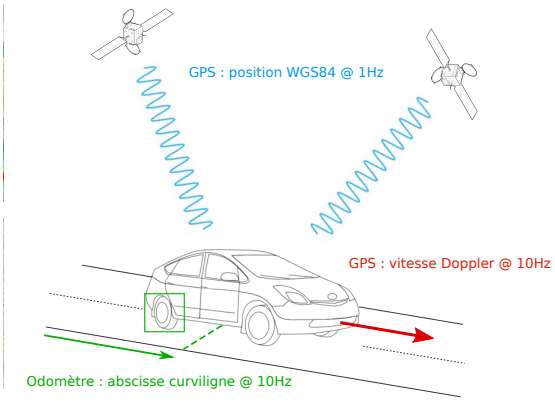


FIGURE 2.1 – Variables de position et de vitesse mesurées au cours de l'expérience *eco-Driver*

2.2.1 Les systèmes GNSS

Le système GPS (pour *Global Positioning System*), lancé en 1978 et déclaré pleinement opérationnel en 1995, a été mis au point par le département de la défense américain, et appartient aujourd'hui à un système plus large dit de GNSS (pour *Global Navigation Satellite System*) dont il était le seul représentant jusqu'en 2007, mais auquel il faut à présent ajouter les systèmes GLONASS (Russie), Galileo (Union Européenne) et bientôt le système chinois Beidou qui devrait être opérationnel en 2020 (Bosser, 2011).

En pratique, ces différents systèmes ne sont pas exclusifs, et la plupart des récepteurs GNSS modernes sont capables d'effectuer leurs calculs de positionnement en tenant compte des signaux reçus par les satellites GLONASS et Galileo qui viennent densifier la constellation GPS. Un système GNSS est déclaré opérationnel quand il comprend à lui seul suffisamment de satellites pour assurer une couverture totale du globe terrestre. La constellation de satellites GNSS totale compte à l'heure actuelle, en plus des 30 satellites américains, 24 satellites russes et 14 satellites européens. Leurs orbites sont situées à environ 20 000 km d'altitude, suivant un plan d'inclinaison calculé de sorte à fournir une couverture optimale.

2.2.2 Mesure de la position

La mesure de la position du récepteur est effectuée à partir de l'observation des distances le séparant des satellites (nous verrons par la suite que 4 satellites sont nécessaires en

pratique). La précision de localisation dépend donc directement de la qualité des mesures de distance, elle même conditionnée par la précision des mesures du temps. À la vitesse de la lumière, une erreur de mesure de 1 μs, entraine une déviation de 300 m. La figure 2.2 illustre le principe de la mesure de code [2] utilisée dans les récepteurs grand public.

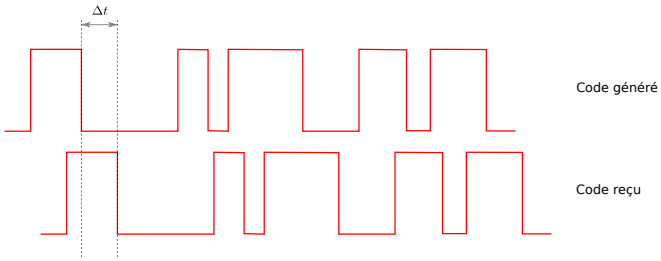


FIGURE 2.2 – Mesure du décalage de temps entre les PRN répliqué et réçu par le récepteur.

Notons $\mathbf{p_r} = (x_r, y_r, z_r)$ la position du récepteur GPS dans le référentiel géocentrique. Les positions des satellites sont indexées par le numéro i du satellite : $\mathbf{p_i} = (x_i, y_i, z_i)$. L'écart de temps mesuré Δt_i entre le signal répliqué et le signal reçu du i^{eme} satellite multiplié par la vitesse de la lumière donne une mesure approximative de la distance parcourue par le signal : $\rho_i = c\Delta t_i$. Cette quantité est liée aux coordonnées du récepteur par l'expression :

$$\rho_i = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2 + (z_r - z_i)^2} + c\delta t_r, \tag{2.1}$$

où le terme δt_r dénote l'erreur de l'horloge à quartz du récepteur. Le satellite est quant à lui muni d'une horloge atomique, dont les dérives sont corrigées en temps réel par le segment de contrôle. On peut donc faire l'hypothèse pour simplifier que les horloges des satellites sont toutes synchronisées sur le temps GPS.

Les positions des satellites (x_i, y_i, z_i) étant connues via les messages de navigation transmis en temps réel, on se trouve face à une équation à quatre inconnues $x_r, y_r, z_r, \delta t_r$. Quatre satellites sont donc nécessaires pour localiser le récepteur. En pratique, tout se passe comme si on souhaitait localiser un point dans l'espace spatio-temporel à 4 dimensions. Si le récepteur a exactement 4 satellites en vue, le système d'équations linéarisées est en général inversible, la solution s'exprimant alors naturellement par inversion matricielle. La constellation GPS contenant à elle seule 30 satellites, il est fréquent, même en milieu urbain, de recevoir un signal en provenance de plus de 4 satellites simultanément (Lee et al., 2008). Les récepteurs GPS proposent alors une détermination de la position par moindres carrés, permettant de résoudre un système d'équations sur-contraint, augmentant ainsi la robustesse de la solution trouvée (Sillard, 2001).

La position obtenue par le récepteur comporte une marge d'erreur que l'on peut imputer

2. Il existe également un procédé de mesure directement sur la phase de la porteuse, dont la mise en œuvre est plus complexe, mais qui permet moyennant une prise en compte fine des erreurs externes, et généralement avec l'aide de stations de base connues aux environs, d'obtenir des précisions de positionnement différentiel de l'ordre du demi-centimètre. Cette fonctionnalité n'est en général accessible que sur des GPS professionnels haut de gamme.

à divers facteurs, parmi lesquels on citera en particulier :

- Le bruit de mesure sur le décalage temporel du code. Les récepteurs actuels permettent de réaliser une mesure de décalage de l'ordre du centième de la longueur d'un bit du code PRN. Pour un cycle complet de 1023 bits transmis en 1 ms, cette précision correspond à une incertitude sur la pseudo-distance de l'ordre de 3 m.

- L'erreur de synchronisation de l'horloge des satellites (5 ns, soit 1.5 m).

- L'erreur sur les paramètres de correction ionosphérique et troposphérique (dépendante de la qualité et de la finesse du modèle utilisé).

- L'erreur sur la position estimée du satellite à l'instant de la mesure, généralement transmise à l'utilisateur via le message de navigation sous forme d'éléments képlériens. L'observation temps réel de ces éléments par le segment de contrôle est entachée d'une erreur que l'on peut réduire en attendant la diffusion des éphémérides précises, disponibles en général quelques jours à quelques semaines après la période d'observation concernée. L'erreur type de la position donnée par le message de navigation est de l'ordre de 1 m.

- Des erreurs liées aux multitrajets des signaux GPS (reflections sur l'environnement).

- Des erreurs externes negligeables dans notre cas d'application, tels que des effets relativistes au second ordre, des décalage de centre de phase des antennes ou encore des phénomènes géophysiques complexes à modéliser tels que les marées terrestres ou la surcharge océanique.

La table 2.1 récapitule les sources d'incertitudes pour un récepteur grand public, avec l'erreur type repercutée sur le calcul de la position finale (Kaplan et Hegarty, 2005).

Source d'erreur	**Type d'erreur**	**Précision** σ (m)
Constellation GPS	Erreur d'horloge des satellites	1.1
	Erreur d'éphémérides des satellites	0.8
Recepteur	Délai ionosphérique	7.0
	Délai troposphérique	0.2
	Bruit du récepteur et résolution	0.1
	Multi-trajets	0.2
Total		7.1

TABLE 2.1 – Sources d'erreurs GPS les plus courantes et impacts sur la mesure de pseudo-distances. La précision finale du positionnement est fonction de la configuration géométrique des satellites à l'instant de la mesure. En conditions nominales, cette précision est de l'ordre de 1.6 fois l'erreur typique sur les mesures de pseudo-distances.

Il faut noter cependant que l'erreur ionosphérique possède une auto-corrélation temporelle relativement élevée, permettant en pratique d'obtenir un écart-type d'erreur totale

plus faible que celui donné ci-dessus (*cf* section 2.4.3.2). En revanche, l'influence exacte des multi-trajets est difficilement prévisible en milieu urbain, en particulier lorsque la densité de bâti est élevée (phénomène de canyon urbain). Pour plus d'informations sur le sujet, nous renvoyons le lecteur au travail de Lee et al. (2008). Dans ce travail de thèse, sauf indication contraire, nous considérerons donc que la mesure des positions fournies par le GPS sont entachées d'une erreur gaussienne d'écart-type 7 m.

2.2.3 Mesure de la vitesse instantanée

Lorsque l'on dispose d'une suite de mesures GPS (x_n, y_n), une estimation simple de la vitesse du mobile découle naturellement de la différenciation des positions successives :

$$\hat{v}_n = \frac{\sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2}}{\Delta t}, \tag{2.2}$$

où Δt représente l'écart de temps entre les mesures.

Supposons sans perte de généralité qu'une mesure soit prise toutes les secondes et que le mobile se déplace entre deux points (x_1, y_1) et (x_2, y_2), dont les positions ont été estimées par GPS avec une matrice de covariance :

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} & 0 & 0 \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 & 0 & 0 \\ 0 & 0 & \sigma_{y_1}^2 & \sigma_{y_1 y_2} \\ 0 & 0 & \sigma_{y_1 y_2} & \sigma_{y_2}^2 \end{bmatrix} \tag{2.3}$$

L'estimateur de la vitesse s'exprime : $f(x_1, x_2, y_1, y_2) = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2}$. Par linéarisation au premier ordre, on obtient une approximation de la variance associée :

$$\sigma_v^2 = \boldsymbol{J}\mathbf{\Sigma}\boldsymbol{J}^T = \sum_i \Big(\frac{\partial f}{\partial \mathbf{x}_i}\Big)^2 \sigma_{\mathbf{x}_i}^2 + \sum_i \sum_{j \neq i} \Big(\frac{\partial f}{\partial \mathbf{x}_i}\Big)\Big(\frac{\partial f}{\partial \mathbf{x}_j}\Big)\sigma_{\mathbf{x}_i \mathbf{x}_j}, \tag{2.4}$$

où $\boldsymbol{J}$ désigne la matrice jacobienne de l'application f prise en $\mathbf{x} = [x_1, x_2, y_1, y_2]$:

$$\boldsymbol{J}(\mathbf{x}) = \frac{1}{v}[-\Delta x, \Delta x, -\Delta y, \Delta y],$$

en ayant posé les différences : $\Delta x = x_2 - x_1$ et $\Delta y = y_2 - y_1$. On obtient alors :

$$\sigma_v^2 = \Big(\frac{\Delta x}{v}\Big)^2 (\sigma_{x_1}^2 + \sigma_{x_2}^2) + \Big(\frac{\Delta y}{v}\Big)^2 (\sigma_{y_1}^2 + \sigma_{y_2}^2) - 2\Big(\frac{\Delta x}{v}\Big)^2 \sigma_{x_1 x_2} - 2\Big(\frac{\Delta y}{v}\Big)^2 \sigma_{y_1 y_2}. \tag{2.5}$$

Dans le cadre des mesures GPS sur le code dans un environnement homogène en termes de densité de couvert, on peut admettre que les écarts-types d'erreur suivant les axes x et y sont identiques et indépendant du point de mesure : $\sigma_{x_1} = \sigma_{x_2} = \sigma_{y_1} = \sigma_{y_2} = \sigma$ où σ représente l'erreur typique du GPS sur chaque axe, et dépend des spécifications du constructeur. Par exemple, pour un GPS d'erreur typique totale (root mean square error, ou rmse) de 7 m, on aura $\sigma = 7/\sqrt{2} = 4.95$ m. En effet, en supposant les erreurs distribuées suivant une loi normale de moyenne nulle et d'écart-type σ, et en notant X et Y les erreurs sur chaque axe et Z l'erreur totale, on a par linéarité de l'espérance :

$$\mathrm{RMSE}[Z]^2 = \mathbb{E}[Z^2] = \mathbb{E}[X^2] + \mathbb{E}[Y^2] = (\mathbb{E}[X^2] - \mathbb{E}[X]^2) + (\mathbb{E}[Y^2] - \mathbb{E}[Y]^2) = 2\sigma^2.$$

D'où on obtient :

$$\sigma = \frac{\mathrm{RMSE}[Z]}{\sqrt{2}}.$$

On peut également supposer que l'autocorrélation est identique suivant les deux axes et constante : $\sigma_{x_1x_2} = \sigma_{y_1y_2} = \rho$. L'équation 2.5 est alors réduite en :

$$\sigma_v^2 = 2\sigma^2 - 2\rho, \tag{2.6}$$

dans laquelle nous avons éliminé en apparence la dépendance en vitesse. En réalité, la covariance ρ entre deux mesures successives est une fonction sensiblement décroissante des intervalles de temps et de distance les séparant (nous reviendrons sur ce point dans la section 2.4.3.2)

En règle générale, sauf à poser des hypothèses réductrices très contraignantes, il est impossible d'exprimer analytiquement cette covariance, mais on conçoit aisément qu'elle ne peut être négligée, sachant que les deux mesures successives ont été faites dans un environnement similaire avec une configuration géométrique des satellites par rapport au récepteur quasiment identique. Ranacher et al. (2016) proposent une méthode expérimentale pour déterminer une estimation de cette covariance pour un modèle de récepteur donné. En reprenant les résultats de leurs expérimentations pour les vitesses typiques qui nous intéressent (30 km/h) et en considérant que l'écart-type de mesure sur un axe est de l'ordre de 3 m, on obtient $\rho \approx 2.5$ m^2, soit :

$$\sigma(\hat{v}_n) = \sqrt{2 \times (9 - 2.5)} \approx 3.6 \text{ m.s}^{-1}.$$

On remarque que l'imprécision sur la mesure de vitesse obtenue par différenciation des positions mesurées par un GPS classique, est de l'ordre de 13 km/h ce qui, en particulier à faible vitesse, représente une erreur conséquente.

Pour remédier à ce problème, certains récepteurs GPS estiment la vitesse instantanée de déplacement à l'aide du décalage Doppler (Petovello, 2015), qui stipule que la fréquence d'un

signal réçu dépend de la vitesse radiale entre la source du signal et la position du récepteur.

D'une manière similaire à celle employée pour le calcul de la position, on peut poser un système d'équations dont les inconnues à estimer sont les vitesses du récepteur suivant les trois axes $\dot{x}_r$, $\dot{y}_r$, $\dot{z}_r$ et sa dérive d'horloge $\dot{t}_r$. La fréquence d'émission des signaux f_{sat} est connue et identique pour tous les satellites. Les positions x_i, y_i, et z_i des satellites sont connues grâce aux éléments képleriens transmis via le message de navigation et le récepteur en déduit directement les vitesses $\dot{x}_i$, $\dot{y}_i$, et $\dot{z}_i$ (Remondi, 2004). En supposant que la position du récepteur a été calculée au préalable, il en découle par la même occasion les distances aux satellites $\hat{r}_i$ et les valeurs des composantes des vecteurs unitaires récepteur-satellites : a_x^i, a_y^i et a_z^i. À nouveau, la résolution du système est effectuée par la technique des moindres carrés (Sillard, 2001).

L'erreur de mesure du décalage Doppler est en général de l'ordre du Hz, impliquant un écart-type sur la vitesse du récepteur estimé à quelques cm/s (Chalko, 2009; Aminian et al., 2010), ce qui représente un gain considérable par rapport à la différenciation brute des positions observées (équation 2.6). Pour la suite du travail, lorsque cela sera nécessaire, nous considérerons que les mesures des vitesse fournies par le GPS sont entachées d'une erreur gaussienne d'écart-type 0.1 $\mathrm{m.s}^{-1}$.

2.2.4 Mesure de la distance cumulée

Les mesures de distance cumulée dont on dispose dans les données sont directement issues de l'odomètre du véhicule, dont la tâche consiste à convertir les tours de roues en impulsions éléctriques comptabilisées. Cette mesure peut donc être assimilée à peu de chose près à l'intégration sur le temps des valeurs fournies par le compteur de vitesse du tableau de bord.

Le décompte du nombre de tours de roue est fiable, l'intégralité du bruit de mesure provient de la méthode de conversion de ce décompte en une valeur de distance parcourue. Deux sources principales d'erreur peuvent être identifiées.

- La distance cumulée est obtenue en multipliant le décompte des tours par la circonférence de la roue. Cela suppose une mesure fiable et constante du diamètre de la roue, dont on sait très bien que celui-ci est en pratique sensible aux fluctuations de conditions de température et de pression.

- Le nombre de tours de roue obtenus par l'odomètre ne correspond à la distance cumulée réellement parcourue uniquement dans le cas où le déplacement du véhicule s'effectue en roulement sans glissement. Dans le cas contraire, la mesure de l'odomètre sera perturbée par une erreur aléatoire de moyenne non nulle (positive) et de variance a priori fonction des conducteurs, des conditions météorologiques et du circuit parcouru. Cette seconde erreur est en particulier beaucoup plus difficile à modéliser et à corriger.

L'erreur typique associée à un odomètre de véhicule est de l'ordre de 1 à 2%, ce qui, pour un circuit de 25 km, correspond à une erreur de l'ordre de 250 à 500 m

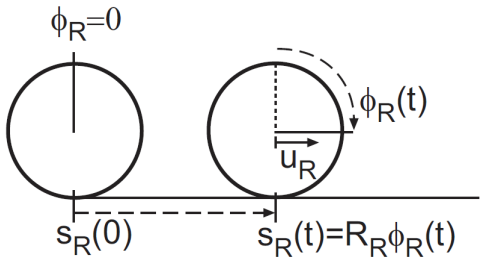


FIGURE 2.3 – Principe de la mesure odométrique

2.3 Modélisation fonctionnelle des profils de vitesse

Dans cette section, nous discutons de la modélisation et de la construction numérique des profils de vitesse GPS issus de véhicules traceurs. Nous supposons avoir à disposition une séquence (chronologiquement ordonnée) de n enregistrements $\{(x_i, y_i, t_i)\}_{i=1..n}$, où $(x_i, y_i) \in \mathbb{R}^2$ désigne la position planimétrique du véhicule à l'instant $t_i \in \mathbb{N}$ (pour simplifier, on suppose que les timestamps ont été préalablement convertis en nombres de secondes écoulées depuis une date arbitrairement choisie [3]).

Nous avons vu précédemment dans le chapitre 1 qu'une description fonctionnelle des objets à traiter permet bien souvent de contourner les limites de la statistique multivariée dans l'analyse des données de grande dimension. C'est le cas ici, avec des trajectoires qui peuvent être échantillonnées jusqu'à 1 Hz (voire plus dans le cas de fusions multi-capteurs), l'objectif in fine étant de représenter chaque trajectoire comme un élément d'un même espace fonctionnel, permettant ainsi de les analyser simultanément et de les comparer.

2.3.1 Les différents types de représentations

Trois variables interdépendantes entrent en jeu dans une trajectoire : le temps, la distance parcourue et la vitesse, ce qui permet une représentation de la trajectoire dans trois espaces de profils différents (Andrieu, 2013).

- L'espace **temps** × **distance** : on parle de *diagramme temps-espace* (*time space diagram*), un outil de visualisation et d'analyse fréquemment utilisé dans les problématiques liées au trafic (Anwar et al., 2014; Protschky et al., 2015).

  Dans cet espace, les profils sont des fonctions monotones croissantes. La figure 2.4 donne un exemple schématique de diagramme temps-espace.

- L'espace **temps** × **vitesse** des *profils temporels de vitesse*, qui représente l'évolution de la vitesse instantanée du mobile en fonction du temps écoulé. Cette représentation est certainement la plus intuitive et la plus directe à calculer. Elle est particulièrement utile lorsque la cinématique du véhicule est au centre de l'étude, par exemple dans des travaux de détection de situations de conduite anormales. Wolfermann et al.

3. En général le 1er Janvier 1970 dans la norme POSIX

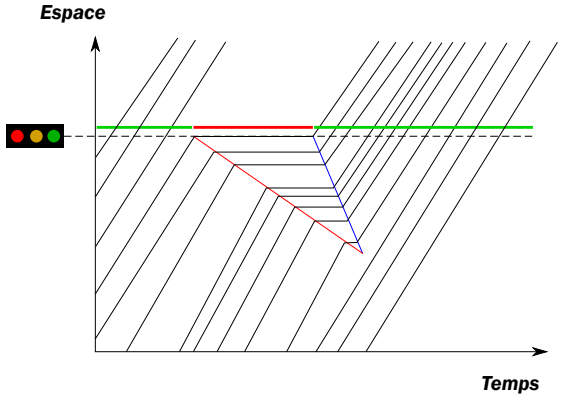


FIGURE 2.4 – Représentation d'un ensemble de trajectoires GPS dans l'espace temps × distance. La ligne rouge oblique représente l'accumulation des véhicules en amont du feu. La ligne bleue représente la propagation de l'onde de démarrage après le passage du feu au vert. Les pentes de toutes les droites correspondent aux vitesses (vitesses de déplacement pour les véhicules et vitesse de propagation des fronts pour les extrémités de la file d'attente).

(2011) par exemple, utilisent ce profil pour caractériser le comportement des véhicules au niveau des virages d'intersection. Cependant, le temps écoulé est en général propre à chaque véhicule, et elle ne permet donc pas une représentation simultanée des profils de vitesse de plusieurs véhicules, quand bien même ces véhicules auraient suivi le même parcours.

- L'espace **distance** × **vitesse** des *profils spatiaux de vitesse*, qui représente l'évolution de la vitesse instantanée du mobile en fonction de la distance parcourue. Contrairement au cas du profil temporel, l'abscisse du profil spatial correspond au parcours effectué, et la représentation simultanée de plusieurs véhicules ayant suivi le même circuit possède à présent une signification. C'est le profil le plus classiquement utilisé dans les études cherchant à établir un lien entre l'infrastructure routière et les véhicules, comme par exemple dans Moreno et García (2013).

La figure 2.5 donne un exemple de représentation des trois types de profils sur une trace GPS de randonneur. On remarque en particulier que les profils temporel et spatial ont une allure similaire avec une différence notable : le profil temporel est plus contracté (horizontalement) dans les zones à forte vitesse, et à l'inverse plus dilaté dans les zones à faible vitesse. Dans le cas extrême d'une vitesse nulle, un point d'arrêt se manifeste par un point de rebroussement de première espèce dans le profil spatial. Dans le profil temps × vitesse, le point d'arrêt perd son caractère ponctuel, et se compose d'un segment de droite horizontal, confondu avec l'axe des abscisses, et de longueur égale à la durée de l'arrêt.

En pratique, disposant de 2 des 3 variables que sont le temps, la distance parcourue et la vitesse, on peut généralement reconstruire la troisième. Par exemple, disposant de la suite de points (t_i, x_i) d'un diagramme temps-espace, on reconstruit aisément la vitesse par dérivation numérique. Par exemple, avec une différence finie avant :

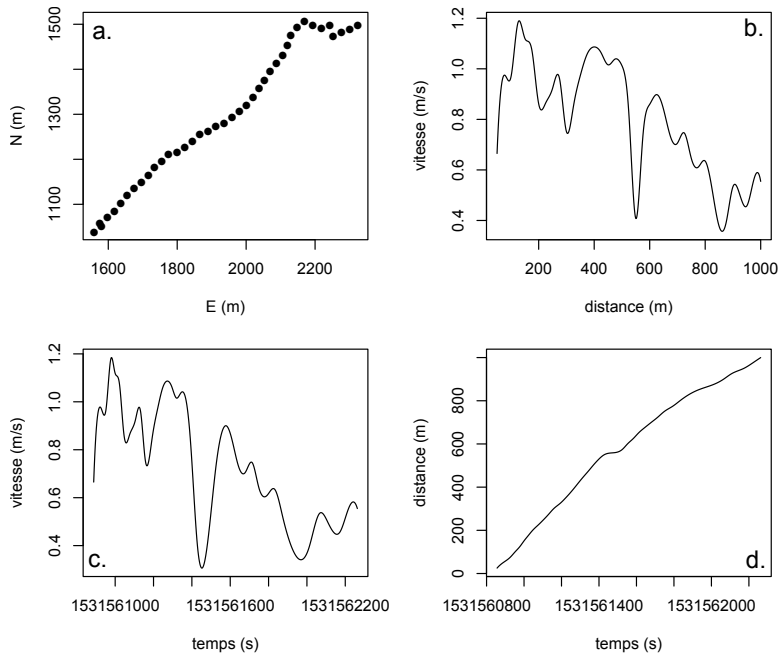


FIGURE 2.5 – a. Trajectoire GPS dans l'espace géographique. b. Profil spatial de vitesse. c. Profil temporel de vitesse. d. Diagramme temps-espace.

$$v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i}. \tag{2.7}$$

Cette opération est possible car on suppose n'avoir qu'une seule mesure à chaque instant, garantissant ainsi la stricte positivité du dénominateur.

Disposant d'un profil temporel de vitesse (t_i, v_i) la variable abscisse curviligne x est reconstruite par intégration numérique :

$$x_i = x_{i-1} + v_i(t_i - t_{i-1}). \tag{2.8}$$

Enfin, lorsque seul le profil spatial (x_i, v_i) est donné, la variable temps se reconstruit par :

$$t_i = t_{i-1} + \frac{x_i - x_{i-1}}{v_i}. \tag{2.9}$$

à condition que $v_i \neq 0$. Ceci montre la perte d'information à vitesse nulle du profil spatial par rapport au profil temporel et au diagramme temps-espace. Cette limite est très clairement mise en avant dans la modélisation d'Andrieu (2013) :

Définition 2.1. *Soit $F : \mathbb{R} \to \mathbb{R}^+$ une fonction croissante (au sens large). On appelle inverse généralisée de F la fonction $F^{-1} : \mathbb{R}^+ \to \mathbb{R} \cup \{-\infty\}$ définie par :*

$$F^{-1}(t) = \inf_{x \in \mathbb{R}} \{x \mid F(x) \geqslant t\}.$$

Définition 2.2. *Soient $a, b \in \mathbb{R}$. Soit v une fonction définie sur $[a, b]$ et à valeur dans $\mathbb{R}^+$. On dit que v est un profil de vitesse, si et seulement s'il existe une valeur $T \in \mathbb{R}^+$, ainsi qu'une fonction de classe $\mathcal{C}^2$ croissante au sens large $F : [0, T] \to [a, b]$ telle que $F(0) = a$ et :*

$$v(x) = F' \circ F^{-1}(x),$$

où $F'(t) = \frac{dF(t)}{dt}$ représente la dérivée de F par rapport à t et F^{-1} son inverse généralisée.

Cette définition théorique permet de mettre en évidence d'un point de vue analytique, les liens entre les trois types de représentation de trajectoires. La fonction croissante F correspond à une courbe dans le diagramme temps-espace (figure 2.4). Les intervalles de temps où F n'est pas strictement croissante correspondent aux arrêts du véhicule. Réciproquement, l'inverse généralisée F^{-1} permet de convertir des distances en temps. Notons qu'elle est discontinue au niveau de ces mêmes points d'arrêts. La conversion du temps en vitesse correspond classiquement à la dérivée temporelle F'. Par transitivité, le passage des distances en vitesse, que modélise le profil spatial, s'exprime par la composition de fonction $F' \circ F^{-1}$. Ces relations sont résumées sur la figure 2.6.

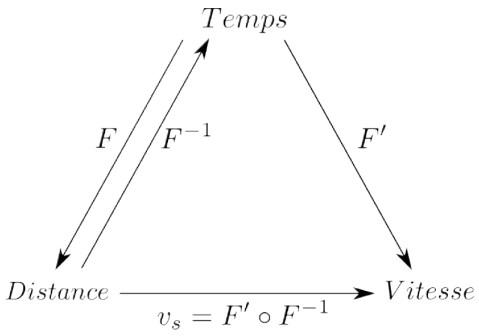


FIGURE 2.6 – Relations analytiques entre les différents types de représentation de la trajectoire d'un véhicule. D'après Andrieu (2013).

La connaissance du diagramme temps-espace est donc équivalente à la connaissance de F, que l'on peut dériver et inverser pour calculer les profils temporel et spatial. Un raisonnement similaire peut être tenu dans le cas où la connaissance initiale porte sur le profil temporel, et donc sur la dérivée F', qui peut être intégrée en F et inversée. En revanche, l'information portée par le profil spatial est moins riche. Déterminer la fonction F à partir du profil v nécessite de résoudre l'équation différentielle du premier ordre :

$$y' = v(y), \tag{2.10}$$

avec la condition initiale $y(0) = a$. Notons que cette caractérisation correspond à une forme fonctionnelle de la relation de passage aux différences finies énoncée dans l'équation 2.8.

Sous certaines conditions de régularité sur la fonction v, le théorème de Cauchy-Lipschitz (Boyer, 2012) nous garantit que 2.10 admet une unique solution maximale sur l'intervalle $[0, T]$ (la propriété d'existence de solutions maximales persiste sous la seule hypothèse de continuité de F, d'après le théorème de Cauchy-Arzela). Malheureusement, un profil spatial de vitesse ne vérifie pas toujours ces conditions en pratique.

Propriété 2.1. Un profil spatial de vitesse s'annule de manière non-dérivable.

Supposons que v soit dérivable en un point $x_0 \in \{x \in [a, b] \mid v(x_0) = 0\}$. Si v est un profil spatial de vitesse, d'après la définition 2.2, il existe une fonction F de classe $\mathcal{C}^2$ sur $[0, T]$ (avec $T \in \mathbb{R}^+$) telle que $v(x) = F' \circ F^{-1}(x)$. Calculons la dérivée de v par rapport à l'abscisse curviligne :

$$\frac{d}{dx}\left[\frac{dF}{dt} \circ F^{-1}(x)\right] = \frac{d}{dF^{-1}}\left[\frac{dF}{dt}\right]\frac{dF^{-1}}{dx} = \frac{d^2F}{dt^2}\frac{dt}{dx}.$$

D'où :

$$\frac{dv}{dx}(x) = \frac{1}{v(x)}\frac{d^2F}{dt^2}[F^{-1}(x)]. \tag{2.11}$$

Or, $v(x_0) = 0$, donc le caclul de dv/dx en x_0 résulte en une quantité infinie ou en une forme indéterminée (dans le cas où la dérivée seconde de F s'annule également). Le profil spatial de vitesse n'est donc pas dérivable sur $H = \{x \in [a, b] \mid v(x_0) = 0\}$.

Plus précisement, on peut montrer moyennant quelques hypothèses supplémentaires, que le profil de vitesse est continu en tout point, et que sa dérivée tend vers l'infini au voisinage d'un point d'arrêt (Andrieu, 2013). Ceci permet d'obtenir une caractérisation des points d'arrêt dans l'espace des profils spatiaux de vitesse, comme étant des points de rebroussement de première espèce, comme illustré sur la figure 2.7.

La fonction v est n'est donc pas lipschitzienne localement au niveau des points d'arrêt, ce qui exclut l'unicité d'une solution maximale, et montre donc que le profil de vitesse est moins informatif que les deux autres types de représentation. Prenons un exemple simple pour illustrer ce point : considérons un mobile se déplaçant suivant une courbe paramétrée par x, avec une accélération constante : $v(t) = t$. La position du mobile s'exprime alors par $x(t) = t^2/2$ et on en déduit immédiatement l'expression du profil spatial de vitesse :

$$v(x) = \sqrt{2x}.$$

Représentons cette trajectoire dans les trois espaces de profils (figure 2.8).

Supposons connaître uniquement le profil spatial de vitesse. La relation 2.10 nous dit que la fonction F caractérisant la représentation dans le diagramme temps-espace est solution de

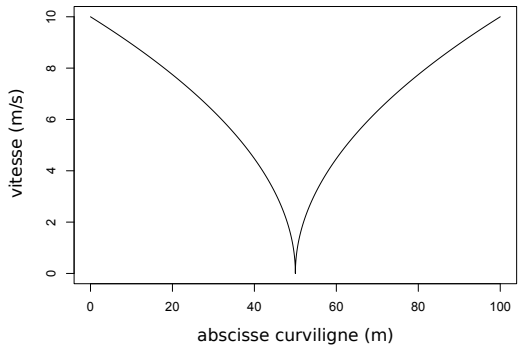


FIGURE 2.7 – Illustration d'un arrêt typique à l'abscisse $x_0 = 50$ dans l'espace des profils spatiaux de vitesse définis sur la fenêtre $[0, 100]$. La fonction analytique choisie ici pour modéliser le profil est : $v(x) = (2|x - x_0|)^{0.5}$.

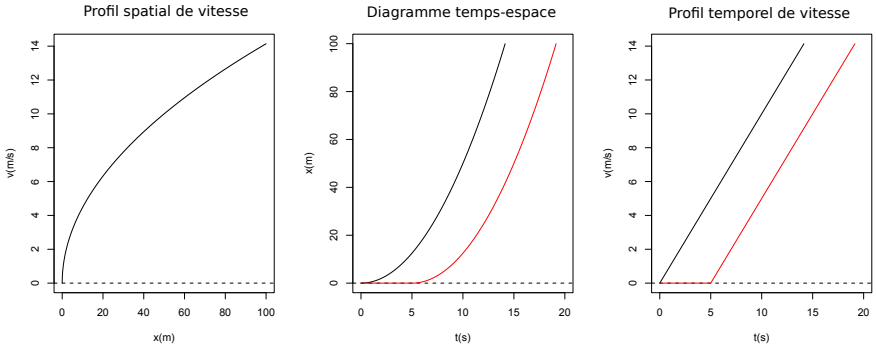


FIGURE 2.8 – Illustration de la non-unicité de la reconstruction à partir d'un profil spatial pour un mobile à vitesse linéaire. En rouge : reconstruction alternative des profils temps × espace et temps × vitesse à partir du profil spatial.

l'équation différentielle : $y' = \sqrt{2y}$. Une solution naturelle de cette équation est la fonction $y_0(t) = \frac{1}{2}t^2$, mais on peut montrer également que toute fonction y_d de la forme ci-dessous (avec $d \in \mathbb{R}^+$) est également solution.

$$y_d(t) = \begin{cases} 0 & \text{si } t \in [0, d] \\ \frac{1}{2}(t-d)^2 & \text{sinon.} \end{cases}$$

La figure 2.8 donne un exemple de multiplicité des solutions, pour $d = 5$ m.

On retiendra donc que le profil spatial de vitesse est moins complet, mais qu'il permet une représentation plus naturelle d'un ensemble de trajectoires dans un même espace. D'autre part, l'abscisse du profil étant une position géographique (exprimée dans un référentiel linéaire propre au réseau routier), la détection de l'infrastructure routière à partir des profils pourra se transcrire directement en information géoréférencée.

Notons qu'il existe un quatrième type de courbe fréquemment utilisée en FDA : Ramsay et Silverman (2005) recommandent une représentation des profils dans l'espace des phases ($\ddot{x} = f(\dot{x})$), susceptible de faire apparaître certains motifs qui seraient invisibles dans les autres modes (fig. 2.9). Ce type de représentation n'est malheureusement pas compatible avec la plupart des techniques d'apprentissage fonctionnel, et nous ne l'utiliserons donc pas dans le cadre de ces travaux de thèse.

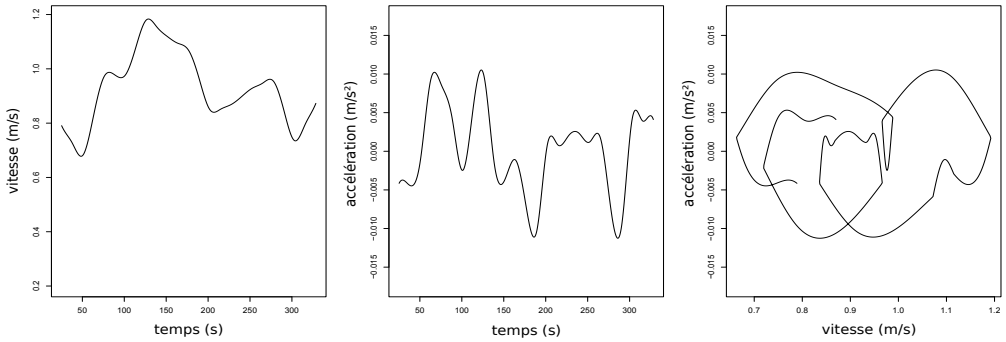


FIGURE 2.9 – Transformation d'un profil de vitesse temporel dans l'espace des phases.

2.3.2 Calcul numérique du profil spatial de vitesse

L'algorithme de calcul du profil spatial se décompose en 4 étapes :

- ➔ Conversion des coordonnées GPS dans un système planimétrique.
- ➔ Calcul de l'abscisse curviligne des points de la trajectoire.
- ➔ Estimation de la vitesse par dérivation numérique.
- ➔ Interpolation spatiale et/ou lissage du profil

2.3.2.1 Projection cartographique

Le système GPS est lié à un référentiel de coordonnées dites géocentriques, qui suppose une origine du repère au centre des masses de la Terre, et est muni d'un axe "vertical" Z paralèlle à l'axe des pôles. L'axe X est tel que le plan $(O, \vec{i}, \vec{k})$ est confondu avec celui du méridien de Greenwich. Finalement, le dernier axe Y est défini de sorte à ce que $\mathcal{R} = (O, \vec{i}, \vec{j}, \vec{k})$ constitue un repère orthonormé direct (cf figure 2.10). Le calcul du positionnement GPS est effectué dans ce système de coordonnées (pour sa simplicité) puis la position obtenue par le récepteur doit être convertie en coordonnées géographiques (longitude, latitude). Il faut noter toutefois que l'algorithme de conversion est intrinséquement dépendant d'un modèle d'ellipsoïde terrestre. Dans le cadre du GPS, le modèle d'ellipsoïde utilisé est le WGS84, associé au système géodésique du même nom, dont les spécifications peuvent être trouvées aisément dans la littérature spécialisé.

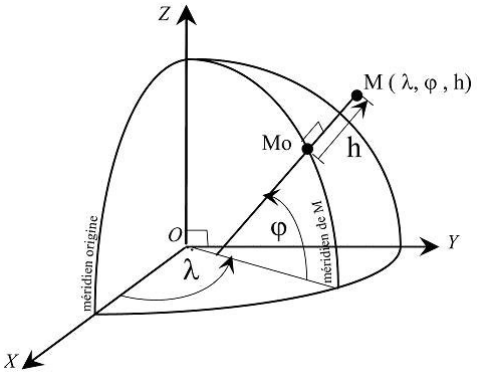


FIGURE 2.10 – Coordonnées géocentriques et géographiques. [Source : IGN].

Tous les traitements qui suivent sont effectués sur une représentation en projection plane des coordonnées. En effet, les coordonnées géographiques (λ, ϕ, h) constituent une manière commode et intuitive de décrire un point à la surface de la Terre ou en son voisinage, mais deviennent inopérantes dès lors qu'il est souhaitable de calculer des distances et des angles entre différents points donnés. Les systèmes de projection cartographique permettent d'apporter une réponse à ce problème, en appliquant une transformation de l'espace des coordonnées géographiques vers un plan que l'on peut se représenter comme étant localement tangeant à l'ellipsoïde terrestre. Des résultats importants de mathématiques montrent qu'il est impossible de développer une sphère ou *a fortiori* un ellipsoïde sur un plan et de fait, toute projection cartographique aussi sophistiquée soit elle, est vouée à introduire des erreurs de déformation. Une projection ne peut être au mieux et de manière exclusive, que *conforme* ou *équivalente*, c'est-à-dire qu'elle doit faire un choix entre conserver les angles ou les surfaces. Tout l'enjeu consiste alors à définir des projections qui minimisent les déformations (traditionnellement mesurées en parties par millions, ou ppm).

Dans le cadre de nos travaux d'étude, nous exprimerons les coordonnées en projection Lambert 93 (pour les données *ecoDriver* en France), adaptée à l'ensemble du territoire métropolitain et en UTM 54 (pour les données de Navitime au Japon). Pour ces deux projections, les déformations maximales, de l'ordre d'une cinquantaine de ppm, seront considérées comme négligeables. Pour une description plus complète du principe de fonctionnement de ces différentes représentations, nous renvoyons le lecteur à Girres (2012) ou encore à Lannuzel (2000).

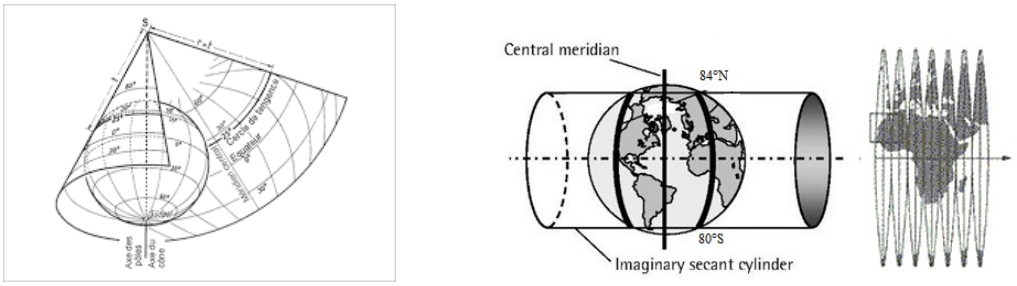


FIGURE 2.11 – Projection planimétrique des coordonnnées GPS : conique conforme Lambert 93 (à gauche) et cylindrique transverse UTM (à droite). [Sources : *Swiss Topo* et *Art of Directional Drilling*]

Pour une étude locale (quelques centaines de mètres), on pourra utiliser une projection locale grossière : supposons que le centre de l'emprise de la zone soit situé aux coordonnées géographiques (λ_c, φ_c) exprimés en degrés décimaux. En notant R_T le rayon moyen de la Terre, on peut déterminer des coordonnées planimétriques à l'aide des formules suivantes :

$$X_i(\lambda_i) = \frac{\pi R_T}{180} \cos\left(\frac{\pi}{180}\varphi_c\right)(\lambda_i - \lambda_0), \tag{2.12}$$

$$Y_i(\varphi_i) = \frac{\pi R_T}{180}(\varphi_i - \varphi_0), \tag{2.13}$$

avec λ_0 et ϕ_0 les coordonnées de l'origine de la projection, que l'on pourra fixer par exemple de sorte à ne traiter que des coordonnées positives :

$$\lambda_0 = \min\{\lambda_i\} \qquad \varphi_0 = \min\{\varphi_i\} \qquad \varphi_c = \frac{1}{n}\sum_{i=1}^{n}\varphi_i. \tag{2.14}$$

Pour le rayon de la Terre, on pourra prendre la valeur moyenne de $R_T = 6.371.10^6$ m. En pratique, le rayon $R(\varphi)$ à la latitude de travail, est compris dans l'intervalle $[R_T - \Delta R_T; R_T + \Delta R_T]$ avec le rapport $\Delta R_T / R_T$ de l'ordre de 0.3%. Les expressions de X_i et Y_i étant linéaire en R_T, les erreurs propagée sur les distances en projection seront du même ordre de grandeur, ce qui reste acceptable dans notre cadre de travail.

2.3.2.2 Calcul de l'abscisse curviligne

Pour chaque point (x_i, y_i, t_i) où x_i et y_i sont les coordonnées en représentation plane obtenues à l'issue de la phase décrite dans le paragraphe précédent, on souhaite adjoindre une abscisse curviligne $s_i \in \mathbb{R}^+$, définie comme la distance cumulée parcourue depuis le temps t_1. Si les coordonnées (x_i, y_i) sont suffisamment précises, le calcul des variables s_i pourra se réduire à une somme des distances entre les points d'observation de la séquence :

$$s_i = \sum_{j=1}^{i-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \tag{2.15}$$

Si les coordonnnées sont entachées d'erreurs significatives, l'équation 2.15 devient alors insuffisante. On pourra utiliser l'une ou/et l'autre des deux solutions proposées ci-dessous.

- Une source de données plus précise : par exemple l'odométrie du véhicule (nous testerons cette solution dans la section 2.4) ou l'intégration des mesures de vitesse Doppler. Nous verrons que même avec cette donnée plus précise, la comparaison de plusieurs profils de vitesse sur une longue portion de circuit, pose des problèmes de décalage.

- Des algorithmes de correction de trajectoire : recalage sur un réseau routier de référence (section 2.4) et filtrage de Kalman (2.5). Notons que ces algorithmes se subdivisent en deux catégories principales (Bijleveld et al., 2011) : les algorithmes **temps**

réel, qui doivent estimer les paramètres de l'époque courante uniquement à l'aide des observations passées, et les algorithmes de **post-traitement**, qui ont accès à l'intégralité de la trajectoire pour chaque correction locale, ce qui produit en général des algorithmes moins rapides, mais plus optimaux en termes de qualité de correction.

Dans ce travail de thèse, où on suppose avoir à disposition un ensemble de trajectoires GPS sur une zone donnée pour y inférer la présence d'un éventuel élément de signalisation routière, nous utiliserons des méthodes de cette seconde catégorie.

2.3.2.3 Calcul de la vitesse

Dans une logique temps réel, les informations de position doivent permettre le calcul de la vitesse à mesure que les données GPS sont acquises. Cette estimation ne peut donc se faire qu'avec une différence finie avant, avec une erreur d'approximation inversement proportionnelle à la fréquence d'acquisition des données :

$$v(t) = s'(t) = \frac{s(t+h) - s(t)}{h} + \mathcal{O}(h). \tag{2.16}$$

Dans le cadre d'un post-traitrement, on estime la vitesse par différence finie centrée, avec une erreur d'ordre quadratique :

$$v(t) = s'(t) = \frac{s(t+\frac{h}{2}) - s(t-\frac{h}{2})}{h} + \mathcal{O}(h^2). \tag{2.17}$$

Théoriquement, l'erreur d'approximation de 2.17 n'est valable que si s est dérivable jusqu'à l'ordre 3 ou plus, ce qui permet d'écrire, par un double développement de Taylor :

$$\begin{aligned}
s(t+h) &= s(t) + hs'(t) + \frac{h^2}{2}s''(t) + \frac{h^3}{6}(s'''(t) + \varepsilon^+(t,h)),\\
s(t-h) &= s(t) - hs'(t) + \frac{h^2}{2}s''(t) - \frac{h^3}{6}(s'''(t) + \varepsilon^-(t,h)),\\
\frac{s(t+h) - s(t-h)}{2h} &= s'(t) + \frac{h^2}{6}\Big(s'''(t) + \varepsilon(t,h)\Big) = s'(t) + \mathcal{O}(h^2).
\end{aligned}$$

La fonction s réprésentant l'évolution d'une grandeur physique, on peut facilement emmettre l'hypothèse qu'elle est $\mathcal{C}_\infty$ sur tout son domaine.

Numériquement, en prenant $h = 2\Delta t = 2(t_1 - t_0)$, où Δt est l'intervalle de temps (constant) entre deux acquisitions GPS successives :

$$v_i = \frac{s_{i+1} - s_{i-1}}{2\Delta t}. \tag{2.18}$$

On pourra se référer au paragraphe 2.2.3 pour quantifier l'incertitude sur le calcul de v_i à partir des caractéristiques du capteur GPS. Dans l'hypothèse où l'erreur commise sur v_i est

inacceptable pour la suite du travail, à nouveau, on pourra se reporter sur deux solutions alternatives :

- Utiliser une autre source de données plus précise : par exemple la mesure de vitesse Doppler (à l'instar du chapitre 3) ou la mesure odométrique du véhicule.

- Appliquer des algorithmes de correction de trajectoire, en particulier si d'autres types d'observations sont disponibles, comme par exemple l'accélération longitudinale et le cap (*cf* paragraphe 2.6).

2.3.2.4 Interpolation et lissage

À l'issue de l'étape d'estimation des vitesses, on dispose de quintuplets $(x_i, y_i, t_i, s_i, v_i)$ à partir desquels on souhaite construire un profil de vitesse $v = f(s)$. Dans la suite, on notera x l'abscisse curviligne du véhicule. L'objectif est d'obtenir les valeurs $v(x_i)$ pour une suite régulièrement échantillonnée de valeurs le long du parcours : $x_i = x_0 + i\Delta x$.

Le but de l'interpolation est de trouver une fonction f, appartenant à une certaine classe de fonctions, et telle que $f(x_i) = v_i$ pour tout couple (x_i, v_i) de l'échantillon de données (Negulescu, 2007). On parle d'*interpolation* lorsque le modèle est exact, c'est-à-dire lorsque les valeurs qu'il prédit au niveau des sites observés sont égales aux valeurs observées. Dans le cas contraire, on suppose un degré d'incertitude sur les observations, et on parle d'un modèle de *lissage*. Dans notre cadre, nous avons décidé d'interpoler les profils de vitesse à l'aide de splines.

En mathématiques appliquées, une spline désigne une fonction polynomiale par morceaux, définie à partir d'un ensemble de nœuds (ou points d'observation) et de contraintes de régularité sur ses dérivées. Les interpolateurs constants par morceaux et affines par morceaux correspondent donc respectivement à des splines d'ordre 1 et 2.

En pratique, pour un échantillon de points observés (x_i, y_i), le calcul numérique d'une spline d'interpolation s'effectue en considérant l'espace des polynômes par morceaux comme un sous-espace vectoriel de $\mathcal{C}^m([a, b])$ de dimension $N = m + L$, dont on extrait une base de fonctions $\{\phi_1, \phi_2, ...\phi_N\}$, permettant d'écrire une fonction spline sous la forme linéaire :

$$f(x) = \sum_{k=1}^{N} \beta_k \phi_k(x) = \mathbf{B}\boldsymbol{\phi}^T, \tag{2.19}$$

où $\boldsymbol{\phi} = [\phi_1(x), \phi_2(x)...\phi_N(x)]$ est le vecteur des fonctions de bases calculées en x et $\mathbf{B}$ est le vecteur des coordonnées β_k de f dans cette base.

L'estimation d'une spline est alors réduit à la résolution du problème d'algèbre linéaire : $\mathbf{Y} = \mathbf{AB}$ où $\mathbf{Y} = [y_1, y_2...y_n]^T$ est le vecteur des observations et $\mathbf{A}$ est une matrice de $\mathbb{R}^{n\times N}$, dans laquelle l'élément a_{ij} désigne la valeur prise par la fonction de base ϕ_j au niveau du point x_i. Si exactement N points (distincts) ont été observés, le problème a une unique solution. Sinon, on calcule une estimation par moindres carrés.

La question se pose alors sur le choix de la base $\boldsymbol{\phi}$. Une solution simple consiste à utiliser la base des *puissances tronquées*. En pratique, cette dernière pose des problèmes d'instabilité numérique (Eubank, 1999), et on lui préfère souvent la base des B-splines, dont on donne une illustration en figure 2.12, et dont on pourra trouver une description formelle dans Andrieu (2013).

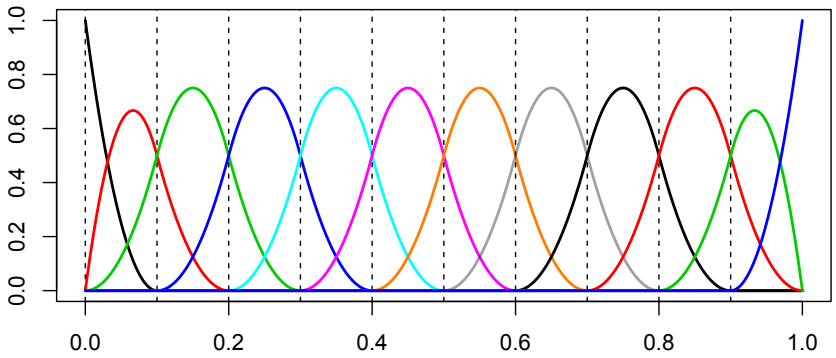


FIGURE 2.12 – Les 12 fonctions de base B-splines d'ordre $m = 3$ avec une subdivision uniforme sur l'intervalle [0,1].

Un avantage pratique des B-splines, réside dans la compactité des supports des fonctions de base. Plus spécifiquement, une B-spline d'ordre m est systématiquement nulle en dehors d'un intervalle couvrant m portions de la subdivision (Ramsay et Silverman, 2005). Cette propriété autorise une formulation du problème à l'aide de matrices bandes, peu coûteuses en stockage mémoire et rapidement solubles.

On peut montrer que les splines d'ordre 3 sont solutions du problème :

$$\underset{f \in W^{m,2}}{\operatorname{argmin}} \ \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int_a^b f^{(m)}(x)^2 dx. \tag{2.20}$$

où $W^{m,2}$ est l'espace de Sobolev : $W^{m,2} = \{f \in \mathcal{C}^{(m-1)}([a,b]) \mid f^{(m)} \in L^2([a,b])\}$.

La solution s'obtient alors immédiatement, en posant $B \in \mathbb{R}^{n \times N}$ la matrice des fonctions de bases prises aux points $\mathbf{x}$: $B_{ij} = \phi_j(x_i)$ et $\Omega \in \mathbb{R}^{N \times N}$ la matrice des produits scalaires des fonctions de base : $\Omega_{ij} = \langle \phi_i, \phi_j \rangle$:

$$\beta^* = (B^T \Sigma^{-1} B + \lambda \Omega)^{-1} B^T \Sigma^{-1} Y, \tag{2.21}$$

où Y est le vecteur des observations y_i de matrice de covariance Σ.

Notons que lorsque $\lambda = 0$, le problème variationnel 2.20 est dégénéré, et toute fonction passant par tous les points du jeu de données sera solution. En pratique, comme on cherche la solution dans un espace de splines, la solution 2.21 est réduite à une régression par

moindres carrés sans pénalité sur la régularité du modèle. À l'inverse, quand $\lambda \to \infty$, seul le terme de régularisation importe dans le processus de minimisation, et la solution 2.21 est la droite de régression des données (Friedman et al., 2001). La figure 2.13 illustre ce compromis sur des données générées aléatoirement.

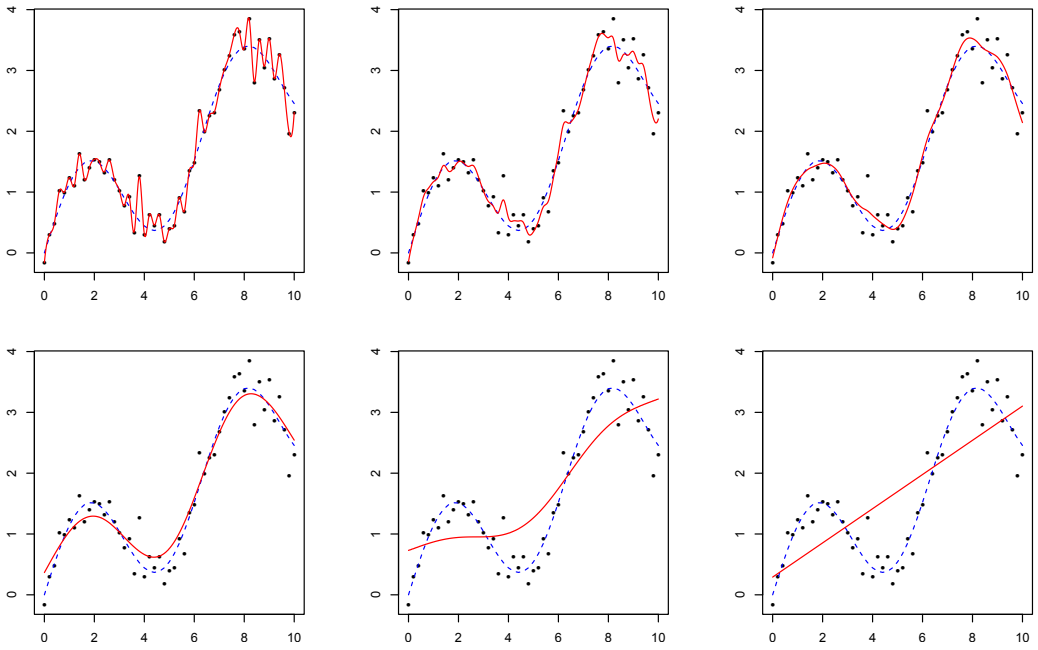


FIGURE 2.13 – Calcul d'une spline cubique (en rouge) pour différentes valeurs de λ (croissantes de gauche à droite et de haut en bas). La fonction cible (en pointillés bleus) est $f(x) = \sin x + 0.3x$. Les y_i sont entachées d'erreurs normales *i.i.d.* d'écart-type 0.4.

Si la formulation variationnelle des splines de lissage permet d'évacuer le problème du choix de la subdivision ξ de l'intervalle de travail, on doit en retour calibrer le paramètre λ. Pragmatiquement, on souhaite choisir λ de sorte à minimiser l'erreur quadratique moyenne entre la fonction f à estimer, et son estimation par spline f^* :

$$\lambda^* = \underset{\lambda \in \mathbb{R}^+}{\operatorname{argmin}} \, \mathbb{E}[(f(x) - f^*(x))^2]. \tag{2.22}$$

En pratique, f est inconnue et l'équation 2.22 est inopérante. Plusieurs stratégies ont été développées pour contourner ce problème, la plupart requierant des hypothèses sur la distribution des erreurs (Besse et Thomas-Agnan, 1989). Dans notre cadre de travail, où les erreurs peuvent difficilement être considérées comme étant *i.i.d.*, nous utiliserons la méthode de la validation croisée (CV), décrite pour la première fois par Allen (1971). L'objectif principal consiste à minimiser l'erreur quadratique entre une observation y_k et la prédiction réalisée par un estimateur f^*_{-k} ayant été construit à partir du jeu de données dont on a retiré la k-eme observation :

$$\lambda^*_{CV} = \underset{\lambda \in \mathbb{R}^+}{\operatorname{argmin}} \sum_{k=1}^{n} (y_k - f^*_{-k}(x_k))^2. \tag{2.23}$$

Cette méthode de sélection est analogue à la procédure de validation *leave-one-out* d'un algorithme d'apprentissage dont nous parlerons dans le chapitre 3.

On donne en figure 2.14 un exemple d'interpolation par splines des profils de vitesse utilisés dans la suite de ce travail de thèse. La figure 2.15 présente une illustration de la méthode de validation croisée pour le choix du paramètre de lissage optimal $\lambda^* = 32.9$ sur le cas des profils de vitesse.

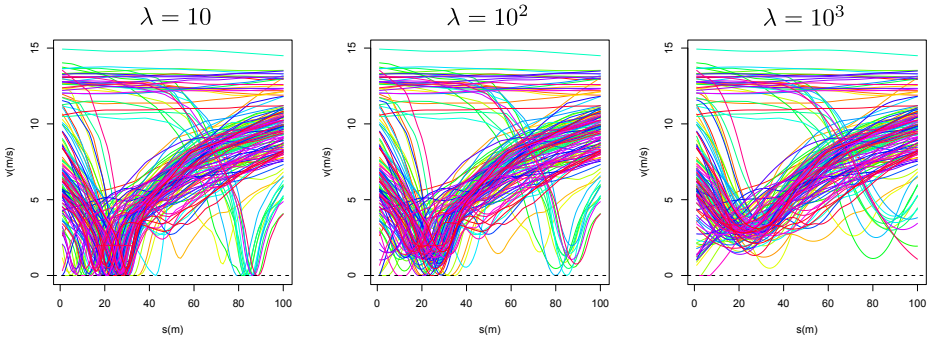


FIGURE 2.14 – Exemples d'interpolation des profils de vitesse par splines cubiques de lissage, pour trois paramètres λ.

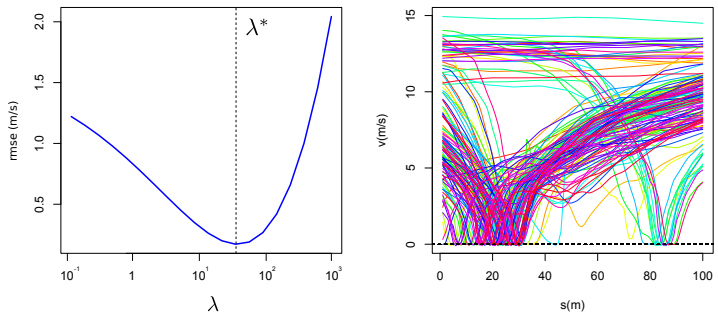


FIGURE 2.15 – À gauche : validation croisée et sélection d'un paramètre optimal $\lambda^* = 32.9$. À droite : représentation des profils de vitesse interpolés et lissés avec le paramètre λ^* sur une fenêtre glissante de 100 m (voir chapitre 3 pour plus de détails).

Lissage par splines sous contraintes

Par hypothèse, la fonction F est croissante (*cf* définition 2.2, section 2.3.1). Lorsque l'abscisse s du profil de vitesse dénote l'abscisse curviligne du véhicule le long de sa trajectoire (numériquement calculée à partir de l'expression 2.15), cette hypothèse est exacte par définition. En revanche, si s désigne l'abscisse curviligne du véhicule après projection

sur un circuit de référence, il n'existe aucune contrainte physique imposant à $F(t) = s(t)$ d'être croissante. Pour un véhicule circulant sur un réseau routier, en condition nominale d'utilisation, l'hypothèse de croissance de F (et donc par suite de positivité de la vitesse v) est satisfaisante en pratique.

Nous voyons d'après l'expression 2.21 que tout point interpolé $\hat{y}(x)$ s'exprime sous forme d'une combinaison linéaire des observations : $\hat{y}(x) = \sum_i l_i(x) y_i$. En pratique, il n'existe aucune garantie sur le fait que $l_i(x) \in [0, 1]$, autrement dit, l'estimation n'est pas nécessairement une combinaison linéaire convexe des observations, et le profil de vitesse interpolé peut être localement négatif, quand bien même toutes les observations y_i sont positives. Une vitesse localement négative impliquerait une abscisse curviligne décroissante, ce qui violerait la définition fonctionnelle du profil de vitesse. On donne une illustration de ce phénomène sur la figure 2.16 à gauche.

Une solution simple à ce problème consiste à seuiller l'estimation f, de sorte à rejeter les valeurs négatives (figure 2.16 au centre). Cette option donne souvent des estimations non-régulières et irréalistes des profils de vitesse.

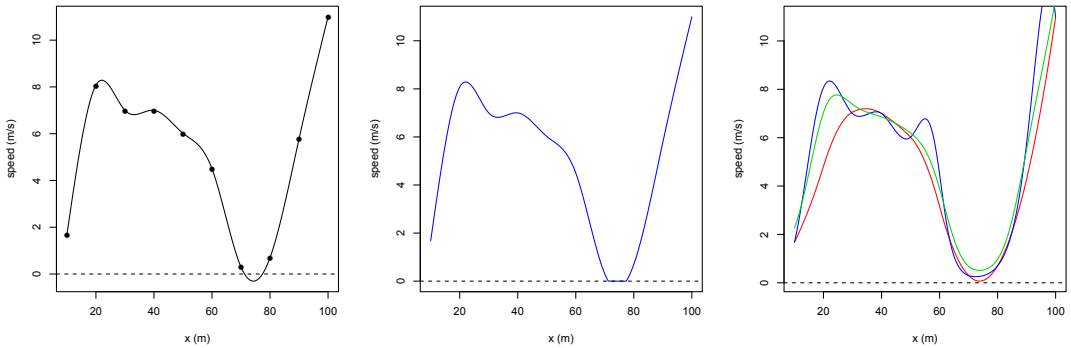


FIGURE 2.16 – À gauche : splines d'interpolation sans contrainte. Au centre : seuillage dur d'une splines d'interpolation sans contrainte. À droite : trois modes d'interpolation avec contrainte de positivité (voir texte pour les détails).

Une méthode plus rationnelle pour poser une contrainte de positivité sur une fonction spline f, consiste à l'exprimer comme l'exponentielle d'une fonction non-contraine w (Ramsay et Silverman, 2005) :

$$f(x) = e^{w(x)}. \tag{2.24}$$

La fonction f peut alors être estimée en minimisant le critère :

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - e^{w(x_i)})^2 + \lambda \int_a^b w''(x)^2 dx, \tag{2.25}$$

où la pénalité de régularisation porte sur la dérivée seconde de la fonction transformée dans l'espace logarithmique, et non sur la fonction f elle-même. L'avantage de cette méthode est que le critère d'attache aux données est exprimé dans l'espace métrique des observations $f(x_i)$. En contre-partie, il n'existe pas de forme close pour la résolution de ce problème, et il devient nécessaire d'utiliser des méthodes d'optimisation numériques. La courbe rouge de la figure 2.16 (figure de droite) donne un exemple de calcul de spline contrainte minimisant l'équation 2.25 sur un profil de vitesse.

Une solution alternative pour éviter le passage par des méthodes numériques itératives, consiste à exprimer le terme d'attache aux données dans l'espace logarithmique :

$$\frac{1}{n}\sum_{i=1}^{n}(\ln\, y_i - w(x_i))^2 + \lambda \int_a^b w''(x)^2 dx. \tag{2.26}$$

Moyennant cette modification, la résolution du problème s'effectue algébriquement, par transformation exponentielle de la solution, obtenue par 2.21. Dans le cadre de ces travaux de thèse, nous utiliserons cette seconde méthode pour interpoler les profils de vitesse GPS. Les courbes bleue et verte illustrent, respectivement, un cas de splines d'interpolation et de lissage, obtenues par cette méthode approchée. La courbe de lissage approche plus grossièrement l'arrêt, mais permet d'éviter les problèmes d'oscillation (type phénomène de Runge) qui affectent la courbe bleue.

Notons qu'il existe d'autre méthodes plus spécifiquement adaptées aux profils de vitesse, passant par exemple par la définition d'une contrainte de monotonie sur le profil temps $\times$ espace F (Ramsay et Silverman, 2005; Andrieu et al., 2013a).

2.4 Correction latérale : recalage sur le réseau routier

2.4.1 Introduction au map-matching

Dans la section précédente, nous avons vu que l'abscisse des profils de vitesse pouvait être calculée de plusieurs manières. La solution retenue dans un premier temps dans le cadre de nos expérimentations (*cf* chapitre 3), a consisté à définir cette abscisse à partir des données fournies par l'odomètre (*cf* section 2.2.4). La représentation des profils de vitesse à disposition met en évidence le manque de fiabilité de l'odomètre dans notre cas d'application (figure 2.17). Pour des véhicules ayant circulé sur une boucle de 25 km, l'écart maximal entre les mesures de distances cumulées sur les 144 profils est de l'ordre de 600 m en fin de parcours (soit une erreur maximale estimée à 300 m, correspondant à une précision de l'ordre de 1 %).

Chaque profil de vitesse de véhicule est donc localement plus ou moins dilaté ou contracté par rapport à une fonction de distance cumulée théorique qui correspondrait à la distance réellement parcourue par le véhicule. La valeur du coefficient d'erreur n'étant pas constante sur l'ensemble de la trajectoire, une correction globale consistant à estimer une homothétie d'abscisse pour chaque profil paraît donc inenvisageable.

Notons toutefois qu'une méthode proposée par Andrieu et al. (2013a) vise à estimer des fonctions monotones (appelées *warping functions* dans la littérature de l'analyse de don-

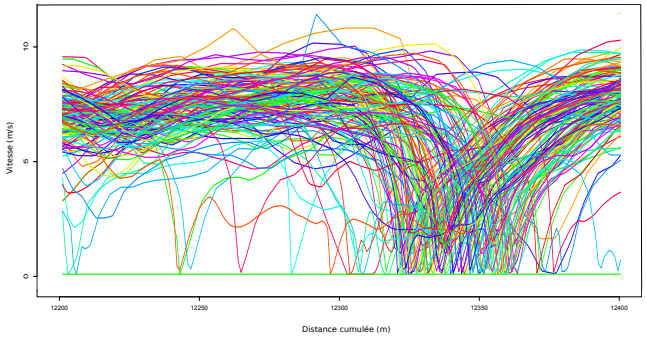


FIGURE 2.17 – Décalage des profils de vitesse induit par le bruit sur les mesures de l'odomètre

nées fonctionnelles) qui, par composition, assurent le recalage des profils de vitesse entre eux. Ces fonctions sont calculées de sorte à aligner les points caractéristiques des profils les uns sur les autres via un critère de type moindres carrés. Cette solution, qui ne suppose que la connaissance des profils de vitesse, permet la construction de profils agrégés plus représentatifs de l'ensemble des profils et est particulièrement intéressante lorsque le réseau routier sur lequel les véhicules circulent n'est pas connu, ce qui est le cas dans un contexte de *map inference*. Dans notre cadre de travail, que l'on peut situer plus en aval, nous supposons avoir à disposition les données topographiques de ce réseau, soit qu'elles aient été acquises par des méthodes de cartographie traditionnelles, soit que la détermination de ce réseau ait déjà été effectuée en amont à partir des traces GPS.

Dans ce contexte particulier, il paraît donc plus avantageux de substituer les mesures bruitées de l'odomètre par une valeur d'abscisse curviligne le long du réseau routier, permettant ainsi de s'affranchir des problèmes de décalage des courbes. Cette solution est d'autant plus intéressante qu'elle ne suppose pas de concomittance des points caractéristiques des profils, ce qui est le cas en pratique (e.g. présence de files de voitures devant un feu tricolore). Cette opération de recalage d'observations GPS sur le réseau routier est généralement appelée *map-matching* (Quddus et al., 2007).

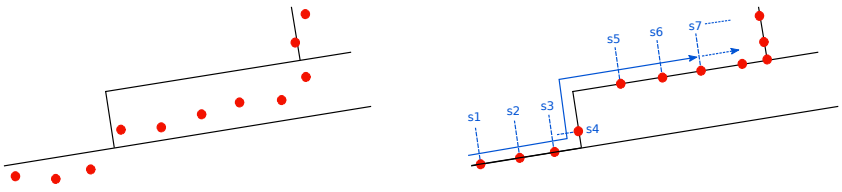


FIGURE 2.18 – Map-matching des points GPS sur le réseau routier et calcul d'une distance cumulée le long du réseau (abscisse curviligne s en bleu).

Pour peu que le réseau routier de référence soit de qualité décente (*cf* section 2.4.4.2), un intérêt fondamental du map-matching est d'améliorer la précision de localisation des points mesurés par GPS, en particulier sur les portions de circuit où les satellites sont susceptibles d'être masqués (tunnel, arbres, canyons urbains...). La couverture de la constellation GPS est telle que même en milieu urbanisé, un récepteur a de fortes chances d'avoir plus de quatre satellites en vue et donc de réunir les conditions pour obtenir une estimation de

sa position courante (Lee et al., 2008). Il peut cependant arriver de rencontrer localement une forte dérive locale de l'estimation de la position lorsque le nombre de satellites en vue est trop faible. Ces erreurs peuvent être compensées par la procédure de map-matching, qui ajoute une information externe afin d'affiner le positionnement.

Étant donnée une séquence de points GPS et un modèle de graphe du réseau routier, le problème générique du map-matching consiste à trouver la séquence de points correspondant aux positions successives d'un mobile sur le graphe et dont les points GPS sont des observations bruitées.

La littérature associée à ce domaine de recherche classifie généralement les solutions en quatre grandes catégories (Quddus et al., 2007; Zheng, 2015) :

- Les **algorithmes géométriques**, réduisent le réseau routier à sa représentation topographique en laissant ainsi de côté la nature topologique du graphe sous-jacent. L'algorithme le plus fréquemment utilisé dans cette catégorie, est la projection sur le réseau suivant la plus courte distance. Cet algorithme donne de bons résultats lorsque le réseau routier est simple et que sa densité est relativement faible au regard de l'erreur introduite par le GPS. Dans le cas contraire, la solution obtenue par map-matching peut retourner des résultats incohérents vis-à-vis de la topologie du réseau routier (e.g. franchissement d'un fleuve en une zone dépourvue de pont...).

- La seconde catégorie dite des **algorithmes topologiques** pallie ce problème en considérant en plus la nature de graphe du réseau et refusera ainsi toute solution induisant le parcours par le mobile d'un chemin incompatible avec la topologie des routes.

- Les **algorithmes probabilistes** orientent la résolution du problème à l'aide d'informations statistiques sur les erreurs engendrées par la mesure GPS (par exemple à l'aide d'ellipses de confiance définies autour des points). Ces algorithmes permettent de n'envisager que les cas d'affectations point-tronçon crédibles au regard de la précision du GPS. Lorsqu'un point peut être projeté sur différents tronçons de routes, la décision est faite sur la base de critères additionnels tels que la direction de déplacement du mobile à l'instant de la mesure.

- Enfin, les **algorithmes avancés** combinent au sein d'un même framework cohérent, des techniques issues d'au moins deux des catégories précédentes.

On distingue généralement deux types de calcul de map-matching : le calcul temps réel (qui ne peut bénéficier que des observations précédentes) et le calcul en post-traitement. Dans notre cas de figure, nous nous plaçons dans une configuration de map-matching en post-traitement. La solution utilisée dans notre travail a été proposée par Newson et Krumm (2009) et fait actuellement figure de référence dans le domaine du map-matching. Elle repose sur un modèle de chaîne de Markov à états cachés et fait donc partie de la quatrième des catégories citées précédemment. À noter que depuis, plusieurs versions temps-réel de l'algorithme ont été proposées (Goh et al., 2012; Ahres et al., 2014).

2.4.2 Map-matching par chaîne de Markov cachée

On se donne un modèle de graphe $G(V, E)$ où :

- V représente l'ensemble des sommets du réseau routier.
- $E \subseteq V^2$ représente l'ensemble des tronçons (on notera $|E| = m$).

G représente la structure topologique du réseau routier. Un itinéraire sur G est donc une séquence de sommets $\{v_i\}_{i=1..N}$ telle que tout couple de sommets consécutifs soit relié par un arc :

$$\forall\ i \in [\![1, N-1]\!] \quad (v_i, v_{i+1}) \in E. \tag{2.27}$$

On définit alors à partir de ce graphe un modèle d'automate de Markov en considérant E comme une liste d'états possible. Chaque état représente un tronçon de route du graphe sous-jacent. Soit $X_0, X_1, ...X_n$ une suite de variables aléatoires définies sur un même espace de probabilité $(\Omega, \mathcal{A}, P)$ à valeurs dans $\mathcal{X} = E$, un ensemble fini ou dénombrable.

Définition 2.3. *La suite de variables $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov d'espace d'états $\mathcal{X}$ si pour tout entier $i \geqslant 2$ et pour tout ensemble d'états $(x_1, x_2, ...x_i) \in \mathcal{X}^n$:*

$$P(X_i = x_i, X_{i-1} = x_{i-1}, ...X_1 = x_1) > 0, \tag{2.28}$$

et la probabilité conditionnelle de X_i sachant les états passés peut être résumée par la probabilité conditionnelle de X_i sachant l'état précédent X_{i-1}.

$$P(X_i = x_i \mid X_{i-1} = x_{i-1}, X_{i-2} = x_{i-2}, ...X_1 = x_1) = f(x_i, x_{i-1}). \tag{2.29}$$

Autrement dit, l'état futur est indépendant du passé, sachant la présent.

Alternativement, on peut définir une chaîne de Markov de manière constructive :

Propriété 2.2. *La suite de variables $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov d'espace d'états $\mathcal{X}$ s'il existe une famille de noyaux de transition $Q_i(.,.) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ (avec pour tout $x \in \mathcal{X}$, $Q_i(., x)$ une loi de probabilité sur $\mathcal{X}$) et une probabilité π sur $\mathcal{X}$, telle que la loi jointe des X_i s'exprime sous la forme :*

$$P(X_0 = x_0, X_1 = x_1, ...X_n = x_n) = \pi(x_0) \prod_{i=1}^{n-1} Q_i(x_{i+1}, x_i), \tag{2.30}$$

avec Q_i et $\pi > 0$ de sorte à respecter la contrainte 2.28.

La preuve s'établit aisément : dans le sens direct en écrivant la loi jointe totale à l'aide de la règle de chaînage puis en appliquant la propriété de Markov 2.29 ; dans le sens indirect on obtient immédiatement le résultat en écrivant explicitement la loi conditionnelle puis en simplifiant à l'aide de la caractérisation 2.30.

Définition 2.4. *On dit que la chaîne de Markov est homogène lorsque le noyau de transition ne dépend pas du pas de temps :* $Q_1 = Q_2 = ... = Q_n = Q$.

Dans le cadre du map-matching, Newson et Krumm (2009) propose de modéliser la localisation du véhicule dans le graphe G à l'aide d'une chaîne de Markov homogène d'espace d'états E. On définit un noyau de transition, en supposant les arcs de E indexés par les entiers i et j :

$$Q_{i|j} = \mathbb{P}(x_k = i \mid x_{k-1} = j), \tag{2.31}$$

traduisant ainsi pour tout couple de tronçons $(i, j) \in E^2$, la probabilité de passer du tronçon j au tronçon i. On notera $Q \in \mathbb{R}^{m \times m}$ la matrice correspondante contenant les probabilités des transitions : $q_{ij} = Q(i|j)$. Comme toute valeur de probabilité, elle doit respecter les deux contraintes suivantes :

$$\forall\ (i, j) \in V^2 \quad q_{ij} \geqslant 0, \tag{2.32}$$

$$\forall\ j \in V \quad \sum_{i=1}^{m} q_{ij} = 1. \tag{2.33}$$

La matrice Q est dite *stochastique*.

L'automate ainsi défini représente les déplacements *probables* d'un véhicule donné sur le réseau routier. En toute logique, un noyau de transition semblera cohérent si une valeur de probabilité relativement haute est affectée à deux arcs partageant un même sommet, et à l'inverse si une probabilité quasi-nulle est affectée à deux arcs difficilement joignables (indépendamment de la distance géométrique les séparant).

Définition 2.5. *On appelle modèle de Markov caché, ou plus formellement modèle de Markov à états cachés (en anglais HMM pour Hidden Markov Model), une chaîne de Markov* $(X_n)_{n\in\mathbb{N}}$ *à laquelle on adjoint une seconde suite de variables aléatoires* (Y_n) *à valeurs dans un espace* $\mathcal{Y}$ *(potentiellement non-dénombrable) et telle que chaque variable* Y_i *ne dépende que de l'état* X_i *courant* :

$$P(Y_i = y_i \mid X_{0:i} = x_{0:i}, Y_{0:i} = Y_{0:i}) = P(Y_i = y_i \mid X_i = x_i), \tag{2.34}$$

avec la convention : $x_{0:i} = (x_0, x_1, x_2...x_i)$.

Newson et Krumm (2009) proposent d'utiliser un modèle d'émission permettant de relier les observations GPS aux tronçons de route. On note $Y = (y_1, y_2, ... y_n)$ la séquence des observations GPS, où y_k est un point de $\mathbb{R}^2$. L'erreur introduite par un GPS est modélisée par une loi normale [4], si bien que pour tout tronçon i et pour toute observation y_k, on peut écrire :

4. Cette modélisation est justifiée par le théorème de Cochran (*cf* section 2.4.4).

$$p_{y_k|i} = \frac{1}{2\pi\sigma} \exp\Big(-\frac{1}{2}\big(\frac{\min\{||y_k - p||_2,\ p \in E_i\}}{\sigma}\big)^2\Big), \tag{2.35}$$

où σ est l'erreur typique du GPS, $||.||$ est la norme euclidienne classique de $\mathbb{R}^2$ et p est un point du tronçon E_i. La quantité $\min\{||y_k - p||,\ p \in E_i\}$ s'interprète alors comme la distance minimale entre l'observation y_k et le tronçon i. On vérifie bien que $p^e_{y_k|i}$ est maximale quand la mesure y_k est située sur le tronçon i et tend vers 0 à mesure qu'elle s'en éloigne, ce qui paraît cohérent pour un modèle d'observation. En pratique le modèle d'erreur devrait aussi prendre en compte l'imprécision géométrique des polylignes du réseau routier (*cf* section 2.4.4) : $\sigma^2 = \sigma^2_{gps} + \sigma^2_{network}$

On complète le modèle avec une probabilité initiale π_0 sur E, telle que $\pi_0(i) = P(x_0 = i)$ traduit la probabilité que le véhicule parte du tronçon i à l'instant initial. Lorsque l'on n'a pas d'information sur le point de départ d'un véhicule, π_0 est uniforme sur E.

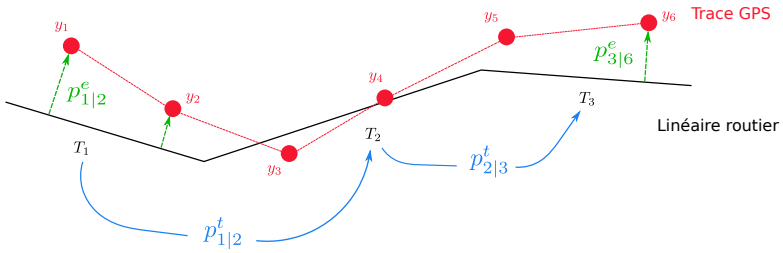


FIGURE 2.19 – Modèle de Markov caché sur un graphe routier, avec les probabilité de transition (en bleu) et d'émission (en vert).

La probabilité de transition est quant à elle modélisée pour tout couple de tronçons (i, j) par une loi exponentielle :

$$Q_{i|j} = \frac{1}{\beta} \exp\Big(\frac{d_v(i,j) - d_r(i,j)}{\beta}\Big), \tag{2.36}$$

où β est un paramètre à ajuster tandis que $d_v(i,j)$ et $d_r(i,j)$ représentent respectivement les distances à vol d'oiseau et suivant le réseau entre les tronçons i et j. Pour fixer β, les auteurs suggèrent d'utiliser la méthode d'estimation robuste du paramètre d'échelle d'une loi exponentielle proposée par Gather et Schultze (1999). On remarquera par exemple, qu'avec ce modèle, deux tronçons de route adjacents situés de part et d'autre d'un fleuve auront une probabilité de transition très faible s'il n'existe pas de pont traversant le fleuve aux environs ($d_v \ll d_r$). Notons que d_v est calculée suivant la distance euclidienne tandis que la détermination de d_r nécessite l'emploi d'algorithmes de routing tels que Dijkstra (Dijkstra, 1959) ou, sous certaines conditions sur la métrique du graphe, des algorithmes heuristiques plus rapides tels que A* (Hart et al., 1968a).

En pratique, d_v est calculée entre les observations GPS, tandis que d_r est calculée sur le réseau entre les points projetés des observations. Le noyau de transition dépend donc en toute rigueur des observations et du pas de temps :

$$Q^t_{i|j} = \frac{1}{\beta} \exp\Big(\frac{||y_{t+1} - y_t||_2 - d_r(p_j(y_i), p_i(y_{t+1})}{\beta}\Big), \tag{2.37}$$

où $|.|_r$ représente la distance suivant le réseau routier, et $p_i : \mathbb{R}^2 \to \mathbb{R}^2$ est le projeté (suivant la plus courte distance) d'une observation GPS sur l'arc i du réseau. Cette simplification consiste à supposer que y_i est un paramètre pour la transition (et non une variable conditionnante) et une variable aléatoire pour le modèle d'emission. Ceci n'a aucun impact sur la résolution numérique du problème, et nous utiliserons une hypothèse similaire par la suite lorsque nous étendrons ce modèle dans la section 2.4.3.

Notons enfin qu'avec cette définition, la loi de probabilité de transition sur le graphe n'est pas nécessairement symétrique puisque : $d_r(i, j)$ est en général différent de $d_r(j, i)$ (sens uniques, non-communications, etc). Ces probabilités de transition 2.36 sont normalisées de sorte à respecter la contrainte 2.33.

Le problème peut alors se formuler ainsi :

Problème : étant donnée la séquence d'observations $(y_1, y_2...y_n) \in \mathbb{R}^2$, trouver la séquence d'états $X^* \in E^n$ qui maximise la loi conditionnelle des états, sachant les observations :

$$X^* = \underset{X \in E^n}{\operatorname{argmax}}\ P(x_1, x_2, ...x_n \mid y_1, y_2, ...y_n). \tag{2.38}$$

De manière plus informelle, le problème consiste à trouver l'itinéraire parcouru par le véhicule, compte tenu de la séquence des observations GPS bruitées, et de la topologie du réseau. La solution la plus probable X^* sera donc celle qui explique au mieux les observations (p) tout en restant cohérente avec la topologie du réseau routier (q).

Il apparaît clairement que la recherche exhaustive de X dans E^n est inenvisageable dès lors que la séquence de déplacements contient plus d'une vingtaine de points. En revanche, par application de la formule de Bayes :

$$P(x_1, x_2, ...x_n \mid y_1, y_2, ...y_n) = \frac{P(y_1, y_2, ...y_n \mid x_1, x_2, ...x_n)\ P(x_1, x_2, ...x_n)}{P(y_1, y_2, ...y_n)}, \tag{2.39}$$

et grâce aux indépendances conditionnelles du modèle HMM, le problème 2.38 peut se réexprimer sous la forme :

$$X^* = \underset{X \in E^n}{\operatorname{argmax}} \prod_{k=1}^{n} P(y_k|x_k)P(x_k|x_{k-1}) \tag{2.40}$$

$$= \underset{X \in E^n}{\operatorname{argmax}} \prod_{k=1}^{n} p_{x_k|x_{k-1}} q_{y_k|x_k}. \tag{2.41}$$

Ce problème est en général résolu par l'algorithme de Viterbi (1967), fondé sur un paradigme de programmation dynamique et qui procède de manière itérative : à chaque instant

t, et pour chaque état e_i atteignable en t, on détermine la séquence $x_{0:t}$ qui maximise la probabilité 2.41 tronquée au pas de temps t et qui se termine en $x_t = e_i$. On propage alors le résultat au temps $t+1$ à l'aide des probabilités calculées en t, et ainsi de suite jusqu'à $t = n$. La séquence qui maximise la probabilité d'apparition est enfin reconstruite par *back-tracking*. Il faut noter toutefois que lorsque la séquence d'observations est trop longue, les produits de probabilités tendent à s'annuler en calcul machine. La transformation logarithmique du problème (effectuée en amont de la résolution) permet de se prémunir contre cet éceuil tout en offrant l'avantage de convertir les produits en sommes, plus rapides à calculer.

$$X^* = \underset{X \in E^n}{\operatorname{argmin}} \; -\sum_{k=1}^{n} \Big[\log p_{x_k|x_{k-1}} + \log q_{y_k|x_k} \Big]. \tag{2.42}$$

Cette modélisation permet aussi de concevoir la résolution comme un calcul d'un plus court chemin dans un graphe valué par la somme des logarithmes des probabilités de transition et d'emission. Le chemin le plus court minimise alors la somme des logarithmes, ce qui revient à maximiser la probabilité d'apparition de la séquence.

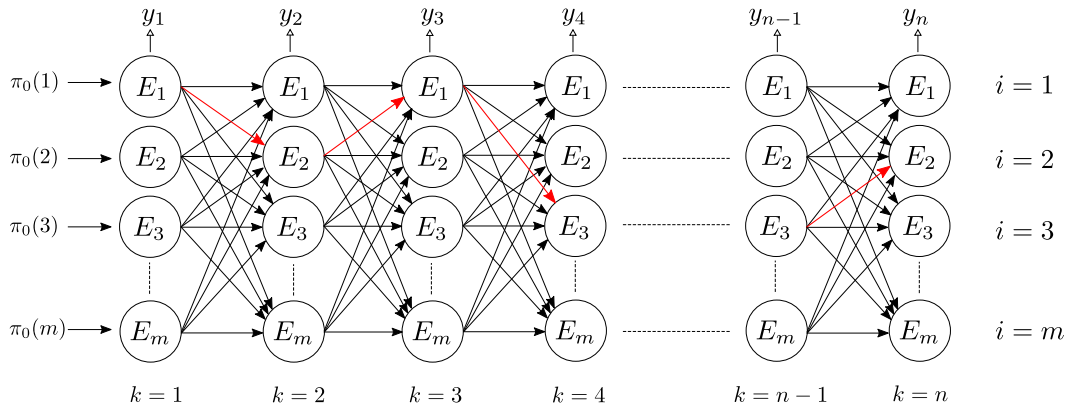


FIGURE 2.20 – Résolution numérique d'un modèle de Markov à états cachés. Chaque colonne représente une étape k, à laquelle on dispose de l'observation y_k. Chaque ligne représente les différents états possibles à l'étape k.

Pour réduire le nombre d'états du modèle et limiter les temps de calcul, il est indispensable de sélectionner en amont les sous-ensembles $E_t \subset E$ des tronçons de route situés dans un voisinage [5] des observations GPS à chaque pas de temps t.

Les coordonnées finales map-matchées sont alors obtenues par projection suivant la plus courte distance des observations GPS Y, sur les tronçons $\hat{X}$ auxquels elles ont été rattachées par l'algorithme de Viterbi (figure 2.24). On peut quantifier la qualité d'un map-matching à l'aide de la racine carrée de l'erreur quadratique moyenne (rmse) des déplacement induits par le recalage :

5. En pratique, on peut prendre un voisinage circulaire de rayon égal à 5 à 10 fois l'écart-type du GPS

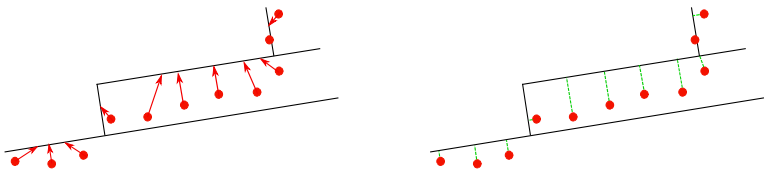


FIGURE 2.21 – Map-matching des observations GPS sur le réseau routier par Viterbi (à gauche) et projection géométrique sur les tronçons auxquels elles ont été affectées (à droite).

$$r_{mm} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{2.43}$$

où la séquence des $\hat{y}_i$ correspond aux projections des points GPS sur le réseau routier.

Le rmse est un bon indicateur de la qualité d'un map-matching, mais il ne peut être utilisé pour paramétrer l'algorithme. En effet, en prenant l'exemple de la figure 2.22 qui donne un exemple d'application de l'algorithme de map-matching sur des données GPS réelles, on observe que le rmse du recalage par plus proche voisin (ppv) vaut 9.75 m, alors que celui du recalage par HMM vaut 12 m. En observant les résultats produits, on dénombre 7 points recalés de manière incorrecte pour la méthode ppv. Avec la méthode HMM, tous les points sont recalés sur le bon segment.

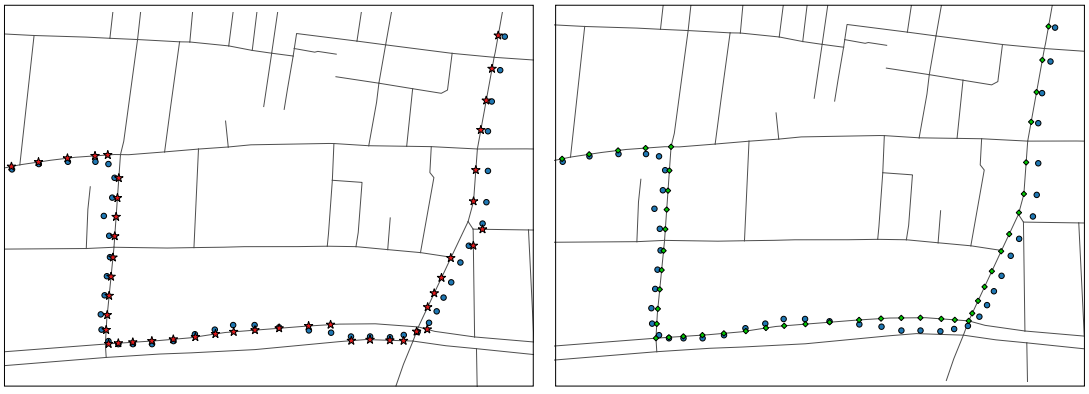


FIGURE 2.22 – Recalage d'une trajectoire GPS (en bleu) sur un réseau routier de référence : par plus proche voisin (à gauche en rouge) et par HMM (à droite en vert).

Pour compléter le modèle, on peut ajouter des informations de vitesse (lorsque les timestamps sont disponibles), de cap ou encore de limitations de vitesse sur le réseau.

2.4.3 Quelques contributions au map-matching

2.4.3.1 Optimisation du temps de calcul

a. Map-matching par morceaux

La complexité de l'algoritme de Viterbi est un $\mathcal{O}(m^2)$ avec m le nombre d'états du modèle, c'est-à-dire avec le nombre de tronçons de route que comporte le graphe du réseau. La taille de l'emprise géographique couverte par la zone d'étude peut donc être prohibitive pour le map-matching des données. Une solution consiste à diviser la zone et à appliquer l'algorithme de Viterbi indépendamment sur des sous-ensembles du graphe. La probabilité initiale de chaque section est déterminée grâce à la position du dernier point map-matché de la section précédente. Cette solution est efficace puisqu'elle opère une résolution de type *diviser pour régner* sur le nombre de tronçons du réseau, qui intervient dans la complexité de l'algorithme de manière supralinéaire. On notera cependant que des erreurs peuvent survenir au niveau des jonctions de zones. Une solution efficace pour gérer ce problème consiste à créer un recouvrement entre les différents secteurs, permettant ainsi à 2 applications successives de l'algorithme de Viterbi de travailler sur des données en partie commune et donc d'obtenir des résultats concordants, comme illustré sur la figure 2.23.

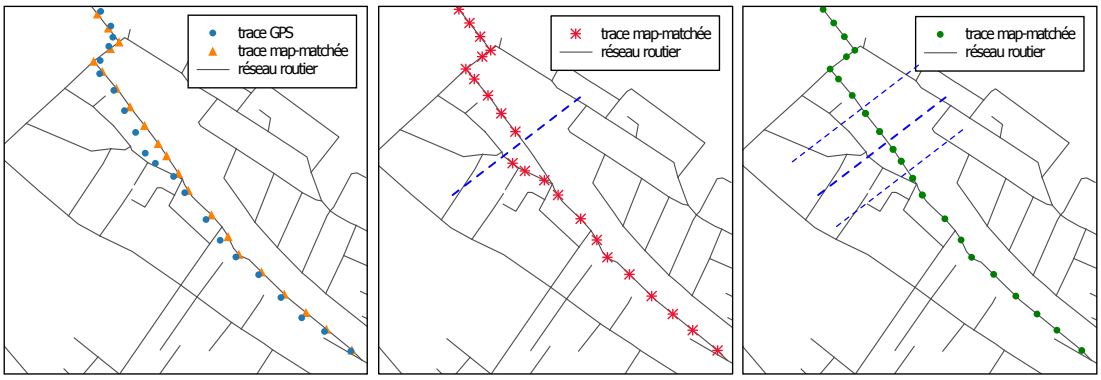


FIGURE 2.23 – À gauche : map-matching d'une trajectoire GPS en un seul bloc. Au centre : map-matching par blocs sans recouvrement. La ligne en pointillés bleue représente la coupure entre les deux blocs. À droite : map-matching par blocs avec recouvrement de 2 points. Les lignes en pointillés bleues représentent la zone de recouvrement.

Cette optimisation convient parfaitement pour traiter des trajectoires contenant un grand nombre de points, ou plus spécifiquement des trajectoires s'étendant sur une large emprise géométrique.

b. Map-matching par sous-échantillonnage

Le nombre n de points à recaler est quant à lui moins pénalisant puisqu'il n'intervient qu'en $\mathcal{O}(n)$ dans la complexité de l'algorithme. Toutefois, à haute fréquence d'acquisition comme c'est le cas ici, un gain de temps non-négligeable peut être obtenu en sous-échantillonnant la trajectoire, par exemple d'un facteur dix. Ainsi, seul un dixième des points sont effectivement utilisés dans la procédure de map-matching, les autres sont ajustés par interpolation linéaire sur le réseau entre les points repositionnés par Viterbi (figure 2.24). Le facteur de sous-échantillonnage est directement dépendant de la fréquence d'acquisition des données et de la longueur minimale d'un arc du graphe. Par exemple,

pour des données acquises à 10 Hz, sur un réseau routier en milieu urbain (vitesse maximale supposée à 50 km/h) avec une longueur minimale de 25 m pour un tronçon de route, un facteur de sous-échantillonnage de l'ordre de 10 à 20 est suffisant pour assurer d'avoir au moins un point map-matché sur chaque arc effectivement parcouru par le véhicule.

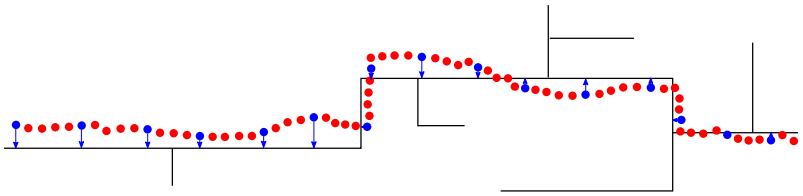


FIGURE 2.24 – Map-matching des observations GPS sous-échantillonnées sur le réseau routier par Viterbi (en bleu) et interpolation des points intermédiaires (en rouge).

c. Pré-calcul des distances sur le graphe

Dans le cas de figure où on souhaite recaler un grand nombre de traces sur une zone d'emprise limitée (c'est le cas dans la plupart des travaux menés dans cette thèse, et en général dans le cadre du map inference), il arrivera nécessairement d'avoir à calculer plusieurs fois une distance suivant le réseau entre deux même points. Pour optimiser le temps de traitement, une approche possible consiste à pré-calculer en amont les distances entre tous les couples de sommets du graphe routier. Par simplicité, le calcul peut être effectué à l'aide de l'algorithme de Floyd-Warshall (Floyd, 1962) qui évalue les distances entre tous les couples de sommets d'un graphe $G(V, E)$ en un temps $\mathcal{O}(|V|^3)$. Le stockage des distances calculées nécessite un espace mémoire qui croît $\mathcal{O}(|V|^2)$, rendant ainsi cette solution inadaptée dès lors que le réseau routier compte plus d'une vingtaine de milliers d'intersections [6], ce qui correspond à une zone urbaine d'une vingtaine de km². Deux options principales permettent de contourner ce problème :

- Supprimer tous les nœuds de degré 2 dans le graphe routier, en particulier pour les réseaux informatiquement mal conditionnés, dans lesquels les vertex (au sens géométrique du terme, *i.e.* les sommets des polylignes modélisant la géométrie du réseau routier) sont modélisés par des nœuds topologiques. Les expérimentations menées sur les réseaux routiers des villes de Mitaka et Tsukuba (*cf* chapitre 4) ont montré que cette solution permettait de supprimer jusqu'à 37% des nœuds et 28% des arcs.

- Stocker les distances en forme matricielle creuse, par exemple dans une table de hachage, permettant ainsi un accès en temps constant à chaque distance du réseau, sans imposer un stockage des distances dépassant un certain seuil. On pourra fixer cette valeur de seuil en prenant la distance maximale qu'un véhicule peut parcourir entre deux points d'une trajectoire. En fixant à r ce seuil, et en considérant un réseau urbain de densité moyenne [7] d, la taille nécessaire au stockage dans une table de hachage devient $K\pi r^2 d|V|$ où K est une constante dépendant de l'implémentation

6. Avec un stockage des distances en nombres flottants simple précision, la taille recquise pour le pré-calcul de la table est de 1 Go pour un réseau contenant 16 384 nœuds

7. Typiquement entre 100 et 400 nœuds par km² pour un réseau numérique de qualité décente et suivant que la zone à traiter est composée de péri-urbain ou d'urbain dense

nb pts GPS	100	1000	10000	jeu complet ($8.5\ 10^6$)
sans pré-calcul	1"76	8"50	59"13	14 h
avec pré-calcul	8"40	8"62	9"68	22 min

TABLE 2.2 – Temps de calcul pour le map-matching sur un réseau de 4884 points (16 km²).

du mécanisme de hachage. Par exemple, en Java, avec les configurations par défaut et un enregistrement des distances en flottants simple précision : $K \approx 67$ octets. En prenant $r = 0.1$ km et $d = 200$ nœuds/km², l'espace mémoire nécessaire s'élève à 420 octets par point. Un espace mémoire alloué de seulement 10 Mo devient suffisant pour traiter un graphe contenant de l'ordre de 25 000 points, soit l'équivalent de la surface de Paris en urbain moyen.

Un intérêt subsidiaire de cette manière de procéder, réside dans la possibilité d'écrire la matrice des distances calculées dans un fichier pour une utilisation ultérieure, ce qui peut constituer un avantage considérable dans un contexte de map inference incrémental.

En contre-partie, avec cette stratégie, l'algorithme de Floyd-Warshall ne peut plus être utilisé, et le calcul des distances doit être effectué par appels itératifs de l'algorithme de Dijkstra, avec une complexité totale en $\mathcal{O}(r^4 \log r)$.

L'analyse de la complexité permet de répondre à la question de savoir à partir de combien de traces la méthode de pré-calcul des distances sur le réseau devient plus avantageuse. En fixant les paramètres r et d (respectivement à 0.1 km et 200 points/km²), on obtient une complexité égale à $18N|V| \log V$ (où N est le nombre de points GPS à traiter) pour la méthode sans pré-calcul et à $3|V|^2 \log |V| + N$ avec pré-calcul. On peut montrer que, la méthode avec pré-calcul devient plus avantageuse dès lors que :

$$\frac{|V|}{6N} + \frac{1}{18|V|} \approx \frac{|V|}{6N} \leqslant 1, \tag{2.44}$$

où le facteur 6 est une approximation, dépendant de l'implémentation informatique de l'algorithme de Dijkstra. En pratique, on pourra retenir que le pré-calcul des distances est intéressant dès que le nombre de points à traiter et d'un ordre de grandeur sensiblement plus grand que le nombre d'intersections topologiques du réseau.

Nous avons testé les deux méthodes sur le réseau routier de la ville de Mitaka (Japon), avec les traces GPS fournies par la compagnie Navitime (*cf* chapitre 4). Les résultats sont donnés dans le tableau 2.2 et montrent un gain significatif dans ce cas d'application.

d. Bufferisation du graphe

Enfin, une dernière option pour acclérer le calcul du map-matching consiste à extraire un *buffer* du réseau autour des trajectoires en amont du pré-calcul des distances et/ou du recalage. Lorsque toutes les traces passent sensiblement par le même chemin, on peut ne calculer que le buffer de la première trace du jeu de données. C'est la méthode que nous avons employée dans le chapitre 3. Si les trajectoires sont réparties de manière éparse sur

la zone, on peut calculer l'union des buffers de toutes les trajectoires, et extraire l'intersection du graphe routier avec ce buffer (figure 2.25). Cette optimisation ne présente plus d'avantage notable lorsque la couverture du réseau par les traces est relativement dense.

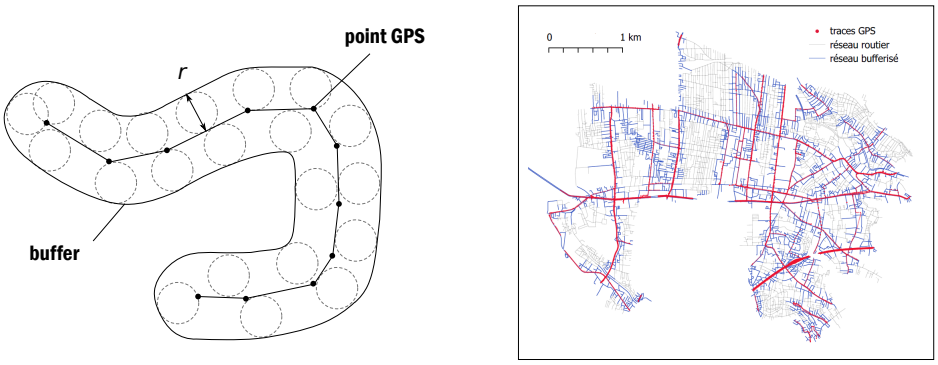


FIGURE 2.25 – À gauche : illustration schématique du calcul d'un buffer sur une trajectoire GPS. À droite : extraction de l'intersection du réseau routier avec 100 traces *bufferisées* ($r = 50$ m) en amont de la procédure de map-matching.

Le gain en complexité (mémoire et temps) est grossièrement proportionnel au ratio de la zone couverte par la réunion des buffers sur la surface totale du réseau routier.

2.4.3.2 Autocorrélation spatio-temporelle des erreurs de mesure

Nous avons vu dans la section 2.2.2 que la matrice de covariance de la position estimée par GPS est intrinsèquement dépendante de la position relative des satellites par rapport au récepteur. Or, la configuration des satellites de la constellation évolue de manière continue dans le temps, impliquant ainsi que les erreurs de positionnement d'un point fixe dans le référentiel GPS sont temporellement autocorrélées (Roberts, 1993) [8]. De manière similaire, au niveau local, la qualité du positionnement dépend des obstacles environnant le récepteur, notamment via l'obstruction des lignes du vue (diminuant ainsi le nombre de satellites disponibles pour le calcul de la position) et les multi-trajets. La continuité locale de cet environnement physique se traduit également par une autocorrélation spatiale des erreurs à tout instant entre deux points fixes distincts dans le référentiel GPS (Hoang et al., 2016). Dans notre cadre d'étude, la combinaison de ces deux effets se traduit par une dépendance entre les erreurs de positionnement d'un vehicule le long de sa trajectoire, et par suite, par une dépendance entre les résidus de l'équation 2.43.

Dans certaines applications, plus particulièrement celles où on souhaite calculer une grandeur cumulée par intégration, l'autocorrélation d'une suite de variables aléatoires est une source d'erreur supplémentaire. À l'inverse, nous avons vu dans le paragraphe 2.2.3 qu'elle peut aussi parfois réduire les incertitudes sur le résultat recherché.

Dans cette section, nous tentons de prendre en compte cette autocorrélation pour raffiner les résultats obtenus par l'algorithme de map-matching de (Newson et Krumm, 2009).

8. Les erreurs d'horloge sont également autocorrélées dans le temps mais d'effet secondaire dans le cadre du positionnement GPS par code.

Dans un premier temps, introduisons quelques définitions afin de pouvoir modéliser la série des erreurs des résidus du recalage.

Soient $(\Omega, \mathcal{A}, P)$ un espace probabilisé complet et $\mathcal{D}$ un espace topologique (en général $\mathcal{D} \subseteq \mathbb{R}^n$ désigne un intervalle d'espace ou de temps).

Définition 2.6. *On appelle processus stochastique à temps continu et à valeurs réelles, une fonction $Z : \Omega \times \mathcal{D} \to \mathbb{R}$ telle que :*

- *Pour tout point $x \in \mathcal{D}$, $Z(., x)$ est une variable aléatoire représentant l'incertitude sur la valeur prise par un signal aléatoire en x.*
- *Pour tout évènement élémentaire $\omega \in \Omega$, $Z(\omega, .)$ est une fonction classique de $\mathcal{D}$ dans $\mathbb{R}$. On parle d'une* **trajectoire** *du processus stochastique.*

Ainsi, le processus stochastique est une généralisation continue et à tout ordre des chaînes de Markov introduites dans la définition 2.4. La figure 2.26 donne un exemple de 2 trajectoires d'un processus stochastique.

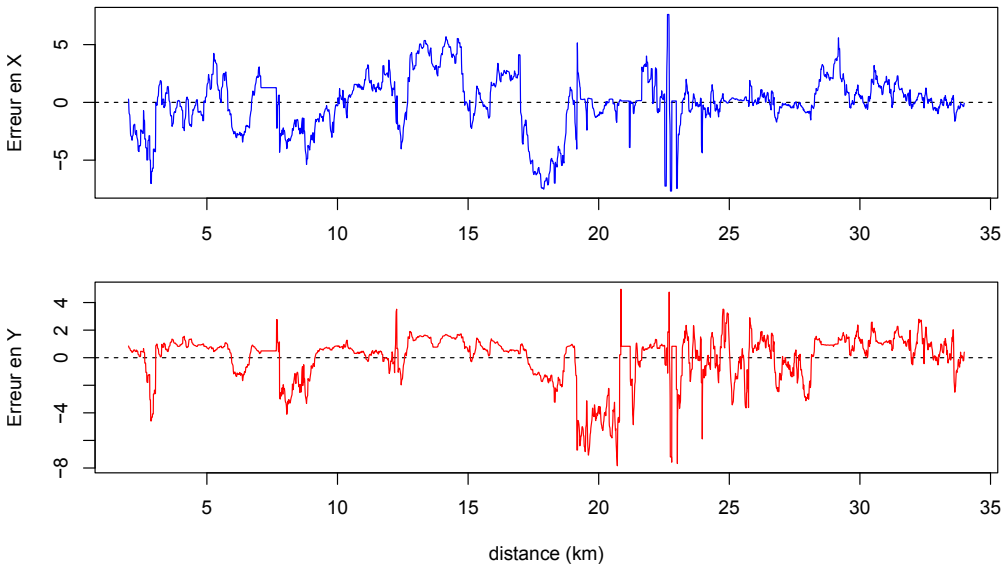


FIGURE 2.26 – Profils spatiaux des résidus (en X et en Y) de map-matching sur le réseau routier de Tsukuba : 2 trajectoires d'un processus stochastique continu à valeurs réelles.

Un processus Z est entièrement déterminé par ses lois jointes *fini-dimensionnelles* : $F^Z_{x_1,...x_k}(z_1, ...z_k) = P[Z(\omega, x_1) \leqslant z_1, ...Z(\omega, x_k) \leqslant z_k]$. La connaissance de la famille (infinie) de ces lois est impossible en pratique, et on cherche plutôt à caractériser Z à l'aide de ses premiers moments, plus particulièrement sa moyenne $\mathbb{E}[Z(x)]$ et sa covariance $\mathrm{Cov}[Z(x), Z(x')]$, pour tous les points x et x' du domaine $\mathcal{D}$. Dans la suite, on suppose que Z est d'espérance nulle, éventuellement moyennant un débiaisage de la série en amont

de l'analyse. Cette hypothèse nous permet de confondre autocovariance et autocorrélation :

$$\gamma(x_1, x_2) = \mathbb{E}[Z(x_1)Z(x_2)] = \int_\Omega Z(x_1, \omega)Z(x_2, \omega)dP(\omega). \tag{2.45}$$

Nous avons mentionné en introduction que l'autocovariance des résidus de map-matching était un phénomène spatio-temporel. En toute rigueur, le processus stochastique associé devrait être défini sur un domaine $\mathcal{D} \subseteq \mathbb{R}^2$. Dans notre cadre d'application, nous pouvons emettre l'hypothèse que la principale source d'erreur de positionnement est due aux obstructions du bâti environnant et aux multi-trajets. La vitesse de déplacement des vehicules rend en particulier secondaire l'autocorrélation temporelle des résidus. Nous faisons donc ici le choix de modéliser uniquement la composante spatiale de l'autocorrélation, et en conséquence nous ne représenterons que des profils spatiaux de résidus, à l'instar de la figure 2.26, que nous considérons comme plus informatifs que les profils temporels.

Une propriété souhaitable pour les processus stochastiques, est de se comporter identiquement (d'un point de vue probabiliste) en tout point du domaine d'analyse : l'hypothèse de *stationnarité* au sens strict fait appel à la caractérisation de Z par ses lois jointes finidimensionnelles, et se révèle donc être beaucoup trop lourde en pratique. On lui préfère la stationnarité à l'ordre 2 (dite aussi stationnarité au sens faible, ou au sens large), qui stipule uniquement une uniformité spatiale des 2 premiers moments du processus.

Définition 2.7. *Un processus stochastique Z est dit stationnaire à l'ordre 2, si et seulement si :*

- (1) *L'espérance de Z est indentique en tout point : $\mathbb{E}[Z(x)] = m$ (indépendant de x).*
- (2) *La covariance de Z est invariante par translation : $\gamma(x_1, x_2) = \gamma(x_1 + t, x_2 + t)$*

Du point (2) de la définition 2.6, il vient immédiatement que sous l'hypothèse de stationnarité, la fonction d'autocorrélation ne dépend que de la distance entre les sites :

$$\gamma(x_1, x_2) = \gamma(x_1 - x_2, x_2 - x_2) = \gamma(x_1 - x_2, 0) = \gamma(\tau), \tag{2.46}$$

où $\tau \in \mathcal{D}$ est le vecteur de translation séparant x_1 et x_2. Pour un processus uni-dimensionnel et stationnaire au sens large, la fonction d'autocorrélation ne dépend que d'une seule variable : la distance entre les deux sites de mesure.

Pour un modèle d'erreur GPS, la stationnarité au sens large peut être admise en première approximation, bien que localement non valide, en particulier pour des tissus urbains fortement inhomogènes, dans lesquels les zones baties denses induisent une plus grande variance ainsi que potentiellement une autocorrélation plus marquée à courte échelle. L'allure des profils d'erreurs de la figure 2.26, relevés sur un long trajet de 35 km dans la ville de Tsukuba (morphologies urbaines très variées), confirme la validité de l'hypothèse de stationnarité au sens large.

La stationnarité étant admise, on peut calculer l'autocorrélation des profils d'erreurs, par estimation numérique :

$$\gamma(j) = \frac{1}{n-j} \sum_{i=j+1}^{n} Z(x_i)Z(x_i - j\Delta_x). \tag{2.47}$$

Notons que pour que l'expression 2.47 soit opérante, il est nécessaire d'interpoler le profil d'erreur régulièrement en $(x_1, x_2, ...x_n)$. Pour ce faire, on utilise les techniques décrites dans la section 2.3.2.4 traitant de l'interpolation et du lissage des profils de vitesse.

Le résultat de l'application de 2.47 au profil des résidus en X de la figure 2.26 est illustré sur le graphe de gauche de la figure 2.27. Il s'agit d'une fonction symétrique dont la valeur en $\gamma(\tau)$ traduit la ressemblance entre deux résidus de map-matching séparés par une distance τ le long du trajet. Conformément à l'intuition, l'autocorrélation est maximale en 0, et tend à s'annuler en l'infini. Notons que la décroissance monotone de γ (sur chaque demi-espace) n'est pas une propriété systématique de l'autocorrélation. D'autre part, le choix de la métrique suivant l'abscisse curviligne du trajet du véhicule découle de l'hypothèse que le parcours ne comporte pas de *marche arrière*.

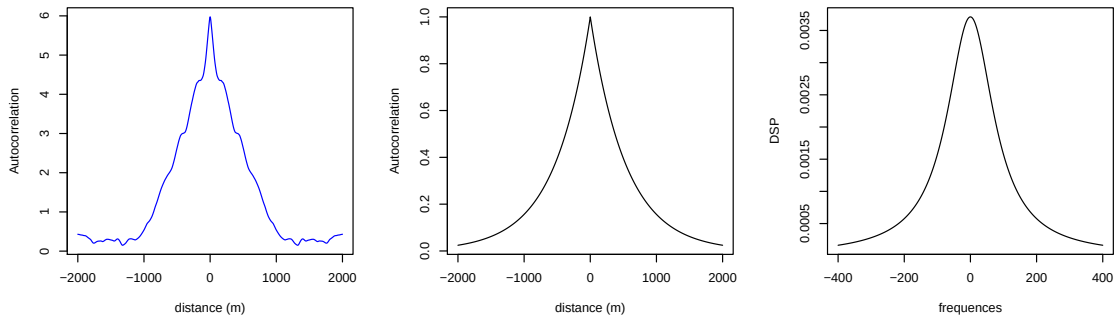


FIGURE 2.27 – À gauche : autocorrélation des résidus du map-matching suivant la direction X pour le profil d'erreur de la figure 2.26. Au centre : modèle exponentiel théorique de la fonction d'autocorrélation. À droite : densité spectrale de puissance du processus.

L'*ergodicité* est une seconde propriété souhaitable des signaux aléatoires, et consiste à supposer que les propriétés statistiques d'un processus Z peuvent être inférées à partir d'un unique trajectoire $Z(\omega, .)$, à condition que le domaine d'observation de Z soit suffisamment grand. Il est généralement très difficile en pratique de démontrer la validité de cette hypothèse. De manière similaire à la stationnarité, on se restraint à la propriété d'ergodicité pour la moyenne et la covariance :

$$\mathbb{E}[Z] = \int_{\Omega} Z(\omega, x)dP(\omega) = \lim_{\Delta x \to \infty} \int_{-\frac{\Delta x}{2}}^{+\frac{\Delta x}{2}} z(x)dx, \tag{2.48}$$

$$\gamma(\tau) = \mathbb{E}[Z(x)Z(x+\tau)] = \lim_{\Delta x \to \infty} \int_{-\frac{\Delta x}{2}}^{+\frac{\Delta x}{2}} z(x)z(x+\tau)dx, \tag{2.49}$$

où $z(x) = Z(\omega, x)$ pour un évènement élémentaire $\omega \in \Omega$ quelconque fixé, désigne une réalisation du processus aléatoire.

Notons que cette hypothèse n'est absolument pas triviale, et ne peut être comparée à la loi des grands nombres : dans le cadre des processus stochastiques, l'ergodicité stipule que les lois de probabilités sont *estimables* à partir d'une unique réalisation ω. Il existe de nombreux contre-exemples théoriques, par exemple le signal aléatoire constant sur son domaine analytique $Z(\omega, x) = Z(\omega)$, pour lesquels l'hypothèse d'ergodicité n'est pas vérifiée. Sous l'hypothèse d'ergodicité, la limite 2.49 garantit la convergence de l'approximation 2.47. Nous supposerons ici que le processus des erreurs GPS est ergodique, et nous tenterons d'argumenter a posteriori sur la validité de ce choix.

Arrêtons-nous un instant sur le coût computationnel du calcul de l'autocorrélation. L'implémentation directe de la formule 2.47 nécessite un nombre de multiplications de l'ordre de $\mathcal{O}(n^2)$ où n est le nombre de points d'une trace GPS. Pour des traces suffisament longues, n peut être de l'ordre de plusieurs dizaines de milliers de points. Le théorème de Wiener-Khintchine (Kschischang, 2017) permet une accélération du calcul.

Définition 2.8. *Soit Z un processus stochastique stationnaire et ergodique (au sens large). On appelle* **densité spectrale de puissance** *(DSP) le carré du module de la transformée de Fourier du signal, divisé par la largeur de l'intervalle d'intégration :*

$$\Gamma(f) = \lim_{\Delta x \to +\infty} \frac{1}{\Delta x} \left| \frac{1}{\sqrt{2\pi}} \int_{-\frac{\Delta x}{2}}^{+\frac{\Delta x}{2}} z(x) e^{2\pi i f x} \right|^2. \tag{2.50}$$

Théorème 2.2 (Wiener-Khintchine). La densité spectrale de puissance d'un processus aléatoire Z stationnaire et ergodique au sens large est analogue à la transformée de Fourier de sa fonction d'autocorrélation.

$$\gamma(\tau) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \Gamma(f) e^{-2\pi i f x}. \tag{2.51}$$

La preuve repose sur l'analogie du produit de convolution et de la multiplication dans les espaces spatiaux et fréquentiels. On pourra trouver plus de détails dans l'ouvrage de Picinbono (1998).

Le calcul pratique de 2.47 peut donc être accéléré en calculant la transformation de Fourier du signal z ($\mathcal{O}(n \log n)$ multiplications avec l'algorithme de FFT), en élevant son module au carré ($\mathcal{O}(n)$ multiplications) puis en prenant la transformation de Fourier inverse (à nouveau $\mathcal{O}(n \log n)$ multiplications), pour une complexité totale quasi-linéaire (à comparer avec le coût quadratique de l'implémentation directe du produit de convolution 2.47) :

$$\gamma = \mathcal{F}^{-1}[(\mathcal{F}[z])^2]. \tag{2.52}$$

Nous avons appliqué ce schéma de calcul sur les 26 000 traces collectées par la compagnie Navitime sur la ville de Tsukuba, pour un temps total de traitement de l'ordre de 10 minutes, contre 6h30 pour le calcul direct.

On ajuste alors un modèle théorique γ sur la fonction d'autocorrélation expérimentale. On sait qu'une fonction d'autocorrélation doit être de type défini positif. Le théorème de Bochner (Barret, 2009) nous dit que γ doit être la transformée de Fourier d'une mesure positive bornée sur $\mathbb{R}$, la densité spectrale de puissance. La fonction $\gamma(\tau) = \exp(-a|\tau|)$ ($a \in \mathbb{R}^+$) est la transformée de Fourier d'une fonction lorentzienne (figure 2.27 à droite) :

$$\Gamma(f) = \frac{2a}{a^2 + 4\pi^2 f^2}. \tag{2.53}$$

La fonction Γ étant strictement positive, le modèle exponentiel symétrique (figure 2.27 au centre) est donc bien un modèle théorique de fonction de corrélation. D'autre part, elle correspond à la forme de l'autocorrélation empirique (figure 2.27 à gauche). Un ajustement par moindres carrés nous donne une estimation du paramètre d'échelle $\hat{a}^{-1} = 539$ m (erreur relative d'ajustement de 4 %). Ce paramètre correspond à la portée de corrélation des erreurs GPS dans notre jeu de données. À la vitesse moyenne de 33 km/h, cette distance correspond à un intervalle de temps de l'ordre de la minute, durée bien inférieure aux temps typiques d'autocorrélation des mesures de pseudo-distances (Roberts, 1993), ce qui confirme a posteriori la prépondérance de la dimension spatiale de l'autocorrélation des erreurs dans notre cadre d'étude. La figure 2.28 illustre la procédure d'ajustement.

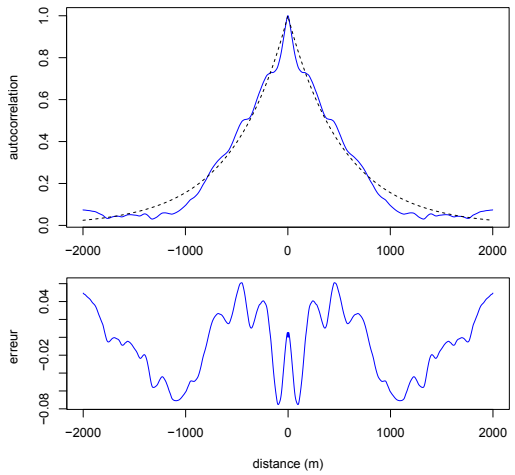


FIGURE 2.28 – En haut : ajustement par moindres carrés du modèle exponentiel symétrique sur la fonction d'autocorrélation empirique. En bas : erreur d'ajustement.

Le modèle exponentiel est classiquement utilisé pour modéliser les erreurs GPS, par exemple dans Grejner-Brzezinska et al. (2005). D'autre part, le fait que la fonction d'autocorrélation calculée sur le jeu de données complet s'ajuste de manière satisfaisante (erreur relative de 7%) à la trajectoire 2.26, appuie (sans valider complètement toutefois) l'hypothèse d'ergodicité du processus.

Cette fonction γ peut être alors directement utilisée pour améliorer les résultats du map-matching, par exemple, en supposant que le processus stochastique des résidus est un processus gaussien de fonction de covariance γ (ce qui semble une hypothèse raisonnable).

Définition 2.9. *Soient $k : \mathcal{D}^2 \to \mathbb{R}_+$, une fonction de type défini-positif et $m : \mathcal{D} \to \mathbb{R}$ une fonction quelconque. Un processus stochastique à temps continu Z est un processus gaussien de moyenne m et de fonction de covariance k si pour tout ensemble fini de points du domaine analytique $\mathbf{x} = (x_1, x_2 ... x_n) \in \mathcal{D}$, le vecteur aléatoire $Z(\mathbf{x})$ est une loi normale multivariée de moyenne $m(\mathbf{x})$ et de covariance $K \in \mathbb{R}^{n \times n}$ avec $K_{ij} = k(x_i, x_j)$.*

Lorsque le processus gaussien est en plus stationnaire, la matrice de covariance des lois jointes fini-dimensionnelles s'exprime par : $K_{ij} = \gamma(x_i - x_j)$. Notons que la fonction de covariance d'un processus gaussien n'est pas nécessairement gaussienne (*cf* figure 2.29). C'est le cas ici, où nous allons utiliser un GP de moyenne nulle et de fonction de covariance exponentielle symétrique.

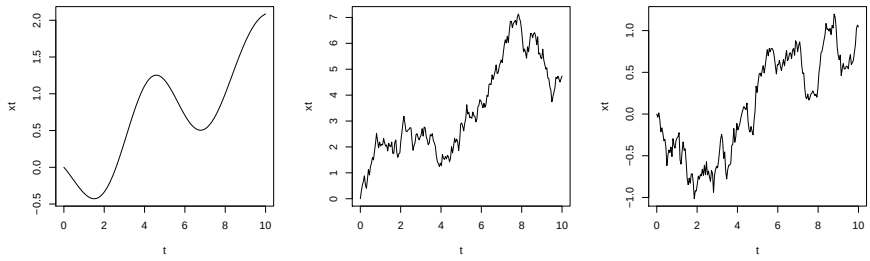


FIGURE 2.29 – Trois exemples de réalisations de processus gaussiens sur le domaine $\mathcal{D} = [0, 10]$, avec trois noyaux de corrélation différents. À gauche : noyau gaussien de variance 10. Au centre : processus (non-stationnaire) de Wiener $\gamma(s, t) = \min(s, t)$, analogue à une marche aléatoire continue. À droite : noyau exponentiel symétrique de portée $a^{-1} = 10$.

Un intérêt fondamental dans l'utilisation d'un processus gaussien (en anglais gaussian process, ou GP), est celui de pouvoir exprimer les lois conditionnelles sous une forme close. Plus spécifiquement, étant donné X un vecteur aléatoire gaussien (de dimension quelconque supérieure ou égale à 2) respectivement de moyenne et de matrice de covariances :

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^T & \Sigma_{22} \end{bmatrix}$$

Alors, la loi conditionnelle de $\mathbf{x}_1$ sachant $\mathbf{x}_2$ est une loi normale respectivement de moyenne et de covariance :

$$\mathbb{E}[\mathbf{x}_1 | \mathbf{x}_2 = a] = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2) \qquad \Sigma_{\mathbf{x}_1|\mathbf{x}_2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^T. \tag{2.54}$$

La preuve de ces résultats pourra être trouvée dans Lindgren et al. (2013).

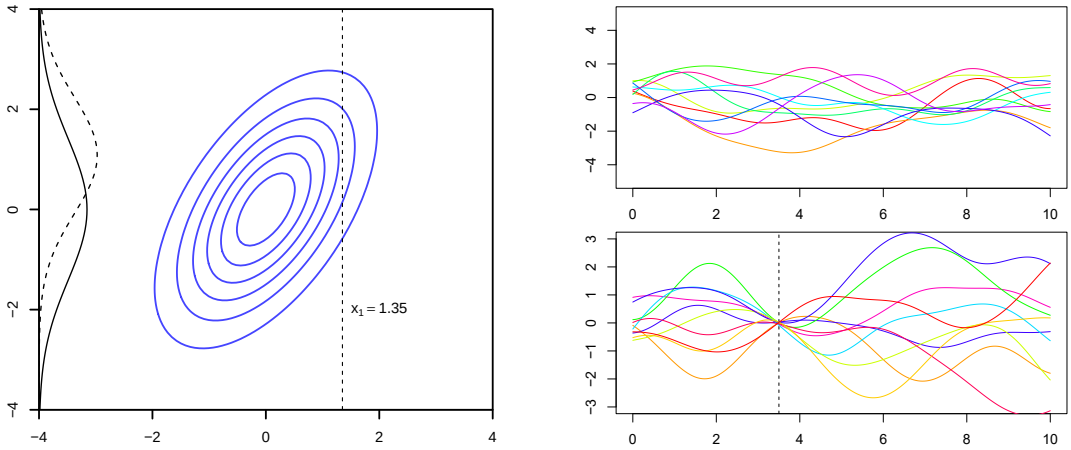


FIGURE 2.30 – À gauche : loi normale multivariée sur $\mathbb{R}^2$ (corrélation $\rho_{12} = 0.57$). À gauche en trait plein : loi marginale de x_2. En pointillés : loi conditionnelle $p(x_2|x_1 = 1.35)$. À droite : exemples de processus gaussiens pour une fonction de corrélation normale (en haut) et processus gaussiens conditionnés à la valeur 0 au point $x = 3.5$ (en bas).

On peut alors exprimer la probabilité de transition des erreurs par une loi normale :

$$e_{n+1}|e_n \sim \mathcal{N}\Big(\frac{\gamma(\tau_n)}{\gamma(0)}e_n, \gamma(0) - \frac{\gamma(\tau_n)^2}{\gamma(0)}\Big), \tag{2.55}$$

où $\mathcal{N}(m, \Sigma)$ désigne la loi normale de moyenne m et de variance Σ, et où $e_n = y_n - \hat{y}_n$ représente l'écart (dans une direction donnée) entre une observation GPS, et sa position estimée sur le réseau routier. La quantité τ_n représente la distance (suivant le réseau) entre deux points $\hat{y}_n$ et $\hat{y}_{n+1}$. On vérifie bien que, pour tout fonction γ, définie-positive et continue en 0 et tel que $\gamma(\tau)$ tend vers 0 en l'infini :

$$p(e_{n+1}|e_n) \underset{\tau_n \to 0}{\to} \mathcal{N}(e_n, 0) = e_n, \tag{2.56}$$

$$p(e_{n+1}|e_n) \underset{\tau_n \to +\infty}{\to} \mathcal{N}(0, \gamma(0)) = p(e_{n+1}), \tag{2.57}$$

où les limites 2.56 et 2.57 sont à comprendre respectivement au sens d'une convergence en probabilités et en loi. La figure 2.31 illustre les situations intermédiaires, sur lesquelles on observe que l'espérance de la loi conditionnelle tend à regresser vers la moyenne, tandis que la variance tend à croître vers la variance a priori, à mesure que la distance parcourue par le véhicule depuis l'observation précédente augmente. La partie droite de la figure 2.30 donne une illustration (générée à l'aide de données synthétiques) de ce phénomène : lorsque le point $x_0 = 3.5$ est observé, la dispersion des réalisations au voisinage de x_0 est moindre comparée à la variance a priori.

L'équation 2.55 peut alors être insérée dans le critère d'optimisation du map-matching. En partant de la forme logarithmique 2.42, X^* appartient à l'argmin sur E^n de :

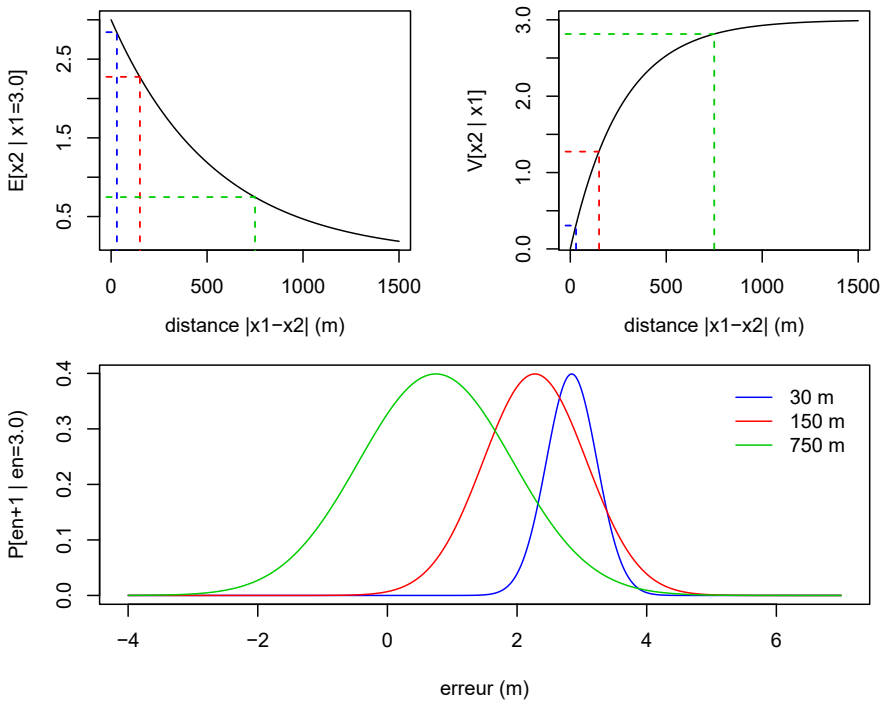


FIGURE 2.31 – En haut : évolution de l'espérance $\mathbb{E}[x_2|x_1 = 3]$ (gauche) et de la variance $\mathrm{Var}[x_2|x_1]$ (à droite) conditionnelles en fonction de la distance $\tau = |x_1 - x_2|$. En bas : 3 exemple de densités de probabilités conditionnelles pour 3 distances τ différentes, sachant que l'erreur précédente est $e_n = 3$ m.

$$\frac{1}{2\sigma^2}\sum_{i=1}^{n-1}(y_i - x_i)^2 + \frac{1}{\beta}(\tau_i - ||y_{i+1} - y_i||_2) + \frac{1}{2}\sum_{i=1}^{n}\left(\gamma(0) - \frac{\gamma(\tau_i)^2}{\gamma(0)}\right)\left(e_{i+1} - \frac{\gamma(\tau_i)}{\gamma(0)}e_i\right)^2$$

$$\text{avec :} \quad \gamma(\tau) = \sigma^2 e^{-a|\tau|}, \tag{2.58}$$

et où $\tau_i = d_r(x_i, x_{i+1})$ est la pseudo-distance (orientée) suivant le réseau routier entre les points x_i et x_{i+1}. Dans cette formulation 2.58, le premier terme correspond à la fidélité aux données, et modélise l'erreur normale du GPS. Le second terme assure la validité topologique du déplacement sur le réseau, et est dérivé d'une loi exponentielle symétrique (d'où l'abscence d'exposant 2 sur la différence en forme logarithmique). Le troisième et dernier terme représente l'autocorrélation spatiale des erreurs GPS, modélisée par un processus gaussien d'autocorrélation exponentielle symétrique.

La résolution numérique du problème s'effectue de la même manière que précédemment (cf figure 2.20). La méthode a été testée sur un jeu de 20 trajectoires GPS issues du jeu de données fourni par Navitime, étiquetées à la main, et réparties en deux classes : une première classe où le biais des mesures GPS (et/ou du réseau routier) est significatif, et une seconde classe d'exemples plus simples et plus représentatifs des traces typiques du jeu de données complet. La table 2.3 résume les résultats.

La prise en compte de l'autocorrélation spatiale des erreurs augmente donc significativement les scores de performance de la procédure de map-matching. Les expérimentations

Type de problème	sans autocorrélation (%)	avec autocorrélation (%)
simple	96.7 ± 0.23	99.4 ± 0.04
difficile	92.5 ± 0.51	99.8 ± 0.02

TABLE 2.3 – Comparaison des taux d'affectations correctes sur un échantillon de trajectoires map-matchées avec et sans prise en compte de l'autocorrélation des erreurs GPS.

menées ont permis de mettre en évidence le fait que l'apport est particulièrement significatif avec des traces dont les coordonnées sont légèrement biaisées par rapport au réseau routier, avec une fréquence d'acquisition intermédiaire, de l'ordre de 1 point toutes les 10 à 15 secondes. En effet, à haute fréquence (1 Hz - 10 Hz), la majorité des erreurs de map-matching sont évitées à l'aide de l'analyse globale menée par l'algorithme de Viterbi, en particulier lorsque les attributs du réseau routier (sens uniques, non-communications, limitations de vitesse...) sont bien renseignés. À l'inverse, lorsque la fréquence d'acquisition est trop faible, la distance typique entre deux points successifs est trop importante pour que l'effet du terme d'autocorrélation ait une importance décisive dans le processus. De même, à haute fréquence, le nombre de points à recaler est démultiplié et il devient alors plus rentable (du point de vue de la fonction de coût 2.58) de négliger l'autocorrélation spatiale sur une transition pour diminuer drastiquement le terme d'attache aux données. Une autre limite de ce modèle réside dans le fait que l'autocorrélation effectivement mesurée est en réalité le produit scalaire de l'autocorrélation réelle suivant une direction perpendiculaire aux tronçons du réseau routier, ce qui introduit artificiellement un terme de décorrélation, dû à la structure spatiale de l'itinéraire emprunté. Une solution pour contourner ce problème pourrait être de subdiviser les arcs du réseau de sorte à prendre en compte l'erreur parallèle au déplacement (et pas uniquement l'erreur transversale).

Enfin, une question ouverte sur ce modèle de map-matching pourrait être de savoir si l'estimation de la fonction de covariance peut être effectuée sur le jeu de données à recaler. Autrement dit, peut-on améliorer les performances de map-matching en procédant de la manière itérative suivante :

- (1) Map-matching brut du données avec l'algorithme original (fonction de coût 2.42)
- (2) Estimation de la fonction d'autocorrélation du processus gaussien modélisant les erreurs GPS, à l'aide de la formule directe 2.47 ou de l'optimisation 2.52.
- (3) Réestimation du map-matching avec la fonction d'autocorrélation calculée au point (2), et à l'aide de la fonction coût 2.58.

De plus, si les résultats obtenus à la fin de l'étape (3) sont sensiblement différents de ceux de l'étape (1), on peut imaginer boucler la procédure jusqu'à convergence éventuelle des résultats, à la manière d'un algorithme *expectation-maximization*. Nous laisserons cette question en suspens pour des travaux ultérieurs et conclurons cette section avec une illustration du recalage de profils de vitesse de l'expérimentation *ecoDriver* (figure 2.32).

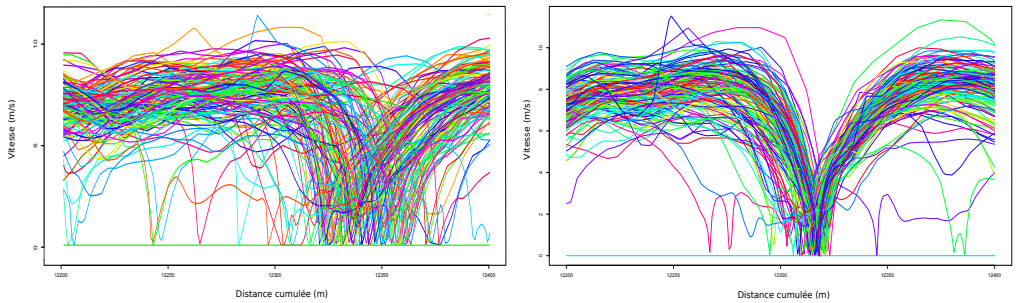


FIGURE 2.32 – Profils de vitesse GPS sur la commune de Versailles avant (à gauche) et après (à droite) recalage des données par map-matching sur un réseau routier de référence.

2.4.4 Analyse du gain de précision géométrique

Dans cette section, nous cherchons à évaluer le gain de précision géométrique (évaluée en termes d'erreur quadratique moyenne sur la position du véhicule) apporté par l'opération de map-matching (2.4.1).

2.4.4.1 Etude préliminaire dans le cas d'un réseau parfait

Afin de mieux appréhender l'impact du map-matching sur la précision des observations, nous avons effectué une simulation, sur un circuit que l'on considérera comme la vérité terrain et sur lequel nous avons simulé aléatoirement une trajectoire à une fréquence de 1 Hz. Pour obtenir une série de points réaliste, la vitesse du véhicule a été fixée sur chaque tronçon comme étant constante et proportionnelle à la longueur du tronçon, de sorte que la vitesse maximale soit de 50 km/h au niveau du tronçon le plus long, contre 25 km/h au niveau du tronçon le plus court. Les mesures GPS sont alors simulées en introduisant une bruit gaussien de moyenne nulle et d'écart-type donné, indépendamment sur chacun des axes X et Y. La séquence d'observations GPS bruitées est alors map-matchée sur le circuit de référence avec la méthode décrite dans les sections précédentes, et les coordonnées obtenues sont comparées avec les coordonnées des points simulés. L'expérience a été conduite dans un premier temps avec un écart-type GPS de 10 m sur chaque axe.

La figure 2.33 montre clairement le gain de précision obtenu par le map-matching lorsque l'on émet l'hypothèse que le circuit de référence correspond à la vérité terrain. Par ailleurs, la mesure de l'écart quadratique moyen entre les observations GPS brutes et les positions réelles du mobile (distribution rouge) vaut 14.10 m, contre 9.75 m pour l'écart entre les observations GPS brutes et les points recalés sur le réseau (distribution verte) soit un gain de 1.45, valeur assez proche de $\sqrt{2}$ que l'on peut conjecturer être la valeur théorique du gain. Éssayons d'expliquer ce résultat en montrant que l'effet du map-matching équivaut à la suppression de la variance du GPS suivant l'une des 2 dimensions du repère.

Commençons par énoncer une version réduite du théorème de Cochran, que l'on peut considérer comme analogue en loi au théorème de Pythagore en géométrie (Bardet, 2006).

Théorème 2.3 (Cochran). *Soient E un sous-espace vectoriel de $\mathbb{R}^n$ de dimension m et $X = (X_1, ..., X_n)^T \in \mathbb{R}^n$ un vecteur gaussien centré de matrice de variances-covariances $\sigma^2 I_n$. On note $\pi_E : \mathbb{R}^n \to E$ l'application linéaire définissant la projection orthogonale sur E, de matrice P_E et $E^\perp = \{\mathbf{x} \in \mathbb{R}^n \mid \forall\ y \in E\ \langle \mathbf{x}, y \rangle = 0\}$, l'espace orthogonal de E dans*

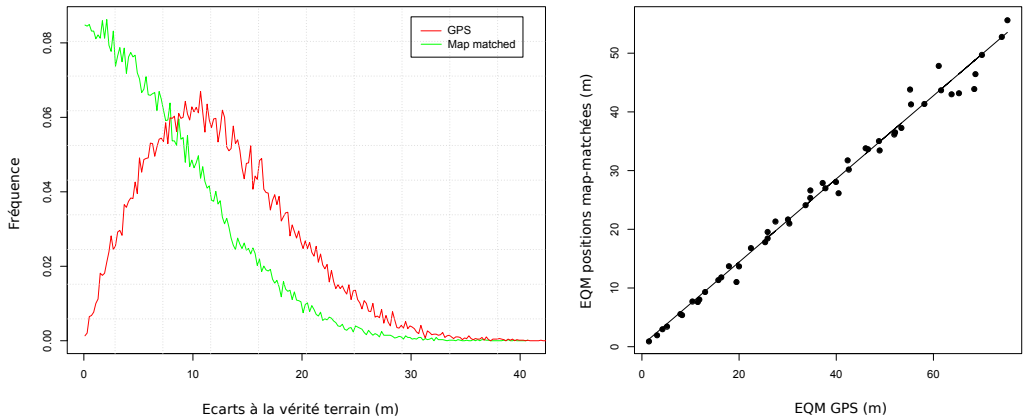


FIGURE 2.33 – À gauche : distribution des écarts entre les points simulés et 1) les observations bruitées (en rouge) avec $\sigma_{gps} = 10\ m$ et 2) les observations recalées sur le réseau routier (en vert). À droite : racine carrée de la moyenne quadratique des erreurs (RMSE) sur la position GPS (en abscisse) et sur la position map-matchée sur le réseau de référence (en ordonnée).

$\mathbb{R}^n$. *On a alors les deux propriétés suivantes :*

(i) les vecteurs aléatoires issus respectivement des projections orthogonales de X *sur* E *et* $E^\perp$ *sont indépendants et suivent une loi normale centrée de covariances repectives* $\sigma^2 P_E$ *et* $\sigma^2 P_{E^\perp}$.

(ii) les variables aléatoires réelles $\frac{||P_E X||^2}{\sigma^2}$ *et* $\frac{||P_{E^\perp} X||^2}{\sigma^2}$ *suivent une loi du* χ^2 *:*

$$\frac{||\pi_E(X)||^2}{\sigma^2} \sim \chi^2(m) \qquad \frac{||\pi_{E^\perp}(X)||^2}{\sigma^2} \sim \chi^2(n-m).$$

Considérons un axe routier E (que l'on suppose de longueur infinie) dans le plan du terrain. La projection π_E définit alors l'étape finale d'un algorithme de map-matching. S'agissant d'une projection orthogonale on a $P_E = P_E^2 = P_E^T$. L'application est donc diagonalisable dans une base orthonormée de vecteurs propres, et la matrice diagonale D est composée d'un 1 et d'un 0. Si les coordonnées (aléatoires) avant map-matching sont notées X, alors $Y = P_E X$ désigne les coordonnées map-matchées du point. Le théorème de Cochran (exprimé dans une base de $\mathbb{R}^2$ qui diagonalise π_E et pour $m = 1$) nous dit alors que :

$$\mathbb{E}\big[||Y||^2\big] = \mathbb{E}\big[||\pi_E(X)||^2\big] = \sigma^2. \tag{2.59}$$

Parallèlement, dans la base canonique de $\mathbb{R}^2$, on a : $\mathbb{E}[||X||^2] = \mathbb{E}[X_1^2] + \mathbb{E}[X_1^2] = 2\sigma^2$ (par définition). On en déduit alors immédiatement le gain en précision sur la racine carrée de l'erreur quadratique moyenne (RMSE) après map-matching :

$$\frac{\mathrm{RMSE}(X_{gps})}{\mathrm{RMSE}(X_{mm})} = \frac{\mathbb{E}[||X||^2]}{\mathbb{E}\left[||Y||^2\right]} = \sqrt{2}. \tag{2.60}$$

Tout se passe donc virtuellement comme si la procédure de map-matching éliminait la variance suivant l'axe transversal à la route, réduisant ainsi l'erreur finale d'un facteur $\sqrt{2}$. On réitère l'expérience pour différentes classes de précision du GPS (*i.e.* pour différentes valeurs de σ_{gps}) et on représente les résultats sur la figure 2.33 (à droite).

La régression donne une droite de pente 1.44 ± 0.05, ce qui est en accord avec notre résultat ($\sqrt{2} \in [1.41, 1.46]$). On peut expliquer cette légère différence par rapport à la valeur de gain théorique de $\sqrt{2}$ par la dimension finie des axes, conduisant des effets de bord pour lesquels la correction intervient alors également dans une moindre mesure suivant le second axe. En pratique, on observe que la qualité du map-matching est croissante avec la fréquence d'aquisition, avec une valeur limite du gain à $\sqrt{2}$ pour les fréquences d'acquisition usuelles (~ 1 Hz). Cette variation de la divergence à la valeur théorique en fonction de la fréquence nécessiterait cependant quelques investigations supplémentaires.

Par ailleurs, un test de normalité de Shapiro-Wilk a été effectué sur une réplication symétrique de la distribution des erreurs après recalage en prenant $n = 4999$ données, les résultats obtenus sont : $W = 0.99973$ ($p = 0.781$), permettant ainsi d'accepter (ou plus formellement, de ne pas rejeter) l'hypothèse nulle H_0 selon laquelle les erreurs après recalage sont distribuées selon une loi normale repliée, comme illustré sur la figure 2.33. Les erreurs des points avant recalage sont distribuées suivant un loi de Rayleigh (Siddiqui, 1962).

Dans cette première section, nous avons émis l'hypothèse de l'exactitude géométrique absolue du réseau routier de référence. En pratique, cette hypothèse est rarement vérifiée, et il est indispensable de considérer également la précision géométrique de ce réseau. En effet, si l'erreur typique du réseau est du même ordre, voire supérieure à celle du récepteur GPS, on peut suspecter que l'opération de map-matching n'apporte aucune amélioration sur la précision de positionnement, quand elle ne dégrade pas cette précision dans les cas les plus défavorables. À ce stade, il est donc intéressant d'évaluer la contribution, dans l'erreur de positionnement final, du choix d'une référence, admise comme étant seulement une approximation de la vérité terrain. Nous tentons de répondre à cette question dans la section suivante.

2.4.4.2 Impact de la qualité géométrique du réseau routier sur le map-matching

Dans le chapitre 3, nous utiliserons les coordonnées map-matchées sur le circuit de référence OSM. La raison logique derrière ce choix est que les données collaboratives OSM recensent également une partie de la signalisation routière. Notre objectif à terme étant de comparer les capacités d'un algorithme de détection sur les profils de vitesses GPS par rapport à la complétude de données actuellement disponibles en termes d'infrastructure de signalisation routière, il a paru plus avantageux de recaler les points observés sur le circuit OSM.

La figure 2.34 représente quelques comparaisons visuelles effectuées dans le but de valider l'utilisation du réseau OSM dans notre cas d'application. Initiée par Girres et Touya en 2010, la question de la qualité géométrique des données collaboratives Open Street Map

par rapport à la référence nationale est encore à l'heure actuelle une question de recherche importante. Afin de s'assurer que le recalage sur les données OSM n'introduisait pas plus d'erreur qu'il ne serait susceptible d'en corriger, nous avons confirmé visuellement que les écarts de position entre les deux références semblaient relativement faibles sur la zone concernée.

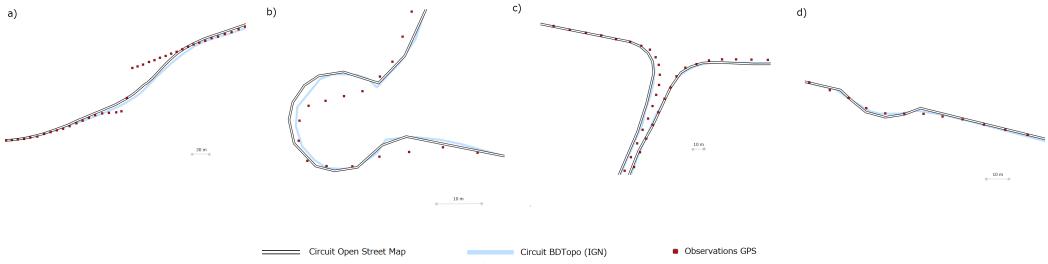


FIGURE 2.34 – Comparaison des géométries des réseaux routiers BDTOPO© et Open Street Map par rapport aux positions mesurées par le GPS.

Si le map-matching améliore indéniablement la qualité géométrique des points mesurés par GPS, sur la plupart des tronçons, et ce quelque soit la référence choisie (figure 2.34 a-b-c), on remarque que sur certaines portions (d), le choix du circuit BDTopo aurait put sembler plus avantageux, sans toutefois afficher une nette différence. Girres et Touya (2010) ont relevé un écart moyen de positionnement entre les lignes de l'ordre de 2 m (suivant la mesure définie par McMaster (1986), consistant à calculer l'aire totale entre les lignes rapportée à la moyenne des longueurs des lignes), mais soulignent la grande variabilité de cette différence en fonction de la zone d'étude avec certains écarts pouvant atteindre jusqu'à 6 m. La mesure d'écart obtenue sur le jeu de données est de 1.34 m, ce qui semble légèrement meilleur (de l'ordre d'un demi-écart-type) que la moyenne observée. Nous avons également mesuré l'écart moyen entre les trajectoires GPS et le circuit BD TOPO (la référence nationale) et nous avons obtenu un résultat de 2.93 m, ce qui signifie que l'écart entre les deux références est en moyenne deux fois inférieur à l'écart de la trajectoire par rapport aux références. Dans cette section, nous cherchons à évaluer l'impact du choix de la référence en fonction des écarts mesurables.

C'est une problématique à la quelle il est particulièrement important de répondre, par exemple dans un contexte où on aurait à disposition deux réseaux différents, l'un d'eux étant plus complet (du point de vue de l'exhaustivité des éléments à répertorier, des attributs, de la toponymie...), l'autre étant géométriquement plus précis [9]. Dans certains cas, il peut arriver également qu'un réseau de référence d'un niveau de précision géométrique supérieur soit disponible moyennant un coût financier ou computationnel. Il est donc nécessaire de disposer d'un critère permettant de déterminer si les erreurs de précision d'un réseau donné sont acceptables au regard de l'application à mener à la suite du map-matching. Notons que certains algorithmes ont été spécifiquement développés pour fusionner plusieurs représentations cartographiques d'un même réseau routier (voir Mustière et Devogele, 2008, par exemple). Cependant, ces algorithmes ne sont pas toujours facile à mettre en œuvre, tandis qu'un critère de décision simple pourrait nous permettre de s'épargner un travail inutile lorsque le réseau le moins précis l'est tout de même suffi-

9. On ne s'intéresse ici qu'aux erreurs en précision relative, le biais global dû à l'erreur de géoréférencement n'est pas pris en compte.

samment dans un cadre d'application donné.

Nous avons vu dans la section 2.4.1 que de nombreux algorithmes de map-matching sont proposés dans la littérature (Quddus et al., 2007). La grande majorité d'entre eux possède toutefois un point commun : l'étape finale du recalage (après l'affectation des observations GPS aux arcs du réseau) consiste en une projection (suivant la plus courte distance) des points sur les polylignes. S'agissant de l'étape effective qui modifie les coordonnées des points GPS (et qui donc améliore potentiellement leur qualité de positionnement), dans un souci de généricité, nous ne traiterons ici que cette ultime étape du map-matching, emettant ainsi *de facto* l'hypothèse que les étapes préliminaires résultent en un ensemble parfait d'affectations.

L'opération de map-matching peut être considérée comme une fonction mathématique f, prenant en entrée une séquence ordonnée de points GPS $X = \{x_1, x_2, ..., x_n,\ x_i \in \mathbb{R}^2\}$, ainsi qu'un réseau de référence $\mathcal{R}$, *i.e.* un graphe topologique dans lequel chaque arc représente un tronçon de route, et dont nous passons sous silence la description rationnelle par souci de concision. L'image de X par la fonction f est la *projection* de X sur $\mathcal{R}$. Le problème peut alors être réduit à l'analyse de sensibilité de la fonction f, c'est-à-dire à la mise en relation de la variabilité des sorties de f par rapport à celle de ses arguments. La méthodologie employée est illustrée sur le schéma 2.35, inspiré de Saltelli et al. (2000b).

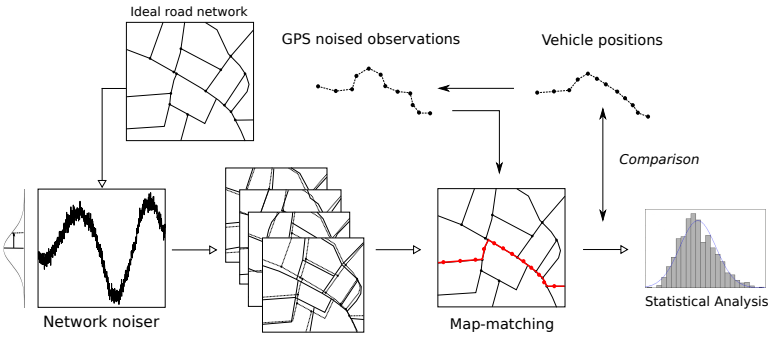


FIGURE 2.35 – Méthodologie globale.

Le réseau routier de référence national, BDTOPO© a été utilisé pour l'expérimentation, et nous le considérerons arbitrairement comme étant un réseau idéal $\mathcal{I}$ de précision absolue. Notons E l'espace de toutes les représentations de $\mathcal{I}$. Une collection de réseaux bruités $\mathcal{R}_1, \mathcal{R}_2, ...\mathcal{R}_k$ est générée à partir de $\mathcal{I}$ à l'aide d'une méthode inspirée par Vauglin (1997), *i.e.* en simulant deux processus stochastiques ε_x et ε_y suivant chacune des directions du plan par factorisation LU de la matrice de covariance du bruit sur les polylignes du réseau (voir section 3.5.4 pour plus de détails). Cette étape a permis de simuler des traces GPS bruitées réalistes tout en évitant les problèmes d'incohérence topologique pouvant survenir avec le modèle d'erreurs en bruit blanc sur les sommets des polylignes (Bonin, 2002).

Pour mesurer l'écart de forme entre deux tronçons routiers du réseau, nous avons besoin d'une distance sur l'espace E, permettant ainsi de considérer que $||\mathcal{R}|| \triangleq d(\mathcal{R}, \mathcal{I}) \in \mathbb{R}^+$ (avec un léger abus de notation) peut être assimilé à une mesure de qualité (absolue) du réseau $\mathcal{R}$ (croissante à mesure que la précision du réseau est mauvaise). D'autre part, il est souhaitable que cette mesure soit relativement pratique à utiliser, et si possible déjà implémentée dans la plupart des logiciels de cartographie. Nous avons donc comparé deux

indicateurs : (1) la distance moyennée de Hausdorff (Girres et Touya, 2010), calculée comme la moyenne sur l'ensemble du réseau des distances de Hausdorff entre tronçons du graphe routier (voir Besse et al., 2016 par exemple pour une définition précise de la distance de Hausdorff dans le contexte des comparaisons de trajectoires automobiles), et (2) la *différence d'aire* (McMaster, 1986), illustrée sur la figure 2.36 et dont nous redonnons la définition ci-dessous.

Définition 2.10. *Soient $\mathcal{R}_1$ et $\mathcal{R}_2$, deux réseaux routiers de référence représentant la même vérité terrain (i.e. $\mathcal{R}_1$ et $\mathcal{R}_2$ ont la même structure topologique et il est possible d'apparier sans ambigüité tout tronçon l_1^i de $\mathcal{R}_1$ avec un tronçon l_2^i de $\mathcal{R}_2$). On définit la différence d'aire entre réseaux par :*

$$d(\mathcal{R}_1, \mathcal{R}_2) = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathcal{A}(l_1^i \circ l_2^i)}{\min(|l_1^i|, |l_2^i|)}, \tag{2.61}$$

où l_j^i dénote la i-eme polyligne du réseau j, $|l_j^i|$ est sa longueur, $\mathcal{A}(l)$ est l'aire intérieure définie par le lacet l et $l_1 \circ l_2$ désigne la concaténation des chemins composables l_1 et l_2.

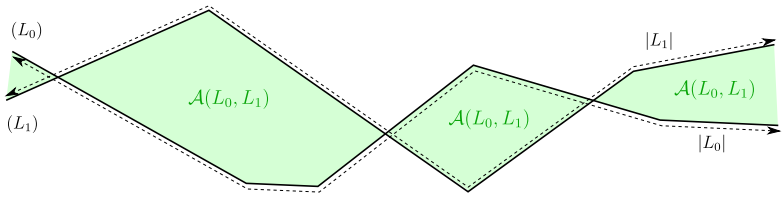


FIGURE 2.36 – Illustration empirique du concept de différence d'aire défini par McMaster (1986) pour mesurer la dissemblance entre 2 polylignes.

Maheureusement, on peut facilement montrer que 2.61 n'est pas une distance [10].

Nous cherchons donc à transformer la définition de McMaster (1986) pour en faire une distance. Considérons dans un premier temps le cas simplifié ou les deux réseaux à comparer sont constitués d'une unique polyligne joignant deux points communs de $\mathbb{R}^2$. Spécifions plus rigoureusement la définition de polyligne.

Soit $\mathcal{U} \subseteq \mathbb{R}^2$ un ensemble ouvert et simplement connexe du plan. Dans ce qui suit nous supposerons que $\mathcal{U} = \mathbb{R}^2$ pour alléger la présentation, mais toutes les définitions et propriétés peuvent se généraliser pour un espace de travail qui soit un sous-ensemble strict du plan.

Définition 2.11. *On appelle courbe plane un chemin de $\mathcal{U}$ sans auto-intersection, i.e. une application continue $\gamma : [0,1] \to \mathcal{U}$ injective, qui à une abscisse curviligne $t \in [0,1]$ associe un point $\gamma(t)$ du plan. L'image $Im(\gamma) \subset \mathcal{U}$ de l'application (parfois notée γ^* ou encore $\gamma([0,1])$) correspond à la représentation graphique du chemin dans le plan.*

Dans le contexte des bases de données cartographiques, une polyligne est en général un chemin continu et dérivable par morceaux. Dans certains systèmes plus fins, un élément

10. Par exemple, en fonctionnel, en considérant une troisième courbe l_3, sinusoïdale et modulée entre l_1 et l_2. On voit alors immédiatement qu'il est possible de faire tendre $|l_3|$ vers l'infini tout en conservant bornées les aires $\mathcal{A}(l_1 \circ l_3)$, et $\mathcal{A}(l_2 \circ l_3)$, violant ainsi l'inégalité triangulaire.

topographique linéaire peut être modélisé par une spline (*cf* section 2.3.2.4), auquel cas γ est dérivable à un ordre égal à celui de la base de splines utilisée dans la modélisation. Ici, nous resterons dans le cadre le plus général et supposerons que $\gamma \in \mathcal{C}^0$.

Considérons une première définition possible de la différence d'aire.

Définition 2.12. *Soient a, b deux points fixés de $\mathcal{U}$. On note E, l'ensemble des applications continues γ de $[0,1] \to \mathcal{U}$, telles que $\gamma(0) = a$ et $\gamma(1) = b$. On définit alors $\delta_1 : E \times E \to \mathbb{R}^+$, l'application qui à deux chemins continus du plan reliant a et b, associe l'aire du complémentaire de la plus grande partie connexe de $\mathcal{U}$.*

Remarquons que la définition 2.12 ci-dessus est légitime. En effet, par un théorème de Jordan généralisé, le lacet $[\gamma_1\gamma_2]$ découpe le plan en n parties connexes dont une seule est non-bornée. Le réel $\delta(\gamma_1, \gamma_2)$ prend alors la valeur (finie) de l'aire de la réunion des $n-1$ autres parties. Il s'agit de la définition formelle la plus simple que l'on puisse associer à la différence d'aire de McMaster (1986). Il est cependant possible d'exhiber un contre-exemple simple invalidant l'inégalité triangulaire, comme illsutré sur la figure 2.37.

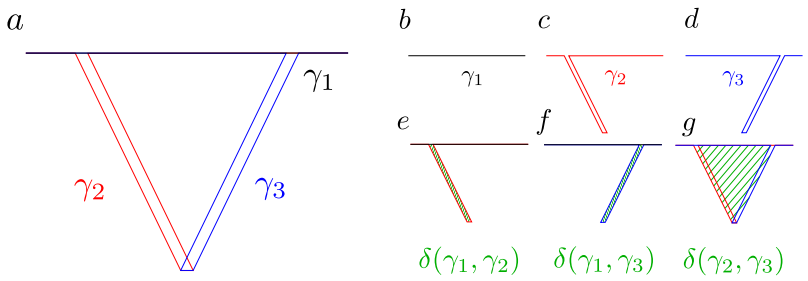


FIGURE 2.37 – Contre-exemple de 3 chemins γ_1, γ_2 et γ_3 ne respectant pas l'inégalité triangulaire pour la définition 2.12. En choisissant γ_2 et γ_3 suffisamment fins, on peut obtenir une quantité $\delta(\gamma_2, \gamma_3)$ (figure g) arbitrairement plus grande que la somme transitive des quantités $\delta(\gamma_2, \gamma_1)$ (figure e) et $\delta(\gamma_1, \gamma_3)$ (figure f). La figure a représente les trois chemins dans le même repère et les figures b, c et d, représentent les chemins individuellement pour plus de clarté.

Considérons une autre définition de différence d'aire qui puisse potentiellement être une distance. Commençons par définir la notion d'homotopie qui caractérise la déformation continue d'une courbe (nécessairement continue elle aussi) vers une autre.

Définition 2.13. *Soient $\gamma_0, \gamma_1 : [0,1] \to \mathcal{U} \subseteq \mathbb{R}^2$, deux chemins continus tels que $\gamma_0(0) = \gamma_1(0)$ et $\gamma_0(1) = \gamma_1(1)$. Les chemins γ_0 et γ_1 sont dits strictement homotopiques, s'il existe une fonction d'homotopie :*

$$\begin{array}{rcl} h : [0,1] \times [0,1] & \to & \mathcal{U} \\ (t,s) & \to & h(t,s) = \gamma_s(t), \end{array}$$

telle que :

- *h est continue (en s et en t)*

- $h(t,0) = \gamma_0(t) \quad et \quad h(t,1) = \gamma_1(t)$
- $\forall\ s \in [0,1]\ h(0,s) = \gamma_0(0) = \gamma_1(0) \quad et \quad h(1,s) = \gamma_0(1) = \gamma_1(1)$

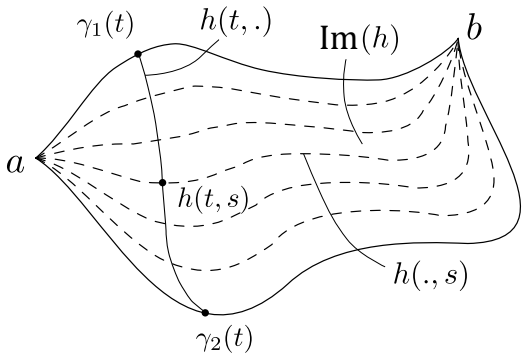


FIGURE 2.38 – Homotopie stricte h entre deux chemins γ_1 et γ_2 à extrémités fixes entre a et b. Image adaptée des travaux de Resconi (2014).

L'homotopie entre chemins nous permet de définir une nouvelle mesure de dissemblance entre courbes. On se place dans les conditions de la définition 2.12.

Définition 2.14. *On définit la fonction $\delta : E \times E \to \mathbb{R}^+$, l'application qui, à deux chemins continus et rectifiables du plan reliant a et b, associe la borne inférieure de l'ensemble des aires des images d'homotopies strictes entre γ_1 et γ_2 :*

$$\delta(\gamma_1, \gamma_2) = \inf_{h \in \mathcal{H}_{12}} \int_{u \in \mathcal{U}} \mathbb{1}\{\exists\ (t,s) \in [0,1]^2 \mid u = h(t,s)\}(u)\ du, \tag{2.62}$$

où $\mathcal{H}_{12}$ désigne l'ensemble des homotopies strictes de γ_1 vers γ_2.

Remarquons que $\mathcal{U}$ étant simplement connexe, tout couple de chemins aux extrémités communes est strictement homotopique. Pour s'en convaincre, il suffit de prendre l'homotopie suivante, dont on montre aisément qu'elle respecte les propriétés de la définition 2.13 :

$$h(t,s) = s\gamma_1(t) + (1-s)\gamma_2(t). \tag{2.63}$$

En conséquence, $\mathcal{H}_{12}$ est non-vide. D'autre part, l'aire étant définie par l'intégrale d'une fonction positive ou nulle, δ est la borne inférieure d'une partie non-vide et minorée, ce qui assure son existence selon la définition 2.14. En revanche, rien ne garantit que cette borne est atteinte, et on pourrait parfaitement imaginer une suite d'homotopies $(h_n)_{n\in\mathbb{N}}$ strictes entre γ_1 et γ_2 dont les aires des images soient strictement décroissantes avec n.

Propriété 2.3. *L'application δ (telle que formulée dans la définition 2.14) est une distance.*

Démonstration. Nous avons établi précédemment que δ est à valeurs dans $\mathbb{R}^+$. Il nous reste à prouver les 3 propositions suivantes ($\forall\ \gamma_1, \gamma_2, \gamma_3 \in E$) :

- Symétrie : $\delta(\gamma_1, \gamma_2) = \delta(\gamma_2, \gamma_1)$
- Séparation : $\delta(\gamma_1, \gamma_2) = 0 \Leftrightarrow \gamma_1^* = \gamma_2^*$ (*i.e.* $\gamma_1 = \gamma_2$ au paramétrage près)
- Inégalité triangulaire : $\delta(\gamma_1, \gamma_3) \leqslant \delta(\gamma_1, \gamma_2) + \delta(\gamma_2, \gamma_3)$

Notons $\mathcal{A} : \mathcal{H} \to \mathbb{R}^+$, l'application qui à une homotopie $h \in \mathcal{H}$ associe l'aire de h^* :

$$\mathcal{A}(h) = \int_{u \in \mathcal{U}} \mathbb{1}\{\exists\ (t,s) \in [0,1]^2 \mid u = h(t,s)\}(u)\ du. \tag{2.64}$$

Soient $\gamma_1, \gamma_2 \in E$ quelconques. Supposons que $\delta(\gamma_1, \gamma_2) = d \in \mathbb{R}^+$. Alors, on peut trouver une suite d'homotopies h_n, de γ_1 vers γ_2, dont l'aire des images $\mathcal{A}(h_n)$ converge vers d. En conséquence, on peut aussi trouver une suite d'homotopies inverses h_n^-, de γ_2 vers γ_1, simplement en inversant le paramétrage des homotopies directes : $h_n^-(t,s) = h_n(t, 1-s)$. Les homotopies h_n et h_n^- ayant même image (pour chaque indice n), on a immédiatement :

$$\lim_{n\to\infty} \mathcal{A}(h_n^-) = \lim_{n\to\infty} \mathcal{A}(h_n) = d,$$

et donc $\delta(\gamma_2, \gamma_1) \leqslant d$. Par un raisonnement similaire, on montre que $\delta(\gamma_2, \gamma_1)$ ne peut être inférieure à d, ce qui achève de montrer la symétrie de $\delta(.,.)$.

De manière évidente, si $\gamma_1^* = \gamma_2^*$ (et *a fortiori* si $\gamma_1 = \gamma_2$) la fonction $h(t,s) = \gamma_1(t)$ est une homotopie stricte de γ_1 vers γ_2 (moyennant éventuellement la composition par l'isomorphisme de paramétrage reliant γ_1 et γ_2) et donc $\mathrm{Im}(h) = \gamma_1^*$ est d'aire nulle. On déduit alors de la définition 2.14 que $\delta(\gamma_1, \gamma_1) \leqslant \lambda(\gamma_1 *)$, avec λ la mesure de Lebesgue sur $\mathbb{R}^d$. Le chemin γ_1 étant rectifiable par hypothèse (définition 2.14), on peut trouver un homéomorphisme $\varphi : [0,1] \to [0,1]$, tel que $\gamma_1 \circ \varphi$ est lipshitzienne, pour une constante k donnée dépendant de la longueur (finie) de la courbe. D'autre part, on peut utiliser le lemme suivant :

Lemme 2.1 : *Soit un chemin lipschitzien $\gamma : [0,1] \to \mathbb{R}^d$ (avec $d \geqslant 2$). Alors $\lambda(\gamma^*) = 0$.*

Démonstration : Considérons un point $c \in [0,1]$ et un intervalle $I = [c - \frac{\varepsilon}{2}, c + \frac{\varepsilon}{2}] \cap [0,1]$. On a alors automatiquement : $\forall\ x_1, x_2 \in I\ |x_1 - x_2| \leqslant \varepsilon$. La fonction γ étant k-lipschitzienne, on peut écrire que $\gamma(I) \subset B_{k\varepsilon}(\gamma(c))$, où $B_r(y)$ désigne la boule ouverte de centre $y \in \mathbb{R}^d$ et de rayon r. En notant λ la mesure de Lebesgue sur $\mathbb{R}^d$, on obtient l'inégalité suivante :

$$\lambda(I) \leqslant \omega_d (k\varepsilon)^d,$$

où ω_d est une constante dépendant de la dimension d considéré : $\omega_d = \frac{\pi^{d/2}}{\Gamma(d/2+1)}$.

En subdivisant l'intervalle $[0,1]$ en $n \in \mathbb{N}$ sous-intervalles de longueur $\varepsilon = 1/n$, on peut écrire (par sous-additivité de la mesure de Lebesgue) :

$$\lambda(\gamma^*) \leqslant \sum_{i=1}^{n} \lambda\Big(\Big[\frac{i}{n}, \frac{i+1}{n}\Big]\Big) \leqslant \sum_{i=1}^{n} \frac{\omega_d k^d}{n^d} \leqslant \frac{\omega_d k^d}{n^{d-1}}.$$

L'entier n pouvant être pris arbitrairement grand, nécessairement : $\lambda(\gamma([0,1])) = \lambda(\gamma^*) = 0$ (à condition que $d \geqslant 2$). Donc, la mesure de Lebesgue d'une courbe lipshitzienne dans tout espace $\mathbb{R}^d$ ($d \geqslant 2$) est nulle.

En conséquence, $\delta(\gamma_1, \gamma_1) \leqslant \mathcal{A}(\gamma_1) = 0$ et, δ étant positive, on a : $\delta(\gamma_1, \gamma_1) = 0$.

Inversement, supposons que $\gamma_1^* \neq \gamma_2^*$. Alors nécessairement on a aussi $\gamma_1 \neq \gamma_2$ et il existe $t_0 \in [0,1]$ tel que $\gamma_1(t_0) \neq \gamma_2(t_0)$. Les chemins étant continus par hypothèse, γ_1 et γ_2 ne coïncident pas sur une réunion d'intervalles ouverts de $[0,1]$. On peut montrer que les intervalles de l'espace de départ de γ_1 peuvent être mis en bijection avec ceux de γ_2, et le problème se réduit alors sur un lacet simple $\gamma_1' \circ \gamma_2'$ (avec γ' la restriction de γ à un intervalle de non-coïncidence) pour lequel on doit montrer que toute homotopie stricte entre γ_1' et γ_2' balaye une aire non-nulle.

Pour un lacet simple γ, on note $\mathrm{Int}(\gamma)$ la composante connexe non-bornée de $\mathcal{U} \setminus \mathrm{Im}(\gamma)$, qui est un ouvert non-vide d'après le théorème de Jordan. On peut donc choisir $x \in \mathrm{Int}(\gamma_1' \circ \gamma_2')$, et h, homotopie stricte entre γ_1' et γ_2'. Pour $s \in [0,1]$ $h(.,s) \circ \gamma_2'$ étant aussi un lacet simple, on peut définir une fonction indicatrice $f : [0,1] \to \{0, \frac{1}{2}, 1\}$:

$$f(s) = \begin{cases} 0 & \text{si } x \in \mathrm{Int}(h(.,s) \circ \gamma_2') \\ 1/2 & \text{si } x \in \mathrm{Im}(h(.,s) \circ \gamma_2') \\ 1 & \text{sinon.} \end{cases} \tag{2.65}$$

D'après le théorème de Jordan, les deux composantes connexes définies par $h(.,s) \circ \gamma_2'$ sont des ouverts, et f est continue en tout point s tel que $f(s) = 0$ ou $f(s) = 1$. Supposons que l'image de h ne passe pas par x, alors $f(s) \neq 1/2$ pour tout $s \in [0,1]$, ce qui revient à dire que f est continue sur $[0,1]$. Par ailleurs, $f(0) = 0$ (x est choisi dans le lacet $h(.,0) \circ \gamma_2'$) et $f(1) = 1$ (x n'appartient pas à γ_2' et l'intérieur de $h(.,1) \circ \gamma_2'$ est réduit à γ_2 par définition de l'homotopie). On a donc une fonction continue sur $[0,1]$ prenant ses valeurs dans l'ensemble discret $\{0,1\}$ et telle que $f(0) \neq f(1)$, ce qui est une contradiction. Donc, il existe un couple $(t,s) \in [0,1]^2$ tel que $h(t,s) = x$, et ce raisonnement étant valide pour tout $x \in \mathrm{Int}(\gamma_1' \circ \gamma_2')$, qui est un ouvert non vide de $\mathbb{R}^2$, donc de surface non-nulle, $\delta(\gamma_1, \gamma_2) > 0$, ce qui achève de montrer la propriété de séparation de δ.

L'inégalité triangulaire découle directement de la définition de la borne inférieure. Prenons 3 chemins continus γ_1, γ_2, $\gamma_3 : [0,1] \to \mathbb{R}^2$ d'extrémités communes. Supposons que $\delta(\gamma_1, \gamma_3) > \delta(\gamma_1, \gamma_2) + \delta(\gamma_2, \gamma_3)$. Alors, on peut exhiber 2 homotopies h_{12} (de γ_1 vers γ_2) et h_{23} (de γ_2 vers γ_3) et 2 réels $\varepsilon_1, \varepsilon_2 \in \mathbb{R}^{+*}$ tels que :

$$\begin{aligned} \mathcal{A}(h_{12}) &< \delta(\gamma_1, \gamma_2) + \varepsilon_1 \\ \mathcal{A}(h_{23}) &< \delta(\gamma_2, \gamma_3) + \varepsilon_2, \end{aligned} \tag{2.66}$$

d'où l'existence de $\varepsilon \stackrel{\Delta}{=} \varepsilon_1 + \varepsilon_2 > 0$ vérifiant l'inégalité :

$$\delta(\gamma_1, \gamma_3) > \mathcal{A}(h_{12}) + \mathcal{A}(h_{23}) - \varepsilon. \tag{2.67}$$

Or, h_{12} et h_{23} sont composables par transitivité, et on peut définir l'homotopie h entre les chemins γ_1 et γ_3 par :

$$h(t,s) = \begin{cases} h_{12}(t,s) & \text{si } s \in [0, \frac{1}{2}] \\ h_{23}(t,s) & \text{sinon.} \end{cases} \tag{2.68}$$

On s'assure aisément que h vérifie tous les critères de la définition 2.14.

En notant que pour ε suffisamment petit, la quantité $\mathcal{A}(h) - \varepsilon \geqslant \delta(\gamma_1, \gamma_3)$ (par définition de la borne inférieure) d'une part, et que $\mathcal{A}(h)$ minore (par sous-additivité de la mesure d'aire sur les réunions d'ensembles) $\mathcal{A}(h_{12}) + \mathcal{A}(h_{23})$ d'autre part, en substituant dans l'inéquation 2.67 on obtient l'inégalité :

$$\mathcal{A}(h_{12}) + \mathcal{A}(h_{23}) - \varepsilon \geqslant \mathcal{A}(h) - \varepsilon \geqslant \delta(\gamma_1, \gamma_3) > \mathcal{A}(h_{12}) + \mathcal{A}(h_{23}) - \varepsilon, \tag{2.69}$$

d'où la contradiction :

$$\mathcal{A}(h_{12}) + \mathcal{A}(h_{23}) > \mathcal{A}(h_{12}) + \mathcal{A}(h_{23}). \tag{2.70}$$

L'application δ vérifie donc l'inégalité triangulaire, et l'expression 2.62 définit donc bien une distance sur E.

□

Moyennant quelques conditions techniques [11], on peut montrer que les distances introduites dans les définitions 2.12 et 2.14 sont équivalentes, ce qui suggère que la plupart des définitions correspondant à l'heuristique de McMaster (1986) coïncident pour les exemples rencontrés en pratiques (sous ces conditions techniques la définition 2.12 constitue donc elle aussi une distance). Par ailleurs, afin d'obtenir une mesure homogène à une longueur (permettant ainsi de rendre la mesure intensive et donc peu sensible à la longueur des polylignes mais uniquement à leur écart géométrique), on divise le résultat de δ par la distance (euclidienne classique de $\mathbb{R}^2$) séparant les points a et b.

Dans la suite, on note respectivement $d_H(\mathcal{R}_1, \mathcal{R}_2)$ et $d_A(\mathcal{R}_1, \mathcal{R}_2)$ la distance de Hausdorff et la distance d'aire entre 2 réseaux routiers, chacune étant définie par la moyenne arithmétique des distances (respectives) entre les tronçons routiers e_1^i et e_1^2 appariés [12] entre les réseaux $\mathcal{R}_1$ et $\mathcal{R}_2$.

$$d(\mathcal{R}_1, \mathcal{R}_2) = \frac{1}{N} \sum_{i=1}^{N} d(e_1^i, e_1^2). \tag{2.71}$$

11. En particulier l'ordonnancement des intersections entre les courbes dans $[0, 1]$.
12. Cette définition suppose que les topologies $G(V, E)$ des deux graphes sous-jacents sont identiques.

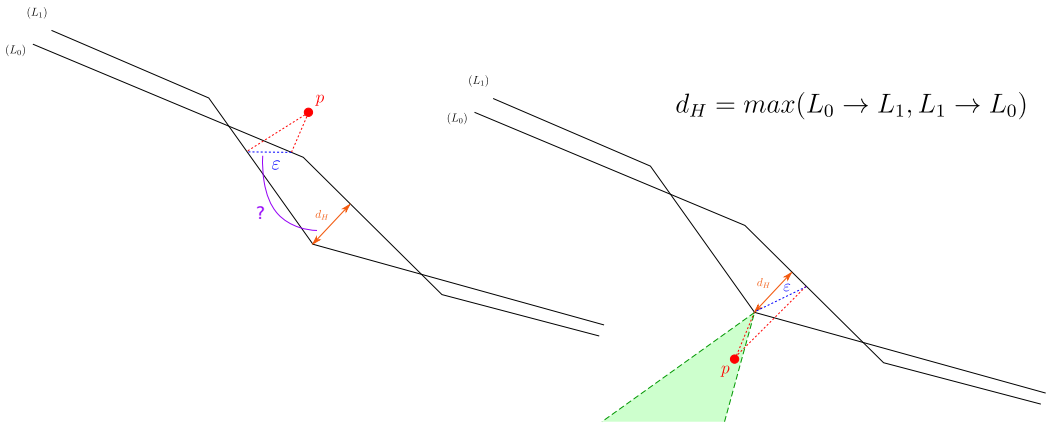


FIGURE 2.39 – À gauche : exemple de map-matching d'un point (en rouge) sur deux polylignes L_1 et L_2. L'objectif de l'étude consiste à relier (par une espérance, une borne probabiliste ou une borne absolue) l'erreur propagée ε à une mesure de distance entre les polylignes. À droite : le choix d'un point dans la zone verte produit automatiquement un dépassement de la distance de Hausdorff : $\varepsilon > d_H(L_1, L_2)$.

On montre aisément que si d est une distance sur l'ensemble des tronçons routiers, alors la généralisation 2.71 définit également une distance sur l'ensemble des réseaux routiers (de graphes topologiques égaux).

Notons que la contrainte d'égalité des topologies des graphes et des positions de nœuds [13] n'est pas si réductrice en pratique. Il est souvent attesté que les bases de données géographiques sont plus précises au niveau des nœuds topologiques que des vertex géométriques (Bonin, 2002; Vauglin, 1997). De plus, si en pratique les extrémités des polylignes ne sont pas communes, on peut définir de nouvelles extrémités en moyennant celles des deux polylignes sans impact significatif sur les mesures d'aire pour des polylignes suffisamment longues (typiquement des lignes de longueur très supérieure à leur précision géométrique).

Notons que cette définition d'une mesure de distance homotopique entre courbes rapproche la distance d'aire de la distance de Fréchet (Alt et al., 2001), qui évalue l'écart entre γ_1 et γ_2 par la longueur maximale des chemins *transverses* $h(t, .)$ tels que représenté en pointillés sur la figure 2.38, et que l'on appelle *hauteur d'homotopie*. Une variante de cette distance consiste à s'intéresser à la *largeur d'homotopie maximale*, *i.e.* au plus long chemin $h(., s)$ obtenu lors de la transformation continue (Chambers et Letscher, 2009). De nombreuses mesures dérivées en découlent, telles que la distance de Fréchet géodésique (Wenk et al., 2010), qui considère des espaces de travail $\mathcal{U}$ plus généraux, la distance de Fréchet isotopique (Chambers et al., 2011) qui ajoute des contraintes supplémentaires sur les transformations homotopiques candidates entre les chemins, ou encore la distance de Fréchet pondérée (Cheung et Daescu, 2009) qui repose sur une métrique non-euclidienne de l'espace de travail.

Cependant, à notre connaissance, aucun développement n'existe dans la littérature de réfé-

13. Du moins, *a minima* des nœuds de degré > 2, les autres pouvant être englobés dans la géométrie des polylignes.

rence sur une métrique fondée sur l'aire d'images d'homotopiques, qui pourtant s'approche le mieux de la définition empirique introduite par McMaster (1986). D'une part, une mesure basée sur une grandeur intégrée est en principe plus robuste que la distance de Hausdorff qui ne prend en compte que le maximum des écarts entre les tronçons, et qui est donc a priori plus difficile à propager (de manière statistique) sur la perte de précision du map-matching. Besse et al. (2016) par exemple, donnent un exemple de configurations pour laquelle la distance classique de Fréchet et la distance de Hausdorff échouent à donner une appréciation correcte des ressemblances entre trois polylignes. D'autre part, on observe que bien qu'étant définie par un maximum, la distance de Hausdorff ne permet pas non plus de majorer la différence de résultats obtenus par le processus de map-matching, comme illustré sur la figure 2.39.

Pour confirmer cette préférence de la distance d'aire, nous avons simulé 10 000 trajectoires, selon un procédé similaire à celui décrit dans la section 2.4.4.1. Chaque trace GPS est alors map-matchée sur le réseau de référence (*i.e.* le réseau brut BD TOPO non bruité) ainsi que sur ses différentes réplications bruitées $\mathcal{R}_1$, $\mathcal{R}_2$,...$\mathcal{R}_k$ (figure 2.35). La phase d'analyse consiste alors à :

- Calculer les indices de qualité des réseaux suivant les deux métriques comparées : $||\mathcal{R}_k||_H = d_H(\mathcal{R}_k, \mathcal{I})$ (pour la distance de Hausdorff moyennée) et $||\mathcal{R}_k||_A = d_A(\mathcal{R}_k, \mathcal{I})$ (pour la distance d'aire moyennée donnée dans la définition 2.14), en rappelant que $\mathcal{I}$ est le réseau brut supposé idéal.

- Calculer le ratio entre la précision du map-matching opérée sur le réseau bruité $\mathcal{R}_k$ et sur le réseau idéal $\mathcal{I}$.

Nous avons alors calculé une fonction *gain de précision* : $g : t \to \mathbb{E}[\sigma/\sigma(\mathcal{R}) \mid ||\mathcal{R}|| = t]$, où σ est l'erreur de la trajectoire simulée, et $\sigma(\mathcal{R})$ est celle de la trajectoire recalée. Plus simplement, g quantifie l'espérance conditionnelle du gain de précision (relativement à la précision GPS) après avoir map-matché les points sur un réseau de qualité connue (et égale au paramètre de la fonction). Notons que σ est un paramètre d'échelle du processus de simulation et n'intervient donc pas explicitement. La fonction g a été estimée statistiquement par une régression polynomiale par morceaux (LOESS) à partir des observations simulées. Les résultats obtenus sont représentés en figure 2.40.

On observe bien que le gain de précision pour un recalage opéré sur un réseau de qualité idéale ($||\mathcal{R}|| = 0$) est proche de 1.4, conformément au résultat obtenu dans la section 2.4.4.1. De plus, on observe effectivement que la mesure d'aire est plus pertinente dans le contexte de la prévision de l'impact de la qualité du map-matching, ce qui se traduit graphiquement par une bande de confiance à 95% nettement plus ressérée, et donc une prévision plus robuste lorsque le seul a priori disponible est la distance du réseau à une référence idéale $\mathcal{I}$.

En pratique, il n'est pas toujours possible de connaître cette distance ($\mathcal{I}$ n'est qu'une abstraction pour les développement théoriques, mais il est en pratique insaisissable, et on dispose seulement de représentations plus ou moins approchées). Reprenons alors la problématique posée dans l'introduction de cette section : nous avons à disposition 2 réseaux, $\mathcal{R}_1$ et $\mathcal{R}_2$, aucun d'eux ne pouvant être considéré comme idéal. Nous sommes capables de mesurer $d_A(\mathcal{R}_1, \mathcal{R}_2)$ (à l'aide de la mesure d'aire moyennée formalisée dans la définition

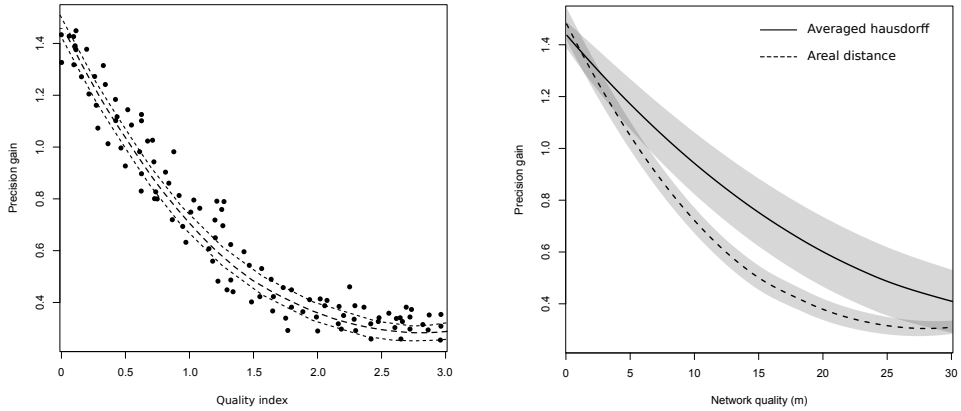


FIGURE 2.40 – À gauche : gain de précision pour différents niveaux de qualité du réseau routier (mesurée à l'aide de la distance d'aire moyennée), régression LOESS et bandes de confiance à 95%. À droite : comparaison des modèles estimés pour la distance d'aire (en pointillés) et la distance de Hausdorff (trait plein).

2.14 et l'équation 2.71, et dont nous avons montré qu'elle respectait les propriétés d'une distance). En utilisant l'inégalité triangulaire sur d_A (illustrée sur la figure 2.41) et le théorème des valeurs intermédiaires sur g on peut établir l'inégalité suivante :

$$\frac{|\Delta\sigma_{12}|}{\sigma} \leqslant \max_{t\in\mathbb{R}^+} \left|\frac{g'(t)}{g(t)^2}\right| \times d(\mathcal{R}_1, \mathcal{R}_2), \tag{2.72}$$

où $\Delta\sigma_{12} = \sigma_{\mathcal{R}_1} - \sigma_{\mathcal{R}_2}$ représente l'écart entre l'écart-type des erreurs des points recalés respectivement sur chacun des deux réseaux $\mathcal{R}_1$ et $\mathcal{R}_2$, σ dénote la précision du GPS, et d est la métrique de comparaison des réseaux utilisée en entrée de la fonction de gain g.

Par dérivation numérique sur les estimations de la figure 2.40, et en supposant que les deux réseaux ont une qualité absolue (suivant les mesures de distance) comprise dans l'intervalle $[0-10]$ m, le facteur de proportionnalité de l'équation 2.72 reliant l'écart entre les réseaux et l'écart relatif entre les erreurs après recalage, vaut 1.08 pour la distance d'aire, et 0.81 pour la distance de Hausdorff.

En reprenant l'exemple d'introduction, avec un réseau OSM (de faible qualité mais plus complet sémantiquement et sur lequel on fait donc le choix de map-matcher nos traces) et le réseau de référence national (en principe plus exact d'un point de vue géométrique, mais pas suffisamment précis pour être considéré comme un réseau idéal). La distance d'aire entre ces deux réseaux nous donnait $d(\mathcal{R}_{osm}, \mathcal{R}_{ref}) = 1.34$ m. Lors du map-matching de traces de précision $\sigma = 10$ m, l'application de l'équation 2.72 nous donne une différence de précision de map-matching entre les deux réseaux de $1.08 \times 1.34 = 1.44$ m.

Dans notre cas d'étude, cet indicateur se traduit par la conclusion suivante : la perte potentielle de précision due au choix d'une référence moins précise (OSM) pour le map-matching est acceptable au regard de la précision attendue pour l'application à suivre.

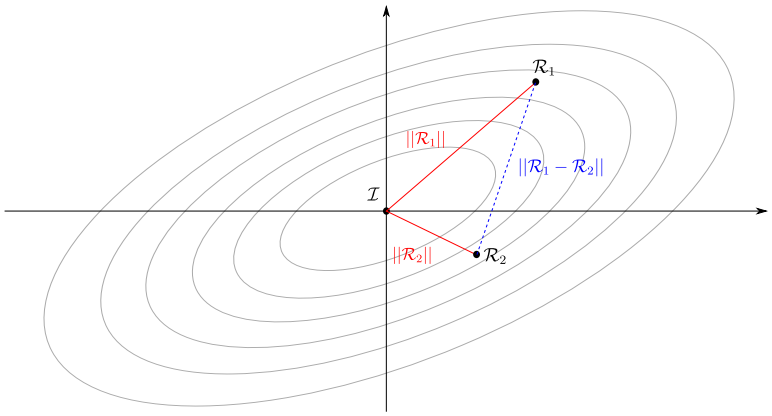


FIGURE 2.41 – Considération de 2 points $\mathcal{R}_1$ et $\mathcal{R}_2$ dans l'espace des réseaux routiers : le réseau idéal $\mathcal{I}$ est inconnu, ce qui exclut la possibilité de mesurer les quantités rouges, mais on peut majorer leur différence par la quantité (accessible) bleue, en utilisant un corollaire de l'inégalité triangulaire $\big|||\mathcal{R}_1|| - ||\mathcal{R}_2||\big| = d(\mathcal{R}_1, \mathcal{I}) - d(\mathcal{R}_2, \mathcal{I}) \leqslant d(\mathcal{R}_1, \mathcal{R}_2)$.

2.5 Correction longitudinale : filtrage et lissage

2.5.1 Introduction

L'objectif des corrections longitudinales est d'ajuster la valeur de l'abscisse curviligne obtenue après map-matching sur le réseau de référence. Ces ajustements ont pour but de :

- Corriger les **incohérences physiques** qui peuvent subsister après le map-matching. C'est le cas par exemple lorsque les mesures GPS d'un véhicule à l'arrêt sont entachées d'une dérive de position dans une direction parallèle à l'axe routier. On se trouve alors en présence d'un système de mesure incohérent du point de vue physique, c'est-à-dire que la dérivée temporelle de l'abscisse curviligne est très supérieure à la vitesse réelle du véhicule mesurée par Doppler : $\dot{s}(t) \gg v(t)$. Cet artefact est caractérisée sur les profils de vitesses par un segment de courbe d'inclinaison anormalement faible.

- Assurer la **monotonicité de la variable d'abscisse curviligne** s. On fait l'hypothèse que le véhicule ne revient jamais en arrière. Lorsque le véhicule est à l'arrêt ou à très faible vitesse, le bruit de mesure sur le GPS peut entraîner une légère diminution de la valeur de s, ce qui se traduit sur les profils de vitesses par une boucle en forme de γ.

- Corriger les **artefacts** systématiques en présence de virages serrés (cf figure 2.42 à gauche). Les spécifications de saisie du réseau routier dans les données topographiques sont telles que certains virages anguleux peuvent ne pas correspondre à la réalité du terrain. C'est le cas en particulier lorsque la route contient une bretelle latérale permettant de tourner en amont d'un carrefour. Le map-matching des données GPS sur le réseau de référence produit alors un effet d'*angle mort* dans lequel aucun point n'est projeté, entraînant un saut dans l'abscisse curviligne, aux effets similaires au premier point ci-avant, à ceci près que l'artefact dépend cette fois du réseau et non plus des capteurs, et qu'il sera donc systématique pour l'ensemble des profils.

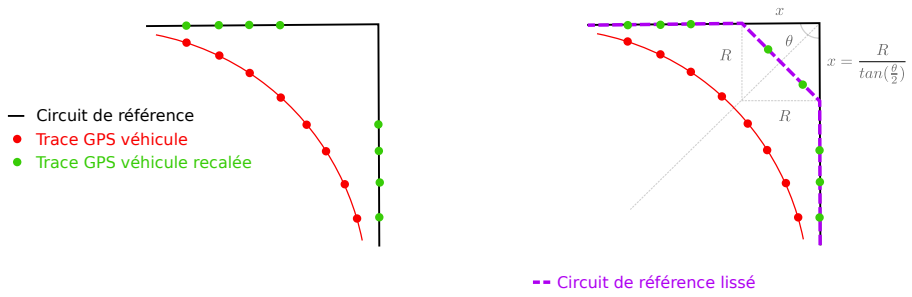


FIGURE 2.42 – À gauche : illustration de l'effet d'angle mort lors du map-matching d'une trajectoire GPS curviligne sur un réseau de référence rectiligne par morceaux. À droite : lissage d'un angle du circuit de sorte à augmenter le rayon de courbure.

Le point (3) est de loin celui qui pose le plus de problèmes du fait de sa systématicité. Il introduit de fait un signal parasite fort dans les profils de vitesse, susceptible de perturber les algorithmes d'apprentissage (*cf* figure 2.47 à la fin du chapitre). Pour corriger cet artefact, nous avons expérimenté deux méthodes différentes : une première approche de lissage du circuit de référence et une seconde approche consistant à lisser les trajectoires map-matchées. Les deux sections suivantes sont consacrées au développement de ces deux approches.

2.5.2 Lissage du circuit de référence

La première solution consiste assez naturellement à modifier le circuit de référence de sorte à le faire correspondre à la réalité du terrain, toute la difficulté de la tâche résidant dans l'automatisation du processus. Une heuristique simple consiste à s'appuyer sur le fait qu'un véhicule ne peut suivre une trajectoire de rayon de courbure infiniment petit. Il s'agit donc de sélectionner tous les couples d'arcs formant un angle inférieur à un seuil critique θ_c.

On choisit alors un rayon type R correspondant au rayon de courbure minimal pour un véhicule circulant en conditions standards (c'est-à-dire que l'on suppose que le franchissement de l'intersection se fait à une vitesse usuelle minimale de 5 à 10 km/h). L'angle du circuit est alors "tronqué" à une distance x du sommet, avec :

$$x = \frac{R}{\tan(\theta/2)}. \tag{2.73}$$

On vérifie alors aisément (cf figure 2.42 à droite) que le rayon de courbure de la trajectoire lissée est de l'ordre de R est se rapproche ainsi de celui de la trajectoire effectivement parcourue. Ce procédé permet alors de *capter* plus de points dans l'angle et d'obtenir de fait un profil de vitesse plus cohérent d'un point de vue physique. L'atout indéniable de cette méthode outre sa simplicité, est de proposer une correction automatique de la référence, permettant ainsi d'obtenir des profils plus proches de la vérité terrain. La méthode a été appliquée avec : $\theta_c = 60$ et $R = 3$ m.

Nous avons constaté que cette solution donnait de très bons résultats au niveau de la correction de l'artefact sur le tracé des profils de vitesse. En revanche, on doit souligner

la difficulté de choisir un rayon de courbure optimal. Lorsque R est trop faible, la correction n'a que peu d'effet. À l'inverse, lorsque R est trop grand, la correction effectuée sur les profils de vitesse reste satisfaisante, mais la position des points recalés peut subir une grosse déviation par rapport au réseau routier.

En conclusion de cette section, nous soulignerons le fait que cette méthode simple et rapide à implémenter permet de supprimer le signal parasite constaté sur les profils de vitesses map-matchés de la figure 2.47, mais au détriment (1) d'une modification des données topographiques de référence et (2) d'une position absolue des points susceptible d'être fortement biaisée par rapport à la vérité terrain. D'autre part, nous noterons que cette première méthode n'est applicable que lorsque tous les véhicules ont parcouru le même itinéraire (à l'instar du chapitre 3) mais devient difficilement extrapolable dans le cas d'un parcours libre sur un réseau car cela augmente la complexité de modélisation du graphe routier de référence.

2.5.3 Lissage du profil de vitesse

Pour pallier ce problème, nous avons testé une seconde méthode. Notons dans un premier temps que la méthode de lissage de la référence au niveau des angles n'est à même de corriger que les artefacts de type 3, que représentent les artefacts induits par le map-matching. Or, ici, nous souhaiterions disposer d'un jeu de données qui soit exprimé dans un système cohérent d'unité, c'est-à-dire que l'on doit avoir la contrainte $\dot{s}(t) \simeq v(t)$ et ce, quelles que soient les perturbations subies sur les mesures de positions GPS et de vitesse Doppler. D'autre part, comme énoncé précedemment, nous souhaiterions interdire les retours en arrière du véhicule, ce qui techniquement se traduit par une contrainte de monotonie au cours du temps sur la variable s. Une méthode par filtrage de Kalman de la trajectoire map-matchée semble bien adaptée au problème.

De par sa formulation, le filtre de Kalman possède de nombreuses propriétés communes avec le modèle de Markov à états cachés introduit dans le paragraphe 2.4.2, tout deux reposant sur un modèle de transition (ou modèle dynamique) et un modèle d'emission (ou d'observation). Le filtrage de Kalman est particulièrement efficace lorsque le modèle décrivant l'évolution du système est linéaire et que les observations sont entachées d'erreurs gaussiennes. On peut montrer que sous ces conditions, il s'agit d'un estimateur optimal.

L'objectif du problème consiste à estimer les états X_k à chaque instant, compte tenu des observations bruitées Y_k et de l'a priori dont on dispose sur le comportement du système. Dans notre cadre d'application, le vecteur X_k contient les mesures d'abscisse curviligne et de vitesse Doppler : $X_k = [s(t_k), v(t_k)]^T$ avec $t_k = k\Delta t = k/f$ où f est la fréquence d'acquisition du signal. Les équations d'état s'ecrivent alors :

$$\text{Équation de modèle :} \qquad X_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} X_{k-1} + \delta_k = AX_{k-1} + \delta_k \tag{2.74}$$

$$\text{Équation de mesure :} \qquad Y_k = X_k + \begin{bmatrix} w_{gps} \\ w_{doppler} \end{bmatrix}_k \tag{2.75}$$

où w_{gps} et w_{dop} sont les bruits de mesures des capteurs GPS et Doppler, supposés indépendants et distribués suivant des lois normales de moyennes nulles et d'écart-types respectifs σ_{gps} et σ_{dop} (cf section 2.2), que l'on supposera constants dans le temps (hypothèse nécessaire en l'abscence de données sur l'évolution de la précision des mesures au cours du temps). ε_k est composé d'un bruit sur l'abscisse curviligne et d'un bruit sur la vitesse, supposés également gaussiens de moyennes nulles et d'écarts-types respectifs σ_s et σ_v. Les matrices Q_k et R_k sont donc indépendantes du pas de temps k et diagonales :

$$Q_k = Q = \begin{bmatrix} \sigma_s^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \qquad \text{et} \qquad R_k = R = \begin{bmatrix} \sigma_{gps}^2 & 0 \\ 0 & \sigma_{dop}^2 \end{bmatrix} \qquad \forall\ k \geqslant 0$$

L'application des équations de Kalman permet alors d'obtenir une réestimation de X_k à chaque état en combinant les informations de positions et de vitesses. À chaque étape, on calcule une prédiction a priori $\hat{X}_{k|k-1}$ à partir de l'état estimé à l'étape précédente $\hat{X}_{k-1|k-1}$ et à l'aide de l'équation 2.74.

$$\hat{X}_{k|k-1} = A\hat{X}_{k-1|k-1}. \tag{2.76}$$

L'incertitude sur la prédiction $\hat{X}_{k|k-1}$ s'obtient alors classiquement par la loi de propagation des variances :

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q, \tag{2.77}$$

avec $P_{k-1|k-1}$ et $P_{k|k-1}$ les matrices de variances-covariances de l'état X respectivement à l'instant $k-1$ et lors de la phase de prédiction de l'instant k.

La phase de mise à jour consiste alors dans un premier temps à calculer un gain optimal de Kalman :

$$K_k = P_{k|k-1}(P_{k|k-1} + R)^{-1}. \tag{2.78}$$

La nouvelle estimation à l'instant k s'exprime alors en combinant la prédiction $\hat{X}_{k|k-1}$ avec la nouvelle observation Y_k :

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k(Y_k - X_{k|k-1}). \tag{2.79}$$

La continuité du calcul nécessite de connaître la variance de l'estimation courante :

$$P_{k|k} = (I_n - K_kA)P_{k|k-1}. \tag{2.80}$$

Enfin, on souhaite interdire les retours en arrière dans les profils, comme présenté en figure 2.43.

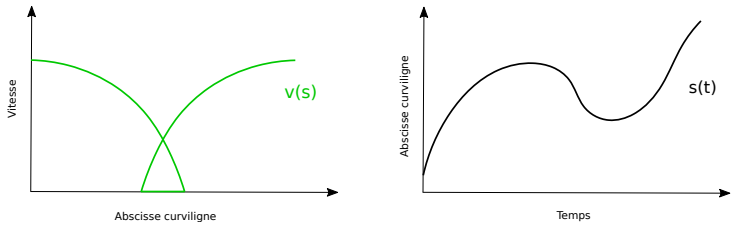


FIGURE 2.43 – Non-monotonie du profil temps × espace (à droite) et impact sur le profil de vitesse résultant (à gauche).

Cette contrainte s'exprime comme une contrainte de monotonie sur l'abscisse curviligne en fonction du temps. Plusieurs solutions de monotonisation en post-traitement de la fonction corrigée par Kalman peuvent être proposées (cf figure 2.44).

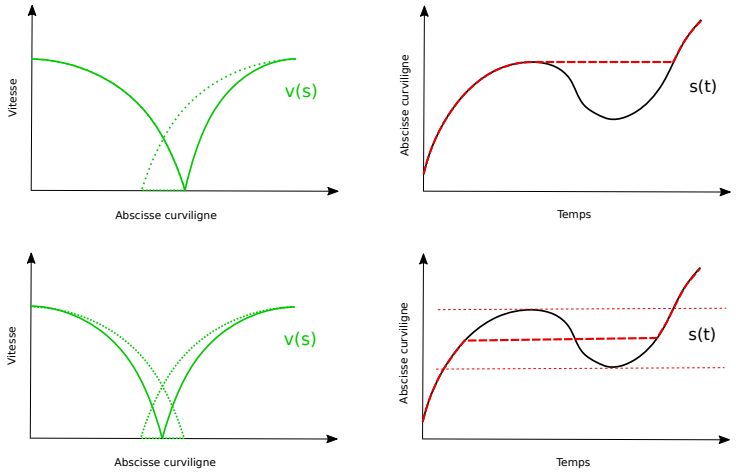


FIGURE 2.44 – Plusieurs solutions de monotonisation de la fonction. En haut : à partir du point le plus avancé. En bas : à partir d'un point intermédiaire.

Toutefois, il faut remarquer que si ces méthodes ont pour objectif commun de mieux localiser l'arrêt réel du véhicule, et donc indirectement la position d'un éventuel élément d'infrastructure rencontré par le véhicule, le critère de monotonisation de la fonction ne repose que sur un choix arbitraire et ne prend pas en compte les informations statistiques disponibles au moment de l'arrêt. Par exemple, la première solution de la figure 2.44, choisi de conserver arbitrairement le point d'arrêt le plus en aval le long de l'abscisse curviligne, ce qui en retour impose une contraction de la partie droite du profil, préalablement corrigée par Kalman. Il peut donc paraître intéressant d'intégrer cette contrainte dans l'estimateur de Kalman de sorte à obtenir les profils les plus cohérents possibles entre les informations de vitesse, de position et la localisation de arrêts. Il existe de nombreuses manières d'intégrer des contraintes à un filtrage de Kalman (Gorinevsky, 2004; Simon, 2010). Ces contraintes sont exprimées sous forme d'égalité ou d'inégalité sur les paramètres à estimer. Les contraintes d'égalité sont en général classiquement gérées en ajoutant à chaque étape un vecteur de multiplicateurs de Lagrange à l'estimation. Les contraintes d'inégalités sont

quant à elles beaucoup plus ardues à intégrer dans la boucle de calcul.

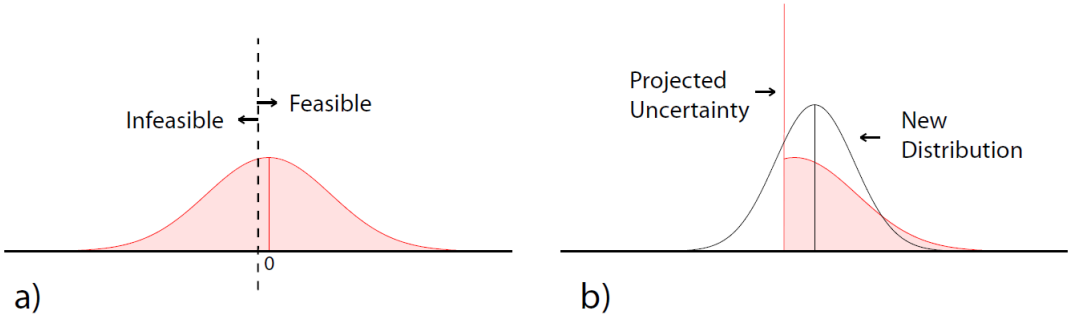


FIGURE 2.45 – Implémentation de la contrainte de monotonie sur l'abscisse curviligne en fonction du temps s. Source : Tully et al. (2011).

La solution utilisée dans notre cadre d'application est celle décrite par (Tully et al., 2011), dans leurs travaux d'automatisation d'un bras chirurgical. À chaque nouvelle estimation la loi normale décrivant la variance d'un paramètre donné est repliée au niveau de la valeur du seuil de la contrainte (figure 2.45). La distribution résultante à l'état suivant est assimilée à une gaussienne de même écart-type que la loi repliée. La valeur contrainte de l'abscisse curviligne à une itération $k+1$ est alors donnée par :

$$s^c_{k+1} = s_k + \sigma_{k+1}\sqrt{\frac{2}{\pi}}\exp\Big(-\frac{(s_{k+1}-s_k)^2}{2\sigma^2_{k+1}}\Big) + \Big(s_{k+1}-s_k\Big)\Big(1-2\Phi\Big(-\frac{s_{k+1}-s_k}{\sigma_{k+1}}\Big)\Big), \quad (2.81)$$

où Φ désigne la fonction de répartition de la loi normale standard et σ_{k+1} est l'écart-type d'estimation de s_{k+1}.

La variance sur l'estimation contrainte s^c_{k+1} s'obtient enfin par :

$${\sigma^c_{k+1}}^2 = (s_{k+1}-s_k)^2 + \sigma^2_{k+1} - (s^c_{k+1}-s_k)^2. \quad (2.82)$$

Nous remarquerons que l'écart-type d'une loi repliée est inférieur à celui de la loi normale d'origine, et ce quelle que soit la position de la contrainte d'inégalité. Cette affirmation peut-être tirée du fait que la contrainte déplace la nouvelle estimation de s plus en aval : $s^c_{k+1} \geqslant s_{k+1}$. En ce sens, l'ajout de la contrainte de monotonie peut être considéré comme une observation supplémentaire, permettant ainsi d'affiner la précision d'estimation des paramètres inconnus. Nous donnons ci-dessous les différentes étapes de l'algorithme de filtrage par Kalman avec contrainte de monotonie

Nous avons appliqué ce filtre sur l'ensemble des profils avec les paramètres suivants : $\sigma_{gps} = 5$ m (écart-type estimé après map-matching), $\sigma_{dop} = 0.1$ m.s^{-1} (valeur couramment admise), $\sigma_s = 0.5$ m et $\sigma_v = 1$ m.s^{-1} (paramètres ajustés empiriquement).

Algorithm 1 Filtre de Kalman avec contrainte de monotonie sur l'abscisse curviligne

Require: Séquence d'observations $\{Y_k\}_{k=1..n}$ et paramètres : σ_{gps}, σ_{dop}, σ_s, σ_v et Δt

Création de la matrice de modèle A et des matrices de variances-covariances Q et R

Départ d'un état initial $X_1 = [s_1, v_1]^T$ avec une variance associée $P_1 = \text{diag}(\sigma_{s_1}^2, \sigma_{v_1}^2)$

for $k = 1$ **to** n **do**

Prédiction de l'état X_{k+1} et calcul de la variance P_{k+1} (équations 2.76 et 2.77)

Réestimation de X_{k+1} et de sa variance associée P_{k+1} par mise-à-jour à partir des nouvelles observations Y_{k+1} (équations 2.78, 2.79 et 2.80)

Contrainte de monotonie sur l'abscisse curviligne : $s_{k+1} \leftarrow s_{k+1}^c$ (équation 2.81)

Approximation de la variance d'estimation finale par une loi normale centrée en s_{k+1}^c et de variance $(\sigma_{k+1}^c)^2$ (équation 2.82)

end for

return Séquence d'observations filtrées $\{X_k\}_{k=1..n}$ avec $X_k = [s_k, v_k]^T$ et variances d'estimation associées : $\text{Var}(s_k) = P_k(1,1)$ et $\text{Var}(v_k) = P_k(2,2)$ à chaque pas de temps $t_k = k\Delta t$.

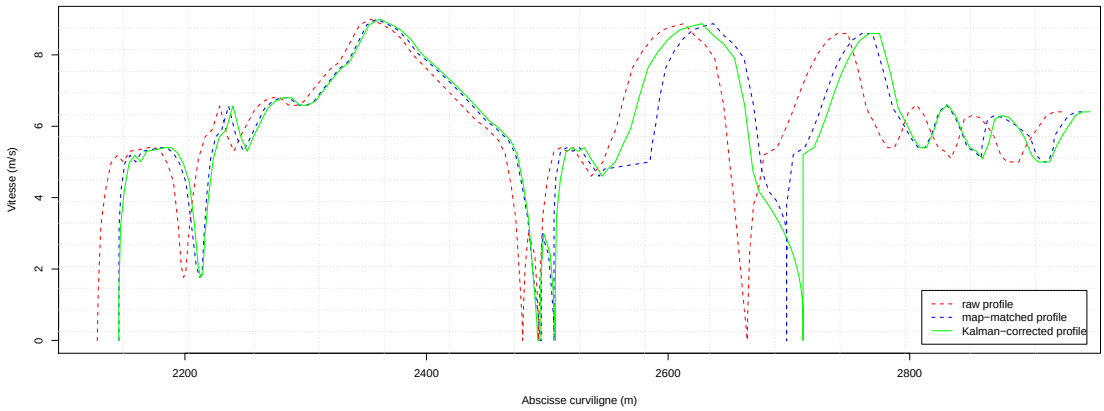


FIGURE 2.46 – Filtrage de Kalman sur un profil de vitesse. Profil brut (en rouge), map-matché (en bleu) et profil map-matché et filtré par Kalman (en vert).

Les résultats obtenus sont tout à fait convaincants et montrent clairement l'intérêt de la méthode.

La figure 2.46 en particulier, montre la comparaison des profils avant map-matching (en rouge), après map-matching (en bleu) et après map-matching et filtrage de Kalman (en vert). On voit clairement que le profil corrigé par Kalman est globalement recalé avec la solution map-matchée sauf au niveau de l'artefact (au niveau $s = 2560\ m$) où il adopte davantage le comportement du profil original. La figure 2.47 (en fin de chapitre) montre l'impact du filtrage de Kalman sur la présence du signal parasite.

Nous noterons enfin qu'une version plus performante du filtrage pourrait être obtenue en incluant également les mesures de l'odomètre, auquel cas, l'équation d'observation serait remplacée par :

$$\begin{bmatrix} y_s \\ v_{dop} \\ v_{odo} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \kappa \end{bmatrix}_k \begin{bmatrix} s \\ v \end{bmatrix}_k + \begin{bmatrix} w_{gps} \\ w_{dop} \\ w_{odo} \end{bmatrix}_k \tag{2.83}$$

Dans cette équation, $\kappa(.)$, le coefficient de dilatation des distances mesurées par l'odomètre, est une fonction paramétrique du pas de temps dont les paramètres sont ajustés par l'algorithme EM en même temps que les écarts-types de l'équation de modèle. Dans notre cadre d'application, étant donnée la difficulté à l'heure actuelle de disposer, en plus des mesures GPS, de données odométriques, nous n'intégrons pas cette seconde information de vitesse dans le lissage des profils.

Notons pour finir, que l'intérêt principal d'effectuer le recalage par map-matching avant le filtrage de Kalman, réside dans le fait que la projection des coordonnées dans un référentiel unidimensionnel (le linéaire routier) permet une linéarisation de l'équation d'observation, garantissant ainsi l'optimalité du filtrage. Une résolution dans l'ordre inversé nécessiterait l'utilisation de méthode non-linéaires, tels qu'un filtre de Kalman étendu (EKF) ou un filtre particulaire. Lamb et Thiébaux (1999) proposent une application simultanée du map-matching et du filtrage.

2.6 Reconstruction d'une trajectoire partielle

Dans cette brève section, nous développons un algorithme permettant de reconstruire une trajectoire GPS dont certaines grandeurs physiques ont été sous-échantillonnées. Cette problématique peut se poser assez fréquemment, par exemple lorsque le fournisseur de données souhaite réduire la quantité de données mise à disposition (pour des raisons commerciales par exemple), où lorsque l'un des capteurs est capable d'opérer à une résolution temporelle sensiblement plus élevée que celles des autres capteurs.

C'est le cas ici, dans un travail connexe à ces travaux de thèse, qui a été effectué en collaboration avec l'incubateur de l'IGN, sur un jeu de données privé et confidentiel. L'objectif final du travail était de pouvoir améliorer la résolution des traces pour une détection des carrefours giratoires. En effet, la spécification des bases de données topographiques de l'IGN font qu'en dessous d'un certain diamètre, ces carrefours sont modélisés par un nœud topologique simple dans le réseau routier. Une action de développement a donc été lancée pour tenter d'inférer la géométrie de ces giratoires à partir des données FCD, ce qui nécessite donc des trajectoires GPS spatialement bien résolues, en particulier dans les zones où le nombre de traces observées est susceptible d'être faible.

Brièvement, les traces du jeu de données utilisées sont composés de trames de navigation, transmises à la fréquence d'une trame par minute depuis les vehicules vers un opérateur central. Chaque trame est composée de 10 positions GPS (en coordonnées géographiques), 60 valeurs de caps et 60 valeurs accélérométrique. Le problème consiste à recomposer la trajectoire à une fréquence de 1 Hz.

Pour résoudre ce problème, nous utilisons une approche d'ajustement par moindres carrés (Sillard, 2001). Le vecteur $\mathbf{x}_t \in \mathbb{R}^{14}$ des paramètres à estimer est :

$$\mathbf{x}_t = (x_{6t+0}, x_{6t+1}, ...x_{6t+6}, y_{6t+0}, y_{6t+1}, ...y_{6t+6})^T,$$

qui correspond aux coordonnées planimétriques de la trajectoire au cours d'une trame t.

On cherche à implémenter trois types de contraintes dans le modèle :

(1) 4 contraintes de position sur les coordonnées initiales et finales de chaque trame, qui sont observées par GPS.

(2) 6 contraintes sur le cap du véhicule, exprimées par une dérivation numérique arrière :

$$y_{i+1} - y_i - \tan\theta_i(x_{i+1} - x_i) = 0 \tag{2.84}$$

$$i = 6t + k \quad k \in [\![0; 5]\!]$$

(3) 10 contraintes sur l'accélération du véhicule, exprimées par une dérivation numérique au second ordre :

$$\frac{d^2x}{dx^2}(t_i) \approx x_{i+1} - 2x_i + x_{i-1} = a_i \cos\theta_i \tag{2.85}$$

$$\frac{d^2y}{dy^2}(t_i) = y_{i+1} - 2y_i + y_{i-1} = a_i \sin\theta_i \tag{2.86}$$

$$i = 6t + k \quad k \in [\![1; 5]\!]$$

On pose la suite de matrices $(A_t)_{t\in\mathbb{N}} \in \mathbb{R}^{20\times14}$:

$$A_t = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & t_1 & -t_1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & t_2 & -t_2 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & t_3 & -t_3 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & t_4 & -t_4 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & t_5 & -t_5 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & t_6 & -t_6 \\
1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1
\end{bmatrix} \tag{2.87}$$

Avec : $t_i = \tan\theta_{6t+i}$. Dans le cas de figure où $\theta = \frac{\pi}{2}\ [\pi]$, on utilise la matrice alternative :

$$A_t = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
\dots & & & & & & & & & & & & & \dots
\end{bmatrix} \tag{2.88}$$

où les lignes 11 à 20 sont inchangées par rapport au cas standard.

Le vecteur des observations $B_t \in \mathbb{R}^{20}$ s'exprime par :

$$B_t = \begin{bmatrix} x_{6t+0} & y_{6t+0} & x_{6(t+1)} & y_{6(t+1)} & 0_{\mathbb{R}^6} & a_{6t+k}\sin\theta_{6t+k} & a_{6t+k}\cos\theta_{6t+k} \end{bmatrix}^T \tag{2.89}$$

où a désigne l'accélération et x_k désigne le vecteur de $\mathbb{R}^5$ contenant les valeurs a_i pour la suite d'entiers i compris entre 1 et 5.

Pour une plus grande régularité entre les trames dans la reconstruction, on peut répéter le schéma décrit par 2.87 et 2.89 sur un plus grand nombre de trames. Par exemple, pour résoudre simultanément trois trames, la matrice A s'étend à 54 lignes ($3 \times 20 - 4$ où le chiffre 4 correspond aux contraintes de position redondantes au début et à la fin de la trame intermédiaire.

On pose enfin une matrice de pondération des observations $P = \Sigma^{-1}$ avec :

$$\Sigma = \text{diag}\Big(\Big[\frac{\sigma_{gps}}{\sqrt{2}}\Big]_4, [\sigma_\theta]_6, [\sigma_a]_{10}\Big), \tag{2.90}$$

où $[x]_m$ désigne le scalaire x répété m fois dans un vecteur de $\mathbb{R}^m$.

La résolution du problème s'effectue alors itérativement à chaque pas de temps t :

$$\hat{\mathbf{x}}_t = \underset{\mathbf{x}\in\mathbb{R}^{14}}{\text{argmin}} \; ||A_t\mathbf{x} - B_t||^2_\Sigma = (A_t^T\Sigma^{-1}A_t)^{-1}A_t^T\Sigma^{-1}B_t, \tag{2.91}$$

où $||.||_\Sigma$ désigne la norme de Mahalanobis associée à Σ (Mahalanobis, 1936). Dans le cas où la matrice normale n'est pas inversible (moins de 1% des tronçons), on calcule les coordonnées sur-échantillonnées par splines d'interpolation (*cf* paragraphe 2.3.2.4). L'écart-type sur les coordonnées estimées s'obtient directement par inversion de la matrice normale :

$$\Sigma_{\mathbf{x}_t} = (A_t^T\Sigma^{-1}A_t)^{-1}, \tag{2.92}$$

ce qui permet de représenter les ellipses d'erreur, centrées sur les points $\mathbf{x}_i$ et d'axes égaux aux vecteurs propres de la matrice de covariance 2×2 réduite aux coordonnées $\mathbf{x}_i$.

Le code a été implémenté en R, puis transcrit en Python à l'incubateur IGN. Le temps de traitement typique est de l'ordre de 800 ms pour une trace GPS de 30 minutes.

La figure 2.48 donne une illustration d'une trajectoire reconstruite avec cette méthode. Le paramètre σ_{gps} a été fixé à 5 m. Les écarts-types σ_θ et σ_a ont été choisi par validation croisée, avec un mode opératoire similaire à celui décrit par l'équation 2.22.

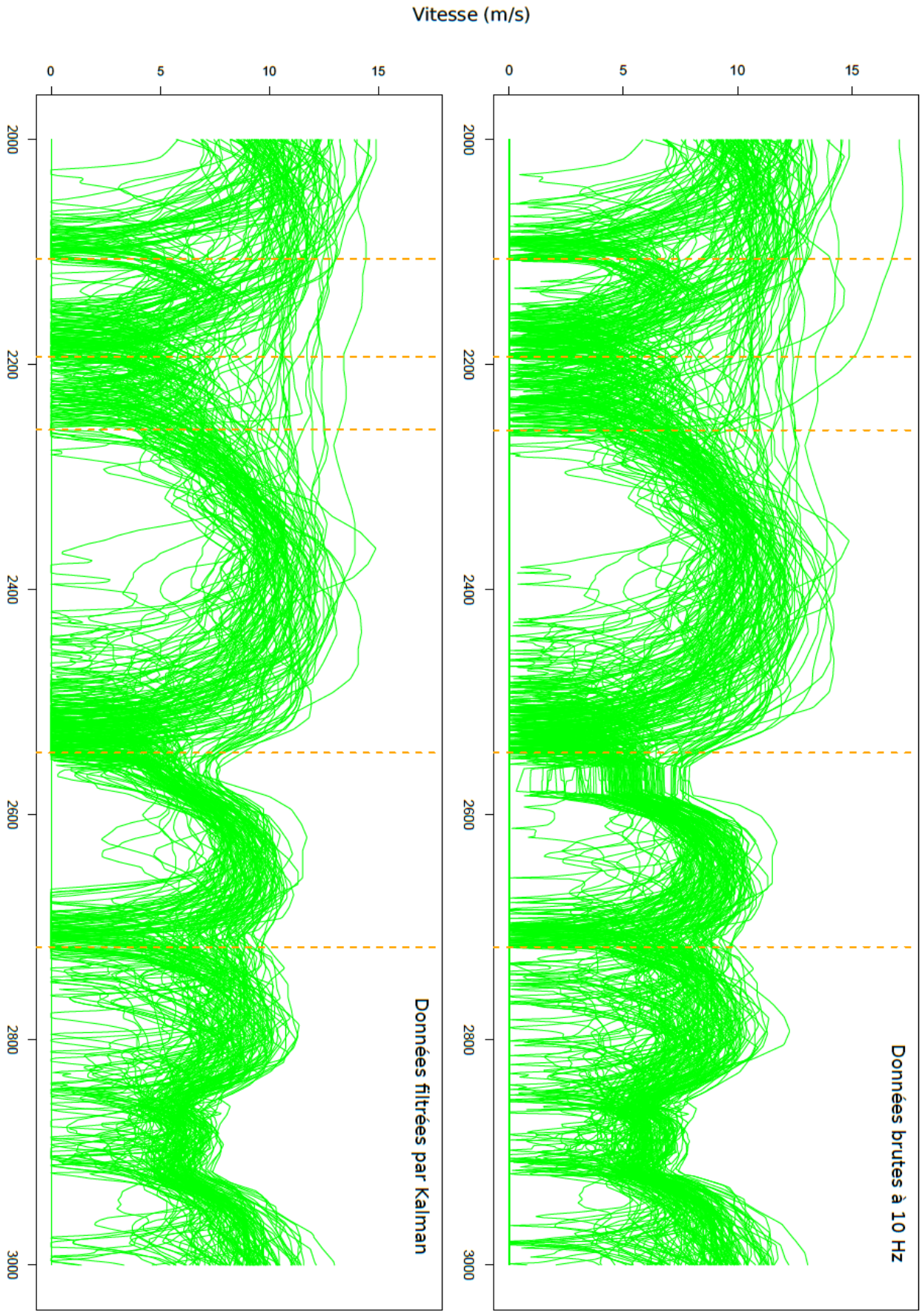


FIGURE 2.47 – À droite : signal parasite dans les profils de vitesse, dû au map-matching de trajectoires curvilignes sur un réseau affine par morceaux. À gauche : élimination de l'artefact de virage par filtrage de Kalman.

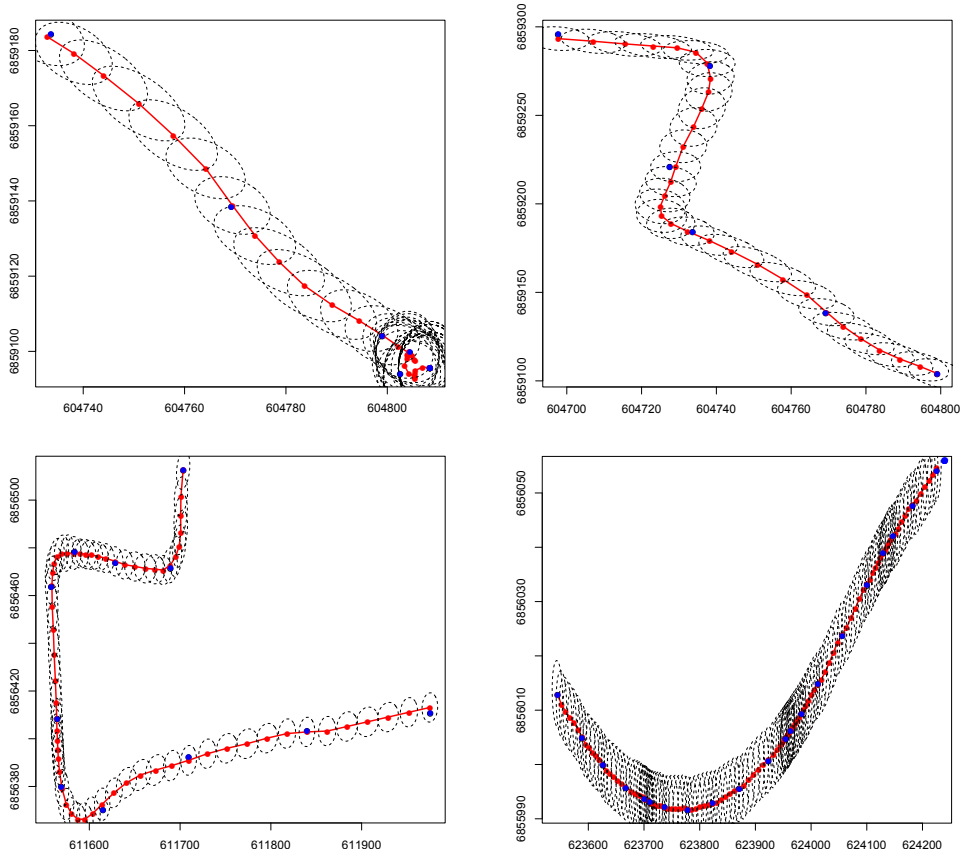


FIGURE 2.48 – Quatre exemples de reconstruction d'une trajectoire partielle. Observations GPS en bleu, trajectoire reconstruite en rouge et ellipses d'erreur en pointillés noirs.

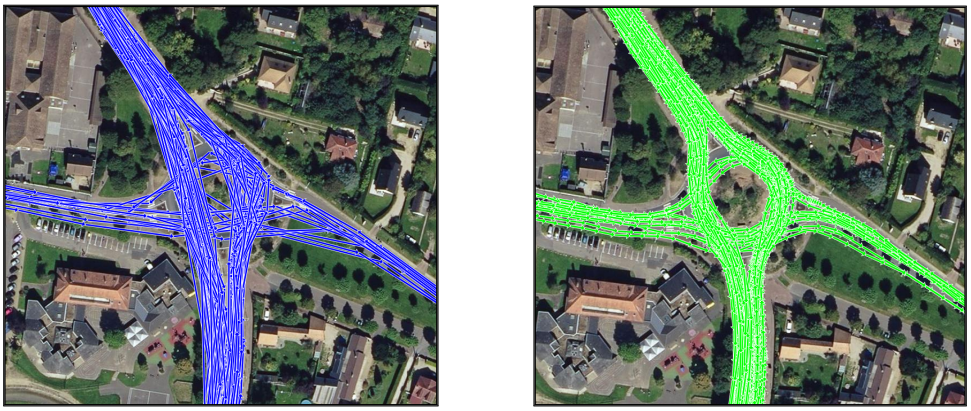


FIGURE 2.49 – Exemple de reconstructions d'une trajectoire partielles pour le cas pratique de la détection de carrefours giratoires.

Chapitre 3

Comparaison des approches image et fonctionnelle en conditions expérimentales

Sommaire

Ce chapitre reprend en grande partie les éléments d'un article publié dans la revue *International Journal of Data Science and Analytics* : Traffic Signal Detection from in-vehicle GPS Speed Profiles using Functional Data Analysis and Machine Learning.

L'accessibilité croissante de données GPS issues de véhicules traceurs permet la conception d'algorithmes de construction automatique du réseau routier à l'aide de techniques d'apprentissage statistique. De nombreux travaux allant dans ce sens peuvent être trouvés dans la littérature de référence sous le nom de *map inference*. Dans ce chapitre, nous tentons d'étendre le map inference par apprentissage au cas de la détection de la signalisation routière, et plus spécifiquement des feux tricolores, pour des données acquises en environnement contrôlé (données expérimentales de bonne qualité, issues du projet *ecoDriver*). Dans notre modélisation, nous découpons un ensemble de profils spatiaux de vitesse en fenêtres glissantes de longueur 100 m. Chaque fenêtre représente une instance, que l'on cherche à étiqueter (présence ou abscence de feu tricolore). L'étiquetage s'appuie sur des méthodes d'apprentissage supervisé, et on considère trois approches différentes pour représenter l'information contenue dans les instances et en extraire des descripteurs : dans une première approche, les données sont considérées comme des rasters (chaque pixel livrant une information zonale en un lieu et une vitesse donnée) sur lesquels nous appliquons des algorithmes classiques de reconnaissance d'image. Dans une deuxième approche, les courbes sont considérées sous leur aspect fonctionnel, et chaque profil de vitesse est un objet à part entière, dont nous allons extraire des descripteurs de plus haut niveau. Ces deux approches sont comparées à une approche *directe*, dans laquelle les descripteurs sont définis par les valeurs ordonnées point-à-point des profils de vitesse. Ce chapitre a deux objectifs principaux :

- Tester différents types de représentation des données
- Tester différents algorithmes de classification

Nous concluons alors le chapitre avec une analyse de sensibilité des résultats obtenus aux différents facteurs variables (précision et fréquence des capteurs embarqués, nombre de trajectoires disponibles...) puis en testant l'application des algorithmes sur d'autres types d'éléments de la signalisation routière.

3.1 Constitution du jeu de données

3.1.1 Protocole d'acquisition

Les traces GPS ont été collectées dans le cadre du projet *ecoDriver*[1], financé par l'Union Européenne, et dont l'objectif est de caractériser la réponse des automobilistes à différentes consignes d'éco-conduite, et d'évaluer l'impact résultant sur la consommation en carburant. À cette fin, 30 conducteurs ont été recrutés pour parcourir un circuit en boucle prédéfini de 25 km dans la commune de Versailles (78) et ses environs. Chaque sujet a répété l'expérimentation (à différentes dates) entre 4 et 6 fois, pour un total de 170 trajectoires collectées à l'issue du processus d'acquisition. La moitié de ces traces a été générée dans des conditions naturelles de conduite, tandis que l'autre moitié a été générée sous l'assistance d'un système embarqué d'aide à l'éco-conduite.

Le circuit de l'expérimentation (représenté sur la figure 3.1, à gauche) a été parcouru dans

1. http ://www.ecodriver-project.eu/

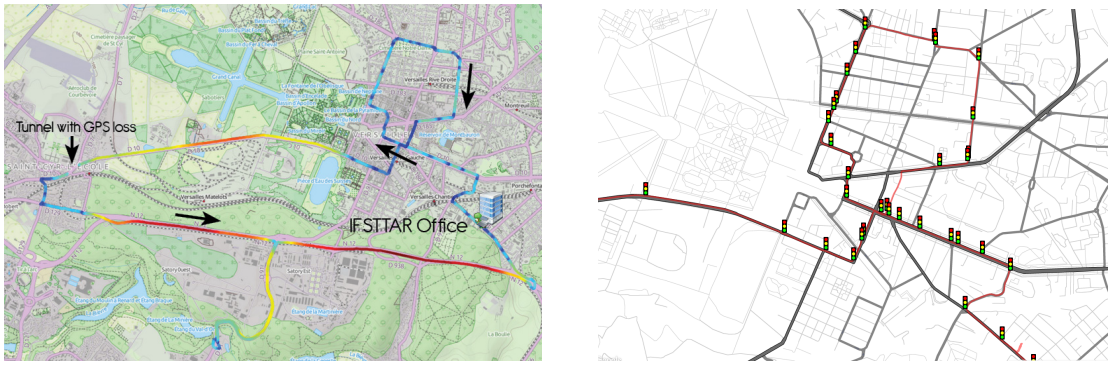


FIGURE 3.1 – À gauche : boucle de l'expérimentation. L'échelle des couleurs représente la vitesse moyenne des véhicules sur le parcours (du bleu vers le rouge à mesure que la vitesse augmente). À droite : positions des feux tricolores sur la partie urbaine du circuit.

le sens horaire au niveau de la boucle supérieure et dans le sens anti-horaire sur la boucle inférieure, et présente une large variété de morphologies, allant de conditions urbaines au nord, à des conditions inter-urbaines et rurales au centre et comporte également une section de type autoroute au sud. Le trajet complet nécessite entre 45 et 60 minutes, en fonction de la fluidité du trafic. Il est important de noter que les automobilistes recrutés n'étaient pas des conducteurs professionnels et n'avaient pas d'expérience particulière en éco-conduite, permettant ainsi de poser l'hypothèse, nécessaire par la suite, que les styles de conduite des participants sont représentatifs de ceux de la population en général.

Deux véhicules du même modèle ont été utilisés pour l'expérimentation, après avoir été équipés d'un *data logger* connecté au bus CAN (*cf* section 1.1.2) du véhicule et à un récepteur GPS Garmin 16x LVC. Sur chaque trajet, 87 paramètres ont été enregistrés, incluant des données cinématiques (les timestamps, les positions du véhicule, les vitesses relevées par Doppler...) ainsi que des paramètres plus spécifiques tels que la consommation de carburant ou encore le régime moteur. La plupart des paramètres issus du bus de données sont échantillonnés à la fréquence de 10 Hz. Comme sur la plupart des récepteurs, les positions et vitesses GPS sont uniquement disponibles toutes les secondes. D'autre part, les temps de démarrage respectifs des différents capteurs étant variables, on obtient en toute rigueur des mesures désynchronisées. Un premier pré-traitement a été effectué par l'IFSTTAR pour resynchroniser toutes les mesures sur une fréquence de base à 1 Hz.

Il est important de noter également que les mesures de vitesse sont effectuées par le Doppler du GPS (Kaplan et Hegarty, 2005), et non par différenciation numérique des positions GPS, ce qui en pratique permet une mesure un ordre de magnitude plus précise à ± 0.1 m/s (*cf* section 2.2.3). En pratique, tous les recepteurs GPS ne sont pas équipés de cette fonctionnalité de mesure Doppler, cependant comme mentionné par Schroedl et al. (2004), le coût décroissant des technologies de géolocalisation incite à faire le pari qu'à l'avenir, les véhicules seront équipés de systèmes tels que des GPS différentiels ou/et à mesure de phase (Warnant et al., 2018), permettant des degrés de précision équivalents. Dans les traitements qui suivent, les longitudes et latitudes GPS ont été converties en coordonnées planimétriques dans le système de projection légal français Lambert 93.

Pour construire la vérité terrain, 235 éléments de signalisation routières ont été relevés sur le terrain, dont 44 feux tricolores, 5 stops, 9 cassis et 92 passages piétons. Chacun de ces

éléments a été géoréférencé précisément sur l'orthoimagerie du Géoportail [2] avec une erreur de localisation inférieure au mètre.

3.1.2 Phase de prétraitements

Les profils spatiaux de vitesse constituent un outil classique dans l'analyse de l'impact des éléments d'infrastructure routière sur les styles de conduites (Laureshyn et al., 2009; Moreno et García, 2013) et nous avons passé en revue leurs propriétés et méthodes d'estimation dans la section 2.3. Notons $x \in \mathbb{R}$ l'abscisse curviligne d'un véhicule le long du linéaire routier. Un profil spatial de vitesse est défini par une fonction $v : x \to v(x) \in \mathbb{R}^+$, la vitesse instantanée du véhicule au point x. La figure 3.2 représente un exemple d'un ensemble de profils de vitesse sur une portion de route.

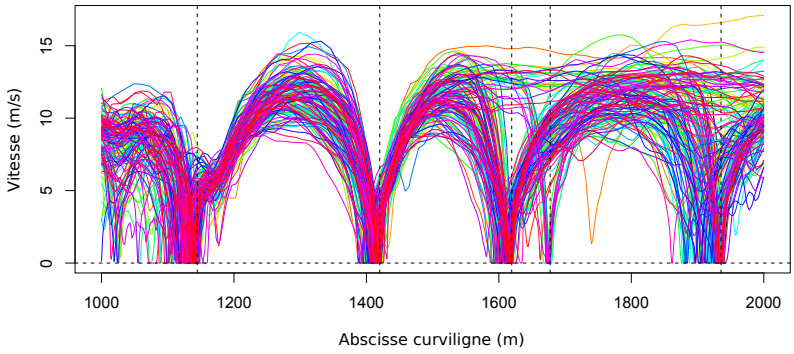


FIGURE 3.2 – Un ensemble de profils spatiaux de vitesse. Chaque couleur correspond à un véhicule individuel. Les positions des feux sont figurées par les lignes pointillées verticales.

Pour calculer les profils de vitesse, on pourrait considérer que l'abscisse curviligne est la distance parcourue, mais le graphique résultant n'a alors de sens que si tous les véhicules ont suivi exactement la même trajectoire. Cette hypothèse est bien évidemment irréaliste, et la distance réelle parcourue par chaque véhicule est fortement liée au comportement du conducteur. D'autre part, si l'abscisse curviligne est calculée uniquement à l'aide des observations données par le récepteur GPS, le bruit de mesure sur les positions induit une erreur sur l'abscisse curviligne, dont la variance d'ensemble croît linéairement avec la distance parcourue (*cf* section 2.4.3.2), conduisant ainsi les profils spatiaux de vitesse à ne plus être mutuellement alignés. Une solution pour remédier à ce problème consiste à recaler les données sur un circuit de référence à l'aide d'un algorithme de map-matching (2.4). Cette opération permet a) d'exprimer la position du véhicule dans un système de référence unidimensionnel et b) d'améliorer la précision du positionnement, dans la mesure où le réseau routier de référence ajoute une information de localisation supplémentaire (*cf* 2.4.4.1). Ce deuxième point est particulièrement important en ville, où les phénomènes de canyons urbains peuvent significativement perturber la réception GPS.

Pour réaliser ce recalage, nous avons utilisé l'algorithme de Newson et Krumm (2009), qui constitue une référence dans la littérature des algorithmes de map-matching. Cependant,

2. Web service cartographique national français : https ://www.geoportail.gouv.fr/

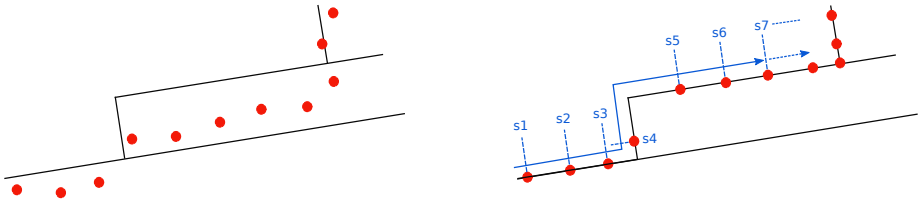


FIGURE 3.3 – Map-matching d'une trajectoire GPS (points rouges) sur un réseau routier de référence (en noir) et calcul d'une abscisse curviligne (lignes pointillées bleues).

le map-matching d'une trajectoire curviligne sur un circuit de référence linéaire par morceaux peut produire des artefacts systématiques. Par exemple sur la figure 3.3, les points 4 et 5 sont notablement plus distants après projection sur le circuit de référence. Pour résoudre ce problème, les profils de vitesse ont été corrigés à l'aide d'un lissage de Kalman avec contrainte de monotonie (2.5). Cette étape permet d'obtenir des profils beaucoup plus naturels.

À l'issue de cette étape, le jeu de données contient 144 profils de vitesse, pour une durée totale de 126 heures et une distance parcourue de 3650 km.

3.1.3 Calcul des fenêtres glissantes

L'objectif consiste à entraîner un algorithme d'apprentissage automatique à détecter la présence d'un feu tricolore sur une portion de route donnée, à partir d'un ensemble de profils de vitesse. Pour ce faire, on découpe les profils de vitesse complets suivant un système de fenêtres glissantes. Chaque fenêtre contient 144 courbes, et couvre une portion de longueur L du circuit. Deux fenêtres consécutives sont séparées par une distance t, choisie inférieure à L, de sorte que les fenêtres se recouvrent partiellement (figure 3.4). Cette augmentation artificielle du nombre de données d'entraînement va permettre aux algorithmes d'apprendre à détecter un feu tricolore indépendamment de sa position dans la fenêtre. Naturellement, puisque les instances voisines sont significativement corrélées entre elles, il faut s'attendre à ce que le gain en précision ne soit pas en commune mesure avec l'augmentation du nombre de données. De plus, il faudra prendre cette corrélation en compte lors de la phase de découpage entraînement/validation, afin de ne pas valider l'algorithme avec des instances (même partiellement) utilisées pour l'apprentissage.

Chaque fenêtre i représente une instance individuelle $X^{(i)}$, et différentes méthodes vont être employées pour en extraire un vecteur de descripteurs numériques $(X_1^{(i)}, X_2^{(i)}, ...X_p^{(i)}) \in \mathbb{R}^p$, dont la dimension p dépend également de l'approche choisie. Il reste alors à calculer la variable cible (ou variable d'intérêt) Y. Puisque nous traitons un problème de classification binaire, $Y^{(i)}$ prend ses valeurs dans l'ensemble $\{0, 1\}$, avec la convention $Y^{(i)} = 1$ si la fenêtre i contient un feu, et $Y^{(i)} = 0$ dans le cas contraire.

Pour l'expérimentation, nous avons fixé $L = 100$ m et $t = 10$ m (90% de recouvrement). Ce choix a été principalement motivé par le fait que la longueur de 100 m semble un compromis acceptable pour obtenir des fenêtres suffisamment larges pour *capturer* le signal nécessaire à la détection d'un feux tricolore, tout en limitant le risque d'inclure plus d'un

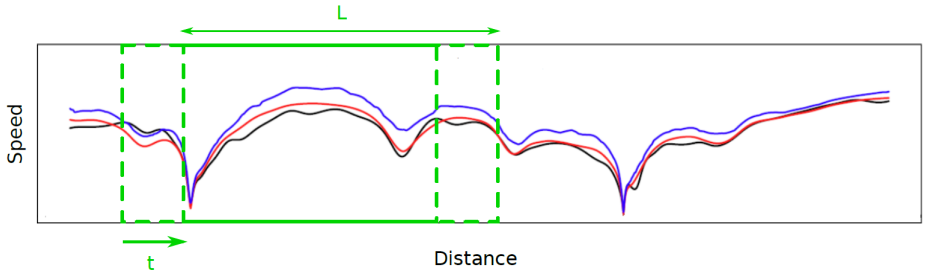


FIGURE 3.4 – Fenêtres glissantes extraites sur un ensemble de 3 profils de vitesse. Chaque fenêtre couvre une portion L du parcours, et est décalée d'une distance t par rapport à sa voisine de gauche.

feu dans une même fenêtre. D'autre part, un recouvrement de 90% permet une couverture dense des positions relatives des feux tricolores dans les fenêtres glissantes, sur l'ensemble du jeu de données.

Avec cette méthode, 2505 fenêtres ont été extraites, dont 402 (16%) représentent des instances positives.

Dans une dernière étape, nous avons procédé à l'élimination des profils outliers à l'aide de la méthode des box-plots. Introduites par Sun et Genton (2011), les box-plots fonctionnelles constituent une généralisation de la notion de percentiles aux processus stochastiques, initialement fondée sur les travaux de López-Pintado et Romo (2009). Étant données n réalisations $Y = \{y_1, y_2, ...y_n\}$ d'un processus à temps continu sur $\mathcal{X} \subseteq \mathbb{R}$ et à valeurs réelles, on définit la bande de Y par le sous-ensemble de $\mathbb{R}^2$ compris entre les courbes : $B(Y) = \{(x, y) \mid \exists\, m, M \in [\![1; n]\!] \mid y_m(x) \leqslant y \leqslant y_M(x)\}$ (figure 3.5).

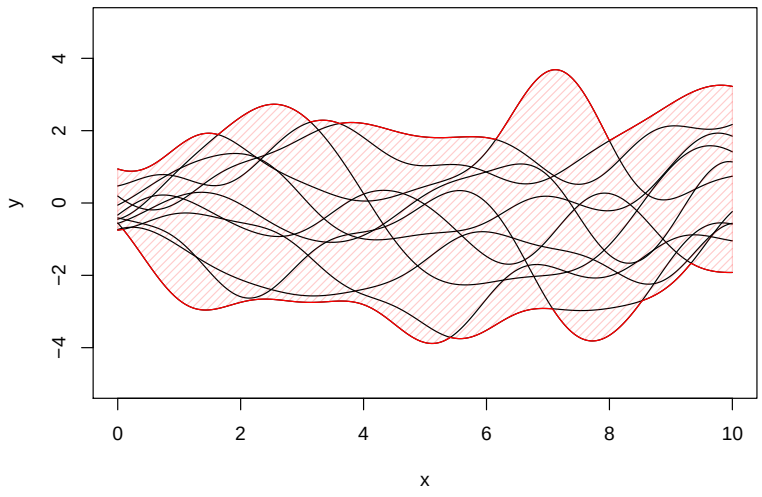


FIGURE 3.5 – Bande B d'un ensemble de 10 réalisations d'un processus stochastiques.

En prenant J un entier fixé compris entre 2 et le nombre n de courbes, on définit la profondeur de bande $BD(y)$ d'une courbe y par :

$$BD^{(J)}(y) = \sum_{j=2}^{J} \mathbb{P}[y(X) \in B(Y_1, Y_2, ...Y_j)], \tag{3.1}$$

où $B(Y_1, Y_2, ...Y_j)$ est la bande générée par j réalisations aléatoires du processus, et où X est tiré aléatoirement et uniformément sur $\mathcal{X}$.

D'un point de vue intuitif, la profondeur de bande traduit la centralité d'une courbe y au sein d'un ensemble de courbes. Plus $BD(y)$ est grand, plus il est probable que la réalisation y soit (au moins partiellement) comprise dans la bande de Y et plus elle peut-être considérée comme centrale. Notons que pour un jeu de courbes Y contenant n courbes (avec $n \gg J$, l'estimation numérique 3.1 nécessite un nombre de comparaisons en $\mathcal{O}(pn^J)$, où p est le nombre de points d'échantillon des courbes. En pratique, Sun et Genton (2011) indiquent que les résultats obtenus sont relativement stables indépendamment de la valeur retenue pour J. On se restreint donc en pratique à $J = 2$, ce qui présente en plus l'avantage de fournir des valeurs de profondeurs interprétables en tant que probabilités.

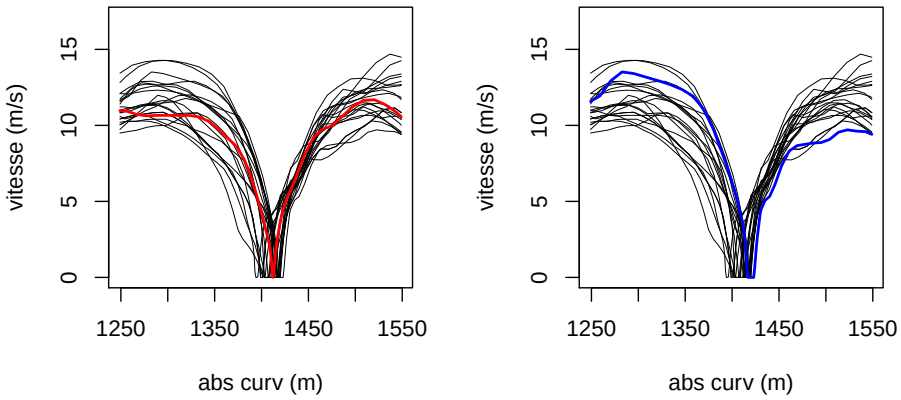


FIGURE 3.6 – Ensemble de 20 profils de vitesse et courbe la plus centrale (à gauche) avec une profondeur de 42.9 % et la moins centrale (à droite) avec une profondeur de 15.1 %.

Cette méthode d'élimination des outliers a permis de retirer environ 15% des profils sur l'ensemble des fenêtres, dont la plupart correspondaient à des problèmes dus au capteurs ou à l'enregistrement des données par la data logger.

La figure 3.7 donne 4 exemples de fenêtres glissantes. Les deux images supérieures présentent des instances négatives (un stop à gauche et un passage piéton à droite), tandis que les deux images inférieures présentent des instances positives.

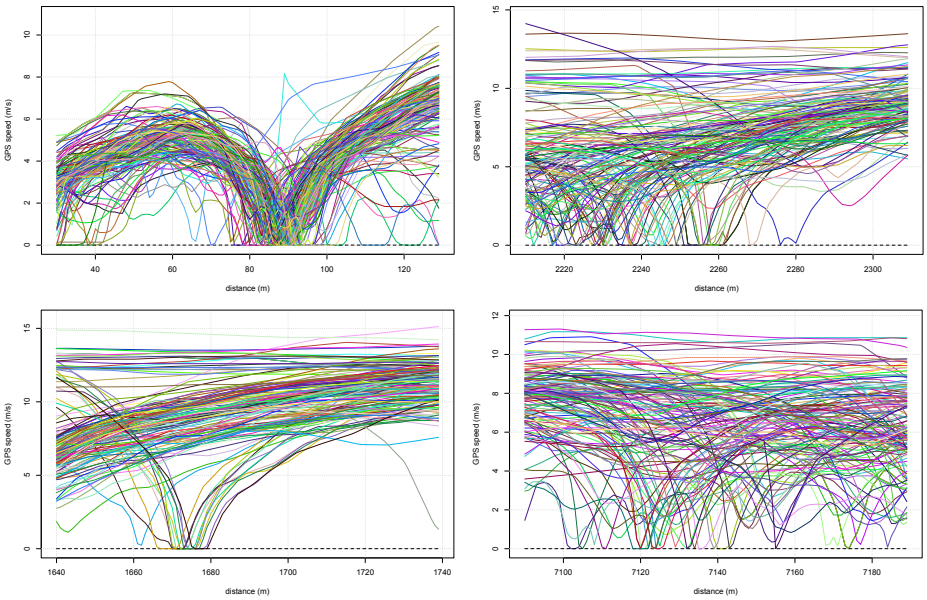


FIGURE 3.7 – Quatre exemples de fenêtres glissantes

3.2 Éléments d'apprentissage statistique

3.2.1 Méthodes d'apprentissage supervisé

Dans cette section, nous exposons brièvement les cinq algorithmes d'apprentissage supervisé utilisés dans cette section, pour le cas de la classification binaire pour des descripteurs réels. Pour la plupart de ces algorithmes, la généralisation au cas de la classification catégorielle est immédiate, en remarquant qu'un problème univarié à k classes peut toujours être transformé en un problème de classification binaire à k variables (Dietterich et Bakiri, 1991). Nous verrons un cas d'extension à la régression dans le chapitre 4.

On désigne par $X = (X_1, X_2, ...X_p) \in \mathbb{R}^p$ l'ensemble des descripteurs d'une instance d'étiquette $Y \in \{0, 1\}$. Le jeu d'entraînement $\mathcal{D}_n = \{(X^1, Y^1)...(X^n, Y^n)\}$ est un n-échantillon de réalisations de la loi jointe inconnue $p(X, Y)$. On note $X^{(n+1)}$ le vecteur de descripteurs d'une nouvelle instance d'étiquette $Y^{(n+1)}$ inconnue et à déterminer.

3.2.1.1 Classifieur bayésien naïf

Le classifieur bayésien naïf (ou *Naive Bayes* dans la littérature anglo-saxonne) pose l'hypothèse d'indépendance des descripteurs conditionnellement à l'étiquette :

$$P(X_i|Y, X_{j\neq i}) = P(X_i|Y). \tag{3.2}$$

À l'aide de la règle de chaînage, on tire facilement de la relation 3.2 la simplification suivante (pour $i \neq j$) : $P(X_i, X_j|Y) = P(X_i|Y)P(X_j|Y, X_i) = P(X_i|Y)P(X_j|Y)$, d'où

l'expression factorisée de la loi jointe :

$$P(X,Y) = P(Y)P(X|Y) = P(Y)\prod_{i=1}^{p} P(X_i|Y). \tag{3.3}$$

La figure 3.8 donne une illustration de modèle graphique probabiliste associé au classifieur bayésien naïf, schématisant l'ensemble des distributions qui se factorisent sous la forme 3.3.

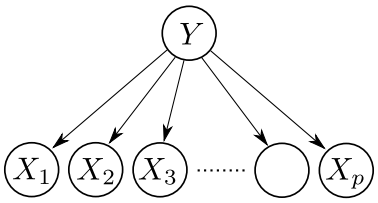


FIGURE 3.8 – Modèle graphique du *Naive Bayes* sur p descripteurs. Les flèches indiquent toutes les dépendances conditionnelles de la distribution de probabilité $p(X,Y)$.

La factorisation 3.3 couplée à la formule de Bayes permet d'exprimer la loi de probabilité sur l'étiquette conditionnellement aux descripteurs :

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} = \frac{P(Y)\prod_{i=1}^{p} P(X_i|Y)}{\sum_y P(y)\prod_{i=1}^{p} P(X_i|y)}, \tag{3.4}$$

où le dénominateur correspond à la loi marginale de X, *i.e.* au numérateur sommé sur toutes les affectations possibles pour y (en classification binaire la somme opère sur $\{0,1\}$).

La phase d'entraînement du modèle consiste alors à :

- Estimer les probabilités a priori $P(Y=0)$ et $P(Y=1)$ à partir des ratios d'instance positives et négatives. Lorsque le jeu d'entraînement est équilibré, $P(Y=0) = P(Y=1)$ et les termes $P(Y)$ s'annulent [3] dans 3.4.

- Estimer les $2p$ lois conditionnelles $P(X_i|Y)$. La littérature est vaste sur le sujet, mais la méthode la plus classiquement utilisée consiste à modéliser ces lois par des gaussiennes (Rogers et Girolami, 2016), de moyennes et variances à déterminer, soit un total de $4p$ paramètres à estimer, à savoir μ_{ij} et σ_{ij} pour $(i,j) \in [\![1;p]\!] \times \{0,1\}$:

$$P(X_i = x|Y = j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\Big(-\frac{1}{2}\frac{(x-\mu_{ij})^2}{\sigma_{ij}^2}\Big).$$

  On montre aisément que les estimateurs par maximum de vraisemblance de μ_{ij} et σ_{ij} sont respectivement la moyenne et l'écart-type (biaisé) empiriques des valeurs prises

3. Sauf à décider expressement d'ajouter un poids sur les classes d'étiquettes pour compenser un éventuel défaut de représentativité statistique de l'échantillon d'entraînement.

par le i^{eme} descripteur pour les données de la classe j (Zivot, 2009).

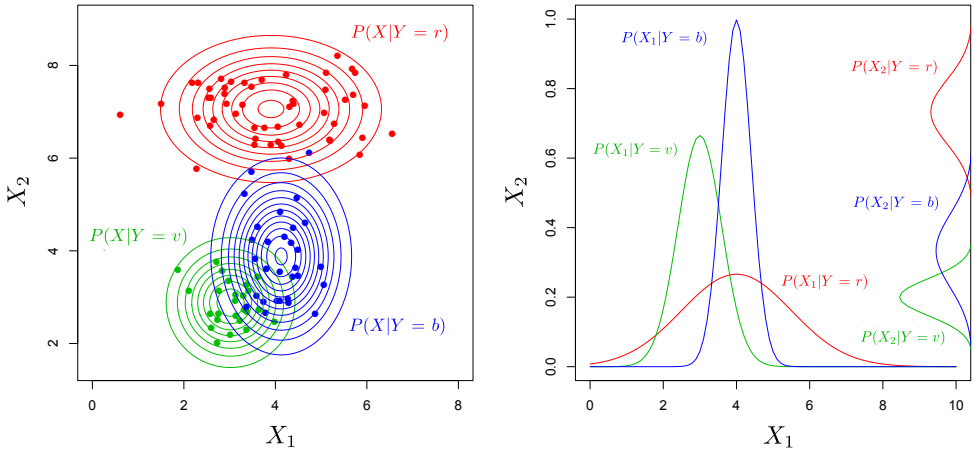


FIGURE 3.9 – Exemple d'apprentissage d'un modèle bayésien naïf pour un problème de classification à 3 classes (rouge, verte et bleue) et 2 descripteurs (X_1, X_2).

La figure 3.9 donne un exemple graphique d'apprentissage du modèle sur un problème simple de classification à 3 classes et 2 descripteurs. L'inférence porte sur les paramètres de 6 distributions (figure de droite) : $\boldsymbol{\mu}_1 = (3.90, 3.02, 4.13)$, $\boldsymbol{\mu}_2 = (7.06, 2.88, 3.88)$, $\boldsymbol{\sigma}_1 = (1.31, 0.45, 0.47)$, $\boldsymbol{\sigma}_2 = (0.56, 0.45, 0.90)$, ainsi que sur les 3 probabilités a priori : $\boldsymbol{\theta} = (0.48, 0.24, 0.28)$. L'hypothèse naïve implique l'alignement des axes principaux des gaussiennes sur les axes du repère (figure de gauche).

L'indépendance conditionnelle des descripteurs est une hypothèse très forte et bien souvent irréaliste en pratique, en particulier lorsque les descripteurs sont significativement corrélés. En pratique, le classifieur bayésien naïf retourne des décisions robustes, y compris sur des problèmes modérément difficiles, mais avec des valeurs de probabilité associées bien souvent trop catégoriques (Rish et al., 2001).

3.2.1.2 Les k plus proches voisins

La méthode des k-ppv (ou k-NN pour k-Nearest-Neighbors) est un algorithme simple, requierant uniquement la définition d'un entier k (typiquement 5 à 10) ainsi que d'une métrique sur l'espace des descripteurs. Il s'agit d'une généralisation de la méthode d'interpolation par ppv (méthode de 1-ppv) décrite dans la partie pré-traitements (section ??).

Pour chaque nouvelle donnée $X^{(n+1)}$, la loi conditionnelle $P(Y^{(n+1)}|X^{(n+1)}, \mathcal{D}_n)$ est estimée empiriquement sur un jeu de données réduit composé des k données d'entraînement les plus proches de $X^{(n+1)}$ dans l'espace des descripteurs.

On remarque que dans le cas des k-ppv, la procédure d'entraînement à proprement parler est exécutée avant chaque inférence (ce qui n'exclut pas une phase de pré-calculs, par

exemple avec la construction d'un index spatial destiné à optimiser les requêtes de recherche des voisins).

Le nombre k de voisins à sélectionner est un paramètre à régler pour obtenir un compromis entre généralisation et sur-apprentissage (*cf* 3.2.1.6).

3.2.1.3 Les arbres de décision

Introduits par Breiman et al. (1984) sous l'acronyme CART, pour *Classification And Regression Trees*, les arbres de décision peuvent être vus comme une version élaborée des k-ppv.

Le concept de la méthode repose sur un découpage de l'espace des descripteurs à l'aide d'hyperplans séparateurs, de sorte à minimiser une certaine fonction d'*impureté* traduisant l'hétérogénéité des étiquettes des données situées de part et d'autre de la séparation. À chaque nœud j de l'arbre de décision, une donnée $\mathbf{x}$ quelconque est affectée d'un côté ou de l'autre par la règle (Louppe, 2014) :

$$h(\mathbf{x}, \boldsymbol{\theta}_j) = [\phi_j \mathbf{x} > \tau_j] \quad \in \{0, 1\}, \tag{3.5}$$

où ϕ_j est un vecteur unitaire de dimension p et $\boldsymbol{\theta}_j = (\phi_j, \tau_j)$ est le vecteur des paramètres du nœud j, contenant l'indicateur de direction ϕ_j et le seuil de coupure $\tau_j \in \mathbb{R}$. Le vecteur θ_j est déterminé par :

$$\theta_j^* \in \underset{\theta_j}{\operatorname{argmax}} \left\{ H(\mathcal{S}_j) - \sum_{i \in \{0,1\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i) \right\}, \tag{3.6}$$

avec i qui indice les deux ensembles de la partition créée par l'hyperplan 3.5, où H désigne l'entropie des étiquettes des données d'entraînement et $\mathcal{S}_j^i$ est l'ensemble des données d'entraînement $\mathbf{x}$ telles que $h(\mathbf{x}, \boldsymbol{\theta}_j) = i$. En général, on impose aux hyperplans séparateurs d'être alignés avec les axes du repère, ce qui implique que ϕ est un vecteur indicateur de $\mathbb{R}^p$ ne contenant que des 0 excepté sur la dimension à tester. Dans le cas plus général, on parle d'arbres obliques (Do et al., 2009).

Notons que l'entropie n'est pas le seul choix possible de fonctions d'impureté (Scornet et al., 2015). La profondeur de l'arbre est paramétrée de sorte à obtenir un compromis entre généralisation et sur-apprentissage (*cf* 3.2.1.6).

Lorsqu'une nouvelle donnée $X^{(n+1)}$ est passée en entrée de l'algorithme, on teste successivement ses descripteurs (en fonction du schéma de l'arbre de décision), jusqu'à être capable de l'affecter à une feuille de l'arbre. La donnée est alors classée selon l'étiquette majoritaire des données d'entraînement situées dans la même feuille.

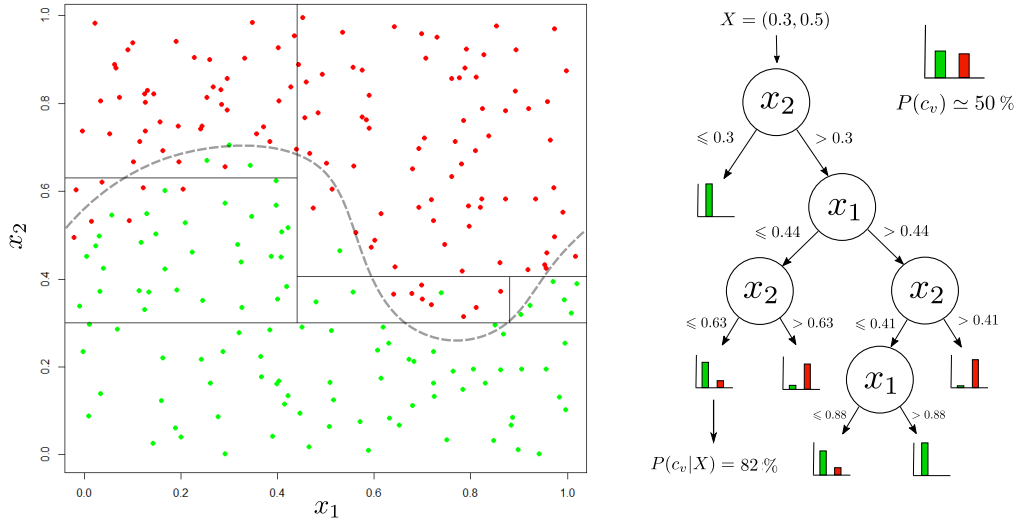


FIGURE 3.10 – Données d'entraînement et arbre de décision (à droite)

Prenons un exemple simple en figure 3.10 avec un problème de classification binaire : rouge/vert. Supposons que l'on souhaite attribuer une couleur à un nouveau point : $(0.3, 0.5)$.

La probabilité a priori que la couleur de ce point soit verte sachant les données d'exemple $P(y = \text{vert})$ est proche de 50% (les proportions des deux classes sont à peu près égales). La structure d'arbre représentant une partition de l'espace au niveau de ses feuilles, elle nous permet de prendre en compte la position du point dans le processus de décision. Il suffit de suivre le cheminement de l'arbre en répondant aux questions 3.5.

Par exemple, la première intersection de l'arbre teste si $x_2 \geqslant 0.3$, la réponse est positive ($x_2 = 0.5$) et on se déplace donc vers le fils droit du nœud. En suivant le cheminement jusqu'à une racine de l'arbre, l'histogramme des couleurs est beaucoup plus discriminé et on obtient une probabilité a posteriori $P(y = \text{vert} \mid x) = 82\%$ que le point soit vert. On l'affecte donc à la classe verte. En effet, dans la cellule correspondante (figure de gauche), 82% des points appartiennent à la classe verte.

Parmi les nombreux avantages des arbres de décision, on citera notamment leur interprétabilité et leur faible coût computationnel.

3.2.1.4 Les Random Ferns

La méthode des *Random Ferns* a été introduite par Ozuysal et al. (2007) dans un papier traitant d'une problématique de reconnaissance d'images. Depuis lors, elle a été utilisée dans de nombreux travaux (Villamizar et al., 2012; Aniruddha et Babu, 2014). On peut la considérer comme une généralisation du classifieur bayésien naïf.

Le principe général consiste à relaxer l'hypothèse d'indépendance conditionnelle des descripteurs, au profit de l'hypothèse plus lâche, et plus réaliste en pratique, d'indépendance entre groupes de variables. Plus formellement, étant donnée une partition F de l'ensemble

$[\![1;p]\!]$, avec $\text{card}(F) = d$, le nombre de groupes indépendants, l'hypothèse d'indépendance de groupes permet de réécrire 3.4 sous la forme modifiée :

$$P(Y|X) \propto P(Y) \prod_{k=1}^{d} P(\{X_i \ ; \ i \in F_k\}|Y), \tag{3.7}$$

où à nouveau, le facteur de proportionnalité ne dépend pas de Y et s'obtient par calcul de la loi marginale de X. Si $d = 1$ on retrouve l'expression générique de la loi jointe (intractable d'un point de vue numérique). À l'inverse, si $d = p$, chaque groupe ne contient qu'une variable, et l'expression 3.7 dégénère en un classifieur bayésien naïf du type 3.4. Pour cette raison, la méthode des Random Ferns est parfois qaulifée de *semi-naive bayes*, avec un compromis entre expressivité statistique des distributions modélisées et rapidité de calcul. En règle générale, le cardinal d de la partition est choisi de sorte à limiter les effectifs des groupes à un nombre de descripteurs de l'ordre de 5 à 10.

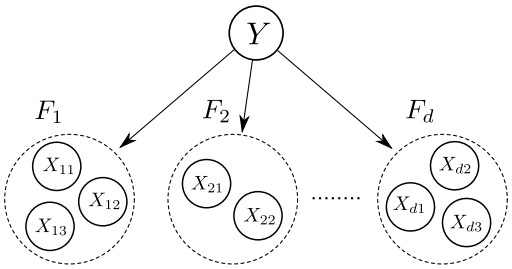


FIGURE 3.11 – Modèle d'indépendance conditionnelle entre groupes. On ne pose aucune hypothèse supplémentaire sur les distributions conditionnelles de F_k sachant Y.

Ayant posé le modèle 3.7 et l'entier d, il reste à définir la partition F. Ozuysal et al. (2007) proposent une stratégie de type fusion de prédicteurs *faibles*, en multipliant les probabilités postérieures fournies par un ensemble de modèles dont les partitions ont été tirées aléatoirement. Par exemple, pour L partitions générées $F^{(1)}, F^{(2)}, ...F^{(L)}$:

$$P(Y|X) = \frac{1}{Z} \prod_{l=1}^{L} P(Y) \prod_{k=1}^{d} P(\{X_i \ ; \ i \in F_k^{(l)}\}|Y), \tag{3.8}$$

où Z est une constante de normalisation permettant à la distribution de se sommer à 1.

Cette méthode de combinaison de prédicteurs faibles partiellement générés aléatoirement rapproche la méthode des Random Ferns des forêts d'arbres aléatoires.

3.2.1.5 Forêts d'arbres aléatoires

Le concept de stabilité, introduit empiriquement par Breiman et al. (1996), désigne la capacité d'un estimateur à produire des résultats similaires après de petites variations dans

le jeu de données d'entraînement. Dans ce même papier, il est mis en évidence (analytiquement ou par simulations) que certains algorithmes, tels que les k-ppv, sont stables par nature, au contraire d'autres estimateurs, comme celui des arbres CART, introduit dans le paragraphe 3.2.1.3.

Pour les estimateurs instables, l'agrégation des résultats d'un grand nombre de prédicteurs individuels peut permettre d'améliorer les performances de prédiction (Breiman, 1996). Le concept est illustré de manière simplifiée par Friedman et al. (2001) : étant donnée une variable $\bar{x}$ calculée empiriquement par la moyenne d'un échantillon de n réalisations x_i identiquement distribuées (selon une loi quelconque de variance σ mais non-indépendantes (supposons un facteur de corrélation ρ). La variance de $\bar{x}$ s'exprime alors classiquement par :

$$\begin{aligned} \mathrm{Var}(\bar{x}) &= \frac{1}{n^2}\Big[\sum_{i\neq j}\mathrm{Cov}(x_i, x_j) + \sum_{i}\mathrm{Var}(x_i)\Big] \\ &= \Big(\rho + \frac{1-\rho}{n}\Big)\sigma^2. \end{aligned} \tag{3.9}$$

L'équation 3.9 montre que lorsque les prédicteurs individuels sont complètement corrélés, la variance de prédiction est limitée (en borne inférieure) par la valeur de σ^2. Pour un nombre n de prédicteurs fixés, la variance est rendue minimale par une valeur spécifique $\rho^* \in [0, 1]$. L'objectif des méthodes d'ensemble, ou méthodes de *stabilisation* selon la terminologie de Breiman (1996), est de décorréler légèrement les arbres de sorte à abaisser la valeur de ρ, idéalement de 1 à ρ^*.

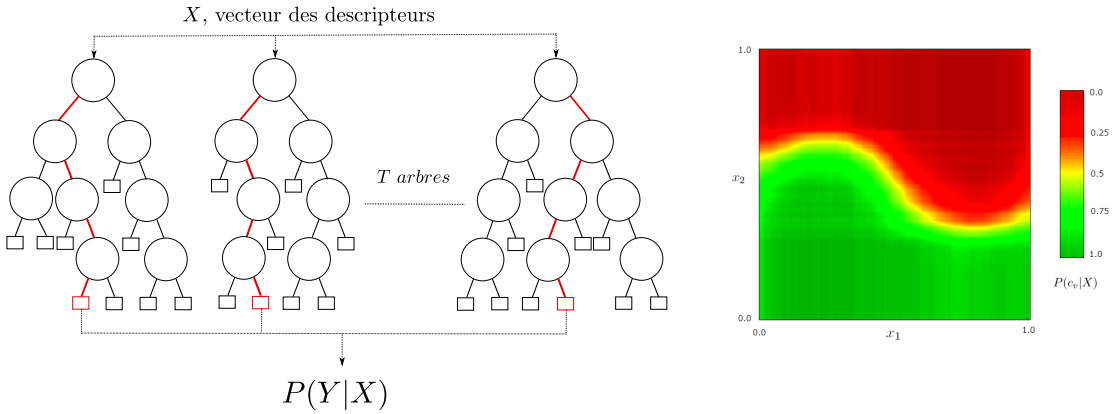


FIGURE 3.12 – À gauche : phase de prédiction du modèle de forêt aléatoire. À droite : résultat de l'inférence sur le problème modèle de classification illustré sur la figure 3.10.

Egalement introduites par Breiman (2001), les forêts aléatoires (*Random Forests*) constituent une version statistiquement robuste des arbres de décision, en s'appuyant au même titre que les Random Ferns sur les concepts de bootstrap statistique (Efron, 1992) et de méthode d'ensemble, pour réduire la variance de prédiction des arbres individuels. En revanche, avec cette *randomisation*, l'interprétabilité des arbres de décision est perdue.

L'idée centrale consiste à générer un nombre T (typiquement plusieurs centaines) d'arbres de décision dans lesquels on introduit une composante aléatoire à deux niveaux :

- À chaque construction d'un nouveau nœud dans l'arbre (equation 3.6), la coupe est réalisée dans le sous-espace vectoriel de $\mathbb{R}^p$ généré par un sous-ensemble de cardinal $m \leqslant p$ déterminé par tirage aléatoire sans remise dans les vecteurs de la base canonique. La littérature de référence (voir Breiman, 2001, par exemple) recommandent de choisir $m = \lceil \sqrt{p} \rceil$ pour les problèmes de classification et $m = \lceil p/3 \rceil$ pour les problèmes de régression.

- Chaque arbre t est construit avec une version *bootstrap* $\mathcal{D}_n^t$ du jeu de données : plus formellement $\mathcal{D}_n^t$ contient n données $(X^{(i)}, Y^{(i)})$, échantillonnées aléatoirement et avec remise dans $\mathcal{D}_n$. Notons que rien n'interdit en pratique la présence de données en doublon dans chaque échantillon bootstrap.

Une fois que la collection de T arbres aléatoires a été construite, l'inférence sur une nouvelle donnée $X^{(n+1)}$ est réalisée en calculant les probabilité conditionnelles sur chaque arbre, puis en moyennant les probabilités obtenues, comme illustré sur la figure 3.12 :

$$P(Y^{(n+1)}|X^{(n+1)}) = \frac{1}{T}\sum_{t=1}^{T} P_t(Y^{(n+1)}|X^{(n+1)}). \tag{3.10}$$

On parle alors de *bootstrap aggregating* (ou en abrégé de *bagging*). Notons que cette méthode d'agrégation par la moyenne des probabilités (Bostrom, 2007) est préférable à celle du vote majoritaire lorsque l'on souhaite une estimation robuste de la probabilité a posteriori, bien que les résultats de classification soit sensiblements identiques in fine, quelle que soit la méthode employée (Breiman, 1996). Une solution alternative pourrait consister à calculer le produit (normalisé) des probabilités individuelles :

$$P(Y^{(n+1)}|X^{(n+1)}) = \frac{1}{Z}\prod_{t=1}^{T} P_t(Y^{(n+1)}|X^{(n+1)}), \tag{3.11}$$

$$\text{avec : } Z = \sum_{y\in\mathcal{Y}}\prod_{t=1}^{T} P_t(Y^{(n+1)} = y|X^{(n+1)}). \tag{3.12}$$

L'expression 3.11 serait optimale si les arbres étaient indépendants. Cependant, de par leur processus de génération, ce n'est en pratique jamais le cas, et l'expression 3.10 donne bien souvent de meilleurs résultats.

L'algorithme des forêts aléatoires présente deux intérêts notables :

- **Erreur OOB :** l'erreur *out-of-bag* (OOB) désigne une procédure de mesure de la performance du modèle sans nécessiter de jeu de données de test. L'étape de bootstrap implique que chaque instance du jeu d'entraînement a une probabilité $(1 - 1/n)^n$ de ne pas être sélectionné dans la base d'apprentissage d'un prédicteur individuel. Lorsque le nombre d'échantillons est suffisament grand, la proportion d'exemples non

utilisés (échantillons out-of-bag) vaut :

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n = e^{-1} \approx 0.367. \tag{3.13}$$

Autrement dit, chaque arbre individuel n'est construit qu'avec 63 % des données d'entraînement, ce qui permet d'utiliser les données restantes pour valider le modèle. On appelle *erreur OOB* le taux d'erreur empirique mesuré sur l'échantillon OOB. Il est généralement admis que cet estimateur est biaisé (Breiman, 2001; Mitchell, 2011), mais suffisant en pratique lorsque l'on souhaite une évaluation approximative de la performance de classification, ou pour comparer plusieurs modèles opérant sur des sous-ensembles différents de descripteurs (Genuer et al., 2010).

- **Mesure d'importance des variables :** les forêts aléatoires offrent des méthodes simples pour estimer l'importance relative des descripteurs dans le processus de classification. On relève deux approches principales dans la littérature (Gregorutti, 2015). Dans une première méthode, l'importance d'une variable X_i donnée est calculée en fonction du gain en homogénéité des étiquettes des sous-arbres à chaque coupe dans laquelle elle intervient. En général, on prend le même critère d'homogénéité que celui qui a été utilisé pour la construction de l'arbre. Dans une deuxième approche plus empirique, on créé un jeu de données artificiel à partir du jeu d'origine, en permutant aléatoirement et pour i fixé, toutes les valeurs des vecteurs X en position i. On mesure alors l'importance de $\mathcal{I}(X_i) \in \mathbb{R}^+$ en évaluant la dégradation des performances (pour la fonction de perte $0-1$) :

$$\begin{aligned}\mathcal{I}(X_i) &= \mathbb{E}\big[|Y - f(\mathbf{x}_{(i)})|\big] - \mathbb{E}\big[|Y - f(\mathbf{x})|\big] \\ &= \mathbb{E}\big[|f(\mathbf{x}_{(i)}) - f(\mathbf{x})|\big],\end{aligned} \tag{3.14}$$

où l'espérance est en pratique estimée empiriquement sur l'échantillon OOB et où $\mathbf{x}_{(i)} = (X_1, ..., \widetilde{X}_i, ...X_p)$ désigne un vecteur de descripteurs dans lequel $\widetilde{X}_i$ est une réplication indépendante et de même loi que X_i. La seconde égalité de 3.14 résulte du fait que la permutation aléatoire de X_i ne peut apporter d'information, impliquant $\mathcal{I}(X_i) \geqslant 0$ (avec égalité uniquement si X_i est complètement non-informative).

Notons que Gregorutti (2015) a mis en évidence le lien existant entre la mesure d'importance par permutation aléatoire, et les indices de Sobol (Saltelli et al., 2000a), démontrant ainsi que les coefficients calculé par 3.14 ont un fondement statistique commun à celui de l'analyse de sensibilité.

Cette mesure d'importance est particulièrement utile dans les problèmes en grande dimension, pour lesquels on ne sait pas a priori quels descripteurs vont être informatifs dans le processus de décision.

Un intérêt subsidiaire des forêts aléatoires, est leur nombre d'hyper-paramètres relativement réduit, avec des règles de paramétrage empiriques établies par Breiman (2001) dans

son papier fondateur.

La complexité du processus de constructions des arbres rend difficile l'établissement de résultats théoriques. On pourra trouver quelques garanties de convergence, moyennant quelques hypothèses simplificatrices, dans les travaux de Breiman (2004), Biau et al. (2008a) ou encore Scornet et al. (2015). En pratique, malgré ces limitations, les forêts aléatoires donnent en général de très bons résultats et ont été utilisées dans de nombreux problèmes concrets, par exemple dans le contexte du véhicule autonome (Zaklouta et al., 2011; Marin et al., 2013), en traitement automatique de la langue (Palomino-Garibay et al., 2015; DeBarr et Wechsler, 2009), en détection de fraudes bancaires (Liu et al., 2015) ou encore en gestion des risques naturels (Wang et al., 2015; Tesfamariam et Liu, 2010).

Pour plus de détails sur les forêts aléatoires, on pourra se référer au travail complet et détaillé de Louppe (2014) pour les aspects théoriques, ou encore à Criminisi et al. (2011) pour une large gamme d'applications pratiques.

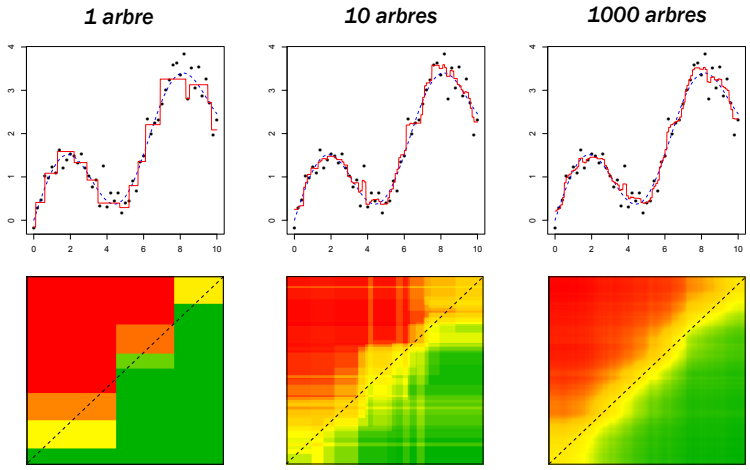


FIGURE 3.13 – Exemples d'application des forêts aléatoires pour différents effectifs d'arbres, sur le problème modèle de régression 2.13 (en haut) et sur le problème de classification $Y(X) = \mathbb{1}\{X_1 + \varepsilon \geqslant X_2\}$ où ε est une v.a. de ratio signal sur bruit égal à 7 (en bas). On remarque la robustesse de l'algorithme au sur-apprentissage (*cf* 3.2.1.6).

3.2.1.6 Le compromis biais-variance

Considérons un cas de problème de régression [4] avec une fonction de perte L quadratique. Le modèle inconnu à estimer est noté f, et on suppose que les descripteurs sont liés aux étiquettes par la relation non-déterministe $Y = f(X) + \varepsilon$ où ε est une variable aléatoire de loi inconnue, de moyenne nulle et de variance σ^2. Un choix naturel pour mesurer les performances d'un modèle $\hat{f}$ retourné par l'algorithme d'apprentissage consiste à évaluer l'espérance de la fonction de perte en un point $\mathbf{x}$ quelconque :

4. Pour un problème de classification binaire dans l'espace des étiquettes $\mathcal{Y} = \{0, 1\}$, la perte quadratique et la perte 0-1 donnent des résultats identiques.

$$\begin{aligned}\mathbb{E}[L(Y,\hat{f}(\mathbf{x}))] &= \mathbb{E}[(Y-\hat{f}(\mathbf{x}))^2] = \mathbb{E}[(Y-f(\mathbf{x})+f(\mathbf{x})-\hat{f}(\mathbf{x}))^2] \\ &= \mathbb{E}[\varepsilon^2] + \mathbb{E}\Big[\Big(f(\mathbf{x})-\mathbb{E}[\hat{f}(\mathbf{x})]+\mathbb{E}[\hat{f}(\mathbf{x})]-\hat{f}(\mathbf{x})\Big)^2\Big] \\ &= \sigma^2 + \big[f(\mathbf{x})-\mathbb{E}[\hat{f}(\mathbf{x})]\big]^2 + \mathbb{E}\big[(\hat{f}(\mathbf{x})-\mathbb{E}[\hat{f}(\mathbf{x})])^2\big] \\ &= \text{Erreur de Bayes } + \text{ Biais}^2(\hat{f}(\mathbf{x})) + \text{Var}(\hat{f}(\mathbf{x})),\end{aligned} \tag{3.15}$$

où les 3 termes de l'expression finale représentent respectivement :

- L'erreur incompressible (due à l'incertitude ε du phénomène à modéliser).
- Le biais du modèle, *i.e.* son erreur moyenne sur un grand nombre de prédictions
- La variance du modèle, *i.e.* sa tendance à reproduire des résultats différents à chaque nouvelle génération d'un jeu de données d'entraînement.

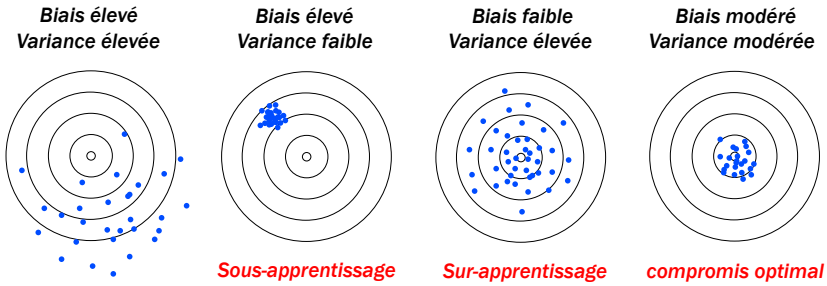


FIGURE 3.14 – Illustration schématique du compromis biais-variance dans le cadre de l'apprentissage statistique.

Dans la situation idéale où le phénomène inconnu est purement déterministe, $\sigma = 0$ et l'équation 3.15 devient :

$$\text{Erreur}(\hat{f}) = \text{Biais}^2(\hat{f}) + \text{Var}(\hat{f}). \tag{3.16}$$

En pratique, plus un modèle est complexe, moins il est biaisé, mais plus sa variance est forte. On parle du *compromis biais-variance*. On pourra trouver des mises en évidences analytiques de ce compromis pour les régressions linéaires, pour les k plus proche voisins, ou encore sous certaines hypothèses réductrices pour les forêts aléatoires dans l'ouvrage de Friedman et al. (2001). On donne en figure 3.15 une illustration graphique du phénomène sur un cas de régression d'un échantillon de points par moindres carrés avec un polynôme de degré variable. La figure 2.13 du chapitre pré-traitements illustre également ce compromis biais variance pour le cas d'une spline de régression.

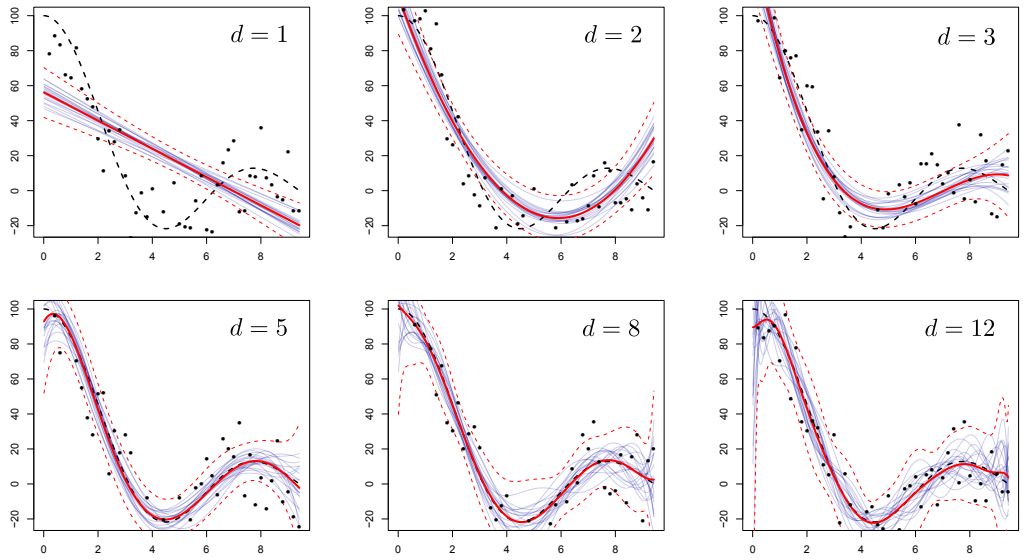


FIGURE 3.15 – Régression d'une fonction f inconnue (en pointillés noirs) par moindres carrés à l'aide de polynômes de degrés croissants d. À mesure que d augmente, l'écart entre l'espérance du modèle et la fonction f (le biais) tend à s'annuler tandis que la dispersion des courbes (la variance) augmente.

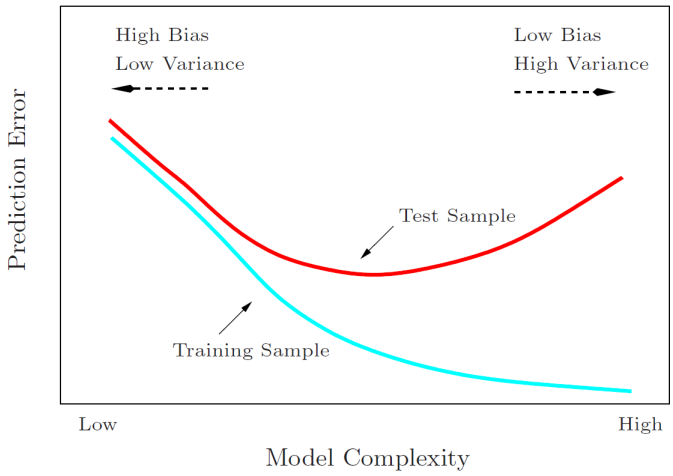


FIGURE 3.16 – Compromis biais-variance sur un modèle d'apprentissage, illustrant le principe d'overfitting. Source : Friedman et al. (2001)

Dans la plupart des algorithmes d'apprentissage, un paramètre permet de régler ce compromis, par exemple la profondeur de l'arbre (pour les arbres de décision et les forêts aléatoires), ou encore le nombre k de voisins séléctionnés dans la méthode des k-ppv. La figure 3.16 illustre la nécessité de contrôler la qualité de l'apprentissage sur un jeu de données séparé (qu'on appelle jeu de données de test). En pratique, lorsqu'il faut régler des hyper-paramètres, on a besoin d'un troisième jeu de données pour se prémunir contre le risque de sur-apprentissage de ces hyper-paramètres : le jeu de validation. Certains algorithmes proposent parfois une régularisation a posteriori. C'est le cas par exemple de

l'analyse en composantes principales (choix de la cascade de valeur propres), de la régression LASSO (sélection de variables), des arbres de décision (élagage des arbres) ou encore des réseaux de neurones artificiels (régularisation par suppression aléatoire de perceptrons).

3.2.2 Apprentissage de données fonctionnelles

Dans cette section nous passons en revue l'extension des algorithmes d'apprentissage au cas des données fonctionnelles. Dans tout ce qui suit, les données sont supposées être des fonctions de l'espace de Hilbert $L^2(\Omega)$, muni du produit scalaire usuel :

$$\forall\ f, g \in L^2(\Omega) \qquad \langle f, g \rangle_{L^2} = \int_\Omega f(x)g(x)dx, \tag{3.17}$$

où $\Omega \subseteq \mathbb{R}$ est un intervalle quelconque (pour la cas des fenêtres glissantes de profils de vitesse, on posera $\Omega = [0, 100]$).

Toute fonction de $L(\Omega)$ peut être approchée à ε près par une suite d'éléments de l'ensemble $\mathcal{F}$ des fonctions étagées à valeurs rationnelles. L'ensemble $\mathcal{F}$ est donc dense dans $L(\Omega)$. D'autre part, $\mathbb{Q}$ étant dénombrable, $\mathcal{F}$ l'est aussi, et par suite $L(\Omega)$ est un espace de Hilbert séparable. Toute fonction f admet donc une représentation sous la forme suivante :

$$\forall\ x \in \Omega\ :\quad f(x) = \sum_{k=1}^{+\infty} \langle f, \varphi_k \rangle \varphi_i(x), \tag{3.18}$$

où $\{\varphi_k\ ; k \in \mathbb{N}\}$ désigne une base de $L^2(\Omega)$ orthogonale.

Dans le cadre de l'apprentissage statistique de données fonctionnelles, nous utiliserons une approche similaire à Gregorutti (2015) en projetant les profils de vitesse sur un sous-espace vectoriel de dimension p finie, ce qui revient à tronquer la série de fonctions 3.18 au rang p (moyennant un réordonnancement *ad hoc* des vecteurs de la base) :

$$\forall\ x \in \Omega\ :\quad f(x) = \sum_{i=1}^{p} \langle f, \varphi_i \rangle \varphi_i(x) + \varepsilon_p(x), \tag{3.19}$$

avec $(\varepsilon_p)_{p \in \mathbb{N}}$ une suite de fonctions d'erreur résiduelle dépendant du degré p de l'approximation et convergeant vers la fonction nulle.

On note alors (pour une subdivision suffisamment fine $\{\omega_k\ ;\ k = 1, .., N\}$ de Ω) :

$$X_i = \langle f, \varphi_i \rangle = \int_\Omega f(x)\varphi_i(x)dx \approx \sum_{k=1}^{N} f(\omega_k)\varphi_i(\omega_k), \tag{3.20}$$

et $X = (X_1, ..., X_p)$ devient un vecteur de descripteurs classique de $\mathbb{R}^p$ qui peut être passé en entrée de l'un des algorithmes de classification passés en revue au paragraphe 3.2.1.

Le théorème suivant montre l'intérêt de travailler dans une base orthogonale :

Théorème 1. Soient $\varphi_1, \varphi_2, ...\varphi_p$ des fonctions deux à deux orthogonales et de norme 1 d'un espace de Hilbert H. Soit F le sous-espace de H engendré les fonctions (φ_i) et $f \in H$ une fonction quelconque. Alors quels que soient $\lambda_1, \lambda_2, ...\lambda_p \in \mathbb{R}$ on a l'inégalité suivante :

$$\left\| f - \sum_{i=1}^{p} \langle f, \varphi_i \rangle \varphi_i \right\| \leqslant \left\| f - \sum_{i=1}^{p} \lambda_i \varphi_i \right\|, \tag{3.21}$$

avec égalité si et seulement si $\lambda_i = \langle f, \varphi_i \rangle$.

Le théorème 1 nous dit que la projection orthogonale de f sur le sous-espace F est le meilleur représentant (au sens de la norme L^2) de f parmi toutes les fonctions de F. Le vecteur des X_i défini par 3.20 est donc un descripteur naturel des fonctions à analyser.

Dans les sections suivantes, nous considérons plusieurs exemples classiques de bases fonctionnelles qui peuvent être employées à des fins d'apprentissage statistique.

3.2.2.1 Base de B-splines

Les fonctions splines, passées en revue dans la section 2.3.2.4, désignent l'ensemble des fonctions polynomiales par morceaux sur une subdivision de Ω. Nous avons vu qu'elles correspondent exactement à l'espace des solutions d'un problème de minimisation de l'energie de flexion, permettant ainsi de donner une interprétation physique à cet ensemble.

La base des B-splines est un ensemble de fonctions à support compact permettant une décomposition des fonctions splines plus aisée (d'un point de vue numérique) qu'avec la base des puissances tronquées. On donne ci-dessous en figure 3.17 une illustration des vecteurs de cette base pour différents ordres m.

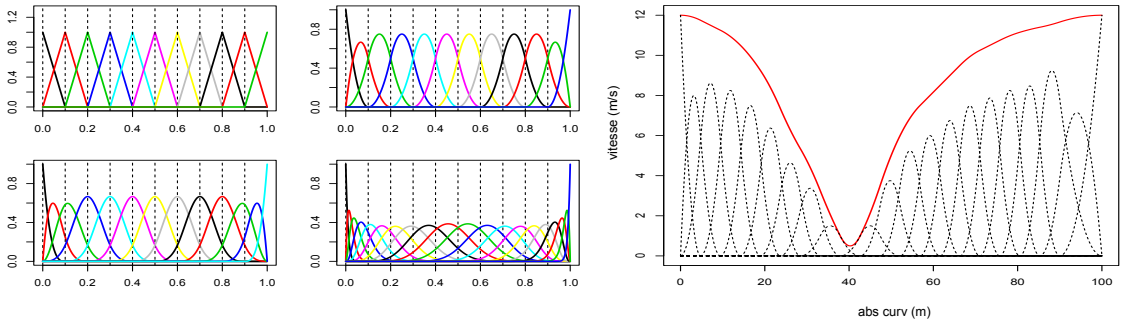


FIGURE 3.17 – À gauche : 4 bases de B-splines (d'ordre 1, 2, 3 et 10, de gauche à droite et de haut en bas). À droite : décomposition d'un profil spatial de vitesse GPS sur une base de B-splines cubiques.

Excepté pour le cas des splines d'ordre 0, une base de B-splines n'est pas orthogonale, et la décomposition 3.18 n'est plus valide. L'estimation des coefficients n'est pas directe, et

doit se faire par inversion d'une système linéaire (*cf* 2.21). Cependant, les fonctions de bases sont à support compact, ce qui permet à la base B-splines de partager d'une certaine manière les avantages numériques de bases orthogonales (Ramsay et Silverman, 2005).

D'autre part, notons que la base des B-splines n'est pas une base de $L^2(\Omega)$. En revanche, le théorème de Stone-Weierstrass nous garantit que toute fonction continue sur un segment est limite uniforme d'une suite de fonctions polynomiales. En conséquence, les B-splines forment une base de l'espace préhilbertien $\mathcal{C}(\Omega)$. Pour un échantillonnage fin de Ω, la base des B-splines peut être utilisée comme une base des profils spatiaux de vitesse (sous réserve de définir a priori le nombre de coefficients nécessaire dans la troncature 3.19).

Une limitation principale des B-splines vient du fait que les fonctions de bases sont spatialement localisées, impliquant ainsi que l'information totale est a priori équitablement répartie dans les coefficients de base, rendant ainsi peu opérante la compression de données par troncature de la série de fonctions, ce qui dans le cadre de l'apprentissage peut poser des problèmes similaires aux données de grandes dimensions (section 1.3.2).

Notons qu'il existe de nombreux autres exemples de bases polynomiales : Legendre, Tchebychev, Laguerre, Hermite... qui constituent des bases de Hilbert des espaces préhilbertiens des fonctions continues et de carré intégrable pour une certaine fonction de poids. Pour plus d'informations sur le sujet, on pourra consulter l'ouvrage de Gilsinger et Jaï (2010).

3.2.2.2 Base de Fourier

Les théorèmes de convergence de Dirichlet nous enseignent que toute fonction f périodique $\mathcal{C}^1$ par morceaux peut s'écrire comme une somme infinie de polynômes trigonométriques (avec une convergence normale de la série de fonctions si f est de plus continue). On appelle spectre du signal f, la suite des coefficients de la décomposition de f en séries de Fourier. La version trigonométrique du théorème de Stone-Weierstrass garantit que toute fonction continue est limite uniforme d'une série de fonctions sinusoïdales.

La transformation de Fourier (TF) est une extension de ce résultat au cas des fonctions non-périodiques, en considérant une valeur de pulsation ω asymptotiquement nulle (Cottet, 1997). Le spectre de f devient alors une fonction continue et il existe une expression intégrale analogue à 3.18. La transformée de Fourier est une fonction complexe dont le module $|\mathcal{F}[f](\omega)|$ indique la contribution des signaux de pulsation ω dans le signal original f. On peut montrer qu'il existe une application réciproque $\mathcal{F}^{-1}$ de forme très similaire à la TF directe, à un signe et une constante près, permettant de calculer la synthèse d'un ensemble infini de signaux de pulsations différentes. En ce sens, la TF opère une décomposition réversible d'une fonction donnée pour la représenter dans un espace fréquentiel.

Les données représentables en machine étant nécessairement de taille finie, il existe une version discrète de la TF, adaptée aux signaux échantillonnées : la transformation de Fourier discrète (TFD). Étant donné un N-échantillon, la TFD évalue N coefficients du spectre uniformément distribués entre la fréquence nulle et la moitié de la fréquence d'échantillonnage (en vertue du théorème de Shannon). Par exemple, la TFD d'un profil spatial de vitesse de résolution 1 m retourne 50 coefficients correspondants à la composante continue ainsi qu'à des fonctions de périodes étalées (en décroissance harmonique) entre 100 et 2 m.

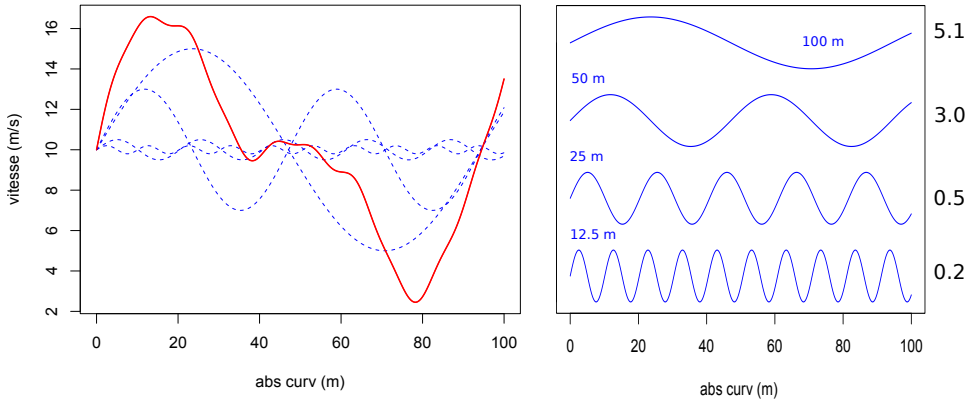


FIGURE 3.18 – Décomposition d'un profil de vitesse comme une somme de 4 fonctions sinusoïdales (de périodicités spatiales indiquées en bleu). Les coefficients à gauche représentent les coordonnées du profil dans la base fonctionnelle.

Toute fonction f de L^2 peut être décrite par la TFD avec une précision arbitrairement choisie en fonction du pas d'échantillonnage. De manière pratique, tout comme pour les splines, le choix de la base fonctionnelle doit être effectué en lien avec de l'erreur acceptable sur la troncature 3.19. En revanche, les decripteurs de Fourier se différencient des coefficients de B-splines dans le sens où ils sont complètement délocalisés dans le domaine spatial (l'information portée par chaque coefficient caractérise uniquement une bande spectrale de la fonction f). Cette limitation rend la TFD peu adaptée à notre cas d'étude.

La TFD possède les avantages notables d'offrir un fort taux de réduction de dimension (pour des signaux suffisamment réguliers) ainsi qu'une extension naturelle au plan complexe, permettant de traiter des courbes non-fonctionnelles, comme par exemple la trajectoire de phase de la figure 2.9. Cette propriété est utilisée en cartographie automatique pour l'analyse des formes baties (voir Bel Hadj Ali, 2001 par exemple).

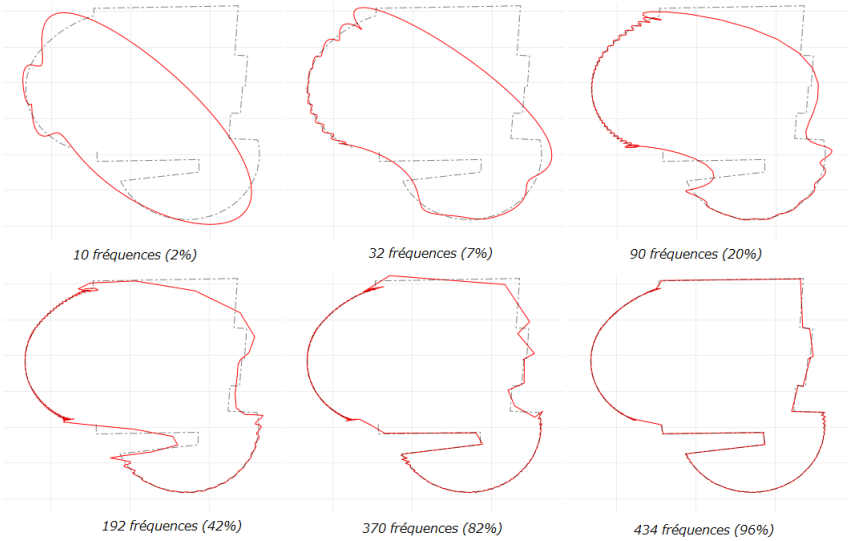


FIGURE 3.19 – Transformation de Fourier Discrète (TFD) d'un bâtiment.

3.2.2.3 Base d'ondelettes

Historiquement, la notion d'ondelettes est apparue du fait des limitations de la transformée de Fourier pour l'étude temporelle et spectrale de signaux (Chun-Lin, 2010). On sait que multiplication et produit de convolution sont analogues dans les deux espaces duaux, et toute analyse localisée du signal induit invariablement une dégradation du spectre calculé. En 1940, Gabor propose de moduler le signal à étudier par une gaussienne, réalisant ainsi le meilleur compromis possible en termes de résolutions dans le domaine joint temps-fréquence (Wei-lun, 2011). Le concept d'ondelettes sera introduit et théorisé dans les années 80 par Jean Morlet et Alex Grossman, en s'appuyant sur les travaux d'Alfred Haar (1909), dont le système d'ondelettes, restera jusqu'en 1985 la seule base connue d'ondelettes orthogonales. Quelques années plus tard, Ingrid Daubechies propose une méthode systématique de construction d'une base d'ondelettes orthogonales à support compact (Daubechies, 1988).

On considère un ensemble de sous-espaces imbriqués : $V_0 \subset V_1 \subset ... \subset L^2(\Omega)$ tels que pour tout j, il existe un espace W_j en somme directe orthogonale avec V_j dans V_{j+1} et engendré par une base de 2^j vecteurs orthonormaux : $\{\psi_{jk}\ ;\ k = 0, 1, ...2^j - 1\}$. L'espace W_j représente le niveau de détails manquant pour passer d'un niveau de résolution fonctionnelle au niveau plus fin suivant. Ces fonctions de base sont définies par des changements d'échelle et des translations d'une fonction ψ appelée *ondelette mère* : $\psi_{jk} = 2^{j/2}\psi(2^j x - k)$.

On peut montrer que l'ensemble des fonctions $\{\psi_{jk}\ ;\ j \in \mathbb{N},\ k = 0, 1, ...2^j - 1\}$ complété par une fonction φ appellée *ondelette père* (ou fonction d'échelle), forme une base orthonormale de $L^2(\Omega)$, avec comme implication directe que toute fonction f_l de $L^2(\Omega)$ peut s'écrire sous la forme :

$$\forall\ x \in \Omega \qquad f_l(x) = \omega_0^l \varphi(x) + \sum_{j=0}^{\infty} \sum_{k=0}^{2^j - 1} \omega_l^{jk} \psi_{jk}(x), \tag{3.22}$$

où les coefficients ω sont donnés par :

$$\omega_0^l = \int_\Omega f_l(x)\varphi(x)dx \qquad \text{et} \qquad \omega_{jk}^l = \int_\Omega f_l(x)\psi_{jk}(x)dx. \tag{3.23}$$

On peut réindicer les vecteurs de base en fonction du niveau de détails souhaité (Berlinet et al., 2008) : $\{\varphi_1, \varphi_2, ...\varphi_p\}$ et le vecteur de descripteurs $X^{(l)}$ d'une donnée fonctionnelle f_l s'écrit $(X_1^{(l)}, X_2^{(l)}, ...X_p^{(l)})$ avec :

$$\forall\ x \in \Omega \qquad f_l(x) = \sum_{i=1}^{p} X_p^{(l)} \varphi_i(x) + \varepsilon_l(x, p). \tag{3.24}$$

Le choix d'ondelettes le plus simple consiste à prendre le système de Haar (Haar, 1910), défini par une fonction d'échelle $\varphi = \mathbb{1}_{[0,1]}$ et une ondelette mère $\psi = \mathbb{1}_{[0,1/2]} - \mathbb{1}_{[1/2,1]}$. Notons qu'il s'agit d'un cas particulier des ondelettes de Daubechies (*cf* figure 3.20).

Une pratique classique pour la réduction de dimension, consiste à ordonner les fonctions de base par niveau de résolution spatial, puis à tronquer la série après les vecteurs de l'espace

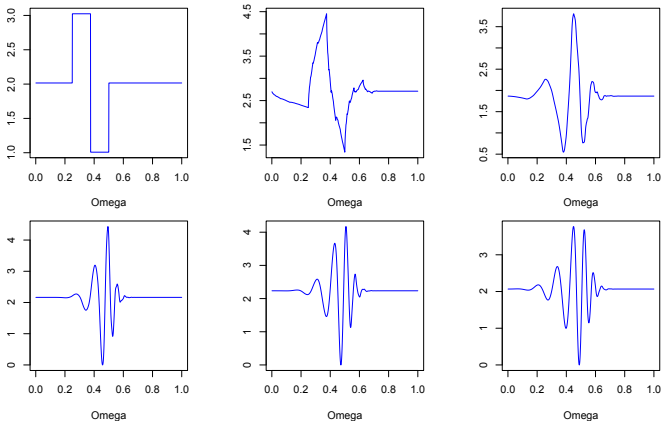


FIGURE 3.20 – Illustration de la forme des ondelettes mères (à différentes échelles) de Daubechies pour un nombre $m = 1, 2, 4, 6, 8$ et 10 de moments dissipants. La cas $m = 1$ (en haut à gauche) correspond au système de Haar.

V_j correspondant au niveau de détail souhaité. Notons qu'il existe d'autres méthodes plus fines, consistant à annuler les coefficients non-significatifs en fonction de différents seuils (déterminés par la théorie ou par validation croisée), donnant ainsi aux ondelettes un fort potentiel dans le domaine de l'estimation statistique non-paramétrique. Pour plus d'informations sur le sujet, on pourra se référer à Donoho et Johnstone (1994) ou encore à Nason (1995).

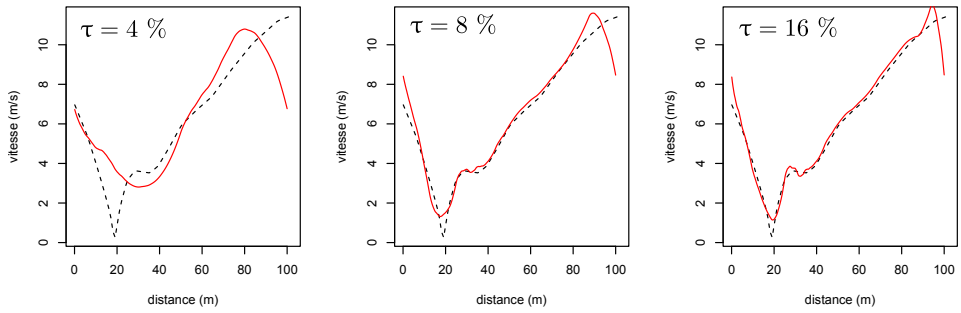


FIGURE 3.21 – Projection d'un profil spatial de vitesse sur l'espace généré par les 4, 8 et 16 premiers vecteurs de la base d'ondelettes de Daubechies d'ordre 4.

De par leur construction, les ondelettes réalisent le meilleur compromis entre localisations spatiale et fréquentielle (Berlinet et al., 2008). Elles sont donc pleinement adaptées au cas d'application de l'apprentissage, pour lequel on souhaite réduire la dimension des vecteurs de descripteurs à analyser, tout en conservant une information localisée (l'objectif *in fine* étant d'obtenir un géoréférencement précis de la signalisation routière). En ce sens, on peut les situer entre les bases de splines et la base de Fourier. Elles sont fréquemment utilisées dans la litérature de la classification de signaux, par exemple en acoustique (Daniels, 2010; Morizet et al., 2016), en diagnostic médical (Aydemir et Kayikcioglu, 2011; Sumathi et al.,

2014) et en reconnaisance d'images (Lotfi et al., 2009).

3.2.2.4 Base de Karhunen-Loève

La transformation de Karhunen-Loève (KL) est une généralisation fonctionnelle due initialement au travaux de Deville (1974) de l'analyse en composantes principales multivariée (Hotelling, 1933). Contrairement aux trois cas de bases fonctionnelles vues précédemment, la base de KL est déterminée à partir des données. Formellement, pour un processus stochastique du deuxième ordre X défini sur $[a, b] \subset \mathbb{R}$, de moyenne nulle et de fonction de covariance continue $K(.,.)$, les fonctions de bases φ_i sont les vecteurs propres de l'opérateur intégral de Hilbert-Schmidt de $L^2(\Omega)$:

$$(Af)(x) = \int_a^b K(x, \tau) f(\tau) d\tau. \tag{3.25}$$

On peut démontrer (voir Giambartolomei, 2015 par exemple) que les fonctions φ_i forment une base orthonormale de l'espace permettant d'écrire tout processus f de $L^2([a, b])$ sous la forme :

$$\forall\ x \in [a, b] \qquad f(x) = \sum_{i=1}^{+\infty} Z_i \varphi_i(x), \tag{3.26}$$

avec $Z_i = \langle f, \varphi_i \rangle$, une collection de variables aléatoires deux à deux orthogonales[5], de moyenne nulle et de variance égale à la valeur propre λ_i associée au vecteur de base φ_i. Cette propriété permet de trouver un ordonnancement naturel des fonctions de base, à partir de la fraction de variance totale expliquée :

$$\mathcal{I}(Z_i) = \frac{\lambda_i}{\sum_{k=1}^{\infty} \lambda_k}, \tag{3.27}$$

où le dénominateur est fini puisque le processus est supposé être du second ordre, et donc de variance intégrable sur l'intervalle $[a, b]$.

Théorème 2. Optimalité de la base de Karhunen-Loève

Étant donnée une base orthonormale $\boldsymbol{\phi} = \{\phi_k\}_{k \in \mathbb{N}}$ de $L^2([a, b])$, on note $\mathcal{E}_p(\boldsymbol{\phi})$ l'intégrale de l'espérance de l'erreur moyenne quadratique entre f et sa projection sur les p premiers vecteurs de la base $\boldsymbol{\phi}$ (on note A_i les coefficients de base) :

$$\mathcal{E}_p(f, \boldsymbol{\phi}) = \int_a^b \mathbb{E}\Big[\big(f(x) - \sum_{i=1}^{p} A_i \phi_i(x)\big)^2\Big] dx. \tag{3.28}$$

5. Pour cette raison, la décomposition KL est dite *doublement orthogonale* : les vecteurs de base (déterministes) sont orthogonaux dans l'espace des fonctions de carré intégrable ; les coefficients de la décomposition le sont dans l'espace des variables aléatoires du deuxième ordre.

Alors, l'erreur $\mathcal{E}_p(f, \boldsymbol{\phi})$ est minimale si et seulement si les fonctions $\{\phi_i\}_{i=1,..,p}$ sont les p premières fonctions de la base de Karhunen-Loève dans l'ordre décroissant des valeurs propres. On pourra trouver la preuve dans Giambartolomei (2015).

Autrement dit, à dimension fixée, la décomposition dans une base KL fournit la meilleure approximation possible. Ce résultat s'explique par le fait que contrairement au cas de la TF ou des ondelettes, la base KL est *data-driven*, et donc naturellement plus optimale. Cette propriété est illustrée sur la figure 3.24 à la fin de cette section.

On donne ci-dessous un exemple d'apprentissage utilisant une projection de profils spatiaux de vitesse sur les 10 premières fonctions de base. On applique un algorithme de classification non-supervisé pour classifier les profils de vitesse dans une fenêtre comprenant un feu tricolore en séparant les trajectoires des véhicules marquant l'arrêt au feu des autres, la vérité terrain ayant été établie par annotation manuelle. Nous avons utilisé la technique des k-means dont l'objectif consiste à répartir les données en k courbes (k étant un paramètre fixé) de sorte à minimiser la variance des données au sein de chaque classe, formellement, on cherche à minimiser un terme de distorsion :

$$W(\mathbf{c}) = \mathbb{E}\Big[\min_{l=1,...|\mathbf{c}|} ||X - \mathbf{c}||_2\Big] \approx \frac{1}{n}\sum_{i=1}^{N} \min_{l=1,...|\mathbf{c}|} ||X_i - \mathbf{c}||_2, \tag{3.29}$$

où l'approximation de droite représente le terme empirique calculable. En réalité, le problème de la minimisation de 3.29 est NP-complet, et on utilise en pratique des méthodes itératives telles que l'algorithme de Lloyd (Lloyd, 1982), issue du domaine de la quantification numérique des signaux.

Des approches similaires ont été utilisées dans de nombreux travaux de classification non-supervisée de courbes : Abraham et al. (2003) par exemple utilisent une projection des données sur la base des B-splines ; Auder et Fischer (2012) comparent différentes stratégies en amont de l'application des k-means, notamment, via les descripteurs de Fourier (3.2.2.2), les ondelettes de Haar (3.2.2.3), Karhunen-Loève (3.2.2.4), ainsi qu'une stratégie de sélection optimale au sein d'une famille de base d'ondelettes ; Barreyre et al. (2016) utilisent une approche similaire à la projection sur une base de Karhunen-Loève mais en substituant à la fonction de covariance, un noyau gaussien de paramètres *ad hoc* dans 3.25, la classification étant alors effectuée par un one-class-SVM. Enfin, dans une approche plus théorique, Biau et al. (2008b) analysent la performance des algorithmes de clustering dans un espace fonctionnel, par projection aléatoire sur un sous-espace vectoriel, en s'appuyant sur le lemme de Johnson–Lindenstrauss [6].

La méthode employée permet d'obtenir de bons résultats dans la classification des feux rouge / feux vert des profils. Dans cet exemple, la part de variance expliquée (équation 3.27) par la base tronquée s'élève à 99.997 %, ce qui permet une représentation très parcimonieuse des données fonctionnelles. Le taux de classifications correctes (évalué sur un ensemble représentatif de 15 fenêtres glissantes) avoisine les 99.6 %, contre 97.9 % avec la méthode par k-means sur les données représentées dans l'espace original.

6. Pour les problèmes en grande dimension, la projection aléatoire de tout ensemble de données sur un sous-espace de dimension logarithmique en le nombre total de données préserve approximativement (et avec une certaine probabilité) les distances entre les données.

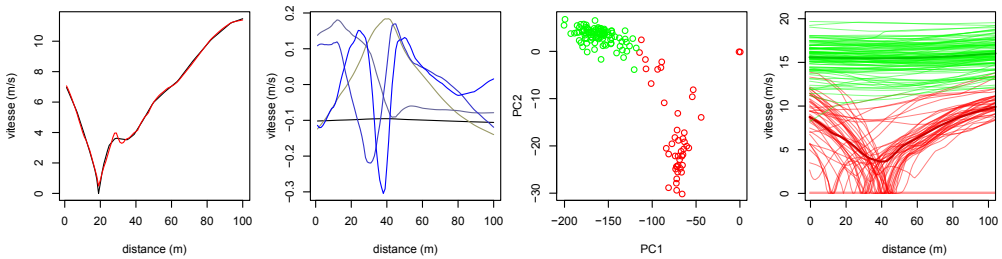


FIGURE 3.22 – De gauche à droite : 1) projection d'un profil sur les 10 premières fonctions de la base KL. 2) 5 premières fonctions de base. 3) Étiquettes inférées par l'algorithme des k-means représentées dans l'espace des 2 premières fonctions de base. 4) Résultat de la classification non-supervisée sur la séparation feu rouge / feu vert des profils.

La figure 3.22 (à droite) illustre également les fonctions centroïdes estimées par k-means [7]. On notera que ces centroïdes ne peuvent être considérés comme des profils de référence (*cf* section 2.3).

La figure 3.23 fournit un autre intérêt notable de la transformation de KL dans un contexte de simulation. Nous avons remarqué que les vecteurs de la base KL formaient une rotation de la base canonique, dans laquelle les coefficients des fonctions sont entièrement décorrélés. La simulation d'un nouveau profil de vitesse peut être effectuée selon un processus décrit par Phoon et al. (2002). Cette possibilité de génération de données synthétiques offre de nombreuses opportunités, par exemple en simulation de trafic, en analyse de sensibilité (nous utiliserons une approche similaire à la fin de ce chapitre) ou encore pour l'équilibrage d'un jeu de données d'entraînement (*cf* chapitre 4).

3.3 Choix des descripteurs et protocole expérimental

3.3.1 Introduction

Les performances de classification obtenues par les algorithmes d'apprentissage automatique peuvent être significativement impactées par le choix de la méthode de transformation d'un ensemble de courbes en un vecteur de variables explicatives réelles. Pour cette raison, trois méthodes différentes ont été expérimentées :

- Une méthode intuitive prenant compte de toutes les mesures de vitesses instantanées de l'ensemble des véhicules dans un unique vecteur ordonné (on parlera d'une *approche directe* dans la suite de ce chapitre).

7. Plus précisément, la projection sur les p premières fonctions de base étant par définition non-réversible, il s'agit de la moyenne d'ensemble des profils par classe dans l'espace original des courbes.

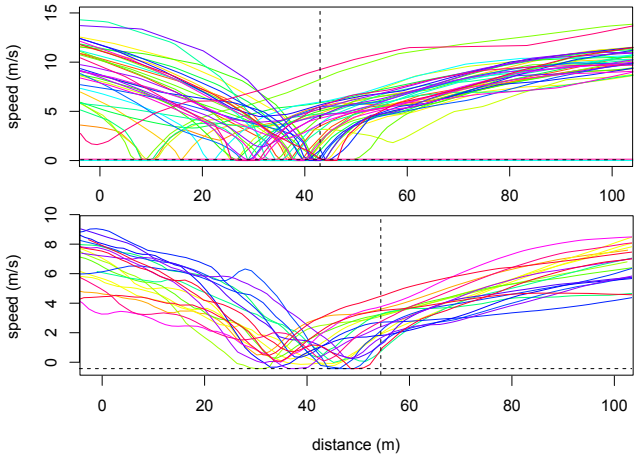


FIGURE 3.23 – En haut : profils de vitesse GPS de véhicules arretés au niveau d'un feu tricolore. En bas : génération synthétique de 20 profils de vitesse. Les positions des feux tricolores sont indiquées par les lignes verticales pointillées.

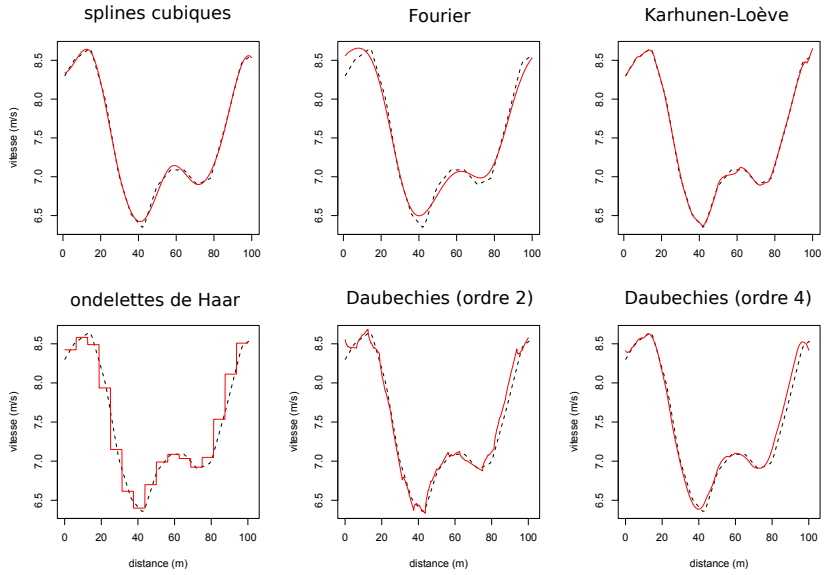


FIGURE 3.24 – Comparaison de plusieures bases fonctionnelles. Toutes les séries sont tronquées après le 16eme terme.

- Une approche impliquant de considérer le graphe des profils dans une fenêtre glissante comme une image (*approche image*).

- Une approche plus spécifique considérant les profils comme des objets individuels à partir desquels on cherchera à extraire des descripteurs de plus haut niveau, en lien avec leur nature fonctionnelle (*approche fonctionnelle*).

Dans cette section, $X \in \mathcal{X} \subseteq \mathbb{R}^p$ représente un vecteur de p variables explicatives caractérisant dans une certaine mesure une fenêtre glissante dont la variable cible (ou variable à expliquer) est $y \in \mathcal{Y} = \{0, 1\}$. Les profils spatiaux de vitesse sont indexés par $i \in \{1, 2..N\}$, de sorte que $v_i(x)$ désigne la vitesse instantanée du véhicule i à la position $x \in [0, L]$. Les profils v_i sont considérés comme étant N réalisations *i.i.d.* d'un processus stochastiques (non-stationnaire) de $\Omega \times [0, L] \to \mathbb{R}^+$ avec la notation abrégée $v_i(.) = v(\omega_i, .)$.

Notre objectif est de concevoir un classifieur opérationnel, adapté à tous les jeux de données, indépendamment du nombre de profils de vitesse disponibles. Cette considération est importante puisque le nombre de traces disponibles sur une zone donnée est en principe inconnu a priori, et que l'entraînement de l'algorithme ne peut raisonnablement être effectué pour toutes les tailles de jeu de données possibles. Plus formellement :

Contrainte 1. *La structure du vecteur de descripteurs X (et donc en particulier sa dimension), doivent être indépendants du nombre N de profils disponibles.*

La contrainte 1 assure en retour que le classifieur est capable de traiter tout ensemble de courbes. On attend cependant du classifieur d'être consistant, c'est-à-dire que la précision de l'inférence doit être une fonction croissante du nombre de traces disponibles sur une fenêtre donnée.

3.3.2 Approche directe

L'approche la plus directe et la plus intuitive, considère l'ensemble des valeurs point-à-point des N profils de vitesse comme un vecteur de descripteurs dont la dimension dépend du pas de discrétisation de l'espace. Pour un pas de 1 m, et un total de $N = 144$ profils, le vecteur X est composé de $p = 14400$ variables explicatives. Chacune de ces variables correspond à l'un des 100×144 mesures de vitesse reportées dans la fenêtre.

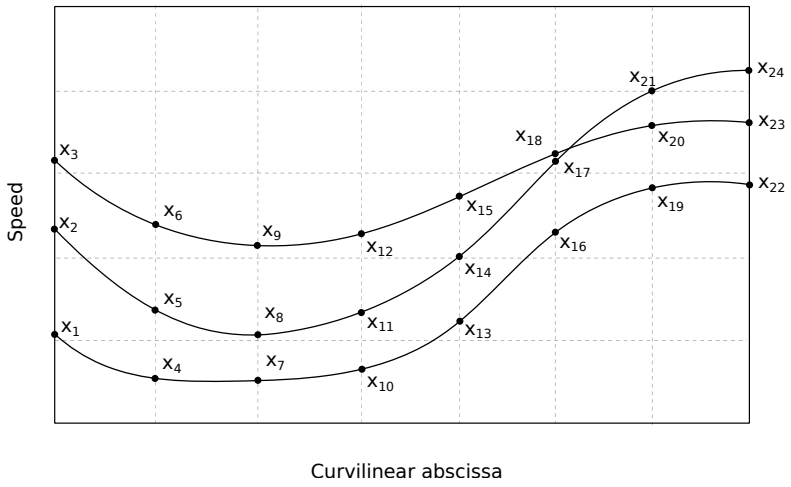


FIGURE 3.25 – Un exemple d'extraction de descripteurs sur 3 profils avec l'approche directe.

Dans une fenêtre donnée, l'ordonnancement des variables est effectué dans l'ordre lexicographique sur les couples abscisse curviligne - vitesse. La figure 3.25 donne un exemple d'extraction de variables explicatives sur une fenêtre glissante fictive contenant 3 profils de vitesse discrétisés au pas de un septième de la longueur totale L de la fenêtre, générant ainsi $3 \times 8 = 24$ descripteurs $(X_1, X_2, ...X_{24})$. Ce choix d'ordonnancement a pour conséquence de perdre le lien existant entre les séquences de points issus de la même trajectoire, mais en retour, il garantit que les fonctions de distance ne sont pas affectées par un ordonnancement arbitraire des trajectoires.

D'autre part, l'inégalité de réarrangement nous dit que, pour tout couple de n-uplets de $\mathbb{R}$: $x_1 \leqslant x_2 \leqslant ... \leqslant x_n$ et $v_1 \leqslant v_2 \leqslant ... \leqslant v_n$:

$$\forall\ \sigma \in \mathfrak{S}_n \qquad \sum_{i=1}^{n} x_i v_i \leqslant \sum_{i=1}^{n} x_i v_{\sigma(i)}, \tag{3.30}$$

où $\mathfrak{S}_n$ désigne l'ensemble des permutations possibles d'un ensemble de n éléments, et avec égalité si et seulement si σ est l'identité.

En conséquence, parmi les ordonnancements respectant la contrainte naturelle que les variables correspondant aux mêmes abscisses doivent partager les mêmes indices, ce choix possède l'avantage de minimiser la distance entre deux fenêtres. Cette propriété indique que les distances ultérieurement calculées (lors des processus d'entraînement et de validation) auront une certaine forme de cohérence. Étant donné que cette structuration du vecteur X ne prend pas en compte les trajectoires dans leur globalité, mais se concentre plutôt sur la distribution des vitesses en un lieu donné, elle peut être considérée comme plus proche d'une approche image que d'une approche fonctionnelle, quand bien même les descripteurs utilisés constituent un échantillonnage direct des valeurs prises par les fonctions.

Il est important de noter également que cet approche de base n'est pas réaliste en pratique puisque la dimension du vecteur X est contrainte à être un multiple entier du nombre de courbes disponibles sur la portion de route considérée, ce qui ne respecte pas la contrainte 1. Cette méthode nous servira d'approche de base dans les comparaisons.

3.3.3 Approche image

La deuxième approche est motivée par le fait qu'une reconnaissance humaine de l'éventuelle présence d'un feu tricolore sur un graphique de profils de vitesse repose essentiellement sur une recherche de motifs de référence simples (*e.g.* un feu tricolore est caractérisé par une annulation d'une partie de l'ensemble des courbes sur un certain intervalle de distance tandis que les courbes restantes ne semblent pas particulièrement impactées). En conséquence, on peut penser qu'une approche orientée image a des chances de pouvoir résoudre de manière satisfaisante le problème de classification posé. Pour ce faire, nous avons utilisé l'algorithme simple et efficace, introduit par Ozuysal et al. (2007), axé sur une comparaison de descripteurs binaires uniformément et aléatoirement échantillonnés sur l'emprise de l'image. Malgré sa concision, il a été démontré qu'il permet d'obtenir de très bons résultats, y compris sur des problèmes complexes de reconnaissance d'images (Villamizar et al., 2012; Aniruddha et Babu, 2014). L'algorithme présenté ci-dessous est une version adaptée pour le cas particulier de la reconnaissance d'un ensemble de données

de nature fonctionnelle. En particulier, nous avons utilisé l'extension proposée par Kursa (2012) pour traiter des descripteurs réels.

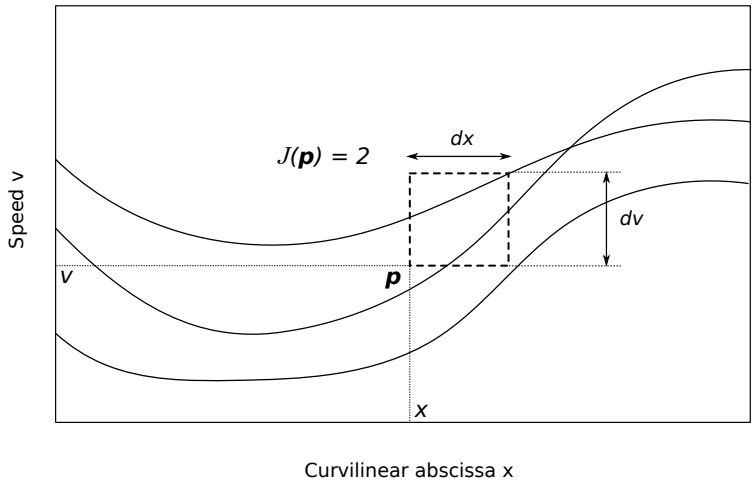


FIGURE 3.26 – Deux profils de vitesse intersectant la cellule rectangulaire (x, v).

Étant donnés une position $\mathbf{p} = (x, v) \in [0, L] \times [0, v_{max}]$ dans la fenêtre glissante, ainsi qu'un couple de paramètres de dimensions (dx, dv), introduisons une fonction $\mathcal{J}$ telle que $\mathcal{J}(\mathbf{p}) \in \mathbb{N}$ retourne le nombre de profils de vitesse intersectant une cellule rectangulaire de taille $dx \times dv$, et dont le coin inférieur gauche est positionné en $\mathbf{p}$ dans le repère de la fenêtre. La figure 3.26 donne un exemple de calcul de la fonction $\mathcal{J}$ pour un cas simple avec 3 courbes.

À l'initialisation de l'algorithme, un nombre p de couples de positions $(\mathbf{p}_j^1, \mathbf{p}_j^2)$ sont échantillonnées aléatoirement suivant une loi uniforme bivariée sur l'espace $[0, L] \times [0, v_{max}]$. À chaque couple, on associe une variable explicative X_j, définie par la différence normalisée des nombres de courbes intersectant les cellules positionnées en $\mathbf{p}_j^1$ et $\mathbf{p}_j^2$:

$$X_j = \frac{\mathcal{J}(\mathbf{p}_j^1) - \mathcal{J}(\mathbf{p}_j^2)}{N}. \tag{3.31}$$

La constante de normalisation N assure que le vecteur de variables explicatives n'est pas sensible au nombre total de profils de vitesse enregistrés dans la fenêtre. Chaque descripteur extrait avec cette méthode représente ainsi la différence entre les *densités de courbes* mesurées en deux positions spécifiques dans l'espace des profils.

Il est important de noter que, bien que les positions aient été initialement échantillonnées aléatoirement, elles restent fixées durant tout le processus d'entraînement et de validation. En conséquence, chaque variable X_j correspond au différentiel d'un couple spécifique de positions dans l'image, et cette définition est invariante sur l'ensemble des fenêtres glissantes que l'algorithme peut avoir à traiter. Pour un nombre de descripteurs suffisant, l'échan-

tillonnage uniforme fournit une couverture homogène et exhaustive de l'image. Puisque cette approche ne distingue pas les courbes individuelles, la contrainte d'indépendance de la structure du vecteur X au nombre de trajectoires disponibles dans la fenêtre, est automatiquement respectée (*cf* paragraphe 3.3.1).

Lorsque le nombre de profils est important (typiquement plus d'une centaine), le processus d'extraction peut être relativement coûteux en temps de calcul. Plusieurs stratégies peuvent être utilisées pour optimiser le calcul.

- Par simplifiation, on peut considérer qu'une courbe intersecte une cellule si et seulement si un point d'échantillon de la courbe est située dans la cellule.

- Avec une approche orientée sur les profils (i.e. on incrémente d'une unité toutes les cellules intersectant ce profil), à résolution spatiale des courbes fixée, la complexité de l'algorithme est de l'ordre de $\mathcal{O}(pNL)$ avec p le nombre de descripteurs à calculer, N le nombre de profils et L la largeur de la fenêtre. Les profils étant échantillonnés à intervalle régulier dans l'espace, il est avantageux de boucler sur les couples de cellules, puis pour chaque couple, de compter le nombre de courbes passant par chacune des cellules en récupérant directement les valeurs prises par les fonctions au niveau des abscisses curvilignes couvertes par les deux cellules. Avec cette approche, la complexité de l'algorithme d'extraction des descripteurs est réduite à $\mathcal{O}(pN)$, ce qui représente un gain considérable dans le temps de calcul.

- Les recherches d'intersections peuvent être encore accélérées en employant un mécanisme d'indexation spatiale : on définit une grille de résolution r sur l'espace des courbes de profils et on affecte en prétraitement à chaque cellule de la grille, les indices des profils intersectant la cellule. La recherche du nombre d'intersections entre une cellule c échantillonnée aléatoirement et les profils est donc réduite sur le sous-ensemble des courbes dont l'indice est compris dans les cellules de la grille d'index intersectant c. La complexité est alors dissociée par rapport au nombre p de cellules à échantillonner et au nombre de courbes. En pratique, $p \gg N$, et avec un choix judicieux de la résolution de la grille, on peut théoriquement obtenir une complexité moyenne réduite à $\mathcal{O}(p)$. Sur notre jeu de données, cette optimisation a permis de réduire le temps de calcul de 4 heures à 25 minutes, avec un gain de temps potentiellement plus marqué en calibrant plus finement la grille d'index (cf figure 3.27).

- Enfin, pour pouvoir générer p descripteurs, on doit en théorie échantillonner $2p$ cellules. En pratique, rien n'interdit de dupliquer certaines cellules pour les partager entre plusieurs descripteurs. En suivant cette approche, générer p descripteurs ne nécessite plus que $\sqrt{p}$ cellules. Cependant, on montre facilement à l'aide du lemme des anniversaires (Flajolet et Sedgewick, 2009) que la probabilité de dupliquer (au moins) un descripteur, est de l'ordre de $1 - e^{-p}$. Une solution intuitive à ce problème peut résider dans une sélection automatique de tous les couples possibles de cellules, parmi un ensemble de $\mathcal{O}(\sqrt{p})$ de cellules échantillonnées uniformément et aléatoirement. Plus précisément, en considérant l'antisymétrie de la définition 3.31, il faut échantillonner un nombre n_c de cellules égal à :

$$n_c = \lceil \frac{1}{2}(1 + \sqrt{1 + 8p}) \rceil. \tag{3.32}$$

Moyennant ces optimisations, la complexité en temps de la phase d'extraction des descripteurs est réduite de $\mathcal{O}(pNL)$ à $\mathcal{O}(\sqrt{p})$. Dans nos expérimentations, nous avons fixé $dx = 5$ m, $dv = 5$ km/h et $v_{max} = 80$ km/h. Un nombre $p = 15000$ descripteurs ont été extraits.

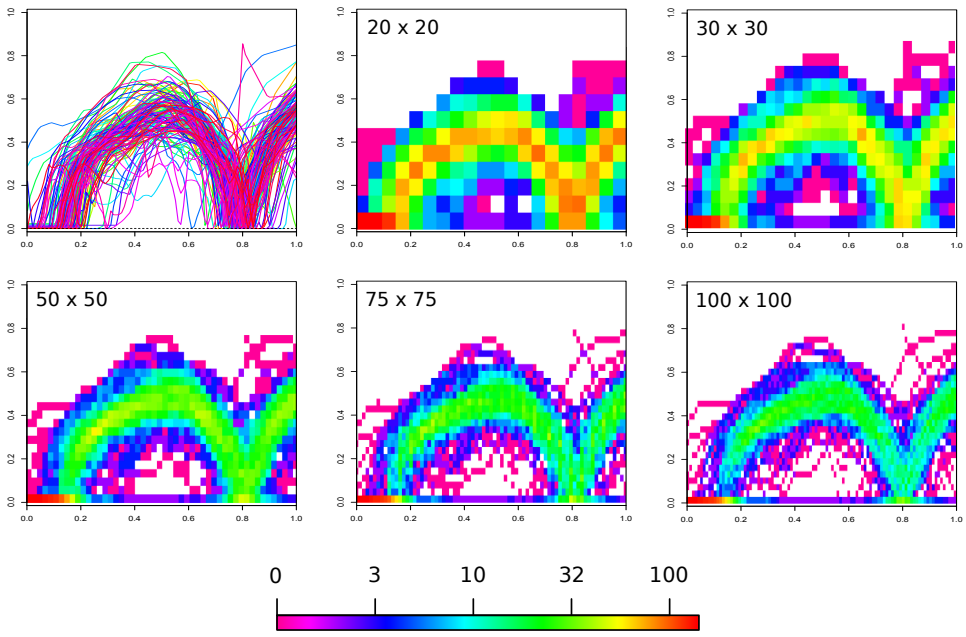


FIGURE 3.27 – Grilles d'index spatial sur une fenêtre glissante pour différentes tailles de cellules. L'échelle de couleurs représente le nombre de profils de vitesse contenus dans l'index de chaque cellule (échelle logarithmique).

3.3.4 Approche fonctionnelle

La particularité de notre tâche de classification réside dans le fait que les données à traiter sont de nature fonctionnelle. Bien qu'une fonction puisse toujours être décrite comme un vecteur de $\mathbb{R}^p$ contenant un ensemble de valeurs régulièrement échantillonnées sur l'espace de départ des fonctions, de nombreux travaux récents ont montrés qu'il ne s'agissait généralement pas là de la description la plus optimale pour l'apprentissage, et qu'une description fonctionnelle des données permet bien souvent d'obtenir de meilleurs résultats qu'une séquence de valeurs prises en un ensemble arbitraire de positions (Berlinet et al., 2008; Ferraty et Vieu, 2006).

Une description fonctionnelle des profils de vitesse vise à extraire une représentation parcimonieuse des données à l'aide d'un nombre réduit de paramètres expressifs qui caractérisent de manière optimale l'ensemble des fonctions à traiter dans la tâche de classification. Mathématiquement, ceci s'exprime par l'introduction d'une base d'un espace fonctionnel sur

laquelle les données sont projetées. Nous avons vu plusieurs exemples de ce type de représentation dans le paragraphe 3.2.2, la plus connue étant certainement la représentation de Fourier, spécifiquement adaptée aux signaux périodiques. Malheureusement, les fonctions de base de descripteurs de Fourier sont complètement délocalisées dans l'espace, ce qui les rend inappropriées pour un problème de détection où l'objectif *in fine* pourrait consister à localiser les éléments de la signalisation routière. À l'inverse les fonctions de la base des B-splines sont complètement localisées, ce qui les rend attractives d'un point de vue numérique par rapport aux autres bases de splines, mais rend difficile la réduction de dimension par troncature de la série de fonctions. Entre ces deux extrêmes, nous avons vu que les bases d'ondelettes offrent un compromis entre localisation spatiale et fréquentielle des signaux à traiter, faisant d'elles un outil de réprésentation puissant pour les problèmes de classification fonctionnelle (Berlinet et al., 2008). Dans ce travail, nous avons choisi d'utiliser la transformation en ondelettes de Haar pour sa simplicité et son efficacité numérique.

Dans le but de respecter la contrainte d'indépendance des vecteurs de descripteurs X au nombre N de courbes contenues dans la fenêtre (*cf* paragraphe 3.3.1), nous proposons ici de fusionner les courbes en un faible nombre de *profils agrégés*. Notons que contrairement à ce que leur nom suggère, ces profils agrégés ne sont pas rigoureusement parlant des profils de vitesse, même quand l'opération de fusion est une opération mathématique simple comme la médiane ou la moyenne *pointwise* (Andrieu et al., 2013a). Nous décrivons ci-dessous les 12 profils agrégés utilisés dans l'approche fonctionnelle :

Pour un nombre $N \in \mathbb{N}^*$ de profils, on définit le profil moyen f_1 par la moyenne point-à-point des vitesses instantanées :

$$f_1(x) = \frac{1}{N} \sum_{i=1}^{N} v_i(x), \tag{3.33}$$

où x appartient au domaine de définition des fonctions v_i, *i.e.* $[0, L]$. En considérant les profils de vitesse comme des réalisations *i.i.d.* d'un processus stochastique $V : [0, L] \times \Omega \to \mathbb{R}^+$, le profil agrégé f_1 est un estimateur empirique de la *moyenne d'ensemble* : $f_1(x) = \int_\Omega V(x, \omega) P(d\omega)$. Ce profil représente la tendance centrale de la collection de courbes. L'analyse de la vitesse moyenne n'est cependant pas suffisante pour distinguer un feu tricolore d'un stop ou encore d'un cassis. Nous définissons donc de la même manière un estimateur empirique de la *dispersion d'ensemble* :

$$f_2(x) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \big(v_i(x) - f_1(x)\big)^2}. \tag{3.34}$$

L'introduction de f_2 est motivée par le fait qu'un stop ou un cassis est caractérisé par un comportement très normatif de la part des conducteurs, induisant ainsi une dispersion moindre dans les profils au niveau de l'élément à détecter. À l'inverse, un feu tricolore sépare généralement grossièrement les profils en deux groupes. Ce second profil agrégé sera cependant moins informatif dans le cas où le phénomène d'onde verte [8] est relativement marqué, ce qui peut se produire en particulier sur les avenues principales. En conséquence,

8. Série de feux coordonnés de manière à faciliter l'écoulement continu des véhicules sur un axe.

nous ajoutons des moments d'ordre supérieur (en supposant f_2 strictement positif sur tout son domaine, *i.e.* qu'il n'existe pas de nœud sur lequel tous les profils s'intersectent) :

$$f_z(x) = \left[\frac{1}{N} \sum_{i=1}^{N} \left(\frac{v_i(x) - f_1(x)}{f_2(x)} \right)^z \right]^{1/z} \qquad z = 3, 4 \tag{3.35}$$

Les 2 moments définis par l'équation 3.35 représentent l'asymétrie (f_3) et l'applatissement (f_4) de la distribution des vitesses en un lieu x, et peuvent être combinées dans l'indice de Sarle (Ellison, 1987) de sorte à fournir un indice de bimodalité de la distribution :

$$f_5(x) = \frac{f_3(x)^6 + 1}{f_4(x)^4}. \tag{3.36}$$

Dans ce travail, nous utilisons l'estimateur empirique proposé par Pearson (1916), permettant d'évaluer $f_5(x)$ à partir d'un nombre fini de réalisations. La figure 3.28 fournit un exemple de trois distributions plus ou moins bimodales avec les indices de Sarle (notés β) estimés. On notera que $\beta \in [0, 1]$ et qu'il prend la valeur $5/9 \approx 0.555$ pour une distribution uniforme (Pfister et al., 2013). On peut également montrer aisément qu'il prend la même valeur pour une distribution exponentielle, et que $\beta = 1/3$ pour une loi normale. En pratique, on considère qu'une distribution telle que $\beta \geqslant 5/9$ est clairement bimodale. Le profil agrégé f_5 pourra aider à distinguer un feu tricolore des autres éléments de l'infrastructure routière.

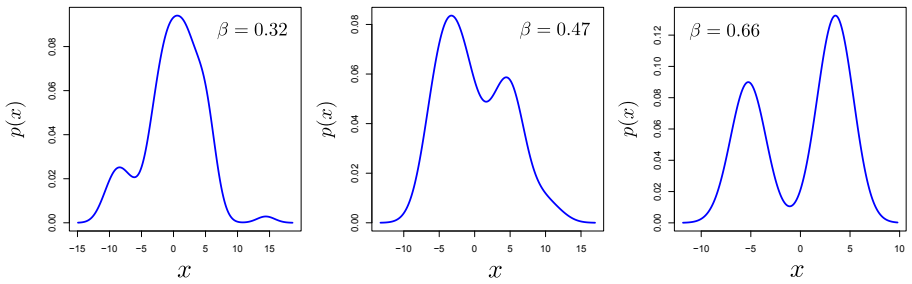


FIGURE 3.28 – Exemples de distributions de probabilités de moyennes nulles, de variances identiques ($\sigma^2 = 22.6$) : unimodale (à gauche), faiblement bimodale (au centre) et bimodale (à droite) avec les valeurs d'indices de bimodalités β associées.

Soit $p \in [0, 1]$ une valeur de probabilité arbitraire. On définit le quantile d'ordre p à la position x par l'unique solution $Q(x, p)$ de l'équation :

$$F_V(x, Q(x, p)) = \mathbb{P}(v(x) \leqslant Q(x, p)) = p, \tag{3.37}$$

où $F_V(x, .)$ est la fonction de répartition du processus stochastique en x, inconnue, et à laquelle on substitue une mesure de probabilité sur les réalisations, *i.e.* sur un tirage aléatoire de l'indice i dans $\{1, N\}$. On définit alors le profil médian par :

$$f_6(x) = Q(x, \frac{1}{2}), \tag{3.38}$$

puis les 15eme et 85eme centiles, que l'on peut soustraire l'un à l'autre pour obtenir un estimateur robuste de la dispersion, généralement moins bruité que f_2 :

$$f_7(x) = Q(x, 0.15), \qquad f_8(x) = Q(x, 0.85), \tag{3.39}$$

$$f_9(x) = f_8(x) - f_7(x). \tag{3.40}$$

Enfin, les valeurs extrémales point-à-point sont ajoutées à l'ensemble des profils agrégés, et également différenciées pour un troisième indicateur (plus sensibles aux valeurs extrêmes) de la dispersion :

$$f_{10}(x) = Q(x, 0), \qquad f_{11}(x) = Q(x, 1), \tag{3.41}$$

$$f_{12}(x) = f_{11}(x) - f_{10}(x). \tag{3.42}$$

Notons que si les profils outliers n'ont pas été préalablement retirés du jeu de données (*cf* paragraphe 3.1.2), f_{10} et f_{11} peuvent être simplement et efficacement estimés par $Q(., \varepsilon)$ et $Q(., 1-\varepsilon)$, respectivement, avec ε une faible valeur de probabilité comme 0.05 par exemple.

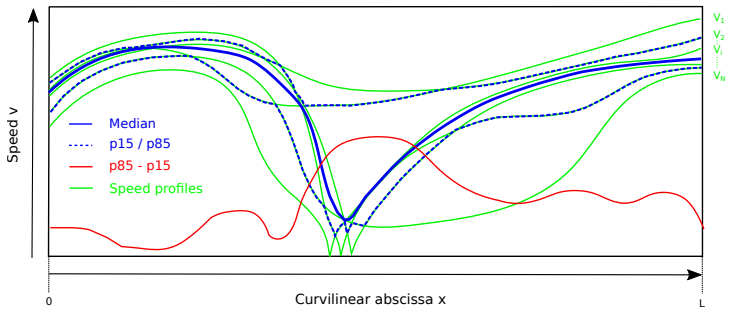


FIGURE 3.29 – Exemple des profils agrégés f_6 (médiane en bleu), f_7, f_8 (percentiles en pointillés) et f_9 (différence des percentiles en rouge). Les courbes vertes représentent les réalisations individuelles.

La figure 3.29 illustre un exemple de calcul de profils agrégés sur un ensemble fictif de 6 profils de vitesse.

Chacun de ces 12 profils contient 100 valeurs, résultant ainsi en un total de 1200 variables explicatives à la fin de cette étape. Cependant, ces variables sont extrêmement corrélées, d'un part du fait de l'inertie du véhicule qui limite considérablement les variations brutales dans les profils individuels, mais également du fait de l'opération d'agrégation qui réduit

les comportements de type *outlier*. Le système d'ondelettes de Haar est alors utilisé pour réduire cette redondance dans les données, tout en conservant l'essentiel de l'information nécessaire à la classification.

Nous avons vu dans le paragraphe 3.2.2.3 que chaque niveau de résolution j d'un système d'ondelettes est un sous-espace vectoriel de dimension 2^j. Pour un entier K arbitraire, la réunion des l premiers niveaux de résolution est donc décrit par $m = 2^{l+1}$ fonctions de base. Considérons un nombre m qui s'exprime comme une puissance de 2. On peut alors construire la matrice $\mathbf{H}_m$, contenant les m premières fonctions de base du système d'ondelettes, chacune d'elles étant discrétisée en m points. Pour le cas du système de Haar, la suite des matrice $(\mathbf{H}_m)_{m\geqslant 2}$ se construit aisément par récurrence de la manière suivante :

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \mathbf{H}_{2m} = \begin{bmatrix} \mathbf{H}_m \otimes [1,1] \\ \mathbf{I}_m \otimes [1,-1] \end{bmatrix}, \tag{3.43}$$

où $\mathbf{I}_m$ désigne la matrice identité de $\mathbb{R}^{m\times m}$ et $\mathbf{A}\otimes\mathbf{B}$ est la produit de Kronecker (*i.e.* la matrice bloc composée des matrices produits des éléments a_{ij} par $\mathbf{B}$). À titre d'exemple, la matrice (normalisée) des deux premiers niveaux de résolution s'écrit :

$$\widetilde{\mathbf{H}}_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \tag{3.44}$$

La figure 3.30 donne une illustration des 8 vecteurs de base $(\mathbf{H}_8)$. Dans notre cadre de travail, nous avons choisi $l = 10$ niveaux, ce qui représente un échantillonnage des profils en 1024 points, assurant ainsi une précision numérique satisfaisante dans la transformation.

Le calcul des coefficients s'effectue alors immédiatement par projection orthogonale sur les vecteurs de base :

$$X_{ij} = \mathbf{H}\cdot f_j = \sum_{k=1}^{m} \mathbf{H}_{jk} f_j(x_k), \tag{3.45}$$

où $i \in \{1, 2, ...p\}$ représente l'indice de la fonction de base, j est l'indice du profil agrégé f_j et $x_k = kL/m$ est la k-ième abscisse échantillonnée dans la fenêtre glissante. On pourra se référer à Stanković et Falkowski (2003) pour plus de détails sur la transformation de Haar discrète. On vérifie aisément dans le cas général que $\mathbf{H}_m$ est orthogonale, d'où $\mathbf{H}_m^{-1} = \mathbf{H}_m^T$, et l'équation 3.45 peut être immédiatement inversée pour reconstruire (avec perte, *i.e.* jusqu'à un certain niveau de résolution) un profil à partir des descripteurs : $f_j(x_k) = \mathbf{H}_k^T \cdot X_{kj}$. En effet, $\mathbf{H}_k^T$ correspond bien au vecteur des m premières fonctions de bases prises en x_k, et on retrouve une version discrétisée de la recomposition 3.19.

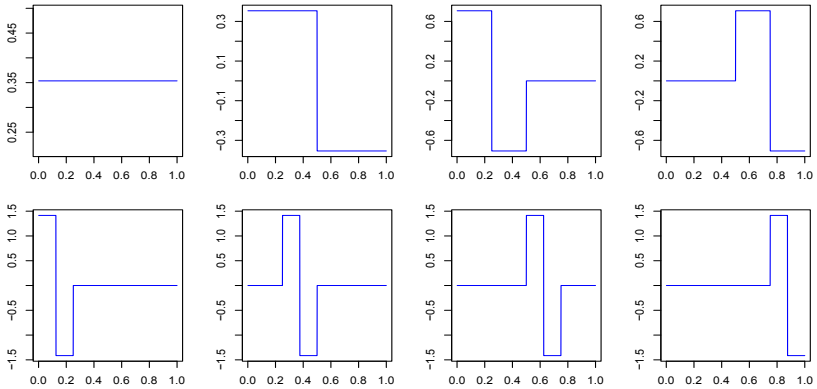


FIGURE 3.30 – Les 8 fonctions de base des 2 premiers niveaux du système d'ondelettes de Haar. De gauche à droite et de haut en bas : $\phi, \psi_{00}, \psi_{10}, \psi_{11}, \psi_{20}, \psi_{21}, \psi_{22}$ et ψ_{23}.

Dans notre cadre d'étude, nous décidons de rejeter tous les coefficients au dessus du niveau $l = 3$, ce qui correspond à une résolution spatiale de l'ordre de 6 m et à un total de 16 coefficients. D'autre part, pour chaque profil agrégé f_j, nous supprimons le coefficient de la première fonction de base, correspondant à la moyenne des signaux. En effet, nous pensons que cette suppression va contraindre les algorithmes d'apprentissage à chercher des critères de discrimination dans les niveaux les plus fins, produisant ainsi potentiellement de meilleurs résultats, ce qui a été vérifié expérimentalement a posteriori. À l'issue de cette étape, nous disposons donc de 15 coefficients pour décrire chacun des 12 profils agrégés, soit un vecteur de descripteurs contenant $p = 12 \times 15 = 180$ variables explicatives :

$$X = \left\{ \omega_{lk}^{z} \;\middle|\; z \in [\![1, 12]\!], \quad l \in [\![0, 3]\!], \quad k \in [\![0, 2^l - 1]\!] \right\}, \tag{3.46}$$

avec :

$$\omega_{lk}^{z} = \int_0^1 f_z(x)\psi_{lk}(x)dx. \tag{3.47}$$

Par exemple, pour une instance de fenêtre glissante donnée, ω_{23}^{5} modélise le comportement du profil de bi-modalité ($z = 5$) au niveau de résolution 12.5 m ($l = 2$) sur la droite de la fenêtre glissante ($k = 3$), *i.e.* pour $x \in [75, 100]$. Le pipeline global de la méthode est représenté sur la figure 3.31.

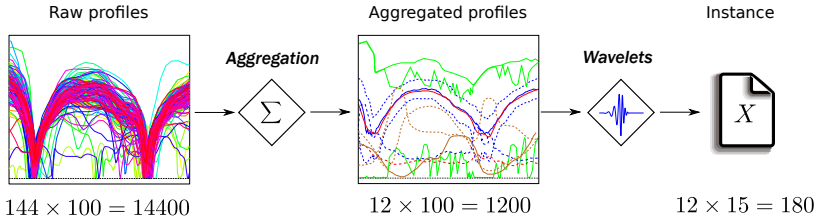


FIGURE 3.31 – Pipeline global de l'approche fonctionnelle.

Pour notre implémentation, le package R *Wavetresh* a été utilisée pour calculer la matrice de Haar (Nason et Maechler, 2006; R Development Core Team, 2008).

3.3.5 Protocole expérimental

Pour chacune de ces 3 approches, nous appliquons les 5 algorithmes d'apprentissage exposés dans la section 3.2.1, ce qui fait au total 15 combinaisons à tester. Les performances de prédiction sont évaluées à l'aide d'une validation croisée, en séparant le jeu de données en 10 fractions (*10-fold cross validation*). L'entraînement est réalisé sur 90 % des instances, et les 10 % restants sont conservés pour valider les modèles. Ce processus est répété 10 fois au total, jusqu'à ce que toutes les instances aient été classifiées en *positive* (contenant un feu tricolore) ou *négative*. Ce procédé permet virtuellement d'utiliser la quasi-totalité (90 %) des données pour l'entraînement, et de valider le modèle sur toutes les données, contrairement au processus de validation classique en 2 folds, qui ne permet de construire chaque modèle qu'avec seulement 50 % des instances. Dans le cas extrême, lorsque l'entraînement est réalisé sur une fraction $r = (N-1)/N$ des données, on parle de *N-fold cross validation*, ou encore de *leave-one-out*, et chaque instance est ré-étiquetée à l'aide d'un modèle construit sur la base de toutes les autres données (dans une approche similaire à celle présentée dans la section 2.3.2.4). Il s'agit d'un procédé idéal, mais coûteux en pratique puisque l'entraînement doit être réalisé autant de fois qu'il y a de données dans le jeu d'entraînement.

À chaque itération du processus de validation croisée, la séparation est effectuée géométriquement, *i.e.* en affectant les 2255 (90 % du nombre total de fenêtres) premières [9] fenêtres au jeu d'entraînement, et les 250 fenêtres suivantes au jeu de validation. Les fenêtres glissantes ayant été construites avec un recouvrement supérieur à 50 %, notons qu'il est impossible de diviser le jeu de données en 2 parties sans que les jeux d'entraînement et de validation contiennent des parties de courbes communes. À chaque séparation, nous retirons donc a posteriori toutes les instances dont le support couvre les 2 fractions. Enfin, l'ensemble du processus décrit ci-dessus est répété 10 fois au total, en tirant aléatoirement le point de départ du circuit, de sorte à se prémunir de la sensibilité du résultat au choix de la position de séparation.

Puisque les étiquettes des données sont significativement déséquilibrées (16 % d'instances positives seulement), certains algorithmes sont susceptibles de sous-performer. Pour contourner ce problème, à chaque construction de modèle, le jeu de données d'entraînement est équilibré par sur-échantillonnage (*i.e.* en répliquant aléatoirement les exemples de la classe minoritaire jusqu'à atteindre l'équilibre) Le jeu de validation peut quant à lui rester déséquilibré puisqu'il n'y a pas de raison de penser qu'une certaine proportion soit plus réaliste qu'une autre. Pour plus de détails sur la marche à adopter face à un jeu de données déséquilibré pour les problèmes de classification binaire, nous renvoyons le lecteur à l'étude exhaustive de Menardi et Torelli (2014). Les prédictions de chaque sous-ensemble de validation sont alors fusionnées dans une unique table de contingence, à partir de laquelle on dérive les scores de performances des differentes approches testées.

Notons que l'ensemble complet des conducteurs est inclus à la fois dans les bases d'entraînement et de validation. Une approche plus précautionneuse aurait été d'effectuer les

9. Dans l'ordre du parcours de la boucle de circuit

séparations à la fois suivant l'axe spatial et l'axe des conducteurs, de façon à ce que les performances de prédiction soient évaluées à partir de données absolument nouvelles. Cependant, il est fréquemment admis que les profils de vitesse sont majoritairement déterminés par l'infrastructure routière, impliquant ainsi que la duplication des comportements individuels des différents conducteurs entre les deux bases n'apporte vraisemblablement aucune aide significative dans le processus de détection. Nous validerons cette hypothèse plus rigoureusement ultérieurement.

L'ensemble de l'expérimentation a été implémentée en R, avec les librairies d'apprentissage statistique les plus communément utilisées (Dimitriadou et al., 2009; Schliep et al., 2007; Therneau et al., 1997; Kursa, 2012; Liaw et Wiener, 2002).

3.4 Résultats

L'expérimentation a été lancée sur une machine équipée d'un processeur Intel Core (TM) i7-3770 à 3.40 GHz et 8 Go de RAM, pour un temps total de calcul de l'ordre de 7 heures. Pour chacun de 15 classifieurs, les indicateurs dérits dans la section suivante ont été calculés et les résultats sont exposés dans la section 3.4.2.

3.4.1 Indicateurs de performance

- La *sensibilité* (parfois appelé rappel), notée **STV** ci-après, définie par la probabilité de détecter une instance positive :

$$\mathbb{P}(\hat{Y} = 1|Y = 1) = \frac{TP}{TP + FN}, \tag{3.48}$$

  où TP et FN représentent repectivement les nombre de vrais positifs et faux négatifs résultant du processus de classification. Cet indicateur mesure l'exhaustivité de la méthode de détection.

- La *spécificité*, notée **SPC**, mesure à l'inverse la probabilité du classifieur de ne pas détecter une instance négative :

$$\mathbb{P}(\hat{Y} = 0|Y = 0) = \frac{TN}{TN + FP}, \tag{3.49}$$

  où TN et FP représentent repectivement les nombre de vrais négatifs et faux positifs.

- La *précision*, notée **PPV** (pour *positive predictive value*), quantifie le nombre de prédictions justes parmi les objects détectés positifs. On parle de mesure de précision *utilisateur* (a posteriori), par opposition aux deux indicateurs ci-dessus, qui sont qualifiés de précision *producteur* (a priori) :

$$\mathbb{P}(Y = 1|\hat{Y} = 1) = \frac{TP}{TP + FP}. \tag{3.50}$$

En particulier, à l'aide de la loi de Bayes, on peut réécrire la précision sous la forme :

$$PPV = \frac{\mathbb{P}(\hat{Y} = 1|Y = 1)\mathbb{P}(Y = 1)}{\mathbb{P}(\hat{Y} = 1)} \propto STV \times \frac{P}{TP + FP}, \tag{3.51}$$

mettant ainsi en évidence le fait que l'indicateur PPV dépend des proportions des classes, et ne fournit donc pas nécessairement une image représentative des performances du classifieur.

- La mesure F_1, notée **F1M**, est la moyenne harmonique des mesures de rappel et de précision :

$$\left(\frac{STV^{-1} + PPV^{-1}}{2}\right)^{-1} = \frac{2TP}{2TP + FP + FN}. \tag{3.52}$$

Le choix de la moyenne harmonique, plutôt que de la moyenne arithmétique, se justifie par une motivation naturelle à pénaliser plus fortement les classifieurs déséquilibrés. On pourra trouver des justifications plus théoriques et générales dans les travaux de Van Rijsbergen (1974), qui mettent en évidence le fait que la famille des mesures F_β (pour un paramètre de pondération relatif entre rappel et précision) est l'unique solution d'un ensemble de contraintes naturelles dans la recherche d'une métrique scalaire de l'évaluation des performances d'un classifieur binaire. La mesure F_1 n'a toutefois pas d'interprétation intrinsèque en termes de valeur de probabilité, contraitrement à tous les autres indicateurs présentés dans cette section.

- La mesure d'*Accuracy*, notée ACC, désigne la probabilité pour le classifieur de retourner une prédiction correcte :

$$\mathbb{P}[\hat{Y} = Y] = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.53}$$

En remarquant qu'on peut réécrire le terme de gauche de 3.53 sous la forme :

$$\begin{aligned} \mathbb{P}[\hat{Y} = Y] &= \mathbb{P}[\hat{Y} = 1|Y = 1]\mathbb{P}[Y = 1] + \mathbb{P}[\hat{Y} = 0|Y = 0]\mathbb{P}[Y = 0] \\ &= STV \times \mathbb{P}[Y = 1] + SPC \times \mathbb{P}[Y = 0], \end{aligned} \tag{3.54}$$

on observe qu'à sensibilité et spécificité fixées, cet indicateur évolue avec les proportions d'instances de chaque classe dans la base de validation. De ce fait, la mesure d'accuracy n'est pas stable.

- Les mesures globales F_1 et accuracy, posent le problème d'être dépendantes de la proportion des étiquettes dans le jeu de données de validation. L'aire sous la courbe

ROC, ou **AUC** (pour *area under curve*) permet de combler cette lacune.

Considérons un classifieur qui à chaque donnée $\mathbf{x} \in \mathcal{X}$ associe un score $f(\mathbf{x})$, d'autant plus élevé (resp. faible) que la donnée porte vraisemblablement l'étiquette 1 (resp. 0). À partir de cet unique classifieur, on peut construire une famille infinie de classifieurs dérivés $(f_t)_{t\in\mathbb{R}}$ en faisant varier le seuil de décision t :

$$\hat{Y} = f_t(\mathbf{x}) = \begin{cases} 1 & \text{si } f(\mathbf{x}) \geqslant t \\ 0 & \text{sinon.} \end{cases} \tag{3.55}$$

On appelle courbe ROC (pour *Receiver Operating Caracteristics*) le lieu des points $(x, y) = (1 - SPC(t), STV(t)) \in [0, 1]^2$ pour l'ensemble des seuils $t \in \mathbb{R}$. En considérant t comme une variable muette, l'aire AUC désigne alors l'intégrale sur $[0, 1]$ de la fonction qui à la variable $x = 1 - SPC$ associe $y = STV$.

On peut montrer que l'AUC est égale à $\mathbb{P}[f(X^{(1)}) \leqslant f(X^{(2)})]$, où $X^{(1)}$ et $X^{(2)}$ sont 2 instances échantillonnées aléatoirement et telles que $Y^{(1)} < Y^{(2)}$ (Hanley et McNeil, 1982). Notons qu'une AUC égale à 50 %, correspond à un classifieur purement aléatoire. Pour plus d'informations sur l'estimation et l'interprétation des courbes ROC, voir Gonçalves et al. (2014).

À cet ensemble classique d'indicateurs, on ajoute 3 indices de performances propres à notre cas d'étude :

- Le taux de confusion avec les signes stop, ou **STP**, désigne la probabilité de classifier un stop en feu tricolore (évalué à partir d'une population de base de 50 fenêtres contenant un stop).

- Le temps de calcul *offline*, ou **OFT**, correspondant au temps machine nécessaire à l'extraction des descripteurs et au processus d'entraînement (on le mesure en secondes par fenêtre).

- Le temps de calcul *online*, ou **ONT**, correspondant au temps machine nécessaire à l'extraction des descripteurs et au processus de validation (également en secondes par fenêtre).

3.4.2 Résultats

Dans les tables de résultats, nous utilisons les sigles suivants pour désigner les algorithmes d'apprentissage : NB (bayésien naïf), CART (arbre de décision), kNN (k plus proches voisins), SNB (Random Ferns, ou Semi-Naive Bayes) et RF (forêts aléatoires).

L'ensemble des performances individuelles des approches sont compilées dans la table 3.1, dans laquelle les valeurs numériques sont exprimées en pourcentage, exceptées les deux dernières colonnes qui le sont en seconde(s) par fenêtre glissante. Les intervalles d'incertitude sont évalués avec un niveau de confiance fixé à 95 %, et sont calculés numériquement sur

Methods	Performance								
	STV	SPC	PPV	F1M	ACC	AUC	STP	OFT	ONT
Direct approach									
NB	88.30 ± 0.10	69.13 ± 1.78	36.26 ± 0.24	51.41 ± 0.25	72.31 ± 0.14	78.31 ± 0.35	90.12 ± 0.22	0.69	0.30
CART	86.56 ± 0.41	87.68 ± 0.20	58.29 ± 0.49	69.66 ± 0.45	87.50 ± 0.22	88.14 ± 1.04	46.53 ± 2.14	0.72	0.30
kNN	62.18 ± 0.22	93.07 ± 0.10	64.10 ± 0.45	63.13 ± 0.27	87.95 ± 0.53	82.54 ± 0.36	37.79 ± 0.45	1.24	0.30
SNB	95.52 ± 0.14	64.19 ± 0.34	34.65 ± 0.23	50.86 ± 0.25	69.38 ± 0.27	88.07 ± 0.59	97.85 ± 2.27	0.40	0.30
RF	72.88 ± 0.27	95.99 ± 0.08	78.34 ± 0.23	75.51 ± 0.16	92.16 ± 0.06	94.69 ± 0.99	4.70 ± 0.53	2.54	0.30
Image approach									
NB	87.56 ± 0.23	67.95 ± 2.90	35.20 ± 1.74	50.21 ± 1.94	71.20 ± 2.47	76.68 ± 1.48	92.05 ± 0.53	1.87	1.49
CART	82.33 ± 1.29	87.09 ± 0.61	55.91 ± 1.29	66.59 ± 1.01	86.30 ± 0.49	85.92 ± 1.26	56.00 ± 4.74	1.87	1.49
kNN	68.40 ± 0.73	91.24 ± 0.20	60.84 ± 1.00	64.40 ± 0.69	87.45 ± 0.18	83.05 ± 0.27	54.12 ± 2.47	3.10	1.49
SNB	94.27 ± 0.71	62.11 ± 3.16	33.10 ± 1.71	48.99 ± 2.04	67.45 ± 2.74	86.59 ± 0.63	99.02 ± 1.67	1.50	1.49
RF	64.17 ± 0.45	96.34 ± 0.18	77.71 ± 0.99	70.29 ± 0.55	91.00 ± 0.22	94.14 ± 0.23	6.13 ± 1.86	5.16	1.49
Functional approach									
NB	88.30 ± 0.22	90.60 ± 0.06	65.13 ± 0.22	74.97 ± 0.18	90.22 ± 0.06	91.88 ± 0.21	67.95 ± 0.47	0.31	0.31
CART	88.30 ± 0.22	91.54 ± 0.31	67.49 ± 0.69	76.50 ± 0.53	91.00 ± 0.27	88.61 ± 0.58	57.52 ± 1.08	0.31	0.31
kNN	79.60 ± 0.22	94.41 ± 0.04	73.90 ± 0.27	76.64 ± 0.20	91.95 ± 0.06	91.5 ± 0.32	48.21 ± 0.53	0.31	0.31
SNB	97.51 ± 0.10	73.88 ± 0.41	42.60 ± 0.45	59.30 ± 0.45	77.80 ± 0.35	95.59 ± 0.25	82.23 ± 1.33	0.31	0.31
RF	82.58 ± 0.23	97.23 ± 0.06	85.56 ± 0.69	84.05 ± 0.27	94.80 ± 0.08	97.28 ± 0.22	2.81 ± 0.80	0.32	0.31

TABLE 3.1 – Performances de prédiction pour les 5 algorithmes de classifieurs avec les 3 approches différentes. Les intervalles de confiance sont calculés à 95%

les 10 réplications du découpage entraînement/validation (Macskassy et Provost, 2004). La table 3.2 indique les performances de chaque mode de calcul des descripteurs, moyennées sur les 5 algorithmes d'apprentissage, tandis que la table 3.3 résume les performances obtenues avec l'algorithme des forêts aléatoires (le plus performant dans notre cas d'étude) sur chacune des trois approches.

La table 3.4 contient les p-valeurs du test de Student pour les données appariées sur les

AUC. Notons que le test de Student est conçu pour le cas de variables distribuées suivant une loi normale. Bien que l'on puisse considérer que les indices AUC sont normalement distribués en vertu du théorème central limite (Dupré, 2013), pour confirmer les résultats du test de Student, nous avons également appliqué le test des rangs signés de Wilcoxon, test non-paramétrique ne recquiérant ainsi pas l'hypothèse de normalité des données à comparer (table 3.5). Le ratio de cohérence globale entre ces deux ensembles de p-valeurs (calculé pour tous les couples de classifieurs) s'élève à 97 % (selon l'indice de Rand), permettant ainsi de supposer réaliste l'hypothèse de normalité, bien que les résultats du test de Wilcoxon puissent parfois être plus robustes dans notre cas, en particulier sur les couples de séries contenant des *outliers*. Étant donnée la concordance des tables 3.4 et 3.5, nous nous référerons ci-après uniquement aux résultats des tests de Student.

Les courbes ROC pour chaque méthode sont illustrées en figure 3.32 : pour chaque classifieur, les 10 courbes obtenues lors de la validation sont moyennées par *threshold averaging*, et les bandes de confiance sont calculées à 95 % avec la méthode *fixed-width band* qui consiste à définir les bandes supérieures et inférieures par une translation de la courbe ROC centrale, suivant une ligne de pente $-\sqrt{m/n}$, où m et n désignent respectivement les nombres d'instances positives et négatives dans le jeu de validation. La distance de translation est évalué empiriquement à partir de la population des 10 courbes, de sorte à ce que le niveau de confiance de la bande à calculer corresponde au ratio de courbes observées comprises entre les bandes. La valeur de la pente se comprend intuitivement : l'écart-type de la moyenne empirique de n valeurs est inversement proportionnel à $\sqrt{n}$. Dans le cadre de l'estimation d'une courbe ROC, chaque classe d'instances disperse la courbe ROC suivant l'un des deux axes du repère. La dispersion totale est donc dirigée suivant une direction de pente égale à la racine carrée des rapports des effectifs de classe. Dans notre cadre d'application en particulier, la classe des instances positives est minoritaire, impliquant ainsi des bandes de confiance plus *étalées* dans la direction verticale. Cette méthode est particulièrement intéressante puisqu'elle ne dépend que de la proportion des classes, et non des effectifs exacts de chaque classe. Dans notre cas d'étude, la superposition des fenêtres glissantes rend difficile l'évaluation précise du nombre d'instances effectivement utilisées par le processus de validation. En effet, ces instances sont significativement corrélées entre elles, impliquant que ce nombre se situe quelque part entre le nombre total de fenêtres générées, et un dixième [10] de ce nombre. Cependant, l'estimation du ratio d'instances positives (16 %) est aisée, et rend la méthode fixed-width band particulièrement bien adaptée ici. Notons qu'il existe de nombreuses autres méthodes pour moyenner une collection de courbes ROC et estimer une bande de confiance (Macskassy et Provost, 2004) avec leurs propres avantages et limites. Contrairement à la plupart de ces méthodes alternatives, la méthode fixed-width band est globale (ou uniforme), *i.e.* à un niveau de confiance de 95 % par exemple, dans 19 cas sur 20, la courbe ROC vraie (et inconnue) est *intégralement* inclue dans la bande. Il en résulte généralement que les bandes FWB sont plus larges qu'avec les méthodes locales.

Les résultats de la table 3.1 et de la figure 3.32 mettent en évidence la supériorité de l'approche fonctionnelle par rapport à ses homologues image et directe, sur 4 des 5 algorithmes appliqués. Il y a une exception notable sur le cas des arbres de décision, pour lesquels l'approche directe affiche une AUC de $88.14\% \pm 1.04$, dont la différence avec l'approche fonctionnelle ($88.61\% \pm 0.58$) semble non-significative. Cependant, ceci peut difficilement être interprété comme une preuve concluante de l'inadéquation de l'algorithme CART au

10. Le recouvrement entre deux fenêtres consécutives est de 90 %.

problème présent, puisque les courbes ROC représentées sur la figure ne semblent pas très robustes, avec une bande de confiance assez lâche. Cette forte variance intrinsèque peut partiellement s'expliquer par le fait que l'algorithme CART est souvent reconnu comme peu stable vis-à-vis du bruit sur les observations (comme mentionné dans l'introduction de la section 3.2.1.5), avec une tendance des arbres de décisions individuels à fournir des modèles sensiblement différents en réponse à une faible modification des donnés.

À l'inverse, sur les 4 algorithmes restants, l'approche fonctionnelle domine clairement les autres approches, avec une différence particulièrement significative sur les algorithmes kNN, NB et SNB. On peut penser qu'un algorithme sophistiqué comme les forêts aléatoires est capable de produire des résultats décents, y compris sur un ensemble de descripteurs peu informatifs, ce qui pourrait expliquer l'écart plus modeste entre les RF dans l'espace des courbes ROC. Cependant, la courbe ROC de l'approche RF en mode fonctionnelle affiche tout de même une aire sous la courbe de 97.28 %, *i.e.* plus de 2 points au dessus de l'approche directe et 3 points au dessus de l'approche image. Ces différences d'AUC sont illustrées sur la figure 3.33, qui montre que si les approches image et directes semblent plus ou moins équivalentes, l'approche fonctionnelle domine significativement sur une large gamme de seuils de décision.

	Performance						
Approach	STV	SPC	PPV	F1M	ACC	AUC	STP
Direct	81.09 ± 13.37	82.01 ± 14.43	54.53 ± 18.78	62.11 ± 10.94	81.86 ± 10.27	87.36 ± 6.66	55.60 ± 38.22
Image	79.35 ± 12.74	80.95 ± 15.04	52.55 ± 18.66	60.10 ± 9.82	80.68 ± 10.59	85.66 ± 5.86	61.60 ± 37.34
Functional	87.26 ± 6.85	89.53 ± 9.12	66.94 ± 15.74	74.38 ± 9.22	89.15 ± 6.59	93.11 ± 3.35	51.20 ± 31.26

TABLE 3.2 – Performances moyennées des algorithmes pour chaque approche.

	Approaches		
Index	Direct	Image	Functional
Sensitivity	72.8	64.2	82.6
Specificity	95.9	96.3	97.2
Precision	78.3	77.7	85.6
F-measure	75.5	70.3	84.1
Accuracy	92.2	91.0	94.8
Roc area	94.7	94.1	97.3
Stops confusion	4.70	6.13	2.81
Offline time	2.50	5.16	0.32
Online time	0.30	1.49	0.31

TABLE 3.3 – Indices de performances pour les 3 approches avec les forêts aléatoires.

En particulier, l'espace entre l'approche fonctionnelle et les deux autres approches est particulièrement large sur la partie opérationnelle de l'espace ROC, *i.e.* au niveau des points

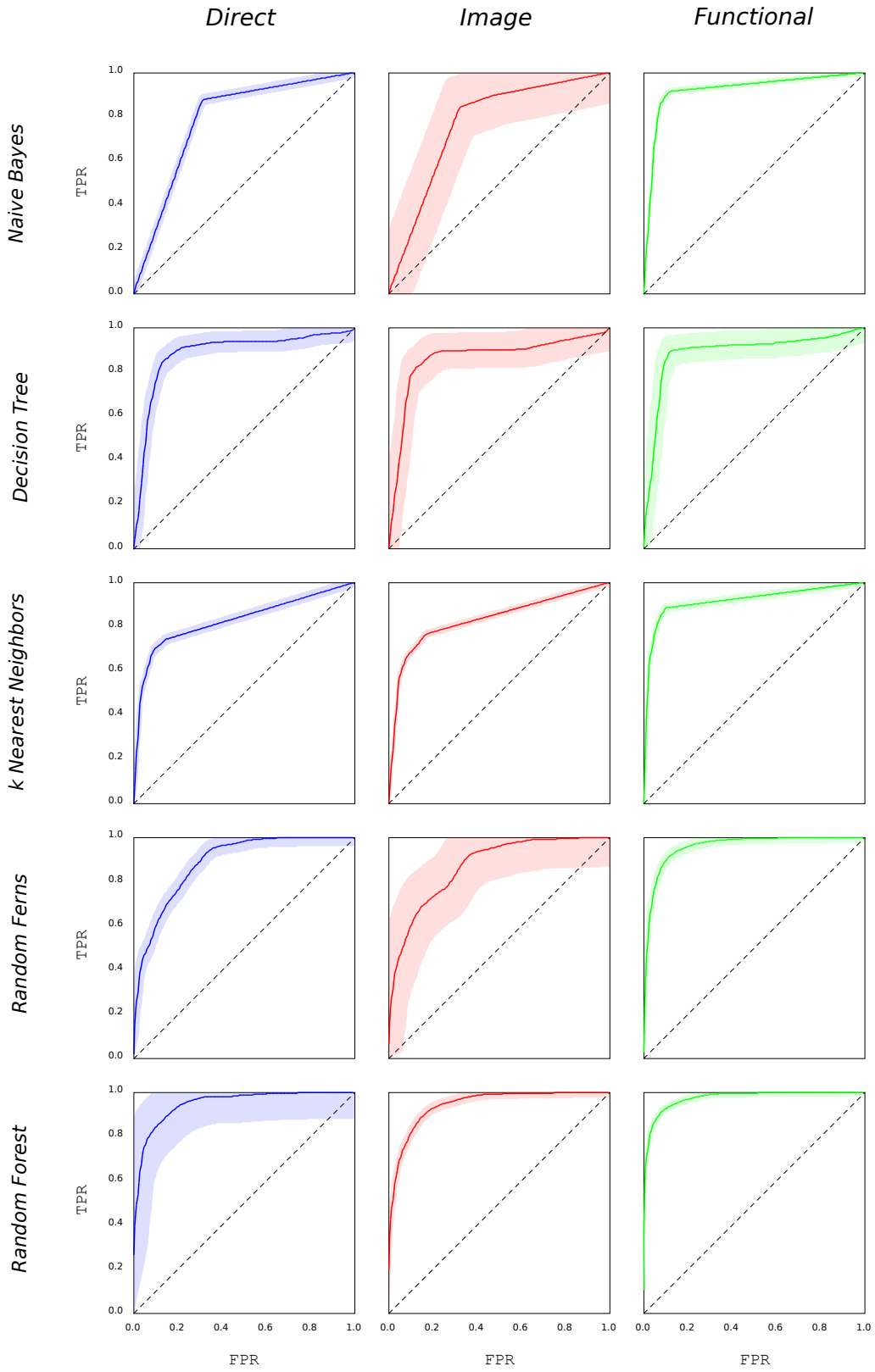


FIGURE 3.32 – Courbe ROC pour chaque croisement approche de calcul des descripteurs × algorithme d'apprentissage, avec bandes de confiance à 95 %.

		Approaches (I-III) × Algorithms (1-5)														
		Direct (I)					**Image** (II)					**Functional** (III)				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
I	1		-	-	-	-	**8.91**	-	-	-	-	-	-	-	-	-
	2	+		+	91.9	-	+	2.11	+	3.07	-	-	52.9	-	-	-
	3	+	-		-	-	+	-	2.71	-	-	-	-	-	-	-
	4	+	91.9	+		-	+	+	+	**1.85**	-	-	23.5	-	-	-
	5	+	+	+	+		+	+	+	+	33.5	+	+	+	16.9	-
II	1	**8.91**	-	-	-	-		-	-	-	-	-	-	-	-	-
	2	+	2.11	+	-	-	+		+	37.9	-	-	-	-	-	-
	3	+	-	2.71	-	-	+	-		-	-	-	-	-	-	-
	4	+	3.07	+	**1.85**	-	+	37.9	+		-	-	-	-	-	-
	5	+	+	+	+	33.5	+	+	+	+		+	+	+	-	-
III	1	+	+	+	+	-	+	+	+	+	-		+	2.02	-	-
	2	+	52.9	+	23.5	-	+	+	+	+	-	-		-	-	-
	3	+	+	+	+	-	+	+	+	+	-	2.02	+		-	-
	4	+	+	+	+	16.9	+	+	+	+	+	+	+	+		-
	5	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
1 = Naive Bayes 2 = Decision Tree 3 = k-Nearest Neighbors 4 = Random Ferns 5 = Random Forest																

TABLE 3.4 – Table des p-valeurs du test de Student pour chaque couple de classifieurs. Le symbole + (resp. -) indique une p-valeur inférieure à 1 %, et que le classifieur de la ligne est plus (resp. moins) performant (en termes d'AUC) que celui de la colonne.

		Approaches (I-III) × Algorithms (1-5)														
		Direct (I)					**Image** (II)					**Functional** (III)				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
I	1		-	-	-	-	+	-	-	-	-	-	-	-	-	-
	2	+		+	87.8	-	+	2.73	+	2.73	-	-	43.2	-	-	-
	3	+	-		-	-	+	-	2.73	-	-	-	-	-	-	-
	4	+	87.8	+		-	+	**2.17**	+	+	-	-	16.0	-	-	-
	5	+	+	+	+		+	+	+	+	84.0	+	+	+	6.45	-
II	1	-	-	-	-	-		-	-	-	-	-	-	-	-	-
	2	+	2.73	+	**2.17**	-	+		+	37.5	-	-	-	-	-	-
	3	+	-	2.73	-	-	+	-		-	-	-	-	-	-	-
	4	+	2.73	+	-	-	+	37.5	+		-	-	-	-	-	-
	5	+	+	+	+	84.0	+	+	+	+		+	+	+	-	-
III	1	+	+	+	+	-	+	+	+	+	-		+	3.66	-	-
	2	+	43.2	+	16.0	-	+	+	+	+	-	-		-	-	-
	3	+	+	+	+	-	+	+	+	+	-	3.66	+		-	-
	4	+	+	+	+	6.45	+	+	+	+	+	+	+	+		-
	5	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
1 = Naive Bayes 2 = Decision Tree 3 = k-Nearest Neighbors 4 = Random Ferns 5 = Random Forest																

TABLE 3.5 – Table des p-valeurs du test de rangs signés de Wilcoxon, pour chaque couple de classifieurs. Le symbole + (resp. -) indique une p-valeur inférieure à 1 %, et que le classifieur de la ligne est plus (resp. moins) performant (en termes d'AUC) que celui de la colonne.

où les taux FPR et TPR ont des valeurs acceptables pour une application à un problème concret. Plus spécifiquement, dans notre cadre applicatif, nous souhaitons minimiser le risque de mettre à jour la base de données de l'infrastructure routière avec une information erronée. De fait, nous pensons que garder un faible taux de faux positifs est plus important que l'exhaustivité de la détection. En conséquence, le point de fonctionnement optimal

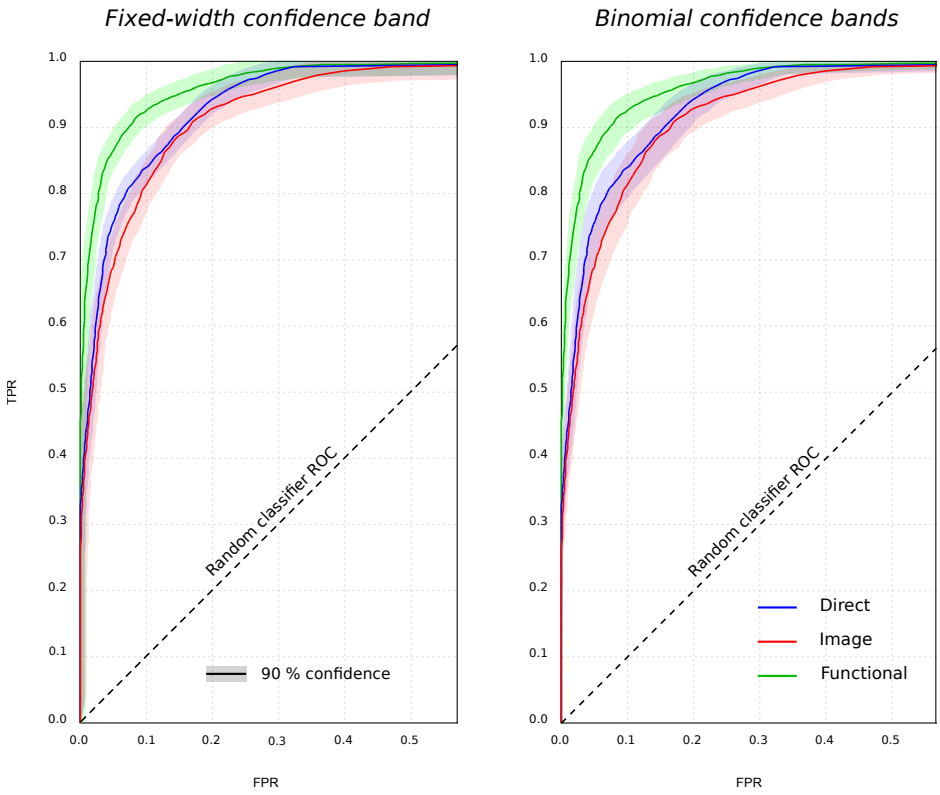


FIGURE 3.33 – Comparison of Random Forest ROC curves at 90 % confidence level

d'un classifieur peut se calculer par l'intersection de sa courbe ROC avec une droite passant par le point $(0, 1)$ et de pente inférieure [11] à -1. C'est aussi dans cette partie de l'espace ROC que la courbe verte est la plus avantagée par rapport aux autres courbes. Ceci vient du fait que nos données sont déséquilibrées en faveurs des instances négatives, impliquant ainsi une bande plus dispersée suivant la direction y, comme indiqué précédemment. Nous mettrons en œuvre par la suite (section 3.4.3) une méthode rigoureuse pour déterminer le point de fonctionnement optimal des classifieurs.

Notons que la largeur des bandes de confiance dépend de la variance introduite par la sensibilité du modèle au choix du découpage géométrique de la boucle de circuit sur le processus de validation croisée. Comme mentionné précédemment, nous avons exclu la variance des modèles due à la taille de l'échantillon de validation en utilisant la méthode FWB. Cependant, pour les comparaisons sur l'algorithme RF, nous souhaitions évaluer cette contribution à l'incertitude, ce que nous avons fait en calculant un second jeu de bandes de confiance à 90% à l'aide de la méthode *threshold averaging* et sous l'hypothèse d'une distribution binomiale des données (intervalle de Wilson), en supposant les instances comme indépendantes. Ces bandes sont illustrées sur la partie droite de la figure 3.33, sur laquelle on peut oberver que l'approche fonctionnelle domine toujours significativement les autres approches.

11. À condition que la proportion des classes ne soit pas trop déséquilibrées en faveur des instances positives. Même en conditions urbaines denses, on peut considérer qu'il s'agit d'une hypothèse raisonnable en pratique.

Les différences entre les 15 combinaisons peuvent être analysées à l'aide des p-valeurs du test apparié de Student (table 3.4). Arbitrairement, nous considérons une différence comme significative lorsque la p-valeur correspondante est inférieure à 1%. On peut observer que clairement, l'approche fonctionnelle est meilleure que les autres approches, en particulier lorsqu'elle est combinée aux RF, et que la domination est souvent significative en dessous du seuil de 1%. Il est intéressant de noter qu'en dépit de sa simplicité, la combinaison RF - approche directe donne de bons résultats.

De manière plus anecdotique, l'approche fonctionnelle est beaucoup plus rapide que les approches directe et image (table 3.2), en particulier en ce qui concerne la phase d'entraînement. La différence en temps de calcul provient du fait qu'il y a significativement moins de descripteurs à traiter dans l'approche fonctionnelle. Cependant, sur les données en grande dimension, on peut voir qu'en ne travaillant que sur des sous-groupes de variables, l'algorithme des Random Ferns (SNB) est capable de construire un modèle en un temps comparativement réduit. De plus, la table 3.2 suggère qu'avec cette approche, le résultat produit est moins sensible au choix de l'algorithme de classification, sans que nous ne puissions expliquer pourquoi.

Sur toutes les approches, les forêts aléatoires sont plus précises (table 3.3), en particulier en termes de taux de confusion avec les stops (entre 3 et 7% des stops sont détectés à tort en feu tricolore). Ceci met en évidence le fait que les RF ne se contentent pas de *scanner* les profils de vitesse à faible vitesse pour détecter les arrêts, mais possèdent un pouvoir de discrimination plus évolué, permettant ainsi d'envisager d'utiliser les RF dans des travaux ultérieurs pour construire un modèle de classification multinomiale spécifiquement entraîné à détecter et classifier les différents types d'élément de la signalisation routière. En contrepartie, dans notre expérimentation, les RF ont été plus coûteuses en temps de calcul pour l'entraînement. Comme nous l'avons vu dans la section 3.2.1.5, la construction de chaque arbre est indépendante, et ce problème peut être partiellement contourné en utilisant des techniques de parallélisation. Le temps d'entraînement peut également être significativement réduit en calibrant plus finement le nombre d'arbres nécessaire à la convergence des résultats.

Avec l'approche fonctionnelle, tous les classifieurs semblent nécessiter le même temps de traitement (la préparation, *i.e.* l'agrégation des profils et la projection sur la base d'ondelettes, est comparativement bien plus coûteuse). Bien que l'algorithme des RF soit le plus performant sur notre cas d'étude, remarquons que la méthode des Random Ferns (SNB) donne également de bons résultats. C'est le cas également de l'algorithme CART (en termes de point de fonctionnement optimal). Cependant, ces deux méthodes sont très peu performantes du point de vue de la discrimination avec les stops.

3.4.3 Point de fonctionnement optimal

On propose ci-dessous une méthode plus rigoureuse, et inspirée du cadre théorique de la décision bayésienne (Körding et Wolpert, 2006), pour déterminer le point de fonctionnement optimal d'un classifieur à partir de sa courbe ROC :

Introduisons les coûts $c_{10},\ c_{01},\ c_{11},\ c_{00} \in \mathbb{R}^+$, sur les décisions entraînant respectivement un faux positif, un faux négatif, un vrai positif et un vrai négatif. Un coût élevé représente

une erreur plus pénalisante, et on doit donc avoir en conséquence $c_{10} \geqslant c_{11}$ et $c_{01} \geqslant c_{00}$. La plupart du temps $c_{11} = c_{00} = 0$, mais il ne s'agit pas d'une obligation, et on pourra dans certains cas bien particuliers considérer qu'une décision correcte entraîne également un coût. L'espérance du coût de la décision sur une instance tirée aléatoirement s'exprime alors :

$$\mathbb{E}[c] = \sum_i \sum_j c_{ij}\pi_{ij}\pi_j, \tag{3.56}$$

où $\pi_{ij} = \mathbb{P}[\hat{y} = i | y = j]$ est la probabilité qu'une instance d'étiquette j ($\in \{0, 1\}$) soit classée i ($\in \{0, 1\}$) et $\pi_i = \mathbb{P}[y = j]$, est la loi a priori sur les étiquettes. Exprimons cette quantité à l'aide de $y = TPR$ et $x = FPR$:

$$\begin{aligned}\mathbb{E}[c] &= \pi_0(c_{10}x + c_{00}(1-x)) + \pi_1(c_{01}y + c_{11}(1-y)) \\ &= \pi_0 \times x(c_{10} - c_{00}) + \pi_1 \times y(c_{01} - c_{11}) + C^{ste}.\end{aligned}$$

En conséquence 2 points (x_1, y_1) et (x_2, y_2) ont la même espérance de coût si et seulement ils sont situés sur une droite de pente :

$$s = \frac{\pi_0(c_{10} - c_{00})}{\pi_1(c_{11} - c_{01})}.$$

En conséquence, les isolignes de l'espérance du coût de la décision dans l'espace ROC, sont des segments de droites de pente s (la valeur numérique de s dépend du contexte).

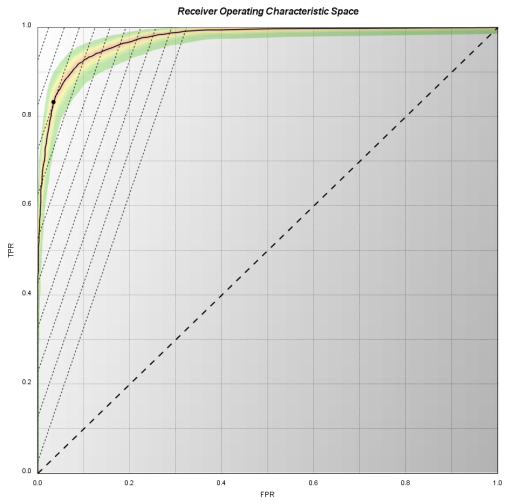


FIGURE 3.34 – Lignes d'isovaleurs du coût dans l'espace ROC (croissantes à mesure qu'on s'approche du point $(0, 1)$) et déterminantion du point de fonctionnement optimal.

Dans notre cadre d'étude, on peut considérer que les bonnes décisions ne sont pas pénalisées : $c_{11} = c_{00} = 0$. D'autre part, souhaitant travailler avec une méthode conservative dans ses détections, on pose que le coût d'un faux positif est 3 fois plus important que celui d'un faux négatif : $c_{01} = 3c_{10}$ (les valeurs numériques de ces 2 coûts n'ont pas d'importance, seul leur rapport intervient dans le calcul de l'espérance). La recherche du point de fonctionnement optimal s'effectue par dichotomie (figure 3.34) en recherchant l'intersection (quand elle existe) de la courbe ROC avec l'isoligne de plus haute valeur. Lorsque cette intersection n'est pas réduite à un point, la solution au problème n'est pas unique, et on pourra arbitrairement sélectionner l'une des intersections. Avec cette méthode, on obtient une sensibilité de 83%, et une spécificité de 97%.

3.5 Tests complémentaires et analyse de sensibilité

Dans cette partie, nous analysons la sensibilité de l'algorithme des forêts aléatoires opérant sur les variables explicatives fonctionnnelles.

3.5.1 Niveau de détail des ondelettes

Dans un premier temps, il paraît naturel d'évaluer l'influence du niveau de représentation des données fonctionnelles sur la qualité de la détection, que nous mesurons à l'aide de l'aire sous la courbe ROC. Ce choix est motivé par le fait que l'AUC est indépendante des proportions des classes dans le jeu de validation, et mesure une performance *intégrée* du classifieur sur tous les choix de seuils possibles.

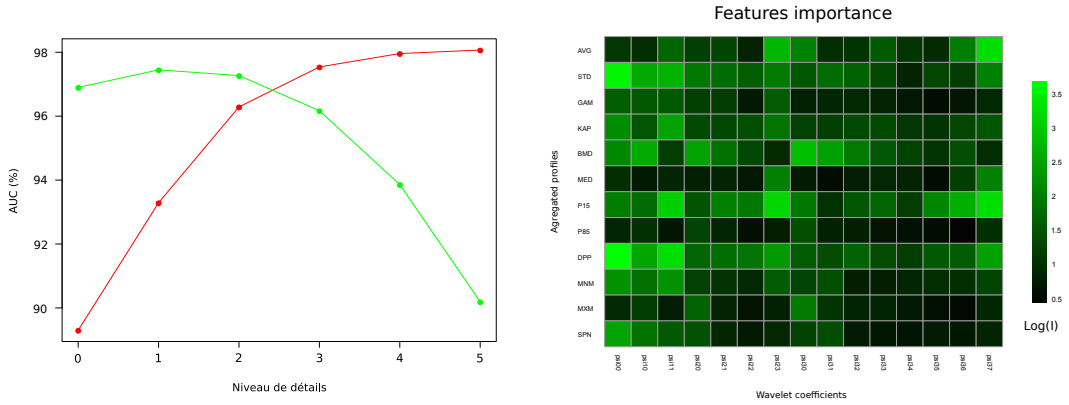


FIGURE 3.35 – A gauche : influence du niveau de l'ondelette père (en vert) et de la plus petite ondelette mère (en rouge) sur l'indicateur AUC du modèle de forêt aléatoire. L'échelle horizontale est logarithmique, avec un niveau h qui représente une échelle métrique égale à $L/2^{h+1}$ A droite : importance calculée par permutation (modèle des forêts aléatoires) pour les 12 profils agrégés, projetés sur les 15 ondelettes de base.

La figure 3.35 (à gauche) illustre la croissance des performances de détection avec le niveau de détail de la base d'ondelettes (courbe rouge). La convergence semble atteinte pour le niveau 3 (6 m), mais les niveaux 4 (3 m) et 5 (1.5 m) permettent d'obtenir des résultats légèrement meilleurs, au prix d'une surchage en temps de calcul (chaque niveau double

le nombre de variables explicatives à traiter). À l'inverse, la courbe verte montre que les performances de l'algorithme sont optimales après suppression du niveau le plus grossier.

3.5.2 Mesure d'importance des descripteurs

Ayant réglé la question du niveau de résolution, intéressons-nous à présent à l'importance relative des différents profils agrégés. Pour ce faire, nous disposons d'un outil de choix grâce aux forêts aléatoires. Nous utilisons l'indice d'importance empirique calculé par Monte-Carlo, et introduit dans le paragraphe 3.2.1.5. La figure 3.35 (à droite) compile l'importance calculée pour tous les descripteurs fonctionnels.

La figure 3.36 représente une version intégrée en lignes de la matrice illustrée à droite de la figure 3.35. On y observe l'importance accrue des profils moyen (AVG), d'écart-type (STD), de bimodalité (BMD), du 15e percentile (P85) ainsi que de la différence P85-P15 (DPP). Le 85e percentile n'est apparemment pas fondamental, probablement à cause de la redondance entre les 3 profils P15, P85 et DPP. Les profils liés aux extremums (MNM, MXN et SPN) semblent dérisoires, invitant ainsi à penser que les données sont trop bruitées. Les moments d'ordre 3 (GAM) et 4 (KAP) semblent également marginaux au regard de leur combinaison BMD. De manière surprenante, le profil médian (MED) est relégué au second plan par le profil moyen, sans que nous ne puissions expliquer pourquoi.

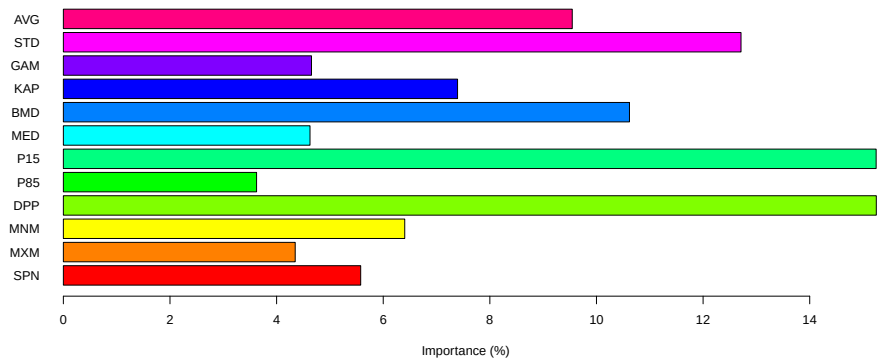


FIGURE 3.36 – Importances relatives des profils agrégés.

On peut également chercher à représenter la figure 3.36 en groupant les profils par type, avec une première classification (à gauche de la figure 3.37) plutôt de nature statistique, et une seconde classification sur la nature des phénomènes représentés par les profils.

Selon la première classification, les profils de type médiane (fondé sur le rang des données) semblent plus informatifs, contrairement à ce qui a été observé sur le graphique précédent. Dans la seconde classification, les indicateurs de type dispersion et outliers semblent plus utiles à la détection.

La même fusion de coefficients peut être effectuée suivant les colonnes de la figure 3.35.

L'histogramme de gauche en figure 3.38 montre de manière intéressante que l'importance

	AVG	STD	GAM	KAP	BMD	MED	P15	P85	DPP	MNM	MXM	SPN
MOMENTS	✓	✓	✓	✓								
BIMOD					✓							
MEDIAN						✓	✓	✓		✓	✓	
SPAN									✓			✓
TENDANCE	✓					✓						
DISPERSION		✓							✓			✓
OUTLIERS										✓	✓	
AUTRES			✓	✓	✓							

TABLE 3.6 – Groupes de profils agrégés.

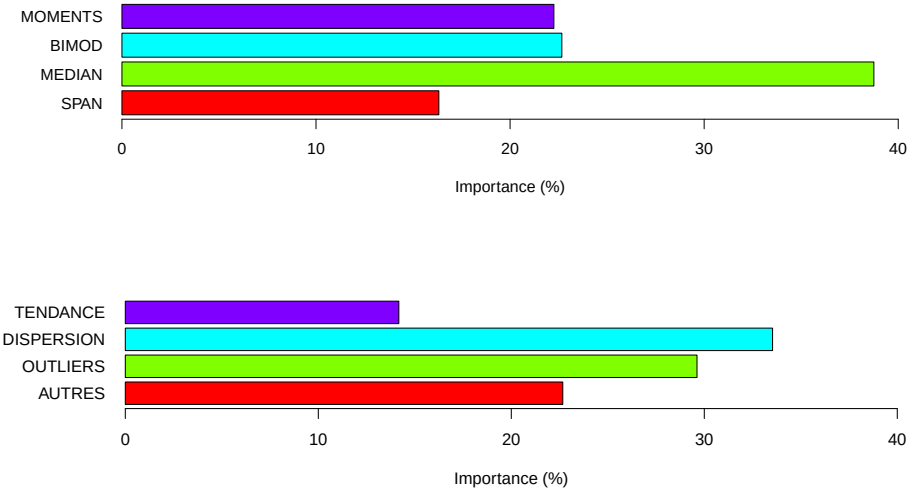


FIGURE 3.37 – Importances relatives des profils agrégés, sommées par type (*cf* table 3.6).

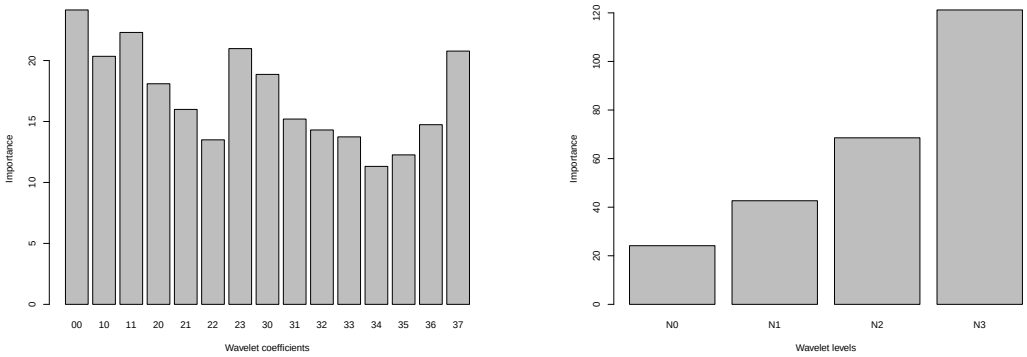


FIGURE 3.38 – Importances relatives groupées par ondelette de base (à gauche) et par niveau d'ondelette (à droite).

des coefficients est plus significative au niveau des ondelettes de base localisées en bordure de fenêtre, ce qui peut être confirmé indépendamment du type de profil sur la figure 3.39.

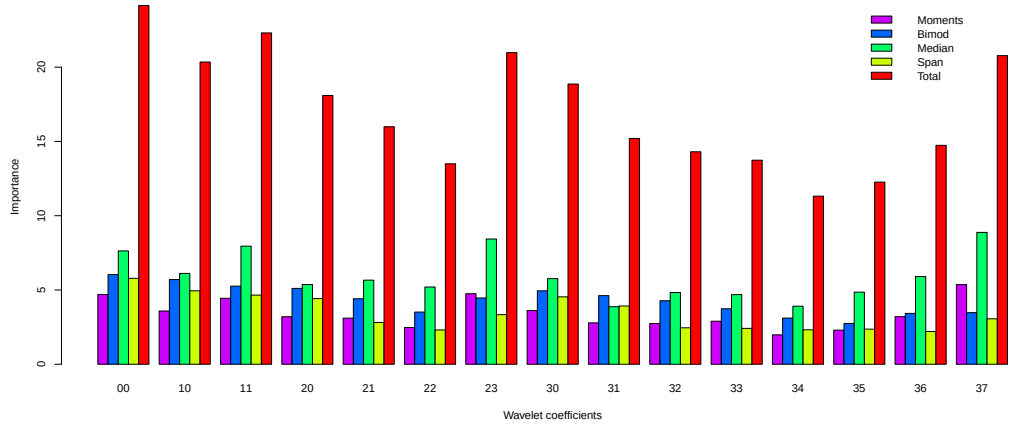


FIGURE 3.39 – Importances relatives groupées par ondelette de base et par types de profils.

3.5.3 Nombre de profils disponibles

Une problématique importante dans les travaux de map inference consiste à essayer d'évaluer la taille du jeu de données nécessaire à l'obtention des résultats souhaités. La réponse à cette question passe par l'analyse de sensibilité de l'algorithme de détection au nombre de profils disponibles. Dans le cadre de notre expérimentations, nous mettons à profit le fait de disposer d'un grand nombre de traces (144 au total) sur l'ensemble des fenêtres glissantes, pour pouvoir réduire ce nombre, et évaluer l'impact sur l'indicateur AUC. Nous avons donc fait varier le nombre de profils N de 2 à 144 (l'algorithme n'est pas défini pour $N = 1$), par pas de une unité. Pour chaque expérimentation, jeu d'entraînement et de validation contiennent tout deux N profils. La figure 3.40 illustre le résultat obtenu.

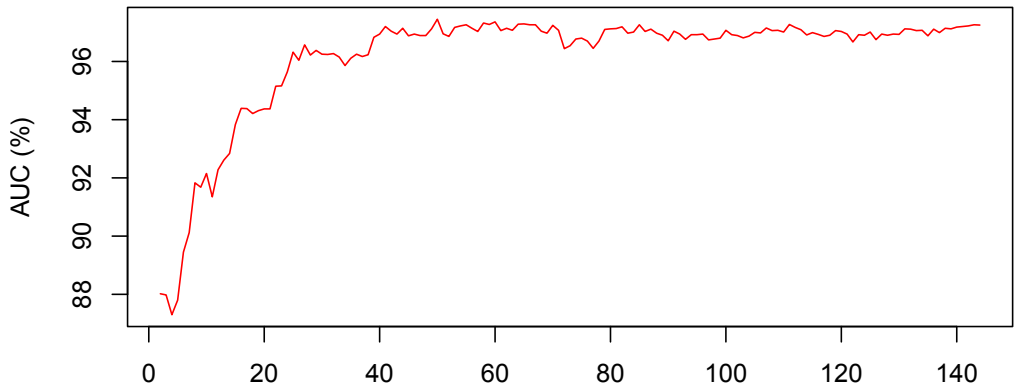


FIGURE 3.40 – Influence du nombre N de profils de vitesse disponibles.

Les résultats mettent clairement en évidence la convergence du score AUC à partir de 30 à 40 profils. Des résultats décents ($\sim$ 92 %) peuvent être obtenus avec seulement une dizaine de profils. Une étude plus poussée à l'avenir pourrait tenter d'évaluer l'impact du nombre de profils disponibles séparément sur les jeux d'entraînement et de validation.

3.5.4 Influence de la précision géométrique des trajectoires

S'il est possible de supposer que les courbes sont des réplications *i.i.d.* d'une variable aléatoire à valeurs fonctionnelles, permettant ainsi de dégrader la *quantité* du jeu de données en éliminant aléatoirement des courbes, il n'en va pas de même pour l'étude de la *qualité* géométrique et nous ne disposons pas de modèle statistique pour générer des jeux de données degradés. En effet, considérer les valeurs prises par les fonctions comme des échantillons *i.i.d.* se résumerait à introduire un bruit blanc dans les profils de vitesse, ce qui produirait des simulations irréalistes, comme illustré sur le graphique en haut au centre de la figure 3.42.

Or, nous avons vu précédemment dans la section 2.4.3.2 que les erreurs de mesure du GPS sont autocorrélées. Nous proposons ici un modèles de bruitage générique, dans lequel on suppose que le profil de vitesse est calculé par différenciation des positions, comme c'est le cas dans la plupart des jeux de données FCD (*cf* chapitre 4) et le processus d'erreur sur le profil est alors directement exprimé en fonction du bruit sur les positions.

3.5.4.1 Calcul de la matrice de covariance des erreurs

Nous supposons que la série des erreurs est une réalisation d'un processus gaussien de moyenne nulle et de fonction de covariance exponentielle symétrique (de portée $a^{-1} = 539$ m), de manière similaire à l'estimation empirique réalisée dans le paragraphe 2.4.3.2. On part alors de la définition 2.2 de (Andrieu, 2013), qui exprime tout profil de vitesse sous la forme : $v(x) = F' \circ F^{-1}(x)$. Le processus d'erreur Z estimé dans la section 2.4.3.2, correspond au bruit (dans la direction transversale) sur la position du véhicule par rapport à l'axe de la route. Par symétrie du problème, on peut supposer qu'il s'agit aussi du bruit sur la composante longitudinale. Sous l'hypothèse d'une vitesse approximativement constante, le processus d'erreur Z sur la position correspond également au bruit qui entâche F et on cherche à calculer le modèle du processus de bruit Y sur F'.

On montre facilement (Bendat et Piersol, 2011) que, lorsqu'elle existe, la fonction d'autocorrélation $\gamma_{Z'}$ de la dérivée d'un signal aléatoire Z, est égale à la dérivée seconde de la fonction d'autocorrélation de Z :

$$\mathbb{E}\Big[\frac{dZ(x)}{dx}\frac{dZ(x+h)}{dx}\Big] = \gamma_{Z'}(h) = \frac{d^2}{dh^2}\big[\gamma_Z(\tau)\big] = \frac{d^2}{dh^2}\Big[\mathbb{E}[Z(x)Z(x+h)]\Big]. \tag{3.57}$$

Cependant, le modèle $\gamma_Z(h) = \exp(-a|h|)$, retenu au chapitre précédent, n'est pas dérivable en $h = 0$, ce qui implique que le processus d'erreur Z n'est pas dérivable en moyenne quadratique. Il est donc impossible d'obtenir une expression $\gamma_{Z'}$ de l'autocorrélation de la dérivée formelle de Z. Intéressons-nous alors aux dérivations numériques. Notons $s(x)$ la lecture de la position 1D (estimée à partir du GPS) au niveau de la position réelle x. Le calcul du profil de vitesse v en x s'exprime par différence finie sur un pas $dx = vdt$:

$$v(x) = \frac{s(x) - s(x - vdt)}{dt}.$$

En considérant $dt = 1$ sec, on peut exprimer la vitesse estimée (entâchée de l'erreur GPS) :

$$\hat{v}(x) = \hat{s}(x) - \hat{s}(x - v) = s(x) + Z(x) - s(x - v) - Z(x - v).$$

L'erreur sur la vitesse estimée est donc égale à la partie stochastique $Y(x) = Z(x) - Z(x - v)$, qui ne dépend pas uniquement de la position x, mais aussi de la vitesse vraie en x. Le processus Y est stationnaire à l'ordre 1 : ($\mathbb{E}[Y(x)] = \mathbb{E}[Z(x)] + \mathbb{E}[Z(x - v)] = 0$, indépendant de x) mais nous allons voir que sa covariance dépend de v et donc, par suite, de la position x. Étudions la corrélation entre les valeurs prises par Y en deux points x_1 et x_2 (en notant $v_i = v(x_i)$, la vitesse vraie du véhicule en x_i) :

$$\gamma_Y(x_1, x_2) = \mathrm{Cov}(Y(x_1), Y(x_2)) = \mathrm{Cov}(Z(x_1) - Z(x_1 - v_1), Z(x_2) - Z(x_2 - v_2)$$

$$= \mathrm{Cov}(Z(x_1), Z(x_2)) - \mathrm{Cov}(Z(x_1), Z(x_2 - v_2))$$
$$-\mathrm{Cov}(Z(x_2), Z(x_1 - v_1)) + \mathrm{Cov}(Z(x_1 - v_1), Z(x_2 - v_2)).$$

Pour un processus Z stationnaire à l'ordre 2, et en notant $h = x_2 - x_1$ l'écart entre les sites, et $\Delta v_{12} = v_2 - v_1$ l'écart entre les vitesses correspondantes, l'expression se réduit à :

$$\gamma_Y(h) = \gamma_Z(h) + \gamma_Z(h - \Delta v_{12}) - \gamma_Z(h + v_1) - \gamma_Z(h - v_2). \tag{3.58}$$

Enfin, en substituant l'expression de γ_Z dans 3.58, on obtient (en notant que Y n'est pas stationnaire et dépend donc des vitesses) :

$$\mathbf{\Gamma}_{ij} = \sigma^2 \left[e^{-a|i-j|} + e^{-a|i-j-(v_j - v_i)|} - e^{-a|i-j+v_i|} - e^{-a|i-j-v_j|} \right], \tag{3.59}$$

où $\sigma^2 = \gamma_X(0)$ correspond à la variance du bruit sur la position.

$$\forall\ i, j \in \{1, 2, ...n\} \qquad \mathbf{\Gamma}_{ij} = \sigma^2 \exp(-a|i - j|). \tag{3.60}$$

3.5.4.2 Simulation du bruit sur le profil de vitesse

On utilise une technique inspirée de Vauglin (1997).

La matrice $\mathbf{\Gamma}$ étant symétrique définie-postive, elle peut être écrite sous la forme d'une factorisation de Cholesky : $\mathbf{\Gamma} = \mathbf{L}\mathbf{L}^T$, avec $\mathbf{L}$ une matrice triangulaire inférieure de $\mathbb{R}^{n \times n}$.

Considérons $\mathbf{Z} \sim \mathcal{N}(0_{\mathbb{R}^n}, I_{\mathbb{R}^n})$ un vecteur de bruit blanc gaussien standard à n éléments, et formons le produit $\mathbf{X} = \mathbf{LZ}$. On vérifie alors que $\mathbf{X}$ est une réalisation d'un processus gaussien de matrice de covariance $\mathbf{\Gamma}$:

$$\mathbf{\Sigma_X} = \mathbf{L\Sigma_Z L^T} = \mathbf{LL^T} = \mathbf{\Gamma},$$

où la seconde égalité résulte du fait $\mathbf{Z}$ est un bruit blanc unitaire.

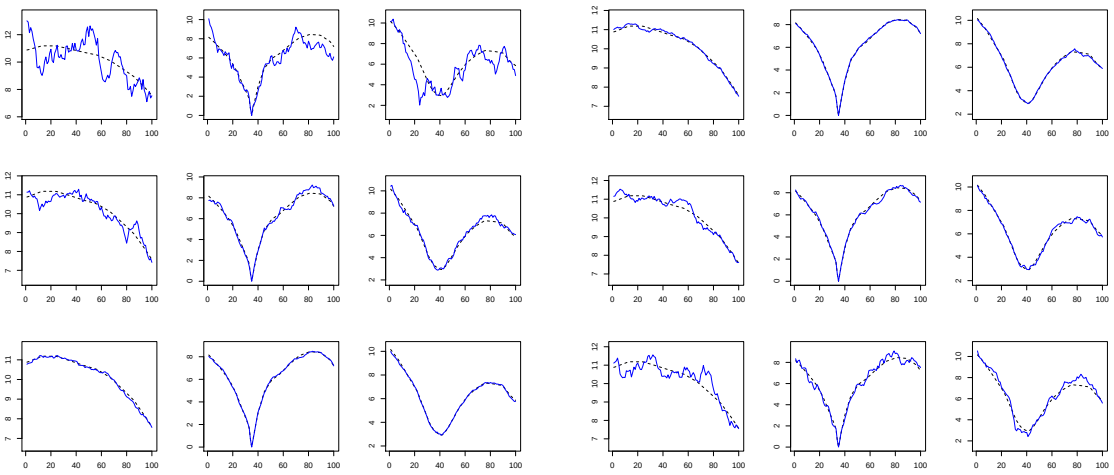


FIGURE 3.41 – Simulation de bruit sur 3 profils (en colonnes), pour 3 portées a^{-1} (en lignes à gauche) et pour 3 amplitudes $\gamma(0)$ différentes (en lignes à droite). Le profil original est représenté en pointillés noirs, le profil bruité en traits pleins bleus.

Notons que le bruit sur le profil de vitesse intervient en réalité à 2 niveaux : (1) le bruit sur la position GPS et (2) le bruit sur la vitesse Doppler. On peut donc décliner l'équation 3.60 en 2 versions, avec σ_p l'erreur typique sur la position et σ_v l'erreur typique sur la vitesse. On calcule alors la décomposition de Cholesky de $\mathbf{\Gamma_v}$ et $\mathbf{\Gamma_p}$. On multiplie les 2 matrices triangulaires obtenues par 2 bruits blanc gaussien unitaires indépendants, puis on ajoute les 2 signaux de bruit obtenus à chacune des 2 colonnes X et V d'un profil de vitesse. On reformate alors le profil par interpolation à l'aide des techniques décrites dans le paragraphe 2.3.2.4. L'opération est réitérée pour chaque profil de la fenêtre. La figure 3.42 donne quelques exemples de bruitage d'un ensemble de profils de vitesse sur une fenêtre glissante.

On étudie alors l'impact sur la qualité de détection en faisant varier les paramètres de bruit, pour un écart type σ_{bruit} compris entre 0 et 12 m. La réponse du classifieur (mesurée par l'aire sous la courbe ROC) est représentée sur la figure3.43.

Les résultats montrent que la méthode est relativement robuste au niveau de bruit, sans variation notable de la précision de détection sur la plage de bruit typique des récepteurs GPS. Elle devrait donc pouvoir être appliquée sur des données FCD dont la vitesse est estimée *a posteriori* par différenciation numérique, et non calculée directement par l'effet Doppler au niveau du récepteur. Nous expliquons cette robustesse par l'étape d'agrégation

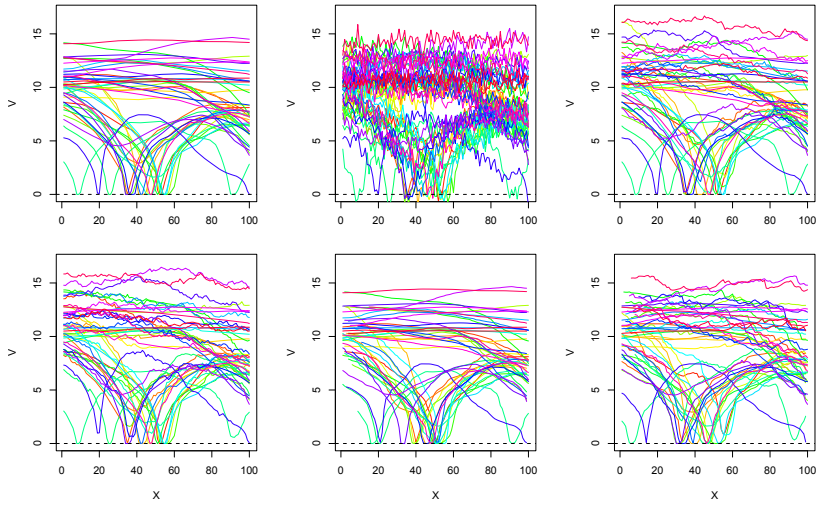


FIGURE 3.42 – Bruitage d'un ensemble de profils de vitesse. De gauche à droite et de haut en bas. 1) Profils originaux. 2) Bruit blanc gaussien ($\sigma = 0.5$m/s). 3 et 4) Bruit corrélé sur la vitesse Doppler. 5) Bruit corrélé sur la position GPS. 6) Bruit corrélé sur la vitesse Doppler et la position GPS.

des profils, qui réduit d'autant la variance du bruit sur les trajectoires individuelles. On remarque que la précision de détection amorce une chute drastique sur des niveaux de bruit élevés (typiquement supérieurs à une dizaine de mètres), indiquant ainsi que la méthode peut plus facilement échouer dans les zones urbaines denses en encaissées.

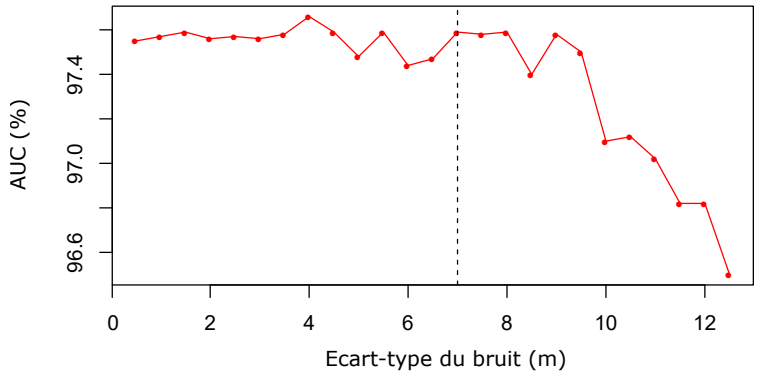


FIGURE 3.43 – Aire sous la courbe ROC du modèle de forêt aléatoire en fonction de l'écart-type du bruit (sur la position). La ligne verticale en pointillés représente la limite haute du niveau de bruit d'un récepteur typique.

Notons qu'une limite de notre modélisation du bruit sur les profils provient de l'hypothèse de stationnarité du processus d'erreurs et d'indépendance des profils. En pratique, la structure géométrique du bâti environnant est susceptible de causer des erreurs fortement corrélées entre les différentes trajectoires GPS, impliquant ainsi *de facto* une corrélation dans les erreurs entâchant les profils de vitesse, limitant ainsi la réduction de la variance par agrégation des profils.

Le résultat graphique du processus de bruitage est illustré sur la figure 3.41.

3.5.5 Prise en compte des accélérations

Dans cette section, nous cherchons à évaluer l'importance des termes d'accélérations dans le processus de classification. Nous avons vu dans la section 3.5.1 que les niveaux d'ondelettes les plus fins ne semblaient pas apporter d'information significative pour la détection (excepté, pour une raison inconnue, sur les bords de la fenêtre).

Nous tentons ici d'étendre l'approche directe pour lui permettre d'avoir accès aux valeurs d'accélérations. En effet, comme mentionné dans la section 3.3.2, à chaque pas d'abscisse curviligne x, les N mesures de vitesses brutes $v_i(x)$ sont ordonnées, puis concaténées au vecteur de variables explicatives, cassant ainsi la relation cinématique existant entre les profils entre les positions x et $x+1$. Avec les decripteurs de l'approche directe, les méthodes de classification ne peuvent alors déduire que des *accélérations de groupe*, *i.e.* une accélération moyenne par abscisse x et pour l'ensemble des véhicules.

Dans leur ouvrage de référence, Ramsay et Silverman (2005) mettent en évidence la difficulté de calculer des dérivées numériques (et a fortiori des dérivées secondes) sur des données fonctionnelles non-traitées. Pour calculer les accélérations, par souci de cohérence et de simplicité, nous partons directement du profil spatial de vitesse, en utilisant la règle de chaînage des dérivations :

$$a(x) = \frac{dv}{dt}(x) = \frac{dv}{dx}\frac{dx}{dt} = v'(x)v(x), \tag{3.61}$$

où $v'(x)$ désigne la dérivée spatiale du profil en x. Le calcul effectif de la dérivée est alors effectué par différence finie centrée ou différence avant (resp. arrière) pour la première (resp. dernière) position de la fenêtre :

$$a_k = \begin{cases} v_k(v_{k+1} - v_k) & \text{si } k = 1 \\ v_k(v_k - v_{k-1}) & \text{si } k = L \\ \frac{1}{2}v_k(v_{k+1} - v_{k-1}) & \text{sinon.} \end{cases} \tag{3.62}$$

Cette solution permet de bénéficier du *pipeline* complet de pré-traitements : interpolation, lissage et extraction du profil spatial d'interpolation, tout en présentant l'avantage d'obtenir des valeurs numériques concordantes entre les valeurs de vitesse et d'accélération.

Formellement, quand $v(x) = 0$, la dérivée spatiale $v'(x)$ est infinie, et la quantité $a(x)$ est alors non-définie. En pratique, l'accélération étant par hypothèse une fonction continue du temps, $a(t) = 0$ quand $v(t) = 0$ et l'accélération spataile $a(x)$ est bien définie en tout point x (y compris quand $v(x) = 0$).

Étudions la stabilité numérique de ce nouveau profil. On montre aisément (à l'aide d'une technique similaire à celle utilisée dans la section 2.2.3, et moyennant l'hypothèse d'un bruit d'autocorrélation exponentielle sur la vitesse) :

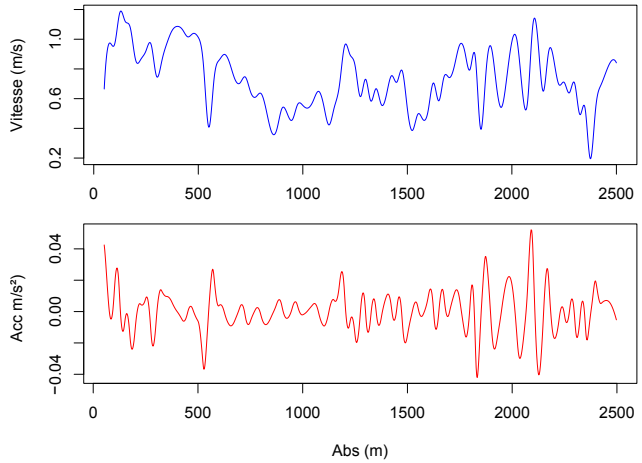


FIGURE 3.44 – En haut : profil spatial de vitesse d'un randonneur. En bas : dérivation numérique du profil spatial d'accélération par les formules 3.61 et 3.62.

$$\sigma_a^2 = \frac{\sigma_v^2}{4}\Big(2v^2(1-\rho^2) + (v_+ - v_-)^2\Big), \tag{3.63}$$

où σ_a et σ_v désignent respectivement les écarts-types sur la vitesse et l'accélération en un point donné x du profil, v_- et v_+ désignent les mesures de vitesse en $x-1$ et $x+1$ et ρ est la corrélation des erreurs entre deux vitesses espacées d'un pas de 1 m. On peut réécrire l'équation 3.63 en l'exprimant sous forme de bande de confiance et fonction de l'accélération à partir de la différence finie centrée 3.62, en supposant $v \neq 0$:

$$CB(x;p) = \frac{\eta(p)\sigma_v}{2}\sqrt{\frac{2v(x)^2(1-\rho^2) + 4(a(x)/v(x))^2}{2}}, \tag{3.64}$$

avec p le niveau de confiance, η la fonction de répartition inverse de la loi normale et $CB(x)$ la demi-largeur de la bande de confiance du profil spatial d'accélération en x, telle qu'illustré sur la figure 3.45 sous la forme de 2 fonctions : $a(x) \pm CB(x; 0.95)$.

Les mesures d'accélérations sont alors ordonnées à chaque pas de distance, et concaténées au vecteur de descripteurs contenant les mesures brutes de vitesse, de sorte à tester l'approche directe augmentée avec les données d'accélération. Les résultats obtenus sont compilés dans la table 3.7.

Nous observons que les données d'accélération n'améliorent pas significativement l'approche directe.

3.5.6 Validation croisée sur les conducteurs

Nous devons valider une hypothèse formulée au paragraphe 3.3.5 qui stipulait que les profils de vitesse ne sont que très peu dépendants des comportements individuels des conducteurs, devant le rôle joué par l'infrastructure routière. En conséquence, nous avons décidé d'inclure

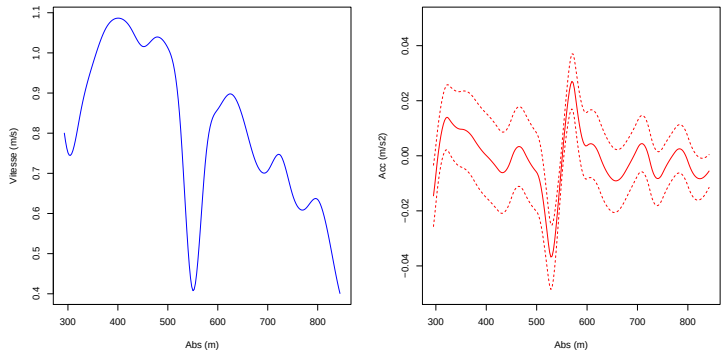


FIGURE 3.45 – À gauche : profil spatial de vitesse. À droite : profils spatial d'accélération (3.62) et bandes de confiances (3.64). pour $\sigma_v = 0.1$ m/s, $\rho = 0.99$ et $p = 0.95$.

Algo.	ACC	AUC	STP	OFT	ONT
NB	71.31	77.12	85.05	1.96	0.44
CART	86.61	88.39	41.33	2.11	0.44
kNN	86.55	83.51	37.08	3.65	0.44
SNB	68.48	88.21	97.49	1.78	0.44
RF	91.79	95.72	4.50	7.55	0.44

TABLE 3.7 – Performances de prédiction pour les 5 classifieurs avec la méthode directe augmentée à l'aide des données d'accélération. Les indicateurs de performance sont les mêmes que ceux esposés dans le paragraphe 3.4.1.

tous les conducteurs dans les deux jeux de données : entraînement et validation. Comme illustré sur la figure 3.46, ceci revient à entraîner chaque modèle sur les jeux 1 et 2, puis à les valider sur les jeux 3 et 4. Afin de confirmer cette hypothèse, nous avons répliqué l'une des expérimentations (celle du classifieur RF avec des variables explicatives fonctionelles) deux fois avec deux protocoles expérimentaux différents. Dans le premier cas, nous avons échantillonné aléatoirement 15 conducteurs, et les 84 profils de vitesse qui leur étaient associés ont été utilisés pour créer à la fois le jeu d'entraînement et de validation. Ce procédé correspond au protocole détaillé dans la section 3.3.5, à la différence près que la moitié des conducteurs ont été retiré des données. Dans le second cas, ces mêmes 84 profils de vitesse ont été utilisés dans le jeu d'entraînement, tandis que les 60 profils restants (provenant de la seconde moitié des conducteurs) ont constitué le jeu de validation. De cette manière, la validation est opérée sur des données complètement nouvelles, à la fois du point de vue de l'infrastructure routière, mais aussi des profils et des conducteurs ayant généré ces profils. Sur la figure 3.46, ce second procédé correspond à un entraînement du modèle sur le jeu 1, et une validation sur le jeu 4. Notons que dans ces deux cas de figure, le découpage de la boucle de circuit est effectué d'une manière similaire à celle décrite précédemment. Cette expérimentation complémentaire vise à capturer (si elle existe) la différence qui pourrait subsister entre les procédés consistant à entraîner et à valider un modèle sur les mêmes ou sur différents sous-ensembles de conducteurs.

Les résultats de l'expérience ont confirmé l'hypothèse, puisqu'aucune différence significative n'a été observée entre les courbes ROC, comme illustré sur la figure 3.46. On peut

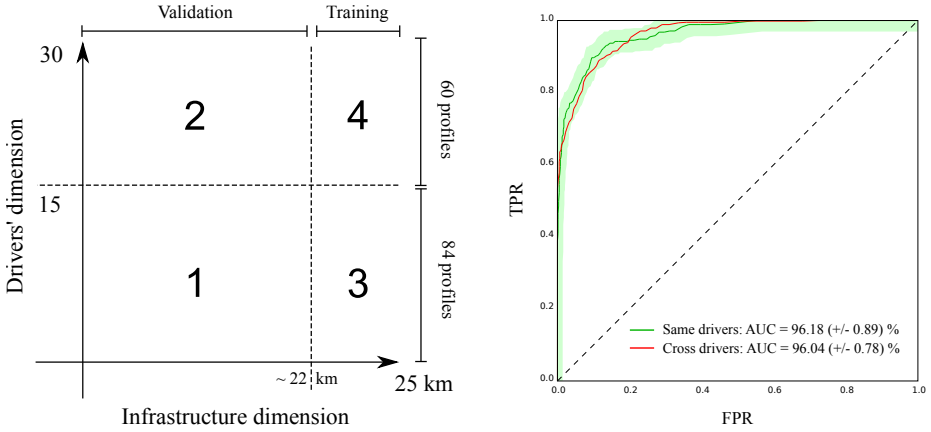


FIGURE 3.46 – À gauche : séparation entraînement/validation suivant 2 dimensions (infrastructure routière et conducteurs). À droite : courbes ROC pour un modèle entraîné sur un jeu contenant des conducteurs identiques (resp. differents) en vert (resp. rouge).

donc en conclure que le fait d'avoir validé le modèle à l'aide des mêmes conducteurs (par souci de simplicité) n'a vraisemblablement pas produit de biais dans les scores de classification qui auraient conduit à surestimer les performances des algorithmes. On peut remarquer également que l'AUC des deux modèles est légèrement inférieure à celle du modèle de l'expérimentation originale (97.3%), dont les performances sont compilées dans la table 3.3. Cette légère différence s'explique par le fait que le nombre de données utilisées pour l'entraînement dans cette expérimentation additionnelle était de moitié inférieur.

3.6 Extensions

Dans cette section, nous tentons d'appliquer nos algorithmes sur d'autres types d'éléments de l'infrastructure routière.

3.6.1 Détection des passage piétions

Le même algorithme (sans changer aucun des descripteurs) est appliqué sur les passages piétons, qui ont été répertoriés avec une méthodologie similaire à celle décrite dans le paragraphe 3.1.1. Nous disposons au total de 92 éléments, *i.e.* 920 instances positives, soit 37% des fenêtres sur la zone. La méthodologie de séparation jeu d'entraînement - jeu de validation est également similaire, et les performances de l'algorithme sont évaluées par 10-fold cross validation. La figure 3.47 illustre la courbe ROC obtenue lors de la validation.

L'aire sous la courbe du modèle s'élève à 86 ($\pm$1.4) %, soit environ 10% de moins que pour le cas de la détection des feux tricolores. Les résultats restent malgré tout acceptables, suggéraint ainsi que la méthode peut potentiellement être étendue à d'autres éléments de l'infrastructure. Le point de fonctionnement optimal (tel que défini dans la section 3.4) est sélectionné avec une sensibilité de 79 % et une spécificité de 80%.

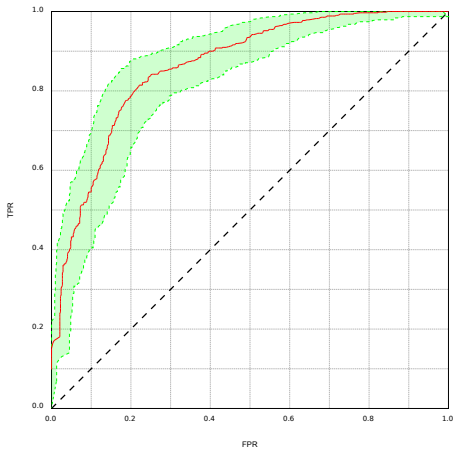


FIGURE 3.47 – Courbe ROC et bandes à 95% pour la détection des passages piétons.

3.6.2 Discrimination feu - stop

Dans ce chapitre, nous avons établi un modèle de classifieur permettant de détecter les feux tricolores à partir d'un ensemble de profils de vitesse. Nous avons vu également que le travail sur une collection de profils impose de trouver des moyens d'agréger les informations disponibles afin de traiter un vecteur de descripteurs de dimension fixée. Dans ce paragraphe, nous cherchons à étudier la quantité d'information portée par les profils individuels, autrement dit :

Peut-on essayer de détecter le type d'infrastructure routière rencontrée par un véhicule à partir de son (unique) profil de vitesse ?

Pour ce faire, nous employons une méthode similaire à celle décrite dans le paragraphe de conclusion de la section 3.2.2.4, pour traiter un problème de classification binaire entre les arrêts de stop et les arrêts de feux tricolores. Plus précisemment, on utilise l'équation 3.25 pour extraire 2 bases de fonctions orthonormales $\{\varphi_i^0\}_{i\in\mathbb{N}}$ et $\{\varphi_i^1\}_{i\in\mathbb{N}}$, de l'espace des processus stochastiques décrivant respectivement un arrêt de stop et un arrêt de feu. La construction des bases est effectuée sur un jeu d'entraînement contenant 50% des profils d'arrêt. Pour chaque profil v de la base de validation, on calcule la projection du profil sur les sous-espaces vectoriels générés par les p premières composantes (dans l'ordre décroissant des valeurs propres) de chacune des 2 bases (et on note $a_{ij}(v)$ le coefficient de la projection de v sur φ_i^j). On évalue alors les erreurs de reconstruction, puis le profil v est affecté à la classe correspondant à la base sur laquelle la reconstruction est la plus précise :

$$\hat{y}(v) = \underset{j\in\{0,1\}}{\operatorname{argmin}} \int_0^L \left(v(x) - \sum_{i=1}^{p} a_{ij}(v)\varphi_i^j(x)\right)^2 dx. \tag{3.65}$$

L'expérimentation est répétée pour un nombre p de composantes variant de 1 à 20. Les résultats obtenus sont illustrés sur la figure 3.48. On observe que les taux de détection sont optimaux pour $p = 4$. Les parts de variances expliquées pour les deux décompositions

sur les 4 premiers vecteurs de base sont de l'ordre de 81 %. Les scores individuels associés sont de $a = 72.2\%$ de taux de détection pour les stops et $b = 84.2\%$ pour les feux, ce qui correspond à un taux de détection global (sous l'hypoythèse d'équirépartition des classes dans le jeu de données) de 78.2 %.

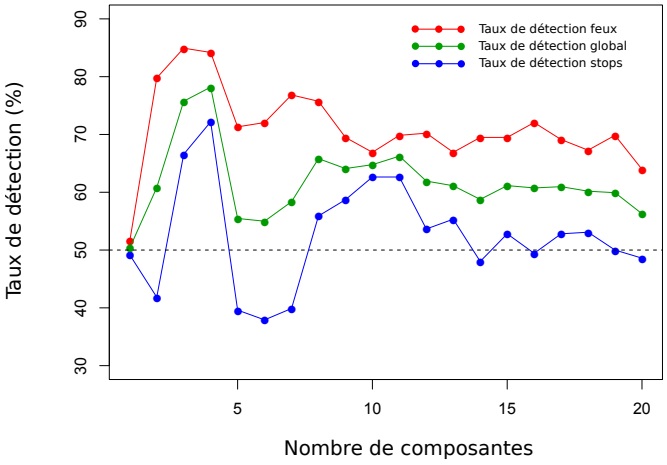


FIGURE 3.48 – Taux de détection de la méthode de discrimination stop/feux sur profil unique, en fonction du nombres de composantes retenues dans la décomposition.

Pour comparer ces scores de performances avec les résultats obtenus sur la méthode principale de ce chapitre, on peut calculer l'aire sous la courbe (le classifieur étant purement discret, la représentation de cette courbe n'apporte aucune information). L'indicateur AUC ne dépend que des valeurs a et b relevées ci-dessus, et on montre aisément que l'AUC est la moyenne des scores :

$$\text{AUC} = 1 - \left[(1-a)(1-b) + \frac{(1-a)b}{2} + \frac{(1-b)a}{2}\right] = \frac{a+b}{2}. \tag{3.66}$$

La courbe verte de la figure 3.48 représente donc également l'AUC du classifieur, et on obtient une AUC optimale de 78.2 %.

Les résultats obtenus sont significativement au dessus de 50 % (ce qui correspondrait à une classifieur purement aléatoire), ce qui incite à penser que la combinaison des classifications opérées sur plusieurs courbes à l'aide de la méthode présentée dans ce paragraphe peut permettre d'obtenir de meilleurs scores que la classification des courbes agrégées [12]. La réponse à cette question sera laissée en suspens, et fera l'objet de travaux ultérieurs.

12. Sous réserve que l'erreur de classification soit de type "variance" (et donc susceptible d'être réduite avec l'augmentation du nombre de courbes) et non de type biais (*cf* 3.2.1.6).

3.7 Conclusions du chapitre

À partir des résultats de cette expérimentation, on peut conclure que l'approche fonctionnelle, en plus d'être remarquablement plus rapide, produit des résultats précis, en particulier avec l'algorithme des forêts aléatoires. On remarque également que ce même algorithme paraît moins sensible à l'approche employée pour extraire les descripteurs.

Cependant, le manque de données est une limite importante dans l'analyse des résultats de cette première expérimentation. Nous avons essayé de prendre en compte ce manque en générant les intervalles de confiance autour des courbes ROC, mais il reste malgré tout difficile de supposer que le jeu de données traité est représentatif de l'ensemble des feux tricolores du territoire national.

Parmi les nombreux avantages proposés par les forêts aléatoires, la validation *out-of-bag* est très certainement l'un des plus attractifs, puisqu'elle permet de calculer un estimateur (pratiquement non-biaisé) de toutes les performances de classification d'un modèle. Cependant, nous avons remarqué dans notre cadre d'application que les estimations OOB du taux d'erreur étaient systématiquement optimistes (avec une valeur de l'erreur 3 à 5 fois plus petite que les estimateurs calculés sur le jeu de validation), montrant ainsi que les fenêtres glissantes ne peuvent être considérées comme étant rigoureusement indépendantes. L'erreur OOB ne peut alors être interprétée que moyennant une modification dans le code des librairies de calcul, de sorte à s'assurer que les arbres n'évaluent ni les échantillons OOB qui ont contribué à les construire, ni les instances vosisines de ces échantillons. En échange, en réduisant d'autant plus l'effectif de la forêt de validation pour chaque instance, on risque à l'inverse d'obtenir un estimateur OOB pessimiste.

Alors que le nombre de conducteurs est également problématique, nous pensons que le fait de disposer de plusieurs trajectoires pour chacun d'entre eux permet de considérer le jeu de données GPS comme représentatif. Dans les travaux ultérieurs (*cf* 3.5.3), nous évaluons l'impact du nombre de profils disponibles sur la qualité des prédictions du modèle, de sorte à déterminer le nombre minimal de trajectoires nécessaire pour obtenir des résultats décents. Nous confirmons que ce nombre n'est pas trop grand (typiquement de l'ordre d'une vingtaine), ce qui nous permettra d'orienter nos recherches suivantes sur des jeux de données acquis en environnement non-contrôlé, avec potentiellement moins de trajectoires par axe routier, mais un plus grand nombre d'instances de feux tricolores, de sorte à être dans une situation plus confortable pour valider statistiquement les performances des algorithmes.

Un nombre plus important de données pourrait également permettre de calibrer plus finement les modèles d'apprentissage, plutôt que de recourir aux paramètres recommandés dans la littérature. De plus, ceci pourrait permettre d'exploiter la corrélation spatiale entre les instances : à courte échelle, les instances sont corrélées positivement (du fait de la superposition des fenêtres). Sur un horizon plus large, la corrélation devient probablement négative avant de tendre à s'annuler. En effet, il est peu probable de trouver deux feux sur des fenêtres voisines (mais qui ne se chevauchent pas). Dans un cas plus général où les fenêtres ne sont pas situées sur un circuit en boucle mais sont distribuées sur un réseau routier complexe, la corrélation spatiale peut être signicativement positive pour différents axes menant à un même carrefour. Dans ce cas, la corrélation entre les instances devient à la fois géométrique et topologique, puisqu'elle est supportée par le réseau routier.

Chapitre 4

Étude du potentiel des méthodes d'apprentissage sur un cas opérationnel

Sommaire

Introduction

Nous présentons ici un cas d'étude opérationnel, avec des données GPS collectées en environnement non-contrôlé, et distribuées sur un réseau routier complexe, sur lequel nous tentons d'adapter la méthode étudiée dans le chapitre précédent (et en utilisant la combinaison algorithme/descripteurs ayant donné les meilleurs résultats). Cette section correspond à une période de mobilité internationale de 4 mois, effectuée au Center for Spatial Information Science de l'Université de Tokyo à l'aide d'une bourse de l'Université Paris-Est et sous la direction de R. Shibasaki et H. Kanasugi. Les données utilisées ont été gracieusement mises à disposition par Y. Kato, de la compagnie NAVITIME. Les études menées dans ce chapitre ont pour objectif d'apporter des éléments de réponses aux trois interrogations suivantes :

- Dans le chapitre 3, nous avons utilisé des données de trajectoires GPS de véhicules le long d'un parcours linéaire en boucle. Dans ce cadre restreint, la définition des instances est naturelle et ne pose pas de problème. Sur un réseau topologique complexe, la notion de fenêtre glissante n'a plus de sens. Peut-on alors trouver une définition satisfaisante pour le fenêtrage [1], permettant de *capturer* de manière parcimonieuse l'intégralité du signal nécessaire à la détection des feux tricolores ? La section 4.1 a pour objectif d'étudier cette question.

- La description mathématique des instances proposée dans le chapitre 3 est elle adaptée à la modélisation des trajectoires GPS de qualité moyenne ? La nature même des données (bruitées, incomplètes...) nous conduira à proposer une nouvelle description dans la section 4.2. L'analyse des résultats de l'expérimentation (4.3) permettra de se faire une idée plus exacte du potentiel de l'approche en contexte opérationnel.

- L'objectif *in fine* étant de détecter et localiser précisemment les éléments de la signalisation routière, nous analyserons également les capacités de l'algorithme d'apprentissage en termes d'inférence de la position des feux.

Notons que la manipulation de données observationnelles [2] nécessitera des procédures de pré-traitement spécifiques, que nous avons choisi de ne pas faire figurer dans le chapitre 2, en particulier en ce qui concerne l'acquisition et le contrôle de la vérité terrain (4.1.2), ainsi que le débiaisage des coordonnées des trajectoires (4.4.1). Nous concluons le chapitre en analysant la sensibilité des performances de l'algorithme aux paramètres extérieurs (4.4.2), puis en proposant quelques extensions naturelles (4.4.3).

4.1 Constitution du jeu de données

4.1.1 Zone d'étude

Les expérimentations ont été menées sur Mitaka (Japon), une ville de la proche banlieue de Tokyo, s'étendant sur une superficie de 16 km^2. Le choix de cette zone d'étude a été motivé par le fait qu'elle contient une large variété de morphologies urbaines, allant de centres urbains denses, jusqu'à des quartiers résidentiels, en passant par des bordures de parcs et d'autoroutes. La ville de Mitaka est représentée sur la figure 4.1, sur laquelle les positions des carrefours contrôlés par (au moins) un feu tricolore sont représentées en rouge.

Le réseau routier sur la zone a été extrait à partir de la base de données produite par la firme *Sumitomo Electric System Solutions*, spécialisée dans la fabrication d'instruments électroniques, en particulier à destination des équipementiers automobiles. Le réseau routier utilisé est donc sensiblement équivalent à ceux utilisés par les navigateurs GPS. Les coordonnées y sont exprimées dans le système de projection cartographique UTM54 Nord.

1. Notons que l'on fait ici uniquement référence au découpage géométrique des traces GPS, et a leur affectation aux instances individuelles. La caractérisation des vecteurs $X^{(i)}$ à l'aide de descripteurs numériques ne fait pas partie du processus de fenêtrage. Le fenêtrage désigne donc la *portée spatiale* des instances individuelles, et non leur représentation mathématique.
2. C'est-à-dire générées en environnement non-contrôlé.

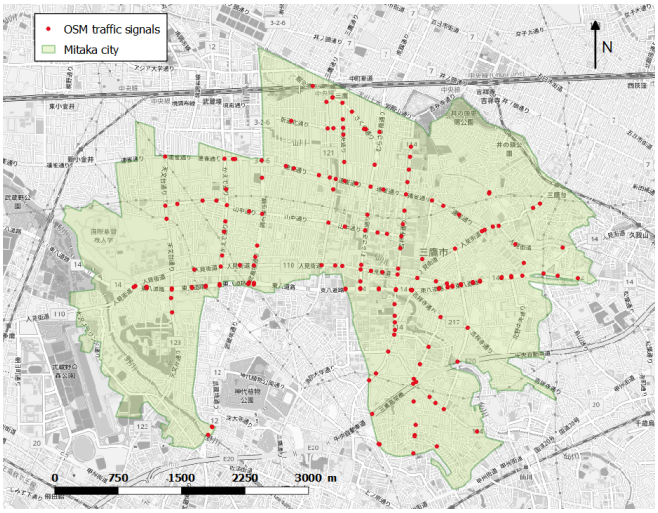


FIGURE 4.1 – Positions des carrefours contrôlés par des feux tricolores de Mitaka et référencés dans la base de données *OpenStreetMap* (OSM). Ce jeu de données étant incomplet (complétude *a posteriori* évaluée à 70 % des carrefours équipés de feux) et insuffisamment résolu, nous avons dû relever manuellement les lignes d'arrêt sur la zone d'étude.

Bien souvent, les graphes topologiques sous-jacents aux bases de données routières sont formatés de sorte qu'un nœud soit présent au niveau de chaque feu tricolore, y compris dans les cas où il n'existe pas d'intersection physique (*e.g.* lorsqu'un feu tricolore est associé à un passage piéton le long d'un axe routier). Pour cette raison, nous avons décidé de supprimer tous les nœuds de degré 2 dans le graphe, afin de pouvoir supposer que le réseau routier numérique a été créé sans connaissance des positions des feux tricolores. De plus, comme nous l'avons mentionné dans la section 2.4.3.1, cette suppression permet d'accélérer les opérations de map-matching des traces GPS sur le réseau.

4.1.2 Vérité terrain

Afin de pouvoir entraîner et valider le modèle d'apprentissage, il est nécessaire de collecter une vérité terrain, à savoir, l'ensemble des positions géoréférencées des feux tricolores de la zone d'étude. Dans l'ensemble de ce chapitre, nous appellerons *ligne d'arrêt* la position (en termes d'abscisse curviligne le long de l'axe routier) devant laquelle les véhicules doivent s'arrêter en amont d'un feu. Au Japon, il n'existe aucune ambigüité quant à la définition précise des lignes d'arrêt, ces dernières étant systématiquement matérialisées par des marquages au sol.

4.1.2.1 Acquisition

Le relevé des lignes d'arrêt a dans un premier temps été réalisé à partir de la base géographique *OpenStreetMap* (figure 4.1). Cette source de donnée n'est malheureusement pas complète. Par ailleurs, chaque point correspond à un carrefour contrôlé par un système de feux tricolores, mais aucune information n'est fournie concernant le nombre d'axes effectivement asujettis à un feu, et *a fortiori* sur les positions individuelles des lignes d'arrêts associées. Nous avons utilisé cette source comme base de référence, que nous avons complétée à l'aide de plusieurs sources d'orthoimages (prises à différentes dates) sur lesquelles

les lignes d'arrêt on été saisies manuellement.

Pour chaque ligne d'arrêt, nous avons également enregistré un attribut binaire indiquant la direction du flux de véhicules asujettis au feu. Il prend la valeur 0 si la ligne d'arrêt s'addresse aux véhicules se déplaçant depuis le nœud *initial* vers le nœud *final*, et la valeur 1 sinon (notons que les définitions des nœuds *initial* et *final* ont été arbitrairement fixées par le fournisseur de données et dépendent de l'ordre de référencement des extrémités des arêtes du graphe dans le fichier du réseau routier [3]).

La saisie manuelle des positions sur un fond de carte pose cependant un problème pratique : comment s'assurer que le jeu de vérité terrain est exhaustif ? Si on peut aisément supposer que la totalité des positions repertoriées correspondent effectivement à des feux tricolores, il est impossible de garantir que la totalité des feux présents sur le terrain ont été numérisés lors de la phase de saisie. Une base de référence incomplète, mènerait inévitablement à la double sanction d'un processus d'entraînement défaillant et d'une validation biaisée.

Pour accélérer le processus de saisie tout en limitant le risque de manquer un nombre trop important de feux tricolores, un programme informatique [4] a été crée au laboratoire en collaboration avec Marie-Dominique Van Damme. Son fonctionnemment est relativement intuitif : on génère une grille régulière de mailles carrées de dimension paramétrable sur l'ensemble de la zone. Dans le cadre de notre application, nous avons constaté que des cellules de 250 m en milieu urbain offrent un bon compromis entre rapidité d'identification des feux et nombre de cellules à traiter. En milieu péri-urbain et rural, la dimension des cellules peut aisément être doublée. Le programme permet ensuite de parcourir l'ensemble des cellules dans un ordre pré-déterminé (en ajustant le niveau de d'échelle de l'orthophotographie à la bordure de la cellule), et facilite la saisie et l'enregistrement des coordonnées des positions dans un fichier.

Ce procédé permet de parcourir l'ensemble de la zone à traiter de manière rapide, exhaustive et sans recouvrement inutile. À titre d'exemple, sur la ville de Mitaka, le paramétrage génère environ 500 cellules, chacune nécessitant en moyenne 15 secondes de traitement, soit un temps de saisie de l'ordre de 2 heures. Le même protocole a été utilisé pour numériser les feux tricolores de la ville de Tsukuba, dont l'emprise est beaucoup plus grande [5], générant environ 2000 cellules, pour un temps de collecte de l'ordre de 10 heures. Le code développé est aisément personnalisable pour d'autres types de saisies de données ponctuelles (*e.g.* passages piétons, centres de bâtiments, bouches de métro... *cf* annexe C exposant les produits informatiques opérationnels développés au cours de la thèse).

Pour clore cette phase d'acquisition de la vérité terrain, chaque position relevée a été projetée orthogonalement sur l'axe routier le plus proche, comme illustré sur la figure 4.2. Les positions (x, y) des feux sont alors ré-exprimées dans un système propre au réseau routier de référence : $(e, s) \in \mathbb{N} \times \mathbb{R}^+$, où e désigne l'indice de l'arc sur lequel a été projetée la position, et s est l'abscisse curviligne (le long de e et dans la direction du nœud final) de

3. Nous avons mis en évidence à l'aide d'analyses statistiques que ces définitions ne sont en réalité pas complètement arbitraires : avec notre réseau de référence, une position de feu a significativement plus de chances d'être étiquetée 1 que 0, confirmant ainsi que le formattage du réseau routier contient des informations implicites liées à la présence (ou à l'absence) des feux. Notons que cet attribut sera définitivement perdu par la suite lors de la construction des instances.

4. `https://mdvandamme.github.io/PoussePousseEditData/`

5. De l'ordre de 250 km², soit la superficie du Val-de-Marne.

la position.

À l'issue de cette étape un total de 669 lignes d'arrêt ont été numérisées sur Mitaka, réparties sur 253 carrefours, soit une moyenne de 2.6 feux tricolores par carrefour contrôlé. Parmi eux, 177 (70%) étaient initialement présents dans la base OSM.

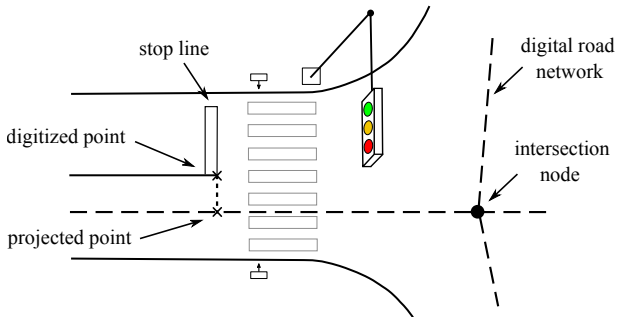


FIGURE 4.2 – Processus d'acquisition et de contrôle de la vérité terrain à partir de l'orthoimagerie et d'un réseau routier de référence.

4.1.2.2 Contrôle

Deux opérations spécifiques ont été effectuées pour contrôler la qualité de saisie :

- **Qualité relative :** la numérisation d'un grand nombre de positions peut conduire à des erreurs (notamment des oublis et des imprécisions géométriques dans les saisies). L'étape de contrôle de la qualité relative a pour objectif de quantifier ces erreurs. À cette fin, le programme présenté ci-dessus a été étendu pour pouvoir échantillonner aléatoirement et uniformément un nombre réduit (typiquement une cinquantaine) de cellules. La saisie manuelle a été effectuée à nouveau sur chacune des cellules de cet échantillon, avec un temps moyen passé par cellules en moyenne trois fois plus long que lors du premier passage. Le programme compare alors ces nouvelles données aux anciennes, sur le plan de l'exhaustivité et de la précision de positionnement. Les techniques classiques d'estimation globale issues du domaine de la Géostatistique (Arnaud et Emery, 2000) permettent d'étendre les résultats obtenus sur la population complète de feux tricolores. Dans notre cas d'étude sur Mitaka, nous obtenons un taux de complétude de 99% et une précision métrique de saisie de l'ordre de 50 cm.

- **Qualité absolue :** cette phase a pour but de contrôler les données au regard de la vérité du terrain. Au total, 30 positions de lignes d'arrêt ont été échantillonnées aléatoirement puis géolocalisées sur site avec un récepteur GPS mono-fréquence à bas coût. L'écart quadratique moyen ρ^2 a été mesuré entre les positions saisies sur les orthophotos et les positions observées par GPS. On peut montrer qu'avec un niveau

de confiance $c \in [0, 1]$, l'erreur σ de saisie sur les orthophotos est contrainte par l'inégalité suivante :

$$\sigma \leqslant \rho\Big(1 + \frac{\Phi^{-1}(c)}{\sqrt{n}}\Big)^{\frac{1}{2}}, \tag{4.1}$$

où Φ est la fonction de répartition de la loi normale standard, et n est le nombre de points GPS observés (*cf* annexe A).

Ce contrôle additionnel a permis de garantir (avec un indice de confiance à 95%) que les positions des lignes ont été saisies dans la base de référence avec une précision sub-métrique. Cette phase a également permis de tester les capacités du récepteur en environnement urbain dense, en mode multi-constellation, et a fait l'objet d'une publication (Méneroux et al., 2017) que nous reproduisons en annexe A.

4.1.3 Données FCD

Pour cette expérimentation, nous avons utilisé les données FCD fournies par NAVITIME JAPAN [6], une compagnie privée développant des technologies de navigation et fournissant des applications et des services web permettant d'assister les personnes dans leur mobilité.

Le jeu de données couvre l'intégralité du territoire japonais, et a été collecté sur une période d'un mois, en octobre 2015. Les trajectoires issues de piétons ont été préalablement retirées, de sorte que le jeu de données utilisé ne contient que des traces issues de véhicules. Nous avons extrait tous les enregistrements GPS intersectant l'emprise géographique de Mitaka. Chaque enregistrement (nominalement 1 par seconde) contient les informations suivantes : un identifiant de véhicule, un identifiant de route, les coordonnées géographiques du point observé et un timestamp. Une route est définie comme une morceau de trajectoire correspondant à une session du récepteur GPS (*i.e.* entre la mise en route et la mise à l'arrêt du récepteur). Pour des raisons de protection de la vie privée, l'identifiant de véhicule est modifié aléatoirement tous les jours à minuit. Les lignes comportant le code -1 dans la colonne des timestamps ou des coordonnées (environ 2% des enregistrements, correspondant a des pertes de signaux GPS ou des problèmes dans la sauvegarde des données) ont été retirées. Les coordonnées des enregistrements restants ont été converties dans le système de projection UTM 54N. Par simplicité, les timestamps ont été convertis en un nombre entier d'époques (exprimées en secondes écoulées depuis une date arbitraire). On donne en figure 4.3 une illustration du résultat des opérations de nettoyage préliminaires.

Le recalage des traces sur le réseau a été effectué par l'algorithme de Newson et Krumm (2009), avec les améliorations exposées dans la section 2.4. Puisque toutes les traces sont situées sur la même zone, il est bénéfique de calculer les distances les plus courtes entre chaque couple de nœuds une seule fois, puis de stocker les résultats dans une table de recherche, avant de recaler toutes les trajectoires. Cette approche a permis d'accélérer le processus et d'atteindre un rythme de 10 traces par seconde (environ 1500 fois plus rapide que la solution naïve qui nécessite de calculer les plus courts chemins au fur et à mesure

6. http ://corporate.navitime.co.jp/en

FIGURE 4.3 – Trajectoires GPS fournies par la compagnie Navitime sur le sud-est de la ville de Mitaka après les opérations de nettoyage préliminaires. Représentation : *Mobmap* - `https://shiba.iis.u-tokyo.ac.jp/member/ueyama/mm/`.

qu'ils sont requêtés). Cependant, pour un réseau routier contenant un nombre n de nœuds, puisque les complexités temporelles et spatiales du calcul de la table de recherche augmentent comme $\mathcal{O}(n^2)$, il devient inévitable de trouver des solutions alternatives lorsque la zone d'intérêt est large. L'une d'entre elles pourrait être d'utiliser la notation matricielle creuse avec une structure de données de type *hashtable*, et de n'enregistrer que les distances plus courtes qu'un seuil prédéfini. Si le temps de création de la table de pré-calcul est lui-même prohibitif, on peut accélérer encore le processus en remplaçant l'algorithme de Dijkstra (traditionnellement utilisé pour les calculs de plus courts chemins) par l'algorithme Dijkstra bi-directionnel (Fu et al., 2003), par l'heuristique [7] A^* (Hart et al., 1968b), ou encore à l'aide de contractions hiérarchiques (Geisberger et al., 2008).

Parallèlement au recalage des trajectoires GPS, on constitue un index spatial sur le réseau routier. Plus spécifiquement : pour chaque arc du graphe routier, on enregistre une liste d'identifiants correspondants aux points observés dans le jeu de données sur l'arc correspondant. Pour un arc i donné, la liste associée se présente sous le format : $L_i = \{a_1, d_1, f_1, a_2, d_2, f_2, ...\}$ avec a_j le numéro d'une trace GPS ayant parcouru le tronçon i, d_j et f_j les indices du premier et dernier point de la trace a_j effectivement situés sur le tronçon i. Notons que L_i peut contenir plusieurs fois le même identifiant a_j (si une même trace parcourt le tronçon i à de multiples reprises). Cet index spatial L (que l'on peut voir comme un map-matching inversé, *i.e.* du réseau vers les traces) va permettre d'accélérer significativement toutes les opérations de requête dans la base pour la construction des instances.

La racine carrée de l'erreur quadratique moyenne des déplacements induits par le recalage est égale à 8.3 m, ce qui donne une idée de la qualité moyenne des récepteurs GPS. Dans l'ensemble, 99 % des points ont été recalés sur le réseau. Finalement, nous avons supprimé toutes les traces recalées sur l'autoroute Chūō, qui longe la partie sud-est de Mitaka et qui ne comporte aucun feu tricolore.

À la fin de la phase de prétraitement, il reste un total de 11 870 trajectoires dans le jeu de données, ce qui représente un peu plus de 7 millions d'enregistrements, environ 42 000 km et 3 122 heures cumulées de données de conduite. Le trajet médian est de 3 km et dure 10 minutes. 95% des trajectoires ont été enregistrées à une fréquence supérieure à 0.2 Hz.

7. Exacte sous certaines conditions qui sont vérifiées ici dans notre cadre d'étude.

4.2 Construction des instances

Dans cette section, nous décrivons notre méthodologie pour construire les instances à partir des trajectoires GPS.

4.2.1 Définition spatiale des instances

Dans la plupart des problèmes d'apprentissage, il existe une définition naturelle d'une instance. Par exemple, dans les tâches de reconnaissance d'images, chaque image individuelle est une instance, et nous pouvons facilement supposer qu'elles sont indépendantes les unes des autres. Dans notre cas d'application, il n'y a pas de telle définition, puisque nous recherchons des objets situés à des endroits inconnus sur un réseau topologique. Cependant, considérant que la plupart des feux de circulation sont situés près des intersections, nous avons choisi de placer les instances sur des segments de route en partant des nœuds.

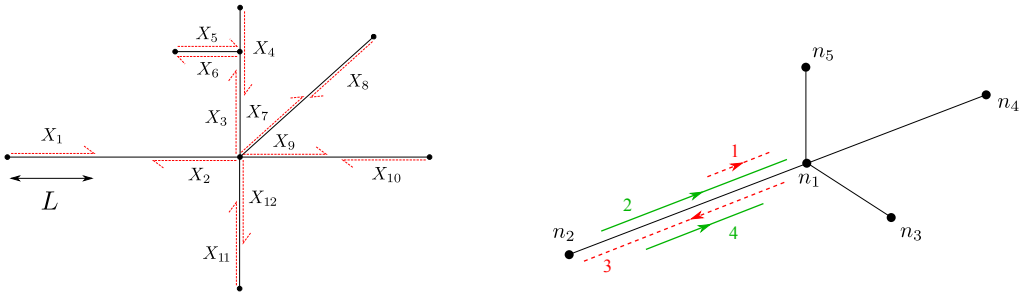


FIGURE 4.4 – Gauche : génération de trames (flèches rouges en pointillés) sur le réseau routier. Chaque trame est calculée à partir des traces GPS se déplaçant vers le nœud d'intersection (*i.e.* dans la direction opposée aux flèches). À droite : sélection des traces (voir le texte pour plus de détails).

Ce choix a été motivé par le fait qu'il résulte en des instances séparées, facilitant ainsi la division du jeu de données entre l'entraînement et la validation. En contre-partie, cet algorithme ne parviendra pas à détecter les feux de circulation situés à grande distance des intersections routières. Étant donné que l'on peut supposer que ce cas de figure ne représente qu'une faible proportion de tous les feux, nous pensons que ce choix n'aura pas trop d'impact négatif. Le nombre total d'instances à générés est au plus égal à la somme des degrés des sommets du graphe, soit (d'après le lemme des *poignées de main*) le double du nombre d'arêtes du réseau routier (certaines arêtes peuvent n'être parcourues par aucune trace, de sorte que ce nombre est un majorant du nombre effectif d'instances générées). Nous désignerons ces segments de route par le terme *fenêtre* ci-après.

Afin d'obtenir des instances homogènes, les fenêtres ont été fixées à une longueur constante L. Si une arête est plus longue que L, seule une partie de longueur L (à partir du nœud) est considérée. À l'inverse, si elle est plus courte que L, la trame est complétée avec des zéros (*e.g.* X_5 et X_6 sur la figure 4.4 à gauche). La valeur numérique de L a été fixée à 100

m, en faisant l'hypothèse raisonnable que les événements situés à plus de 100 m d'un feu tricolore ne sont pas utiles pour la détection.

Seules les traces GPS se déplaçant *globalement* vers le nœud sont ajoutées à la fenêtre. Plus formellement, le dernier point GPS d'une trajectoire de la fenêtre doit être situé plus près du nœud d'intersection que le premier point (par rapport à une mesure de distance calculée comme l'abscisse curviligne le long de la géométrie de l'arête). De plus, nous avons ajouté la contrainte que la distance entre ces deux enregistrements extrêmes soit au moins égale à la moitié de la longueur totale de l'arête. Par exemple, dans la partie droite de la figure 4.4, seules les traces 2 et 4 (lignes pleines) sont prises en compte dans la fenêtre générée à partir de l'intersection n_1 (la trace 1 est trop courte, tandis que la trace 3 se déplace dans le sens inverse au sens requis). Pour l'instance générée à partir du nœud n_2, les traces 1, 2 et 4 sont supprimées.

4.2.2 Calcul des variables explicatives

Une fois que les traces se déplaçant vers le nœud donné ont été identifiées, on peut en extraire des séquences de points correspondants à des arrêts de véhicules.

Définition 4.1 (séquence d'arrêt). *Étant donnés une séquence de points GPS horodatés et deux paramètres : une valeur de vitesse maximale $v_{max} \in \mathbb{R}^+$ et une durée minimale $\tau_{min} \in \mathbb{R}^{+*}$, nous définissons une séquence d'arrêt comme une suite d'enregistrements consécutifs $S = \{(x_i, t_i) \mid p \leqslant i \leqslant q\}$ qui vérifie les deux inégalités suivantes :*

$$t_q - t_p \geqslant \tau_{min} \qquad \text{et} \qquad \forall\, i \in [\![p, q-1]\!] \quad \frac{|x_{i+1} - x_i|}{t_{i+1} - t_i} \leqslant v_{max}, \tag{4.2}$$

où x est l'abscisse curviligne des enregistrements GPS le long de l'arrête. De plus, p et q sont choisis de manière à ce qu'il soit impossible d'ajouter de nouveaux enregistrements à la séquence sans violer les inégalités mentionnées ci-dessus.

Grossièrement, une partie (maximale) de la trajectoire est qualifiée de séquence d'arrêt si elle correspond à un déplacement suffisamment lent du véhicule sur un intervalle de temps suffisamment long.

Définition 4.2 (point d'arrêt). *Pour une séquence d'arrêt donnée, un point d'arrêt est défini comme la position moyenne des points de la séquence, associée à la durée totale de l'arrêt :*

$$sp = \Big(\frac{1}{|S|}\sum_{i=p}^{q} x_i,\ \frac{1}{|S|}\sum_{i=p}^{q} y_i,\ t_q - t_p\Big). \tag{4.3}$$

Pour chaque instance, les points d'arrêt (lorsqu'ils existent) ont été extraits des traces sélectionnées, selon les définitions 4.1 et 4.2 avec les paramètres suivants : $v_{max} = 0.5\ m.s^{-1}$ et $\tau_{min} = 5$ sec.

Puisque le nombre de points d'arrêt est indéterminé a priori, il est impossible d'entraîner un classifieur avec un nombre prédéfini de points d'arrêt. En effet, cette solution implique-

rait fatalement qu'aucune prédiction ne puisse être faite sur des fenêtres en contenant un nombre insuffisant. Inversement, s'il y a trop de points d'arrêt sur une instance donnée, il n'y aurait pas d'autre solution que d'en sélectionner aléatoirement le nombre approprié pour l'adapter au modèle de classifieur. Une solution classique à ce problème consiste à estimer la distribution des durées d'arrêt le long de l'abscisse curviligne de la route avec la méthode des noyaux (KDE), qui remonte aux travaux de Rosenblatt (1956) et Parzen (1962). Dans notre cadre d'étude, nous utilisons une version adaptée du KDE pour pouvoir prendre en compte les dimensions à la fois spatiales et temporelles du phénomène.

On note $K : \mathbb{R} \to \mathbb{R}^+$ une fonction noyau, c'est-à-dire symétrique, à valeurs positives et d'intégrale 1 sur l'ensemble des réels. De plus, on impose que K soit du second ordre, *i.e.* $\int_{\mathbb{R}} u^2 K(u)du < \infty$, et on note κ la quantité correspondante.

Soient $x_1, x_2, ...x_N \in [0, L]$ un ensemble de N points d'arrêt, associés à des temps d'arrêt $t_1, t_2, ...t_N \in \mathbb{R}^+$ (pour des raisons d'efficacité numérique, nous supposerons plus loin que les timestamps sont précis à la seconde près, permettant ainsi de considérer que $t_i \in \mathbb{N}$). Pour un facteur d'échelle $h \in \mathbb{R}^+$, on note K_h la transformée de la fonction K par une homothétie de rapport h :

$$K_h(u) = \frac{1}{h} K\Big(\frac{u}{h}\Big). \tag{4.4}$$

Soit $(X, T) \in \mathbb{R} \times \mathbb{R}^+$ un couple de variables aléatoires de loi jointe $\pi(x, t)$, représentant respectivement la position et le temps d'un arrêt. On note $\pi(x) = \int_{\mathbb{R}^+} \pi(x, t)dt$ la distribution marginale des positions d'arrêt et $\pi(t|x) = \pi(x, t)/\pi(x)$ la distribution conditionnelle des temps d'arrêt sachant la position. Intuitivement, on a tendance à penser que $\pi(t|x) \neq \pi(t)$, *i.e.* que les v.a. X et T ne sont pas indépendantes [8]. On définit alors $\hat{f}_h(x)$, l'estimateur de la densité par noyaux *pondérés* par :

$$\forall\, x \in [0, L] : \qquad \hat{f}_h(x) = \frac{1}{N} \sum_{i=1}^{N} t_i K_h(x - x_i). \tag{4.5}$$

Notons qu'il s'agit d'une légère adaptation par rapport à la méthode KDE standard, puisqu'ici, chaque application du noyau en x_i est pondérée par le temps d'arrêt t_i associé. En conséquence, l'estimation $\hat{f}_h$ n'est pas normalisée et on a :

$$\begin{aligned} \int_{-\infty}^{+\infty} \hat{f}_h(x)dx &= \frac{1}{Nh} \sum_{i=1}^{N} t_i \int_{-\infty}^{+\infty} K_h(x - x_i)dx = \frac{1}{N} \sum_{i=1}^{N} t_i \int_{-\infty}^{+\infty} K(u)du \\ &= \frac{1}{N} \sum_{i=1}^{N} t_i \underset{N\to\infty}{\longrightarrow} \mathbb{E}[T], \end{aligned} \tag{4.6}$$

où la seconde égalité se déduit du changement de variable $u = (x - x_i)/h$. Sous l'hypothèse (raisonnable) que le facteur d'échelle h est petit devant la taille L de la fenêtre, et

8. On peut raisonnablement penser que plus un véhicule stationne à une abscisse proche du feu tricolore, plus son temps d'arrêt est long en moyenne.

que la plupart des points d'arrêts observés ne sont pas voisins des bordures, on peut écrire :

$$\int_0^L \hat{f}_h(x)dx \approx \int_{-\infty}^{+\infty} \hat{f}_h(x)dx = \mathbb{E}[T]. \tag{4.7}$$

Autrement dit, l'aire sous la courbe de l'estimateur $\hat{f}$ sur la fenêtre $[0, L]$ est approximativement égale à l'espérance des temps d'arrêt. De plus, on a :

$$\begin{aligned}\mathbb{E}[\hat{f}_h(y)] &= \int_{-\infty}^{+\infty}\int_0^{+\infty} tK_h(y-x)\pi(x,t)dxdt = \int_{-\infty}^{+\infty}\int_0^{+\infty} tK_h(y-x)\pi(t|x)\pi(x)dxdt \\ &= \int_{-\infty}^{+\infty} K_h(y-x)\Big[\int_0^{+\infty} t\pi(t|x)dt\Big]\pi(x)dx = \int_{-\infty}^{+\infty} K_h(y-x)\mathbb{E}[T|x]\pi(x)dx.\end{aligned} \tag{4.8}$$

En posant $f(x) = \mathbb{E}[T|x]\pi(x)$, et à l'aide d'un développement limité de f à l'ordre de 2 (en utilisant les propriétés intégrales du noyau), on obtient :

$$\mathbb{E}[\hat{f}_h(y)] = \int_{-\infty}^{+\infty} K_h(y-x)f(x)dx = f(x) + \frac{1}{2}f''(x)h^2\kappa^2, \tag{4.9}$$

où κ désigne la variance du noyau, et où le second terme tend vers 0 quand h tend vers 0. L'équation 4.8 montre donc que la fonction calculée par notre estimateur 4.5 est :

$$f(x) = \mathbb{E}[T|x]\pi(x). \tag{4.10}$$

Les calculs sont détaillés dans l'annexe B. En particulier, on vérifie l'égalité intégrale 4.7 de manière théorique en partant de 4.10 : $\int_{\mathbb{R}} f_h(x)dx \approx \int_{\mathbb{R}} \mathbb{E}[T|x]\pi(x)dx = \int_{\mathbb{R}}\int_{\mathbb{R}^+} t\pi(t|x)\pi(x)$ $dxdt = \iint_\Omega t\pi(x,t)d\omega = \mathbb{E}[T]$.

On montre également une propriété similaire à un niveau plus local : exprimons le ratio entre l'aire sous la courbe entre 2 points $a, b \in [0, L]$ (en théorie $\in \mathbb{R}$) et l'aire totale (en négligeant les effets de coupure liés aux bords) :

$$\begin{aligned}\frac{\int_a^b f(x)dx}{\int_{\mathbb{R}} f(x)dx} &= \frac{\int_a^b\int_{\mathbb{R}+} t\pi(t|x)\pi(x)dtdx}{\int_{\mathbb{R}}\int_{\mathbb{R}+} t\pi(t|x)\pi(x)dtdx} = \frac{\mathbb{E}[T.\mathbb{1}_{\{x\in[a,b]\}}]}{\mathbb{E}[T]} = \frac{\mathbb{E}[T|x\in[a,b]]\mathbb{P}[x\in[a,b]]}{\mathbb{E}[T]} \\ &= \frac{n_{[a,b]}/N_{[a,b]}}{n_{\mathbb{R}}/N_{\mathbb{R}}}\frac{N_{[a,b]}}{N_{\mathbb{R}}} = \frac{n_{[a,b]}}{n_{\mathbb{R}}},\end{aligned} \tag{4.11}$$

où, pour tout intervalle $A \subseteq \mathbb{R}$, n_A et N_A sont les variables de comptage représentant respectivement les nombres d'enregistrements GPS et de points d'arrêt sur A. Autrement dit,

l'aire sous la courbe sur tout intervalle est proportionnelle au temps passé par les véhicules à l'arrêt sur ce même intervalle, comme illustré sur la figure 4.5 :

$$\frac{\mathcal{A}_{[a,b]}}{\mathcal{A}_{[0,L]}} \approx \frac{\sum_i t_i \in \{t_i \mid x_i \in [a,b]\}}{\sum_i t_i}. \tag{4.12}$$

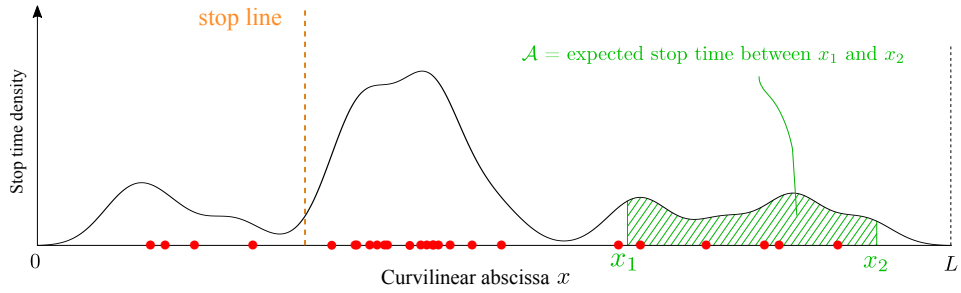


FIGURE 4.5 – Distribution des temps d'arrêt. La ligne verticale orange en pointillé représente la position de la ligne d'arrêt associée au feu tricolore (contrôlant le flux de vehicules sur l'entrée d'un carrefour situé sur la bordure gauche du graphique). Les véhicules se déplacent de la droite vers la gauche.

D'un point de vue pratique, moyennant l'hypothèse que $t_i \in \mathbb{N}$ (si les timestamps sont exprimés en nombres entiers de secondes), le calcul peut-être effectué à l'aide d'une librairie classique de KDE, simplement en *dépliant* les temps d'arrêt, *i.e.* en répliquant chaque point d'arrêt x_i un nombre t_i de fois. Sous certaines conditions de régularité sur le noyau et la densité à estimer, Nadaraya (1965) a démontré la convergence uniforme (presque sûre) de $\hat{f}_h$ vers f à mesure que le nombre de données observées augmente.

Bien qu'il puisse être démontré que l'intégrale de l'erreur quadratique moyenne est minimale avec un noyau d'Epanechnikov (voir Hansen, 2009 par exemple), le choix de la fonction noyau n'est pas critique. C'est pourquoi nous avons utilisé un noyau gaussien. Le paramètre de largeur de bande a été défini indépendamment pour chaque instance, selon la règle de Silverman (Silverman, 1986) : $h_s = 1.06\hat{\sigma}n^{-1/5}$, où $\hat{\sigma}$ désigne l'écart-type empirique des données. En pratique, nous avons remarqué que cette solution résultait en des estimateurs trop lissés (*cf* illustration 4.6 à gauche). Une explication à ce phénomène provient du fait que la règle de Silverman est optimale pour des observations distribuées suivant une loi normale. En théorie, la largeur de bande optimale est inversement proportionnelle à l'intégrale du carré de la dérivée seconde de la densité à estimer. Pour prendre en compte le fait que les densités à estimer dans notre cas d'étude sont vraisemblablement moins régulières qu'une gaussienne, nous avons sélectionné le critère $h^* = 0.75h_s$, comme illustré au centre de la figure 4.6. Notons qu'il existe d'autres solution pour calibrer l'échelle du noyau, en particulier à l'aide de méthodes plug-in (on estime la courbure de f à l'aide d'une première estimation), de bootstrap ou de validation croisées (voir Zambom et Dias, 2012 par exemple). Par souci de simplicité, et pour éviter d'être trop sensible aux données dans un contexte d'apprentissage, nous n'avons pas souhaité mettre en œuvre ces méthodes.

Suivant une méthodologie inspirée de el Habib Boukhobza et Mimi (2016), la fonction $\hat{f}_{h^*}$ résultante est échantillonnée en p emplacements régulièrement espacés pour former le

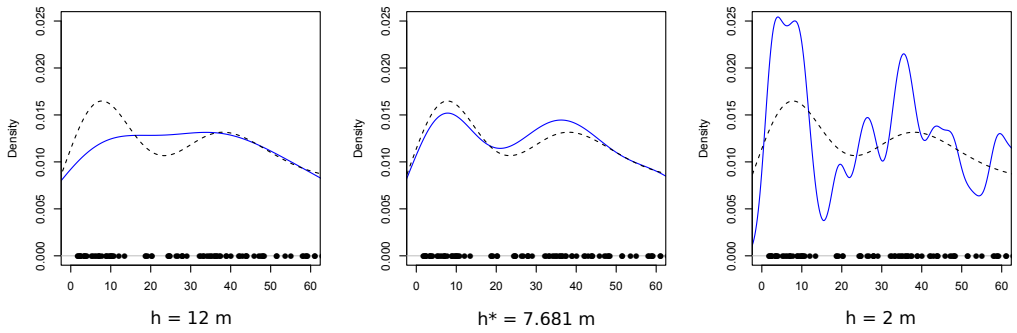


FIGURE 4.6 – Tests d'estimation par noyaux avec un jeu de points d'arrêt sous-échantillonné à 50%. Fonction à estimer (en pointillés), estimation par noyaux (en bleu) pour différentes valeurs de h. De gauche à droite : 12 m, 7.68 m (valeure optimale définie par le critère de sélection) et 2 m. On retrouve un compromis biais-variance similaire à 3.2.1.6.

vecteur de variables explicatives $X \in \mathbb{R}^p$, comme illustré sur la figure 4.7. De plus, étant donné que dans les implémentations efficaces de KDE, le calcul se fait avec l'algorithme de transformation de Fourier rapide, fondé sur le principe algorithmique *diviser pour régner*, il est logique de définir la valeur numérique de p comme une puissance de 2. Dans notre cas, nous avons pris $p = 64$.

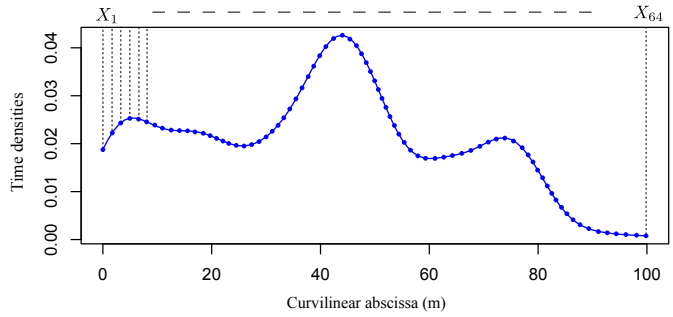


FIGURE 4.7 – Distribution des temps d'arrêt et échantillonnage régulier en 64 points : $X = (X_1, X_2, ...X_{64})$. Les véhicules se déplacent de la droite vers la gauche.

Enfin, les variables cibles sont calculées. La variable de classification binaire $Y_1 \in \{0, 1\}$ indique la présence d'un feu dans l'instance. Si $Y_1 = 1$, la variable de régression $Y_2 \in [0, L]$ spécifie l'emplacement de la ligne d'arrêt, mesuré par sa distance au nœud d'intersection (abscisse de la ligne d'arrêt sur la figure 4.5).

4.2.3 Apprentissage

L'ensemble de données final contient 4611 instances, incluant 662 (14 %) échantillons positifs. Ce déséquilibre important en faveur des instances négatives peut significative-

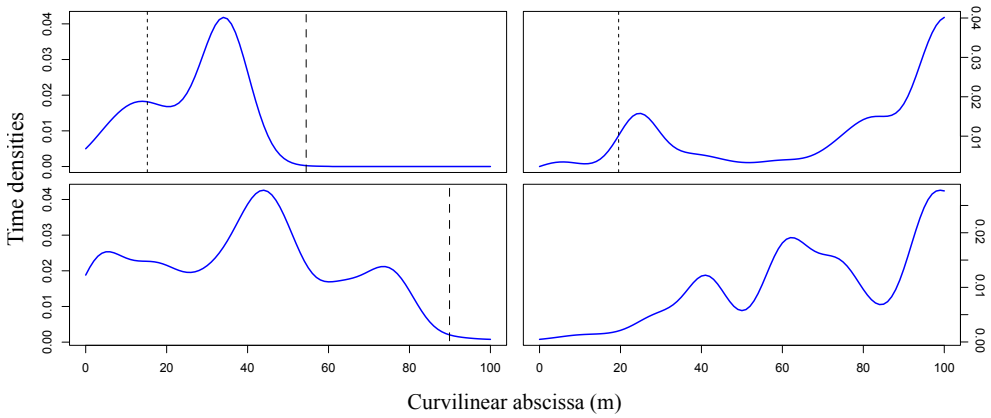


FIGURE 4.8 – Exemples de distributions des temps d'arrêt : les deux instances supérieures sont positives (la ligne en pointillés courts indique la position recherchée), tandis que les deux instances inférieures sont négatives. Lorsque le tronçon de route est plus court que 100 m, la ligne en pointillés longs indique l'extrémité du segment de bord.

ment pénaliser le processus d'entraînement (Batista et al., 2004). Pour contourner ce problème, nous avons évalué différentes stratégies : sous-échantillonnage (suppression aléatoire des échantillons négatifs jusqu'à ce que l'ensemble de données soit équilibré) et sur-échantillonnage (réplication des échantillons positifs : cette seconde stratégie a l'avantage de conserver toutes les informations disponibles des données, au détriment d'une augmentation de la corrélation entre les instances individuelles). Nous avons également appliqué l'algorithme SMOTE (Bowyer et al., 2011), qui est similaire au sur-échantillonnage, mais plutôt que de répliquer les exemples des classes minoritaires, de nouveaux exemples sont générés par interpolation entre des instances voisines de cette classe échantillonnées de manière aléatoire. L'apprentissage a été effectué avec l'algorithme des forêts aléatoires, en mode classification (pour la détection du feu) et régression (pour l'inférence de sa position). Nous avons construit $T = 500$ arbres, et à chaque split $\sqrt{n} = 8$ descripteurs sont pris en compte (*cf* paragraphe 3.2.1.5). Le modèle a été évalué par 10-fold cross validation.

4.3 Résultats et discussion

L'ensemble du processus expérimental a été implémenté en R avec le package *randomForest* (Liaw et al., 2002) et lancé sur un processeur Intel Core(TM) i7-3770 (3.40 GHz RAM 8 Go). Nous avons calculé les scores de performance suivants : spécificité, sensibilité, aire sous la courbe ROC (AUC), temps d'entraînement (pour un seul *fold*, c'est-à-dire sur 90% des données) et accuracy [9] (*cf* paragraphe 3.4).

4.3.1 Analyse des résultats

D'après la table 4.1, nous observons que, comme prévu, la complexité temporelle du processus de formation est à peu près proportionnelle au nombre d'échantillons d'entraînement.

9. On conserve ici le terme anglais *accuracy* pour éviter les confusions avec la *précision ppv* (positive predictive value), également appelée *précision utilisateur*.

Par ailleurs, l'aire sous la courbe (et donc la performance globale) ne semble pas dépendre de la méthode choisie pour équilibrer les données. Tout se passe comme si les quatre classificateurs ci-dessus correspondaient à des seuils de sélection différents du même modèle de classifieur, comme illustré sur la figure 4.9, dont les 4 points sont inclus dans la bande de confiance de la courbe ROC représentée en figure 4.10. Par conséquent, dans le reste de cette section, nous n'utiliserons que le sous-échantillonnage, car il réduit le nombre d'instances à traiter, ce qui donne un temps de calcul minimal. Il convient de noter que si l'estimation OOB est souvent reconnue comme étant assez fiable, elle ne fournit pas une estimation d'erreur réaliste sur les cas de sur-échantillonnage (brut et SMOTE). Cela peut s'expliquer par le fait qu'avec ces deux procédures d'équilibrage, deux données identiques (ou du moins très similaires) peuvent se trouver à la fois dans et en dehors de l'échantillon OOB, ce qui revient à valider un modèle avec des échantillons partiellement contenus dans le jeu d'entraînement.

Scores	Down-sampling	Up-sampling	Brut	SMOTE
Specificité (%)	87.10	95.97	97.23	95.87
Sensitivité (%)	83.25	63.34	57.18	63.98
Accuracy (%)	86.57	91.49	91.73	91.49
AUC (%)	91.38	91.52	91.26	91.25
Temps d'entraînement (s)	1.35	6.98	3.83	7.18
Nombre d'instances	1191	7108	4149	7108
Erreur OOB (%)	14.46	2.00	8.23	2.36

TABLE 4.1 – Performances de détection pour differents modes d'équilibrage des données.

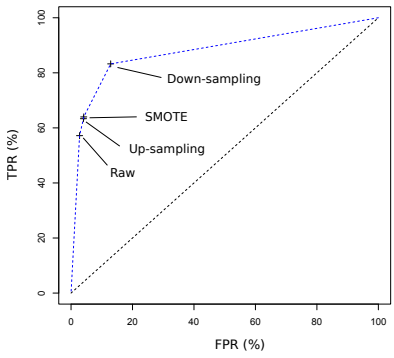


FIGURE 4.9 – Performances dans l'espace ROC des 4 versions du classifieur en fonction de la stratégie d'équilibrage des classes d'instances.

La figure 4.10 illustre les performances de détection et de localisation pour la version à sous-échantillonnage de l'algorithme. L'indice d'aire sous la courbe du classificateur est égal à 91.8 (±1.5) % ce qui peut-être considéré comme un résultat satisfaisant. Bien que la spécificité ne soit pas si élevée (comparée au nombre de faux positifs potentiels qui pourraient être détectés sur un réseau routier typique), la courbe ROC reste proche de la ligne verticale *no false positive* y compris pour une valeur acceptable du taux de vrais positifs. Cette observation permet d'imaginer la possibilité de construire un processus semi-automatique, permettant d'obtenir un rappel (*cf* paragraphe 3.4) satisfaisant et de n'exiger que peu

de corrections manuelles. Cependant, de l'autre côté de la courbe ROC, il semble difficile d'obtenir tous les feux de circulation, sans passer beaucoup de temps à séparer les vrais et les faux positifs. D'un point de vue plus pratique, il convient également de noter que notre rappel peut être comparé aux 70% de la base OSM (avec l'avantage substantiel que notre algorithme effectue la détection sur chaque feu individuel, et non pas sur les carrefours), ce qui constitue un résultat intéressant, au regard de la qualité des données passées en entrée.

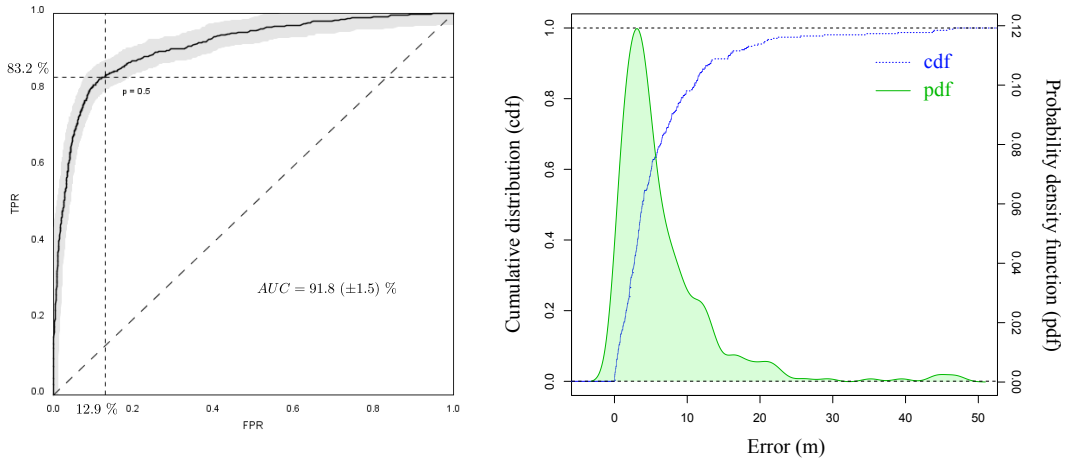


FIGURE 4.10 – À gauche : courbe ROC du classifieur avec bandes de confiance à 95% (calculées par bootstrap). À droite : densité de probabilité et fonction de répartition des erreurs de régression.

Scores	Erreur moyenne	Erreur médiane	Mode des erreurs	RMSE
Estimation (m)	6.22	3.82	2.65	9.51
Ecart-type (m)	±0.4	±0.3	±0.3	±0.8

TABLE 4.2 – Performances de localisation. RMSE : root mean square error.

En outre, comme le montre la fonction de répartition des erreurs de régression, 82 % des erreurs sont inférieures à 10 m, 60 % inférieures à 5 m et 14 % sont sub-métriques. L'erreur quadratique moyenne $\mathbb{E}[(Y_2 - \widehat{Y}_2)^2|X]$ est égale à 9.51 (± 0.8) m, (qui doit être mis en perspective avec l'écart-type *a priori* $\mathbb{E}[(Y_2 - \widehat{Y}_2)^2] = 19.54$ m de la variable expliquée avant régression), alors que les valeurs moyenne, médiane et modale sont beaucoup plus faibles, ce qui indique que la distribution est sensiblement déséquilibrée vers la droite. Cela appelle une discussion plus générale sur ce que signifie "détection". Il pourrait être plus raisonnable de considérer les valeurs aberrantes comme non détectées (une ligne d'arrêt détectée avec une imprécision de 50 m ne peut être légitimement considérée comme détectée), par conséquent, le rappel diminuerait légèrement de 4 % et en retour, le RMSE de localisation tomberait à 6 m et l'erreur moyenne à 4 m.

Comme la plupart des algorithmes d'ensemble, les RF sont robustes au sur-apprentissage, et bien qu'il n'existe pas de lignes directrices pour sélectionner un nombre adéquat d'arbres, il est admis qu'un nombre excessif ne nuit pas à la prédiction (au détriment d'une charge supplémentaire en temps de calcul durant les phases d'entraînement et d'inférence), comme

illustré sur la figure 3.13. La figure 4.11 montre l'évolution de l'estimation de l'erreur OOB à mesure que les arbres sont construits dans le modèle. On peut observer que la convergence des prédictions a été atteinte avec une centaine d'arbres.

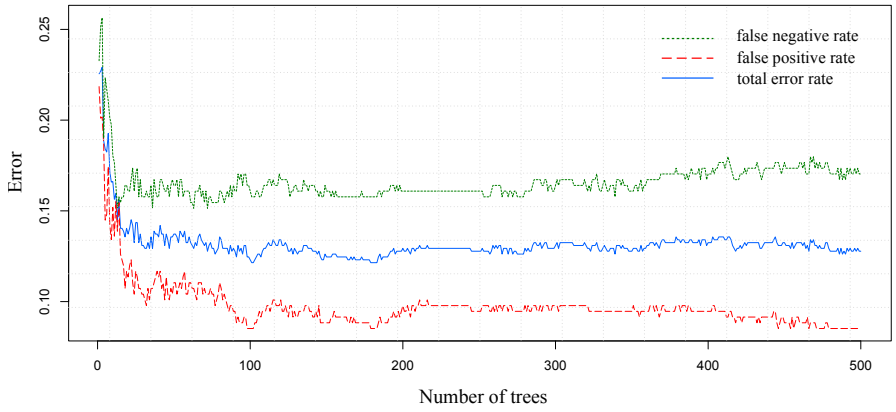


FIGURE 4.11 – Convergence de l'estimateur OOB de l'erreur avec le nombre d'arbres du modèle.

L'examen détaillé des résultats a révélé que de nombreuses fausses détections se sont produites à des endroits où très peu de véhicules circulaient, ce qui implique que l'algorithme n'a pas atteint la convergence en ce qui concerne le nombre de véhicules. Nous traiterons cette question dans la section 4.4.2.1.

4.3.2 Perspectives d'améliorations

Une limitation importante de notre travail est que, comme indiqué dans la section 4.2, notre choix de fenêtre, situé près du nœud d'intersection, rend impossible la détection des feux tricolores situés au milieu des bords. En effet, un nombre relativement important d'erreurs s'est produit sur les feux de circulation activés par les piétons à l'aide d'un bouton-poussoir. Une proposition intéressante pour résoudre ce problème serait de sur-échantillonner le réseau en créant des nœuds artificiels espacés de façon régulière sur les axes routiers dont la longueur est supérieure à la limite L fixée par le paramétrage de l'algorithme. Cette approche peut être efficace pour détecter les feux tricolores restants. Une autre limite importante provient du fait que seule l'information extraite des traces GPS en amont de l'intersection est utilisée pour créer les descripteurs, bien que le comportement des conducteurs en aval d'un feu tricolore puisse également présenter des motifs très spécifiques qui pourraient venir en aide au processus de détection.

Parmi les principales perspectives d'amélioration, nous pouvons tenter d'utiliser l'analyse des données fonctionnelles pour décomposer la distribution temporelle sur une base *ad hoc* de fonctions (par exemple, ondelettes, transformation de Karhunen-Loève...), afin de minimiser la corrélation entre descripteurs. L'extraction d'autres paramètres physiques tels que la vitesse, les accélérations... peut également aider à discriminer les feux tricolores et à les localiser plus précisément. Ceci est possible, à condition que les profils de vitesse des données GPS soient suffisamment lisses. Enfin, un désavantage significatif dans notre

manière de procéder tient au fait que temps et espace sont projetés sur une unique dimension. En pratique, deux véhicules arrêtés pendant 10 secondes sur la même position vont produire localement la même densité de temps d'arrêt qu'un unique véhicule ayant stationné pendant 20 secondes. Pour remédier à ce problème, on peut envisager de construire des vecteurs de descripteurs *spatio-temporels*, avec une estimation bidimensionnelle de la densité par noyau, où une dimension représente le temps d'arrêt et la seconde dimension est la position d'arrêt le long de l'axe routier, comme illustré sur la figure 4.12.

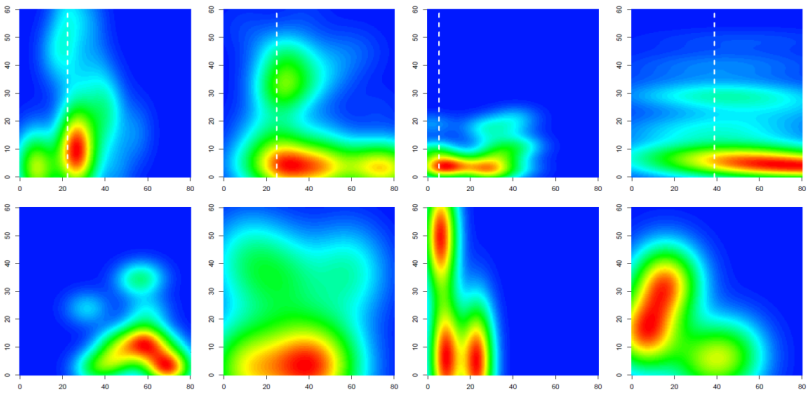


FIGURE 4.12 – Estimation par noyaux 2D de la densité jointe π du couple de variables aléatoires (X, T). L'axe des abscisses représente l'abscisse curviligne (en mètres) de la route, celui des ordonnée, le temps d'arrêt (en secondes). Les quatre instances supérieures sont positives (positions du feu tricolore en pointillés).

Cette représentation permet d'observer des motifs intéressants. Par exemple, sur la figure 4.13, le pic de la densité forme une courbe sur laquelle les variables de temps et de position sont inversement corrélées : le pic de densité est tel qu'à abscisse curviligne x fixée, la durée d'arrêt t la plus fréquemment observée (en termes de densité de probabilité) décroit avec les valeurs de x. Ce phénomène apparaît en général en amont d'un feu et s'explique par le fait que les véhicules arrivant en dernier dans une file d'attente en amont d'un feu tricolore, stationnent moins longtemps que leurs prédécesseurs. Ce motif est donc en lui-même caractéristique de la présence d'un feu tricolore, et on recherche donc ici à détecter une forme, indépendamment de sa position absolue dans le graphique.

Au niveau de certains feux, on observe également une nette bimodalité de la distribution sur la demi-portion inférieure des graphiques, correspondant aux véhicules qui s'arrêtent en amont du feu, et à ceux qui s'arrêtent au centre du carrefour en attendant de tourner à droite [10]. Sur le second point d'arrêt, la forme de corrélation est bien souvent opposée, *i.e.* avec un temps d'arrêt d'autant plus long que l'abscisse est grande. Ce phénomène peut potentiellement s'expliquer par le fait qu'en aval des feux, les vehicules en début de file attendent statistiquement moins longtemps pour tourner à droite.

La reconnaissance dans ces graphiques étant essentiellement visuelle, une perspective de travail consisterait à appliquer un algorithme de réseau de neurones (type CNN, voir chapitre 5) pour détecter la présence de feux tricolores. Quelques expérimentations préliminaires ont été menées en ce sens, avec un CNN classique opérant sur des vignettes de taille

10. Au Japon, les véhicules circulent sur la voie de gauche.

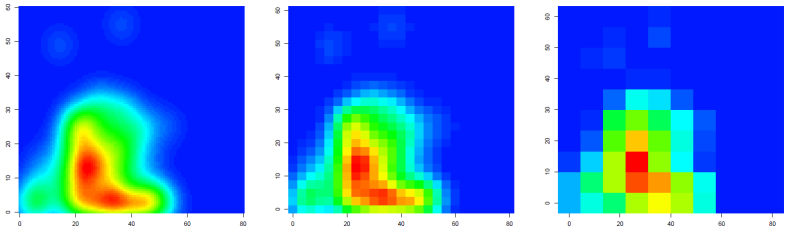


FIGURE 4.13 – Densité de la loi jointe spatio-temporelle des arrêts $\pi(x,t)$ pour trois résolutions de calcul. De gauche à droite : (0.8m, 0.6s), (3.2m, 2.4s) et (8m, 6s).

480 × 480, et composé de 4 enchaînements de convolution/*max-pooling*, la dernière couche, de type *fully connected*, étant reliée à une sortie binaire. Ce nouveau modèle n'a toutefois pas apporté d'amélioration significative. Les travaux ultérieurs viseront à mieux le calibrer. D'autre part, la présence de motifs distincts dans les graphiques, incite à utiliser une approche similaire à celle du paragraphe 3.6.2, en décomposant les distributions sur deux bases distinctes de Karhunen-Loève, construites respectivement avec les instances positives et négatives.

4.4 Études complémentaires

4.4.1 Complément à la préparation des données

Dans cette section, nous étudions un problème spécifiquement rencontré sur les données de ce chapitre, raison pour laquelle les techniques exposées ci-dessous ne figurent pas dans le chapitre 2 traitant de la préparation des trajectoires GPS.

Le problème posé est le suivant : l'inspection visuelle des trajectoires GPS sur le réseau routier (ainsi que sur les photographies aériennes) suggère l'existence d'un biais dans les coordonnées GPS vers le nord, d'une amplitude de l'ordre de 5 à 10 m. L'observation des données sur la ville de Tsukuba, située une soixantaine de kilomètres au nord-est de Mitaka, révèle un comportement similaire, avec une amplitude d'erreur plus marquée (de 10 à 15 m). L'autocorrélation régionale de cette valeur de biais incite à penser que la cause en est une erreur dans la projection cartographique, par exemple le choix d'un mauvais ellipsoïde de référence (*cf* paragraphe 2.3.2.1).

Une solution intuitive pourrait consister à utiliser l'information fournie par le réseau routier via le map-matching pour estimer une valeur numérique de ce biais. Formellement, pour un jeu de données (suffisamment grand) de n points GPS $x_1, x_2 ... x_n \in \mathbb{R}^2$ et une fonction de map-matching $f : \mathbb{R}^2 \to \mathbb{R}^2$, l'estimateur (2D) du vecteur de biais β est donnée par :

$$\hat{\beta} = \frac{1}{n} \sum_{i=1}^{n} \Big[x_i - f(x_i) \Big]. \tag{4.13}$$

Cette solution n'est cependant pas satisfaisante, puisque le biais a tendance à faire échouer l'algorithme de map-matching à identifier le bon arc candidat pour chaque point, y compris avec des algorithmes sophistiqués, tels que celui de Newson et Krumm (2009). En retour, ces erreurs de projection conduisent à sous-estimer (en valeur absolue) le biais dans les

coordonnées. En effet, sur le jeu de données de Mitaka, cette méthode nous donne une estimation $\hat{\beta} = 2.86$ m, bien en dessous de ce qui a pu être observé graphiquement en représentant les trajectoires et le réseau routier.

Nous présentons ci-dessous trois méthodes pour contourner ce problème. Dans ces trois approches, nous utilisons le map-matching (*i.e.* le réseau routier de référence, que nous supposons non-biaisé en termes de position), ainsi qu'un ensemble de trajectoires GPS sur une zone qui doit être à la fois suffisamment grande pour pouvoir avoir suffisamment de données et ainsi moyenner les erreurs des trajectoires et du réseau, mais également suffisamment ciblée pour pouvoir y faire l'hypothèse que le biais est constant.

4.4.1.1 Débiaisage par contrôle manuel

Dans une première approche, on identifie un petit nombre de portions de routes sur lesquelles la situation est relativement claire, *i.e.* où la solution produite par le map-matching ne risque pas d'être significativement impactée par le biais. On récupère ensuite l'ensemble des traces GPS ayant parcouru ces portions de routes, on les recale sur le réseau par map-matching, puis on calcule l'estimateur 4.13. Comme illustré sur la figure 4.14, cet estimateur est biaisé, puisque l'étape finale de la plupart des algorithmes de map-matching consiste en une projection orthogonale des points sur le réseau.

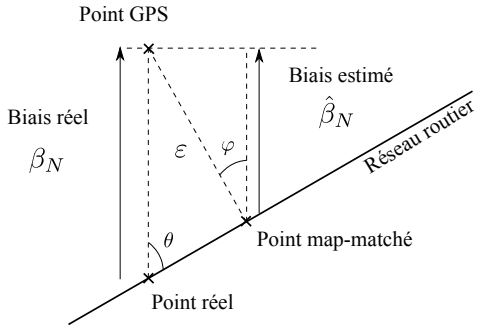


FIGURE 4.14 – Biais estimé $\hat{\beta}_N$ et biais réel β_N suivant la composante Nord.

On peut alors exprimer le biais réel en fonction du biais mesuré par le déplacement du map-matching : $\hat{\beta}_N = \beta_N \sin^2\theta$ d'où :

$$\mathbb{E}_\theta[\hat{\beta}_N] = \frac{2\beta_N}{\pi}\int_0^{\frac{\pi}{2}} \sin^2\theta d\theta \qquad \text{d'où :} \qquad \beta_N = 2\mathbb{E}[\hat{\beta}_N], \tag{4.14}$$

où $\mathbb{E}_\theta$ désigne l'espérance sur la loi de l'azimut θ (supposée uniforme) des axes du réseau routier et $\mathbb{E}$ est l'espérance sur l'ensemble des traces GPS. Le même calcul peut être effectué pour le biais β_E suivant la composante Est.

Nous avons appliqué cette méthode sur 11 zones différentes. Les résultats obtenus (après application de la correction 4.14) sont agrégées dans la figure 4.15. La valeur moyenne

obtenue est de 7.18 m (± 1.46 m).

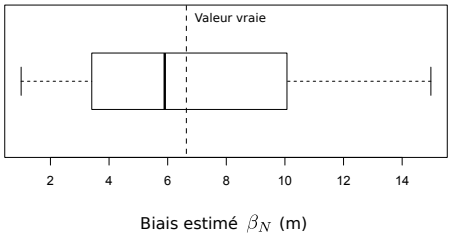


FIGURE 4.15 – Distribution des biais estimés par l'approche manuelle. On considère comme valeur *vraie*, la valeur numérique déterminée par la recherche exhaustive (cf section 4.4.1.2) qui représente *a priori* l'estimation la plus fiable dont on dispose.

La forte dispersion des valeurs estimées et le temps passé à sélectionner les zones limitent fortement le potentiel de cette approche. La valeur moyennée est elle aussi significativement biaisée par rapport à la vérité terrain (*cf* approches suivantes), ce qui s'explique potentiellement par le fait que le nombre de zones sélectionnées est trop faible au regard de l'amplitude des erreurs locales sur les coordonnées du réseau de référence.

4.4.1.2 Débiaisage par optimisation

Nous avons vu dans la section 2.4 que la minimisation de la moyenne quadratique des résidus de recalage n'est généralement pas un bon critère pour régler les paramètres d'un algorithme de map-matching. En revanche, à paramètres fixés, il peut nous permettre d'estimer le biais des coordonnées. Intuitivement, le biais est minimal lorsque les coordonnées sont translatées de l'opposée du biais vrai (inconnu). La démarche consiste donc à définir une séquence de valeurs de biais potentielles $\beta_1, \beta_2, ...\beta_m$. Pour chacune de ces valeurs β_i, on calcule la moyenne quadratique et le biais des résidus du map-matching des données translatées de $-\beta_i$. On obtient donc deux fonctions, échantillonnées en m points et le débiaisage est réduit en un problème de minimisation. Notons que la méthode s'étend aisément au cas des biais en 2 dimensions, en considérant un grille de valeurs $\beta_{ij} \in B \subset \mathbb{R}^2$.

En notant l (1 ou 2) la norme utilisée, l'estimateur s'écrit :

$$\beta_l^* = \underset{\beta \in B}{\operatorname{argmin}} \sum_{i=1}^{n} |x_i - f(x_i)|^l. \tag{4.15}$$

Il existe un vaste choix d'algorithmes pour minimiser 4.15. Dans le cadre de nos tests, nous avons utilisé la recherche exhaustive, la descente de gradient et la méthode de Newton-Raphson (accélération de la descente de gradient par linéarisations successives de la fonction à minimiser). Dans le cas particulier ou β est réduit à une dimension, l'ordre naturel des nombres réels autorise une recherche par dichotomie. Pour plus de détails sur ces méthodes d'optimisation, voir Walter (2016).

Au total, on dispose de 4 algorithmes d'optimisation, pouvant être appliqués sur 2 fonctions objectif différentes, résultant ainsi en 8 estimateurs. Tous les algorithmes (excepté la

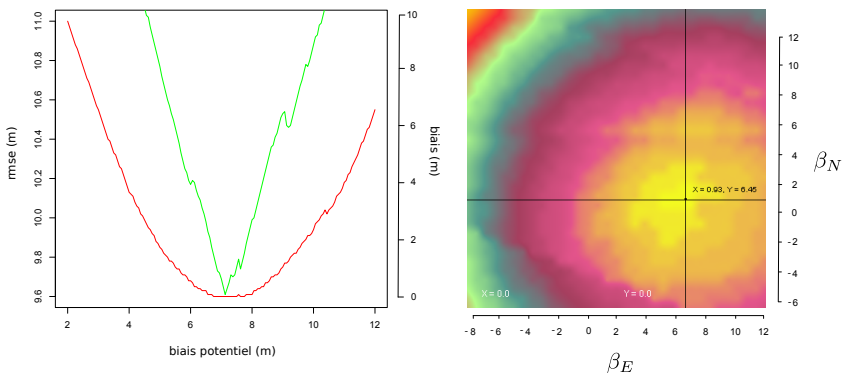


FIGURE 4.16 – À gauche : moyennes L_2 (en rouge) et L_1 (en vert) des résidus de map-matching en fonction du biais potentiel vers le nord. À droite : moyenne quadratique des résidus en fonction d'un biais 2D et point optimal $\beta_E^* = 0.93$ m, $\beta_N^* = 6.45$ m.

recherche dichotomique) convergent. Les fonctions à minimiser n'étant pas rigoureusement convexes (comme illustré sur la figure 4.16), la convergence vers le minimum global n'est pas assurée. Les résultats sont compilés dans la table 4.3 en fin de section.

La méthode la plus efficace semble être celle de Newton-Raphson appliquée sur la moyenne L_1 des résidus, retournant une estimation $\beta_N^* = 6.61$ m en 8 itérations et 6 minutes de calcul. Notons que la vitesse de convergence de cette méthode est quadratique, à condition que la fonction objectif soit deux fois continûment dérivable. En pratique chaque estimation ponctuelle de f est entachée d'un bruit statistique, susceptible de réduire les chances de convergence de l'algorithme. Avec une fonction régulière telle que celle que nous avons eue à gérer dans notre cas d'étude, nous avons pu lisser localement la fonction objectif en calculant 3 estimations ponctuelles au lieu de 2 à chaque itération. Graphiquement sur la figure 4.16, on observe que les dérivées de la fonction objectif sur L_1 sont plus régulières que celles de L_2, expliquant ainsi pourquoi la méthode est plus efficace en norme L_1. Sur un paysage beaucoup plus *chaotique* il est possible que cette méthode ne soit pas suffisante. On peut alors avoir recours à des méthodes plus robustes, comme la descente de gradient, moyennant un temps de calcul plus conséquent.

Lorsque la présence de minimums locaux est fortement pénalisante pour l'algorithme de recherche, on peut se tourner vers des méthodes d'otimisation bayésienne (Mockus, 2012).

4.4.1.3 Débiaisage itératif

Cette méthode est issue d'un constat simple : si le map-matching donne des résultats exacts, la valeur du biais peut être estimée de manière optimale. À l'inverse, si un oracle nous fournit une valeur de biais sur la zone concernée, les résultats retournés par le map-matching seront optimaux. Une idée intuitive peut consister à itérer de la manière suivante : (1) on recale les points GPS sur le réseau par map-matching, (2) on calcule la moyenne des résidus (signés), (3) on retranche ce biais aux coordonnées et (4) on boucle à l'étape (1) tant que le biais calculé en (3) est significativement différent de 0.

Formellement, à chaque itération, le biais partiel β_k est estimé par :

$$\beta_k = \frac{1}{n} \sum_{i=1}^{n} \left[f\left(x_i^{(k)}\right) - x_i^{(k)} \right], \tag{4.16}$$

et les coordonnées sont mises à jour (ou initialisées pour $k = 0$) par :

$$\forall\ i \in [\![1; n]\!] \qquad x_i^k = \begin{cases} x_i^{(k-1)} + \beta_{k-1} & \text{si } k > 0 \\ x_i & \text{sinon.} \end{cases} \tag{4.17}$$

L'estimateur du biais est alors défini par la somme (lorsqu'elle existe) :

$$\hat{\beta} = \sum_{k=0}^{+\infty} \beta_k. \tag{4.18}$$

Il n'existe aucune garantie de convergence de cet algorithme. En pratique, en plus du critère d'arrêt traditionnel sur l'amplitude β_k, on adjoint un critère sur un nombre maximal d'itérations. Dans nos expérimentations, nous avons fixé $k_{max} = 25$ itérations. Pour un modèle théorique simplifié d'un réseau routier en grille régulière (type *Manhattan*) et une distribution arbitraire de points GPS, on peut montrer que l'algorithme présenté dans cette section converge en un nombre d'itérations au maximum égal à l'effectif n du jeu de points.

En supposant la convergence atteinte après L itérations, on a :

$$\hat{\beta} = \sum_{k=0}^{L} \beta_k = \sum_{k=1}^{L} \left[x_i^{(k+1)} - x_i^{(k)} \right] = x_i^{(L+1)} - x_i. \tag{4.19}$$

Les coordonnées finales débiaisées correspondent donc immédiatement aux $x_i^{(L+1)}$. À nouveau, les équations 4.16 à 4.19 restent valides pour le cas général où $\beta \in \mathbb{R}^2$. L'algorithme 2 résume les étapes du débiaisage.

Avec cette méthode, nous avons obtenu une valeur de biais $\beta_N = 6.57$ m en 10 minutes et 13 itérations, ce qui correspond à la valeur vraie issue de la recherche exhaustive. On donne ci-dessous un résumé des résultats obtenus avec les différentes approches proposées.

Nous conclurons cette section sur deux remarques :

- Nous avons supposé que l'erreur sur les coordonnées était entièrement décrite par une translation. En particulier, (et c'est fréquent dans le cas de figure où l'erreur systématique sur les coordonnées provient d'une erreur dans les changements de repère ou les projections cartographiques) il est possible que l'erreur s'exprime plus précisément sous la forme d'une transformation à 4 paramètres (translation, rotation et changement d'échelle). Dans ce cas général, le paramètre β est à rechercher dans un sous-ensemble de $\mathbb{R}^4$ et l'utilisation de méthodes d'optimisation plus performantes devient alors nécessaire.

Algorithm 2 Débiaisage des coordonnées par map-matching itératif

Require: liste de traces GPS $(X_i)_{i=1..N}$ (avec $|X_i| = N_i$ points), fonction de map-matching f, réseau routier de référence $\mathcal{R}$ et paramètres d'arrêt L et EPSILON.

$B = 0$
for $k = 1$ **to** L **do**
  $\beta = 0\,;\ n = 0$
  for $i = 1$ **to** N **do**
    $Y_i = f(X_i, \mathcal{R})$ // map-matching des trajectoires
    for $j = 1$ **to** N_i **do**
      $\beta = \beta + Y_{ij} - X_{ij}$
      $n = n + 1$
    end for
    $\beta = \beta / n$
  end for
  if $\beta \leqslant$ EPSILON **then**
    break // convergence atteinte
  end if
  for $i = 1$ **to** N **do**
    for $j = 1$ **to** N_i **do**
      $X_{ij} = X_{ij} + \beta$ // débiaisage partiel
    end for
  end for
  $B = B + \beta$
end for
return biais estimé B et trajectoire débiaisées $(X_i)_{i=1..N}$.

Méthode	f objectif	Itérations	tps (min)	Estimation (m)
Exhaustive	L_2	400	300	6.59
Exhaustive	L_1	400	300	**6.57**
Manuelle	-	9	7	6.99
Gradient	L_2	180	135	7.02
Gradient	L_1	166	129	6.58
Dichotomie	L_2	20	15	6.58
Dichotomie	L_1	10	8	6.58
Newton	L_2	12	9	6.62
Newton	L_1	**9**	**7**	6.61
Iterative	-	13	10	**6.57**

TABLE 4.3 – Nombre d'itérations, temps de calcul et valeur β_N^* estimée pour différentes méthodes, sur la plage de recherche ± 10 m. Chaque estimation est calculée sur un échantillon bootstrap de 1000 traces. Le critère de convergence a été fixé à 1 cm. Dans chaque colonne, la valeur la plus performante est marquée en gras.

- Dans le cadre restreint de notre jeu de données, nous avons également déterminé un biais d'amplitude plus réduit (sub-métrique) dans la direction est-ouest. Nous n'avons toutefois pas pris en compte cette composante dans la phase de correction, puisque nous pensons qu'il provient d'une erreur dans le réseau routier de référence (un axe de communication nord-sud étant manquant, induisant ainsi une forte composante est-ouest dans l'erreur des points map-matchés sur cette portion de route). Dans des travaux ultérieurs, nous souhaitons investiguer s'il peut être possible de déterminer (par un processus automatique) si les biais mesurés sont significatifs.

En conclusion, dans un cas général, nous recommandons l'emploi de la méthode itérative. En effet, bien que ne possédant pas des garanties de convergence aussi solides que celles associées à la méthode de Newton, elle permet d'obtenir une estimation fiable en un nombre réduit d'itérations et sans nécessiter une quelconque régularité de la fonction de coût à minimiser. Lorsque le jeu de données est suffisamment grand (en termes d'emprise spatiale), nous pensons que l'allure de cette fonction de coût tend à devenir de plus en plus régulière, rendant ainsi la méthode de Newton plus efficace. À l'inverse, pour des jeux de données de petite taille, le temps de calcul ne devient plus un facteur limitant, et la méthode de recherche exhaustive devient alors préférable.

4.4.2 Analyse du comportement de l'algorithme

4.4.2.1 Limites de l'analyse de sensibilité au nombre de traces disponibles

Après la conception d'un algorithme de map inference, une question importante à laquelle on doit répondre, est celle du nombre minimal de traces nécessaires au bon fonctionnement de la méthode. En effet, nous avons vu dans le chapitre 1 que les données FCD peuvent être relativement onéreuses à l'échelle d'une agence nationale de cartographie. D'autre part, les fournisseurs de données proposent bien souvent plusieurs produits différents, avec des prix croissants avec le nombre de véhicules observés et le nombre de jours de données. Pour l'utilisateur des algorithmes de map inference, une décision naturelle consiste à acheter la quantité minimale de données permettant d'inférer de manière

satisfaisante les objets et les attributs manquants dans ses propres bases de données.

Aussi les études de la littérature se concluent-elles bien souvent par une analyse de sensibilité des performances de la méthode au nombre de traces disponibles et/ou à la fraction de véhicules équipés (Buisson, 2017). Dans le cas de figure où la grandeur à inférer est continue (ou du moins ordinale), l'étude ne pose pas de difficulté théorique particulière. Edelkamp et Schrödl (2003), ou encore Zhang et al. (2010) par exemple, montrent que les erreurs d'estimation du nombre et de la largeur des voies sont des fonctions décroissantes du nombre de traces disponibles (avec un gain par trace additionnelle d'autant plus significatif que la précision des capteurs individuels est mauvaise). Dans d'autres travaux, la consistence de l'algorithme (i.e. sa capacité à converger vers la solution exacte lorsque le nombre de traces utilisées tend vers l'infini) est démontrée de manière théorique ou empirique (Li et al., 2018).

La situation se complique lorsque la variable à inférer est une variable binaire locale. Dans le chapitre 3, nous avons déterminé qu'un nombre de 30 traces environ était suffisant pour supposer que les qualités de détection optimales de l'algorithme étaient atteintes. Nous avons pu mener cette étude, car les données utilisées étaient de nature expérimentale, avec un nombre de passages constant, et relativement grand (comparé au nombre minimal attendu), sur toutes les instances.

Dans la suite de cette section, nous poursuivons un objectif plus modeste, en mettant en évidence les faiblesses de cette méthode par sous-échantillonnage pour inférer des informations sur la sensibilité de l'algorithme au nombre de traces.

On donne ci-dessous en figure 4.17 l'histogramme de répartition du nombre N de passages par axe routier comportant un feu tricolore. On y observe la grande variabilité de cette répartition (95% des axes comportent entre 3 et 740 passages).

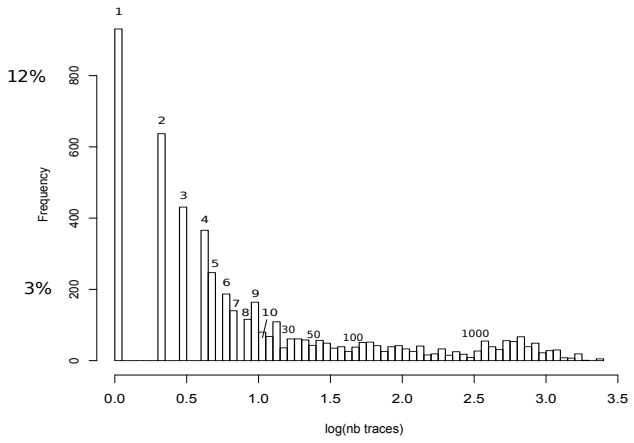


FIGURE 4.17 – Histogramme du nombre de traces GPS disponibles par axe routier comportant un feu tricolore.

Lorsque l'on étudie la sensibilité de l'algorithme, la convergence semble un peu plus lente [11]

11. Mesurée comme l'atteinte de 95% de la sensibilité maximale.

que sur le jeu expérimental du chapitre précédent (de l'ordre de 100 passages, correspondant à une moyenne de 338 traces sur la distribution cumulée). Cependant, poser une contrainte sur le nombre minimal de données disponibles pour traiter une instance, mène à valider l'algorithme sur un jeu de feux tricolores intrinsèquement [12] différent. Cette observation mène à une problématique plus générale.

Lorsqu'un feu tricolore n'est pas parcouru par beaucoup de véhicules (*i.e.* typiquement entre 0 et 10), et qu'il n'est pas détecté par nos algorithmes, on le compte évidemment comme un faux négatif. Cela signifie qu'aucune distinction n'est faite entre les faux négatifs pour lesquels il y avait suffisamment de données mais l'algorithme n'a pas détecté qu'il y avait un feu de ceux du type où il y avait peu de données et donc pour lesquels inévitablement les chances de détecter un feu étaient faibles a priori.

Étant donné que nous ne disposons que de très peu de passages sur ces feux (par définition), on ne peut pas savoir si (hypothèse H_1) ce sont des feux intrinsèquement difficiles à détecter (par exemple, on peut imaginer que les feux dans les lieux plus désertiques ont des comportements relativement atypiques, et que même avec plus de passages, ils resteraient difficiles à détecter) ou bien (H_0) si c'est simplement le nombre de passages qui est insuffisant. Cela pose problème, car si c'est l'hypothèse (H_1) qui est la plus proche de la réalité, cela signifie que les taux de rappel que l'on donne ne sont pas représentatifs des capacités de l'algorithme sur les zones suffisamment couvertes. D'un autre côté, supprimer du jeu de données les feux situés sur des tronçons peu parcourus n'est pas satisfaisant, car si c'est l'hypothèse (H_0) qui est valide, cela signifierait que l'on sur-estimerait les performances de l'algorithme en supposant que tous les feux se valent et que seul le nombre de passages peut faire la différence.

Plus formellement, le type de feu (périphérie/centre ville) ne serait-il pas une variable de confusion, fortement corrélée avec le nombre de passages, et qui ferait qu'on impute uniquement l'échec de l'algorithme sur le manque de données, alors qu'une partie de la cause de cet échec réside justement dans la variable de confusion ?

Pour tester cet hypothèse, nous avons déterminé le nombre maximal de traces nécessaires pour obtenir une sensibilité arbitrairement fixée à 50%. Nous avons obtenu un nombre de 15 traces (pour garantir de disposer d'un nombre suffisant de données, nous avons sélectionné toutes les instances contenant entre 13 et 17 traces). Cela constitue le jeu de données A. Pour former le jeu de données B, nous utilisons toutes les instances contenant plus de 17 traces, et nous réduisons ce nombre à 15 en sous-échantillonnant aléatoirement les données. Nous obtenons donc 2 jeux de données, dont les instances sont parcourues par le même nombre de traces, la seule différence subsistant dans le type d'instance (initialement peu ou beaucoup parcouruue). L'entraînement a été lancé classiquement sur le jeu complet (toujours suivant un mécanisme de validation croisée), puis la validation est opérée séparément sur les jeux A et B. Les résultats du test sont compilés dans la table 4.4.

Sous l'hypothèse (*a priori* acceptable) d'une différence normale, la p-valeur correspondante est inférieure à 10^{-3}, permettant ainsi de rejeter l'hypothèse nulle H_0. En conclusion, les feux peu parcourus sont issus d'une population différente et on ne peut donc pas les supprimer du décompte dans les tests de performances.

12. Par intrinsèque, on entend ici que ces feux tricolores ont des propriétés liées à leur localisation, qui les rend plus facile ou plus difficile à détecter, indépendamment du nombre de traces disponibles.

Jeu de données	sensibilité (%)	écart-type (%)
A	49.95	5.21
B	64.51	2.21
B-A	14.56	3.91

TABLE 4.4 – Résultat du test d'hypothèse (voir texte principal pour les détails).

4.4.2.2 Influence de l'environnement

Dans cette section, nous essayons de déterminer les capacités de généralisation de l'algorithme d'apprentissage. Autrement dit, obtient-on les mêmes performances en apprenant et en validant sur la même zone, ou bien sur deux zones différentes ? Pour ce faire, nous utilisons le second jeu de données pour lequel nous avons relevé la vérité terrain, sur la ville de Tsukuba. Il s'agit d'un jeu beaucoup plus volumineux, avec 1626 feux tricolores répertoriés manuellement (exhaustivité à 96 % ± 3 %). Cette seconde zone d'étude (d'une superficie égale à 17 fois celle de Mitaka), présente également une large variété de morphologies urbaines, avec toutefois un aspect général plus rural que Mitaka.

Nous réalisons 4 tests différents, en entraînant puis validant sur l'une et l'autre des deux villes. Le calcul des instances est effectué suivant un protocole strictement similaire à celui décrit dans la section méthodologie (4.2) de ce chapitre. Nous obtenons les courbes ROC représentées sur le graphique de gauche de la figure 4.18.

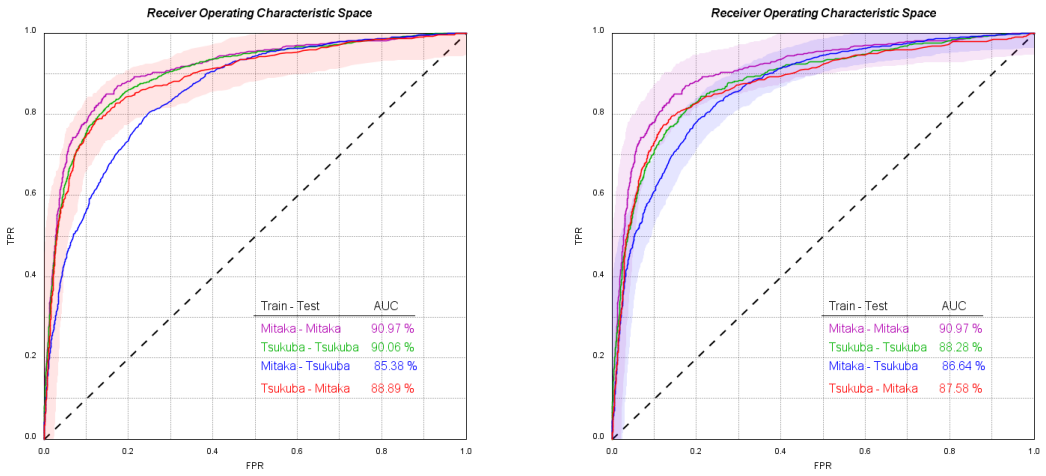


FIGURE 4.18 – Courbes ROC pour les 4 combinaisons entraînement-validation. À gauche : avec les données brutes. À droite : avec les données sous-échantillonnées de sorte à avoir le même nombre de données dans les processus d'entraînement des 4 expérimentations.

On observe que les 4 expérimentations produisent des AUC sensiblement égales, excepté pour le cas d'entraînement sur Mitaka et de validation sur Tsukuba qui retourne une courbe ROC (en bleu) significativement moins performante. Cette différence peut être imputée au fait que les deux jeux de données contiennent des nombres d'instances très différents (6500

instances pour Mitaka, et 53 000 pour Tsukuba). Nous avons réitéré la même expérimentation, en sous-échantillonnant le jeu de Tsukuba pour supprimer l'influence due à la taille de la population d'instances utilisée pour l'entraînement (notons que les proportions d'instances positives dans les deux jeux de données restent inchangées). Les résultats obtenus sont illustrés sur la partie droite de la figure 4.18.

Les scores AUC obtenus sont relativement proches les uns des autres. On remarque toutefois un léger avantage pour les méthode entraînées et validées sur la même zone d'étude, par rapport aux apprentissages croisés, ce qui semble intuitif. La perte d'efficacité en changeant de zone reste cependant modérée, soulignant ainsi les capacités de généralisation de l'algorithme. La superiorité de la combinaison Tsukuba-Mitaka sur Mitaka-Tsukuba suggère également que les instances de Tsukuba sont plus génériques et plus représentatives que celles de Mitaka, sans que nous ne puissions toutefois en être sûr du fait du manque de données et des bandes de confiance relativement larges.

Pour analyser plus finement ces différences, nous cherchons à étudier les zones sur lesquelles l'un des deux modes d'entraînement affiche une nette superiorité par rapport à l'autre. On considère un processus ponctuel marqué de $\mathbb{R}^2 \times \{m, t\}$ (dont les résultats de l'expérimentation décrite ci-avant constituent une réalisation), où un point donné correspond à une instance classifiée de manière erronée par l'un ou l'autre des protocoles d'entraînement. Prenons l'exemple de la zone d'étude de Mitaka. Nous avons deux classifieurs, l'un ayant été entraîné sur Mitaka (en validation croisée) et l'autre sur Tsukuba (suivant le protocole présenté ci-avant). Un point du processus est d'autant plus susceptible d'apparaître en un lieu $x \in \mathbb{R}^2$ où un (et un seul) de ces deux classifieurs retourne un résultat erroné. Le point est alors marqué $Y = t$ si l'entraînement sur Tsukuba a produit l'erreur de classification, et $Y = m$ si c'est au contraire l'entraînement sur Mitaka qui a produit l'erreur. La probabilité du marqueur $Y \in \{m, t\}$ suivant l'emplacement X de l'erreur s'écrit alors à l'aide de la loi de Bayes :

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}. \tag{4.20}$$

où $p(X|Y)$ correspond à la densité spatiale des erreurs commises par l'un ou l'autre des classifieurs, tandis que $p(X)$ est la densité des erreurs (pour lesquelles un seul des classifieurs est fautif) et $p(Y)$ est la probabilité a priori que l'un des classifieurs (plutôt que l'autre) commette une erreur.

En un lieu donné x de Mitaka sur lequel on observe une erreur de classification, la probabilité qu'elle soit commise par le classifieur entraîné sur Mitaka plutôt que sur Tsukuba est :

$$p(Y = m|x) = \frac{p(x|Y = m)p(m)}{p(x|Y = m)p(Y = m) + p(x|Y = t)p(Y = t)}, \tag{4.21}$$

avec les densités spatiales $p(x|Y = m)$ et $p(x|Y = t)$ estimées par la méthode des noyaux, et les probabilités a priori $p(Y = m)$ et $p(Y = t)$ calculées empiriquement à partir des ratios d'erreurs commises par l'un ou l'autre des classifieurs.

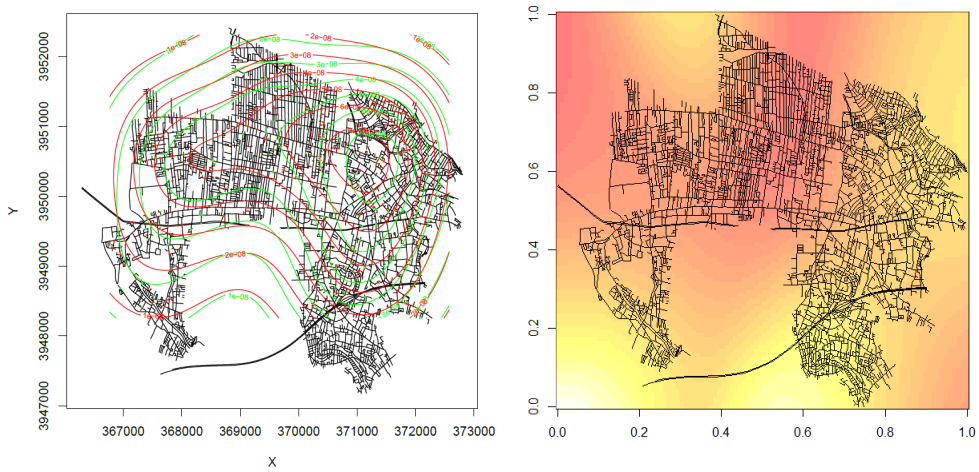


FIGURE 4.19 – À gauche : les densités spatiales d'erreur $p(x|Y = m)$ (en vert) et $p(x|Y = t)$ (en rouge). À droite la carte de probabilité de $\mathbb{R}^2 \to [0, 1]$ qui à x associe $f(x) = p(Y = t|x)$, la probabilité de mauvaise classification par le jeu entraîné sur Tsukuba relativement au jeu entraîné sur Mitaka (du jaune vers le rouge, à mesure que la probabilité augmente).

La figure 4.19 confirme la sensibilité de la méthode aux tissus urbains, en indiquant que le classifieur croisé (entraîné sur la banlieue lointaine de Tsukuba) est plus à même d'opérer sur les zones rurales, ou du moins urbaines peu denses de Mitaka (à l'est et au sud-ouest). À l'inverse, le classifieur non-croisé (entraîné sur la même zone que celle de validation) est plus performant en zone urbaine dense (au centre nord).

4.4.2.3 Comparaison à l'état de l'art

Dans cette section, nous comparons les résultats de la régression de la prosition du feu obtenus dans le paragraphe 4.3, à la méthode proposée par Wang et al. (2017), qui fait actuellement figure de référence dans la littérature. Leur approche repose sur une modélisation des points d'arrêt par un mélange gaussien, ce qui constitue une solution alternative à l'estimation de densité par noyaux, et qui peut être considéré comme son équivalent en statistique paramétrique. L'un de ses intérêts principaux réside dans la possibilité d'inclure des probabilités a priori sur la distribution recherchée. Pour plus de détails sur les mélanges gaussiens, voir Saint Pierre (2003). Dans notre cadre de travail, les auteurs spécifient en particulier que la densité cible est une somme de plusieurs composantes correspondant au rang d'arrêt des véhicules d'attente dans la file. Ils proposent la formulation suivante pour la loi de la variable aléatoire réelle X des points d'arrêt le long de l'axe routier :

$$f(x) = \sum_{k=1}^{m} \pi_k \, \mathcal{N}(x;\ b + (k-1)\mu_s,\ \sigma^2 + (k-1)\sigma_s^2), \tag{4.22}$$

avec m le nombre de composantes considérées, b l'écart entre la ligne d'arrêt et le point d'arrêt du premier véhicule, $\mathcal{N}(\mu_s, \sigma_s^2)$ la loi de l'intervalle entre deux véhicules dans la file d'attente et σ un paramètre contenant l'écart-type de erreurs résiduelles (écart-type de b, erreur GPS...). Afin que la distribution soit normalisée, on impose :

$$\sum_{k=1}^{m} \pi_k = 1, \tag{4.23}$$

où les réels $\pi_k \in [0, 1]$ représentent la loi discrète du rang d'arrêt d'un véhicule dans la file. Pour réduire l'espace de recherche, les auteurs proposent également de poser la contrainte : $\pi_k \geqslant \pi_{k+1}$, ce qui paraît naturel (pour qu'un véhicule s'arrête au rang $k+1$ dans la file, il faut nécessairement que les rangs 1 à k soient également occupés).

La détermination du vecteur de paramètres $\boldsymbol{\theta} = [\boldsymbol{\pi}, \mu_s, \sigma_s, \sigma] \in \boldsymbol{\Theta} \subseteq \mathbb{R}^{m+4}$ permet (entre autres) d'estimer la position de la ligne d'arrêt associée au feu tricolore. Pour un jeu d'observations x_1, x_2,...x_n, l'estimation par maximum de vraisemblance consiste à résoudre le problème d'optimisation non-convexe suivant :

$$\boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmin}} \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta}). \tag{4.24}$$

Une méthode classique consiste à utiliser l'algorithme Expectation-Maximization (EM) pour itérer sur des données complétées jusqu'à trouver un minimum local de la fonction objectif. Pour garantir de déterminer le minimum global, Wang et al. (2017) proposent d'estimer $\boldsymbol{\theta}^*$ à l'aide d'une recherche exhaustive dans l'espace des paramètres $\boldsymbol{\Theta}$, en contraignant le nombre de composantes à $m = 3$ (*i.e.* qu'on ne considère dans la distribution totale que les contributions des véhicules arrêtés dans les trois premières positions dans la file d'attente). Nous avons réimplémenté cet algorithme, et on donne en figure 4.20 un exemple de résultat obtenu sur les carrefours à feu de Mitaka.

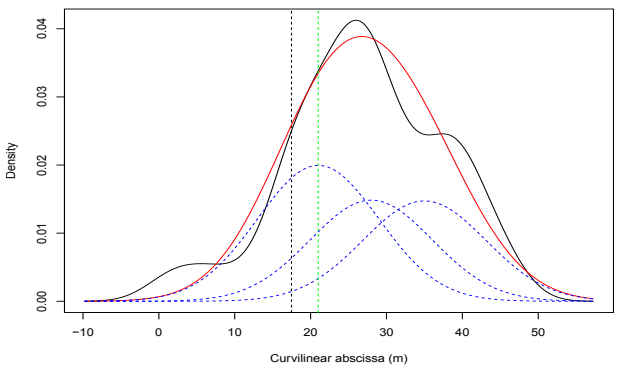


FIGURE 4.20 – Inférence d'un mélange gaussien à partir d'un jeu de points d'arrêt. Composantes élémentaires (en pointillés bleus), densité estimée par kde (en noir), distribution totale estimée par mélange gaussien (en rouge), position estimée du feu tricolore (en pointillés vert) et vérité terrain (en pointillés noirs). Méthode de Wang et al. (2017).

Nous avons observé que l'algorithme rencontrait de grosses difficultés lorsque les données comportent des outliers (par exemple des points d'arrêt correspondant à un autre feu tricolore dans la fenêtre, ou bien à des congestions dues au trafic). Une méthode classique

pour contourner ce problème consiste à définir une composante uniforme (Lathuilière et al., 2018). Dans notre cas, nous avons utilisé une $(k+1)$-eme composante normale avec une variance relativement diffuse (Myronenko et Song, 2010) pour *ramasser* les outliers et éviter de perturber les autres composantes (en fixant la probabilité a priori π_{k+1} égale à la proportion attendue des outliers).

Avec $m = 4$ composantes, la recherche s'effectue dans un espace $\mathbf{\Theta} \subset \mathbb{R}^8$ entraînant ainsi un temps de calcul relativement long (de l'ordre de 2 minutes par fenêtre) même avec une discrétisation plutôt lâche des paramètres à estimer (1 m de résolution pour toutes les grandeurs métriques, une unité pour toutes les grandeurs discrètes, 0.05 pour les valeurs de proportions). L'allongement du temps par rapport au travail de référence de Wang et al. (2017) est principalement dû à l'ajout de la composante supplémentaire, tel qu'expliqué dans le paragraphe ci-dessus. Les résultats obtenus sont compilés dans le tableau 4.5.

Scores	Erreur moyenne	Erreur médiane	RMSE
Estimation (m)	8.89	2.95	13.32
Ecart-type (m)	±0.5	±0.3	±0.9

TABLE 4.5 – Performances de localisation. RMSE : root mean square error.

Ces résultats mettent en lumière la difficulté d'inférer des résultats avec ces données. La présence de bruits parasites rend l'estimationn quasi-aléatoire. On observe cependant que l'erreur médiane (2.95 m) est significativement meilleure qu'avec notre méthode par apprentissage supervisée (3.82 m), montrant ainsi que la méthode de Wang et al. (2017) obtient de très bons résultats dans les situations idéales. En revanche, la présence de bruits sur une partie des instances pénalise fortement les indicateurs moyennés (moyenne et RMSE).

4.4.3 Extensions

4.4.3.1 Autocorrélation spatiale des instances

Dans ce paragraphe, nous cherchons à détecter les carrefours contrôlés par les feux tricolores en agrégant les prédictions individuelles effectuées par l'algorithme des forêts aléatoires.

Considérons un nœud d'intersection du graphe routier, comportant un nombre $n \geqslant 2$ d'arcs entrants, chacun d'entre eux ayant été préalablement classifié par l'algorithme présenté dans la section 4.2, et auquel on a donc associé des prédictions $\psi_i \in [0, 1]$, d'autant plus élevées que l'arc correspondant est susceptible d'être contrôlé par un feu tricolore.

Nous savons que les prédictions agrégées sur les arbres de la forêt aléatoire ne sont pas indépendantes , et que la valeur de prédiction finale est calculée par une somme des prédictions individuelles. En conséquence, les réels ψ_i ne sont pas rigoureusement parlant des valeurs de probabilités. En utilisant la règle de combinaiason de Dempster-Shafer issue de la théorie des fonctions de croyance (Shafer, 1992), on peut montrer que la part de croyance totale allouée à la présence d'un système de feu tricolore sur le carrefour complet s'exprime par la proposition suivante :

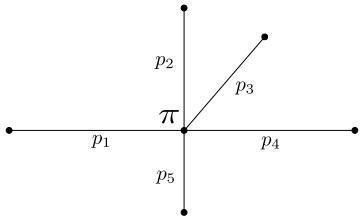


FIGURE 4.21 – Arcs du réseau routier (prédictions individuelles ψ_i) entrant sur un carrefour et agrégés en une prédiction globale pi.

Proposition 4.1.

$$\pi(\psi_1, \psi_2, ...\psi_n) = \prod_{i=1}^{n} \psi_i \times \Big[\prod_{i=1}^{n} \psi_i + \prod_{i=1}^{n}(1-\psi_i)\Big]^{-1}. \tag{4.25}$$

Démonstration. Étudions le cas de base $n = 2$:

La règle de Dempster-Shafer nous dit que la masse jointe $m_{1,2}(A)$ d'un évènement A quelconque non-vide est :

$$m_{1,2}(A) = \frac{1}{1-K} \sum_{B\cap C=A} m_1(B)m_2(C), \tag{4.26}$$

$$\text{avec : } K = \sum_{B\cap C=\varnothing} m_1(B)m_2(C). \tag{4.27}$$

En appliquant 4.26 et 4.27 à l'évènement $A = \{1\}$, on a $B \cap C = A$ pour le couple $(B, C) = (\{1\}, \{1\})$, et $B \cap C = \varnothing$ pour $(B, C) \in \big\{(\{0\}, \{1\}), (\{1\}, \{0\})\big\}$. D'où :

$$m_{1,2}(\{1\}) = \frac{1}{1-K} \sum_{B\cap C=\{1\}} m_1(B)m_2(C) = \frac{m_1(\{1\})m_2(\{1\})}{1-K} = \frac{\psi_1\psi_2}{1-K},$$

$$\text{avec : } K = \sum_{B\cap C=\varnothing} m_1(B)m_2(C) = m_1(\{0\})m_2(\{1\}) + m_1(\{1\})m_2(\{0\})$$

$$= \psi_1(1-\psi_2) + \psi_2(1-\psi_1) = 1 - \big[\psi_1\psi_2 + (1-\psi_1)(1-\psi_2)\big].$$

On obtient la forme recherchée pour $n = 2$:

$$m_{1,2}(\{1\}) = \frac{\psi_1\psi_2}{\psi_1\psi_2 + (1-\psi_1)(1-\psi_2)}.$$

Supposons la relation 4.25 vérifiée pour $n \in \mathbb{N}$ quelconque fixé. La même règle de combinaison nous permet alors d'exprimer la part de croyance allouée à la présence d'un feu à partir des $n+1$ premières prédictions individuelles :

$$m_{n+1}(\{1\}) = m_{n,1}(\{1\}) = \frac{1}{1-K} m_n(\{1\}) m_1(\{1\}) = \frac{\prod_{i=1}^{n} \psi_i}{\prod_{i=1}^{n} \psi_i + \prod_{i=1}^{n}(1-\psi_i)} \frac{\psi_{n+1}}{1-K}$$

$$K = \frac{\prod_{i=1}^{n} \psi_i}{\prod_{i=1}^{n} \psi_i + \prod_{i=1}^{n}(1-\psi_i)}(1-\psi_{n+1}) + \frac{\prod_{i=1}^{n}(1-\psi_i)}{\prod_{i=1}^{n} \psi_i + \prod_{i=1}^{n}(1-\psi_i)}\psi_{n+1},$$

$$\text{d'où :} \quad 1-K = \frac{\prod_{i=1}^{n+1} \psi_i + \prod_{i=1}^{n+1}(1-\psi_i)}{\prod_{i=1}^{n} \psi_i + \prod_{i=1}^{n}(1-\psi_i)}$$

Ce qui implique le résultat recherché au rang $n+1$:

$$\pi(\psi_1, \psi_2, ...\psi_n) = m_{n+1}(\{1\}) = \prod_{i=1}^{n} \psi_i \times \Big[\prod_{i=1}^{n} \psi_i + \prod_{i=1}^{n}(1-\psi_i) \Big]^{-1}.$$

□

Le carrefour analysé est alors classé en tant que carrefour contrôlé par un feu, dès que $\pi \geqslant \frac{1}{2}$. En utilisant cette règle de combinaison, on peut agréger les prédiction sur les rues individuelles en un unique indicateur opérant sur l'intersection globale, échangeant ainsi granularité contre précision. La figure 4.22 illustre les résultats de classification obtenus.

Les indicateurs de sensibilité et spécificité optimaux (tels que calculés dans la fin de la section 3.4) s'élèvent à 87.9% et 96.2% respectivement, ce qui représente une amélioration de plus de 8% par rapport aux classifications individuelles. L'aire sous la courbe ROC (figure 4.22) est de 94.57%. Dans la seconde partie du chapitre suivant, nous adopterons une approche plus générique pour exploiter plus finement les corrélations spatiales (et topologiques) entre les instances à l'aide de l'apprentissage structuré.

4.4.3.2 Régression assistée par orthophotos

Des études ont été engagées pour déterminer dans quelle mesure l'orthophotographie aérienne pouvait permettre de 4.23 d'assister la tâche de localisation. La stratégie testée consiste à détecter l'emplacement approximatif du feu tricolore par la méthode de régression présentée dans le section principale de ce chapitre, puis à affiner cette position par analyse d'image. Les résultats semblent prometteurs mais doivent encore être qualifiés plus

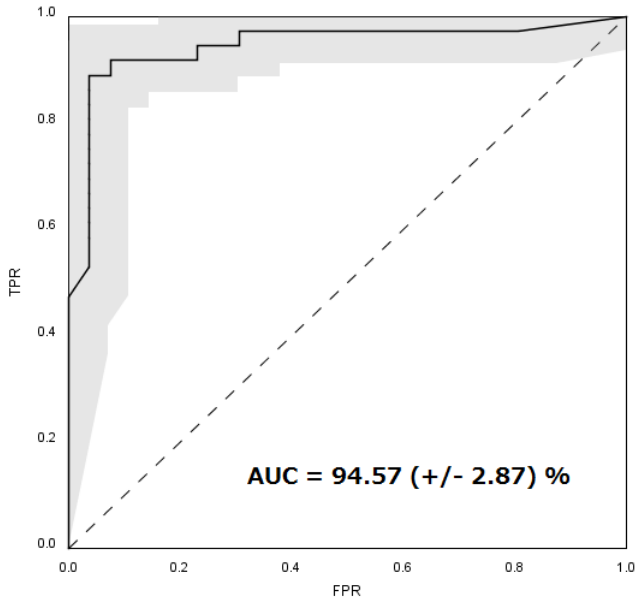


FIGURE 4.22 – Courbe ROC du modèle de classification par carrefour, et bande à 95%.

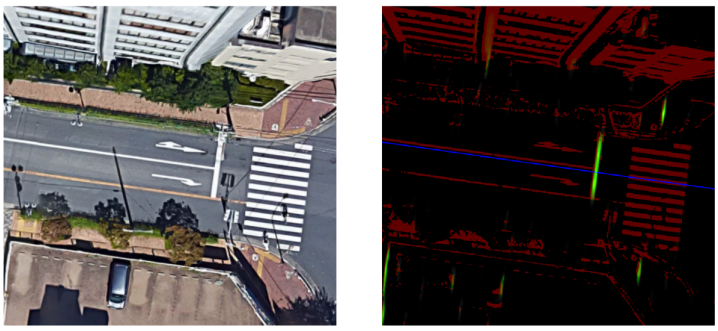


FIGURE 4.23 – Affinage de la position par détection image de la ligne d'arrêt.

précisément dans les travaux futurs.

4.5 Conclusions du chapitre

Dans ce chapitre, nous avons évalué le potentiel des méthodes développées dans le chapitre 3 sur le cas des données opérationnelles, collectées en environnement non-contrôlé et en situation de conduite naturelle. Face à la faible qualité des trajectoires GPS (bruit sur la position, fréquence d'acquisition variable, arrêts intempestifs du récepteur...) nous avons dû adapter notre méthodologie, en focalisant l'approche sur une distribution spatiale des positions d'arrêt des véhicules le long de l'axe routier. Afin de prendre également en compte le caractère temporel du phénomène (notamment à travers les durées des arrêts) nous avons proposé une variation de la méthode d'estimation des noyaux, dont nous avons démontré quelques propriétés mathématiques. La détection du feu et (le cas échant) la

régression de sa position ont été effectuées à l'aide d'une forêt d'arbres aléatoires, dont nous avions expérimentalement mis en évidence l'adéquation face au problème à résoudre dans le chapitre précédent. Les principaux résultats obtenus par les expérimentations sont listés ci-dessous.

- Dans un premier temps, nous avons mis en évidence la difficulté de définir l'emprise géométrique des instances sur la topologie du réseau routier. Trop petites, les instances ne contiennent pas suffisamment d'information pour permettre une détection fiable. Trop grandes, elles entraînent la difficulté de trouver un système de référence commun pour décrire l'ensemble des traces au sein de la fenêtre, tout en présentant le risque d'inclure plusieurs feux tricolores. Nous tenterons de résoudre ce problème dans le chapitre suivant à l'aide d'une approche *globale* de type segmentation d'image.

- Les résultats expérimentaux ont confirmé dans une certaine mesure l'adaptabilité des méthodes employées pour des données de faible qualité, même si les performances obtenues ne permettent pas en l'état actuel d'envisager un processus de détection entièrement automatique. Nous avons cependant dégagé quelques pistes d'amélioration pour prendre en compte de manière plus fine les dimensions à la fois spatiale et temporelle des arrêts de véhicules.

- Une expérimentation a permis de mettre en évidence la possibilité d'exploiter l'autocorrélation des instances individuelles sur le graphe topologique du réseau routier. Dans la deuxième partie du chapitre 5, nous tenterons de rationaliser cette approche à l'aide des méthodes d'apprentissage structuré.

Ce chapitre a également fait l'objet de contributions secondaires, suscitées par le contexte :

- La difficulté à se procurer des données fiables et résolues de la signalisation verticale, a conduit au développement de processus efficaces de relevé manuel et de contrôle de qualité, notamment via la conception d'un programme de saisie méthodique de points d'intérêt sur fonds d'orthophotographies et de relevés terrain à l'aide d'un récepteur low-cost (tout deux présentés en annexe). Ces solutions ont été présentées dans un cadre le plus générique possible pour les rendre applicables à d'autres contextes.

- La présence d'un biais dans les coordonnées du jeu de traces GPS a suscité le développement d'un certain nombre de méthodes de correction à partir des résultats fournis par le processus de map-matching. Les expérimentations menées ont montré l'efficacité de la méthode itérative, dont la mise en évidence de propriétés théoriques (garantie de convergence sous certaines conditions, vitesse de convergence, erreur d'approximation...) feront l'objet de travaux ultérieurs.

Chapitre 5

Approches globales : réseaux de neurones artificiels et apprentissage structuré

Sommaire

Introduction

L'objectif de ce dernier chapitre à visée essentiellement exploratoire, est de tester des méthodes globales, donnant ainsi au modèle de détection un champ de travail plus large que la zone sur laquelle la prédiction est effectuée. Deux approches principales sont testées :

- Dans une première partie, nous expérimentons une approche orientée image, avec l'utilisation d'un réseau de neurones convolutionnel, en mode *prédiction dense* (ou segmentation) permettant ainsi de classifier chaque pixel de la zone de travail en fonction de sa probabilité de contenir un feu tricolore. Cette méthode présente l'avantage de combiner détection et localisation au sein d'un même modèle. En contre-partie, certains post-traitements sont nécessaires pour extraire les positions individuelles des feux tricolores. Cette partie du travail a été initiée par un stage de deux semaines d'élèves-ingénieurs de deuxième année : V. Dizier et M. Margollé, co-encadré avec

M.D. Van Damme.

- Dans une seconde approche, nous utilisons un modèle de champ de Markov défini sur un graphe topologique dérivée du réseau routier, pour affiner les prédictions de la méthode exposée au chapitre précédent. Dans ce framework, détection et localisation peuvent également être gérées au sein d'un même modèle, et nous verrons que l'approche peut être naturellement étendue pour traiter tous les types d'éléments de la signalisation routière dans une problème de classification multinomiale.

5.1 Introduction

Dans cette première partie introductive, nous proposons un petit retour synthétique sur les méthodes expérimentées dans les premiers chapitres 2, 3 et 4 de ce manuscrit. D'une manière schématique, on peut résumer un choix d'algorithme en 5 phases :

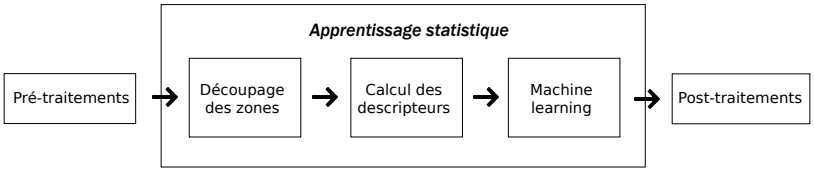


FIGURE 5.1 – Pipeline d'exécution.

- **1. Pré-traitements :** étape préliminaire, dans laquelle on injecte de l'intelligence dans l'algorithme afin d'*aider* (ou de guider) l'apprentissage.

- **2. Découpage des zones :** il s'agit ici de choisir les horizons pour chaque détection. En somme, pour détecter un feu tricolore dans une zone géographique $\mathcal{Z}$, on peut utiliser l'ensemble de l'information contenue dans une seconde zone $\mathcal{Z}' \supseteq \mathcal{Z}$. On notera que $\mathcal{Z}'$ nest pas nécessairement égale à $\mathcal{Z}$. En particulier on peut avoir $\mathcal{Z} \subsetneq \mathcal{Z}'$ quand on souhaite que l'algorithme d'apprentissage considère le voisinage de la fenêtre d'étude pour prendre une décision à l'intérieur de la fenêtre. Dans le chapitre 3, nous avions $\mathcal{Z} = \mathcal{Z}'$ (fenêtre de 100 m de long alignée sur le parcours), mais signalons que dans une première version de la méthodologie, $\mathcal{Z}$ était différente de $\mathcal{Z}'$ (10 m retirés sur chaque côté de la fenêtre). Les résultats nous ont permis de conclure que la configuration $\mathcal{Z} = \mathcal{Z}'$ semblait donner de meilleurs résultats.

- **3. Calcul des descripteurs :** transformation de l'ensemble des données contenues dans $\mathcal{Z}'$ en une entrée compréhensible par l'algorithme d'apprentissage (dans notre cas un vecteur de $\mathbb{R}^p$). Cette étape implique également d'être en mesure de labéliser les zones, *i.e.* de définir sans ambigüité la présence d'un feu dans la zone, et ses coordonnées (dans un système de référence locale à la zone). Si $\mathcal{Z}$ est susceptible de contenir plusieurs feux, les sorties de l'algorithme d'apprentissage devront en tenir compte.

- **4. Passage dans un algorithme d'apprentissage :** étape la plus directe, nécessite de choisir l'algorithme (en fonction des propriétés des modules précédents) et de le paramétrer de manière adéquate.

- **5. Post-traitement :** phase de transformation des sorties de l'algorithme d'apprentissage en un résultat concret exploitable (un certain nombre de feux avec des positions associées sur la carte et éventuellement des indices de confiance associés). En particulier, lorsque les zones de détection $\mathcal{Z}$ se chevauchent, chaque feu réel est détecté plusieurs fois, en général à des positions différentes. L'étape de post-traitement consiste donc dans ce cas-là en une phase de régularisation pour fusionner les solutions.

Ce découpage générique des tâches pose en réalité deux problèmes majeurs. En premier lieu, dans l'étape de découpage des zones : comment transférer des informations d'une zone à l'autre. Plus particulièrement : certains types de signalisation peuvent présenter des signaux caractéristiques bien en amont ou en aval de la présence physique de l'élement) détecter, ce que ne permet pas de prendre en compte un découpage géométrique de la zone de travail. D'autre part, comment prendre en compte les corrélations (positives ou négatives) entre les zones. Les deux parties de ce chapitre ont pour objet de répondre précisemment à ces deux questions. En second lieu, dans l'étape 3, lors du calcul des descripteurs, comment procéder lorsque plusieurs élements sont présents dans une même zone. Cette définition renvoie au choix de la taille des zones : trop petites, elles ne capturent pas le signal nécessaire à la détection. Mais trop grandes, elles risquent d'englober plusieurs feux. Si les zones se recouvrent, les données à passer dans l'algorithme d'apprentissage sont corrélées et donc difficile à valider rigoureusement d'un point de vue statistique. À l'inverse, si on choisit les zones de sorte à réaliser une partition de l'espace (éludant ainsi les problèmes de recouvrement) avec un feu maximum dans chaque zone, à l'inverse on encourt le risque d'une détection très locale et basée uniquement sur des profils à proximité immédiate de la zone sur laquelle on souhaite déterminer un feu. En particulier, nous avons vu dans le chapitre 4 que cette définition était délicate dans le cas d'un réseau routier complexe. Peut-on s'affranchir de la définition des zones à l'aide d'approches globales ?

Remarquons qu'à nouveau, et d'une manière similaire à la démarche adoptée dans le chapitre 3, nous expérimentons l'approche globale suivant deux modes de description complémentaires : un formalisme image, où les relations entre les instances individuelles sont directement encodées par l'agencement spatial des pixels, et un formalisme topologique, plus parcimonieux et potentiellement plus expressif, qui encode les relations par les arrêtes d'un graphe valué, comme illustré sur la figure 5.2.

5.2 Apprentissage image par réseau de neurones convolutionnel

5.2.1 Introduction

Historiquement, les réseaux de neurones artificiels soint issus de la nécessité de modéliser de manière formelle le fonctionnement du cerveau pour tester des hypothèses biologiques simples. Le neurone formel, unité de base du modèle, prend en entrée un certains nombre de signaux, et calcule de manière déterministe une valeur de consigne, qui lui permet de s'activer au delà d'une valeur de seuil prédéterminée. Les poids numériques entrant en jeu

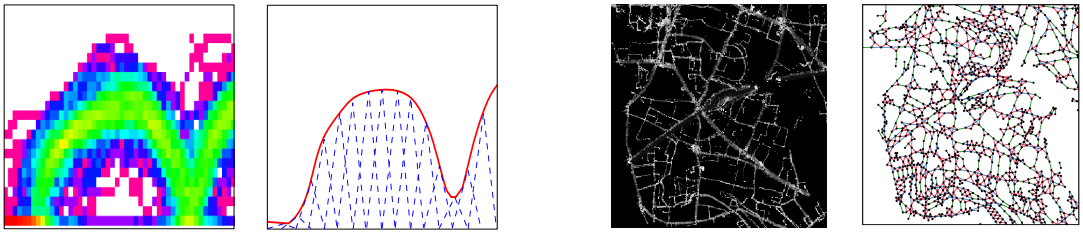


FIGURE 5.2 – À gauche : Approches image (ou raster) et fonctionnelle (ou vecteur) du chapitre 3. À droite : approches globales image (ou géométrique) et relationnelle (ou topologique) du présent chapitre.

dans le calcul sont paramétrables, de manière à simuler la plasticité synaptique du cerveau, et donc sa capacité d'apprentissage en fonction de l'environnement.

En parallèle, les réseaux de neurones peuvent être considérés comme un modèle d'apprentissage statistique qui permet une extension non-linéaire des modèles de régression classiques (Günther et Fritsch, 2010). Étant donné un vecteur de réels en entrée : $\mathbf{x}^i = (x_1^i, x_2^i, ...)$, dans une couche de niveau i du réseau, la sortie $\mathbf{x}^{i+1}$ s'évalue par :

$$x_k^{i+1} = f\left(\omega_{0k}^i + \sum_j \omega_{jk}^i x_j^i\right), \tag{5.1}$$

où ω_{jk}^i désigne le j-eme poids du k-eme neurone de la couche i, et f une fonction d'activation, en général non-linéaire. La sortie du réseau est désignée par $\mathbf{x}^n$, où n représente le nombre total de couches. Le modèle 5.2 est appelé perceptron multicouche.

Étant donné un jeu d'entraînement $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1..n}$ d'instances $\mathbf{x}_i$ étiquetées y_i (variable catégorielle ou continue), l'apprentissage statistique du modèle consiste à calculer les poids ω_{jk}^i du réseau, de sorte à minimiser un critère de coût du type 1.4. L'optimisation est en général effectuée par un algorithme itératif de descente de gradient, à l'aide des expressions de rétropropagation (Kelley, 1960), *i.e.* chaque instance $\mathbf{x}_i$ est passée dans le réseau (avec son paramétrage courant), et on compare le résultat obtenu $f(\mathbf{x}_i; \boldsymbol{\omega})$ avec l'étiquette associée y_i. L'erreur de classification (ou de régression) permet de propager des corrections à effectuer (avec un pas $\eta \in \mathbb{R}^+$, éventuellement variable au cours du temps), en *sens inverse*, sur les poids du réseau (voir voir Friedman et al., 2001 pour les détails) :

$$\forall\ i, j, k \qquad \omega_{jk}^{i(t+1)} = \omega_{jk}^{i(t)} - \eta(t)\ \frac{\partial L(y, f(\mathbf{x}; \boldsymbol{\omega}))}{\partial \omega_{jk}^i}. \tag{5.2}$$

Le théorème d'approximation universelle (voir par exemple voir Hornik et al., 1989 ou voir Cybenko, 1989, garantit que toute fonction continue peut être arbitrairement approchée (au sens de la limite uniforme) par un perceptron monocouche (*i.e.* constitué d'une unique couche de neurones cachés). Cette propriété théorique permet aux réseaux de neurones de partager les avantages des bases fonctionnelles passées en revue dans la section 3.2.2

(Conan-Guez, 2002).

Dans le domaine de la reconnaissance d'images et de signaux audio (entre autres), l'apprentissage profond (ou *deep learning* en anglais) consiste à multiplier le nombres de couches cachées, de manière à améliorer les performances de détection des algorithmes. En particulier, les réseaux de neurones convolutionnels (ou CNN, pour Convolutional Neural Network) consistent à disposer les poids synaptiques de sorte que chaque couche procéde à une convolution de la couche de niveau précédent afin de rechercher dans le signal des motifs de plus en plus complexes. Cette technique permet de partager les poids entre les neurones, et donc de réduire la dimension de la fonction à minimiser, permettant ainsi un gain considérable en temps de calcul, et une réduction du risque de sur-apprentissage des données, tout en conservant un fort potentiel expressif pour l'analyse de données spatialement structurées (Friedman et al., 2001; Guo et al., 2017). En ce sens, les CNN se comportent comme un modèle de régularisation statistique vis-à-vis de l'apprentissage profond classique. En régle général, un modèle classique de CNN, procède à une réduction spatiale de la taille du signal à analyser, en contre-partie d'une augmentation de sa profondeur sémantique.

Les CNN existent depuis les années 1980, mais leurs capacités d'apprentissage n'ont été que récemment mises en évidence avec l'avènement des processeurs graphiques à usage générique (GPGPU), permettant de paralléliser massivement les opérations de matricielles (Li et al., 2016). Depuis lors, les CNN sont utilisés dans de nombreux domaines, par exemple en diagnostique médical (Albarqouni et al., 2016), en cartographie à base d'images satellitaires (Postadjian et al., 2017), en recherche de victimes en montage (Bejiga et al., 2017), en traitement automatique de la langue (Jacovi et al., 2018) ou encore en détection de fraudes bancaires (Lv et al., 2019).

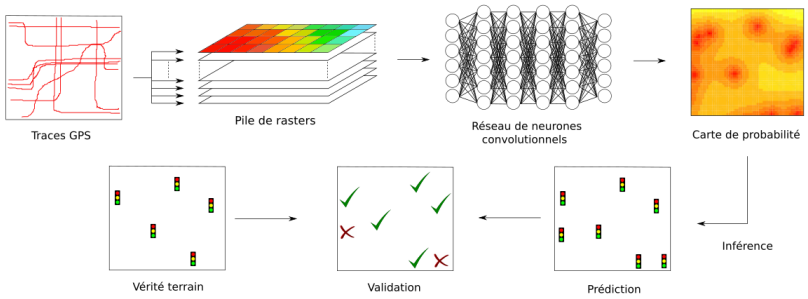


FIGURE 5.3 – Framework global de l'approche par réseau de neurones convolutionnel.

Le framework général de l'expérimentation est représenté sur la figure 5.3. L'idée principale de cette approche consiste à réduire le jeu de trajectoires complet en une pile de grilles de descripteurs zonaux, chaque cellule d'une grille représentant un phénomène physique donné (lié à la cinématique des véhicules traceurs) sur une emprise spatiale plus ou moins réduite. Ces descripteurs pourront être par exemple des vitesses ou des accélérations locales moyennées, des caps majoritaires ou encore des variances de distribution des vitesses.

Dans un second temps, cette pile d'images (que l'on peut considérer de manière équivalente comme une unique image comportant un canal de couleur par descripteur) est passée en entrée d'un réseau de neurones convolutionnel. Dans ce travail, on cherche à détecter et lo-

caliser les feux tricolores simultanément. La sortie du réseau est configurée en mode *dense prediction*, *i.e.* qu'une valeur (dans $[0, 1]$) va être attribuée à chaque zone pour dénoter sa probabilité de contenir un feu tricolore. Ce problème d'apparente à problème de *soft segmentation* (Aksoy et al., 2018) binaire d'une image.

Les positions des feux tricolores sont alors extraites de la carte de probabilité.

L'objectif de cette section est d'évaluer la pertinence de l'utilisation d'un réseau de neurones convolutionnel, pour la détection de la signalisation routière à partir d'un ensemble de trajectoires GPS.

5.2.2 Méthodologie

5.2.2.1 Préparation

Le jeu de données brutes utilisée pour cette expérimentation est identique à celui décrit dans le chapitre 4, à savoir un ensemble de 11 862 trajectoires GPS, localisées sur la ville de Mitaka (16 km^2) dans la proche banlieue de Tokyo (Japon). Les trajectoires comportent typiquement 1 point mesuré toutes les 1 à 3 secondes, et ont été collectées par la compagnie privée NAVITIME JAPAN. La vérité terrain est constituée de 669 feux tricolores, dont les positions ont été numérisées par photo-interprétation, avec une précision sub-métrique.

Contrairement a ce qui a été fait dans les deux chapitres précédents, dans l'approche globale que nous proposons ici, le map-matching des trajectoires sur le réseau routier n'est pas nécessaire [1]. L'approche globale fait donc le pari de réduire les étapes pré-traitements et découpage des zones (figure 5.2) pour laisser l'algorithme d'apprentissage extraire des connaissances à partir de données à l'état brut. La phase de pré-traitement consiste donc en un simple nettoyage du jeu de traces. Plus spécifiquement, les traces contenant moins de 10 points GPS observés (approximativement 1% de l'ensemble des traces) ont été rejetées. Lorsque 2 points consécutifs d'une trace sont espacés de plus d'une distance critique Δ (dans le cadre de nos expérimentations nous avons fixé $\Delta = 150$ m), la trajectoire est divisée en deux sous-portions, la première étant affectée directement à la liste des traces préparées, la seconde étant passée récursivement en entrée de l'algorithme de préparation.

La zone de travail est ensuite découpée suivant une grille régulière de mailles carrées, de résolution 5 m. Cette valeur a été motivée par le fait qu'elle nous est apparue comme un bon compromis afin de disposer de suffisamment de données dans chaque cellule pour y détecter un feu, tout en restant spatialement suffisamment restreinte pour assurer une bonne précision de localisation. Au total, 5 copies de cette grille sont générées, et remplies avec les 5 decripteurs suivants :

- **Vitesse moyenne :** à chaque point GPS observé, on affecte une valeur numérique de vitesse, calculée par différence finie centrée 2.17 à partir des positions et des timestamps. Pour une cellule donnée c et une trajectoire i fixées, on définit la vitesse v_{ic} de i dans c. Si la trajectoire i contient au moins 1 point GPS observé dans c, v_{ic} est définie par la moyenne (arithmétique puisqu'on suppose constante la fréquence

1. Il n'est même d'ailleurs pas souhaitable, puisqu'il réduirait la dispersion transversale des profils, que nous n'avons jusqu'ici pas été en mesure d'exploiter avec les approches à base de fenêtres glissantes longitudinales.

d'acquisition GPS) des vitesses affectées à ces points. Dans le cas contraire, v_{ic} est la moyenne des vitesses des points terminaux des segments intersectants la cellule c. La vitesse v_c dans la cellule c est alors définie par la moyenne des vitesse v_{ic} sur toutes les trajectoires i intersectant c. Si aucune trajectoire n'a été observée dans c, on pose $v_c = -1$ (de sorte à différencier l'abscence de signal de la vitesse moyenne nulle).

- **Accélération moyenne :** on utilise une démarche identique à celle présentée ci-dessus (avec une dérivation numérique d'ordre 2). La valeur obtenue étant cette fois signée, l'abscence de donnée sera marquée par le code -999.

- **Cap dominant :** cet attribut permet de distinguer les deux flots opposés de vehicules sur les artères à double sens. En chaque point, le cap est calculé par différence avant 2.16, *i.e.* par l'azimut de la direction du point suivant de la trajectoire. La valeur retenue pour la cellule est celle du mode de la distribution des caps, quantifée sur 8 niveaux. Pour lever l'ambigüité de 180° sur le calcul de l'arctangente, on utilise la formule des topographes (Bouteloup, 2010) :

$$G_n = 2\tan^{-1}\left[\frac{X_{n+1} - X_n}{\sqrt{(X_{n+1} - X_n)^2 + (Y_{n+1} - Y_n)^2} + (Y_{n+1} - Y_n)}\right]. \tag{5.3}$$

- **Somme des points d'arrêt :** dans chaque cellule, on compte le nombre d'enregistrements GPS correspondants à des points d'arrêt. Pour une fréquence type donnée, ce descripteur est approximativement proportionnel au temps d'arrêt des véhicules dans la cellule, dans une logique similaire à celle des développements 4.11. Si aucune trajectoire n'a été observée on enregistre la valeur -1.

- **Écart-type des vitesses :** obtenu d'une manière similaire à la vitesse moyenne, mais en lui substituant le calcul d'une moyenne par celui d'un écart-type empirique des vitesses locales des trajectoires intersectant la maille. Si aucune trajectoire n'a été observée on enregistre la valeur -1.

La figure 5.4 illustre le processus de calcul des descripteurs sur une grille raster à partir des traces brutes, avec 4 exemples de paramètres physiques.

Ces 6 champs scalaires sont par la suite formatés en niveaux de gris, codés sur 240 niveaux (entre 16 et 255), le niveau 0 étant réservé pour dénoter l'abscence de signal. La vérité terrain est constituée sur une image binaire géoréférencée (dans le même système que les 6 champs de descripteurs), de résolution identique égale à 5 m, un pixel blanc (255) marquant la présence d'un feu tricolore dans la zone correspondante sur le terrain, le reste des pixels étant laissés noirs (0).

L'ensemble constitué des 6 images de descripteurs et de la vérité terrain, sont alors découpées en vignettes de 60×60 pixels (correspondant à une zone carrée d'emprise 300 m au sol). Les premiers tests (effectués avec des vignettes de 30×30 et des pixels de 10 m) étaient plutôt convaincants en termes de temps de calcul, raison pour laquelle nous avons décidé de diviser par 2 la résolution des images. Comme illustré sur le schéma 5.5,

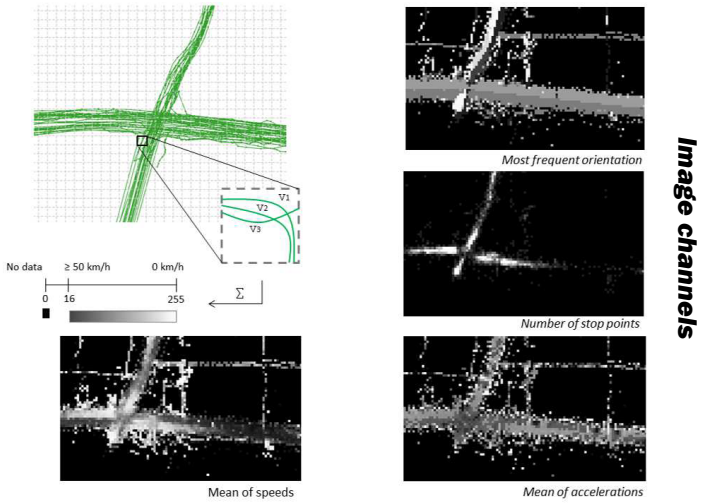


FIGURE 5.4 – Calcul des descripteurs zonaux sur une grille à partir des trajectoires GPS.

chaque vignette comporte donc 5 canaux (correspondants aux 5 champs scalaires décrits ci-avant) plus un canal additionnel pour la vérité terrain (qui est utilisée uniquement lors des processus d'entraînement et d'évaluation de la performance du modèle).

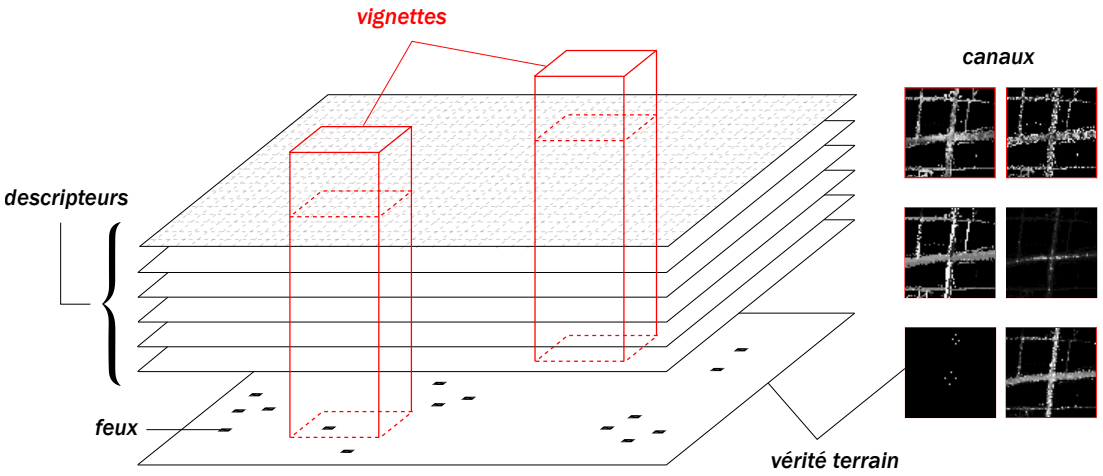


FIGURE 5.5 – Calcul des vignettes à partir des champs scalaires de descripteurs et de la vérité terrain. Les canaux représentent (dans l'ordre conventionnel de lecture) la vitesse moyenne, l'accélération moyenne, le cap dominant, la somme des points d'arrêt, la vérité terrain (en binaire) et l'écart-type des vitesses.

5.2.2.2 Apprentissage

La méthode employée s'inspire du travail de Guo et al. (2017) qui utilisent un réseau CNN de type VGGNet (Simonyan et Zisserman, 2014) pour détecter les constructions

bâties à partir d'images satellites haute-résolution. Cette architecture, composée principalement de deux séquences comportant chacune 3 convolutions et 1 sous-échantillonnage par max-pooling, et terminée par une couche classique dite *fully-connected* (par opposition aux étages de convolutions, pour lesquels chaque neurone n'est sensible qu'aux signaux de la couche précédent situé dans son champ réceptif), est particulièrement bien adaptée au traitement des images de petite taille. Cependant, elle permet uniquement de faire de la classification multi-modale classque (*i.e.* à inférer une étiquette à chaque image). Pour effectuer une segmentation de la scène, nous avons eu recourt à une architecture de type U-Net (Ronneberger et al., 2015), utilisée en particulier pour la segmentation d'images médicales et astronomiques. Pour fournir une sortie qui soit de la même taille que l'image en entrée, le réseau doit *reconstruire* la dimension spatiale de l'image, aui avait été réduite au profit de sa profondeur sémantique lors de son passage dans les premières couches. L'idée fondatrice de l'architecture U-Net, consiste à ajouter lors des phases de déconvolution, l'image correspondante de même dimension obtenue lors du processus de réduction, comme illustré sur la figure 5.6.

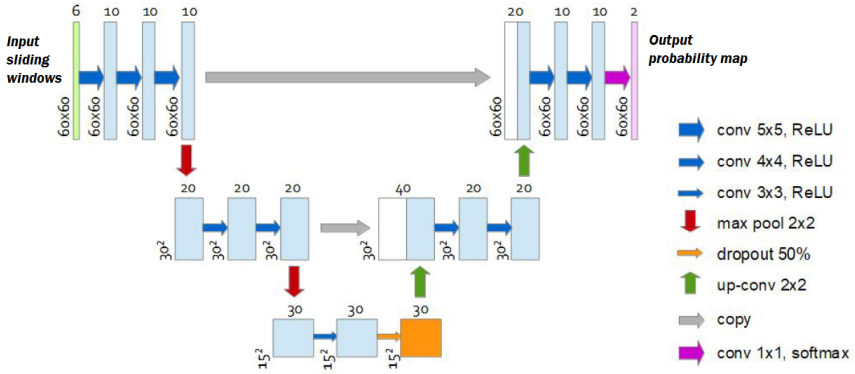


FIGURE 5.6 – Architecture inspirée de U-Net pour la détection des feux tricolores à partir de traces GPS. Les 2 flèches grises représentent le transfert des informations depuis la première partie du réseau pour la reconstruction spatiale de l'image. Illustration extraite de Dizier et Margollé (2018).

Nous utilisons donc ici une architecture *U-Net-like*, en reprenant la double séquence de 3 convolutions et 1 sous-échantillonnage par max-pooling, de l'architecture VGGNet. Notons que la taille de l'image n'est pas réduite entre 2 convolutions, grâce au *zero-padding*. La fonction d'activation dans les neurones des couches de convolution est de type *rectifier linear unit* (ou ReLU) pour sa rapidité de calcul, avec des performances sensiblement équivalentes aux autres fonctions d'activation (Agarap, 2018). Toutes les couches de max-pooling sont opérées en 2×2 (*i.e.* dans chaque paquet de 2×2 pixels, la valeur dominante est affectée au pixel correspondant de l'image sous-échantillonnée d'un facteur 2 dans chaque direction). Nous ajoutons également une phase de régularisation statistique à la fin du processus de réduction spatiale de l'image, consistant à supprimer aléatoirement 50% des neurones de la couche concernée, de manière à forcer l'introduction de redondance dans le modèle et à éviter les problèmes de sur-apprentissage (Srivastava et al., 2014). Le réseau est cette fois terminée par une couche *softmax* afin de convertir les valeurs estimées en probabilités.

À l'initialisation, on attribue des valeurs distribuées suivant une loi normale aux poids des neurones de l'ensemble des couches, suivant une pratique classique dans le domaine de l'apprentissage profond (Kumar, 2017). L'entraînement a été effectuée de sorte à minimiser la fonction d'entropie croisée (à 2 modalités), correspondant à l'espérance sur la loi des étiquettes réelles du logarithme de la probabilité inférée à partir des images. Pour un problème de classification binaire, l'entropie s'écrit :

$$H(p,q) = \mathbb{E}_p[q(x)] \propto \sum_{x \in \mathcal{F}} \log \widehat{y}(x) + \sum_{x \in \mathcal{F}^c} \log(1 - \widehat{y}(x)), \tag{5.4}$$

où x représente un pixel, $\mathcal{F} \subseteq \mathcal{X}$ est l'ensemble des pixels positifs et $\widehat{y}(x)$ est la prédiction effectuée par le réseau de neurones (munit des poids courants) sur le pixel x. Idéalement, la quantité H s'annule lorsque la prédiction $\widehat{y}(x)$ prend la valeur 1 sur tous les pixels de $\mathcal{F}$ et 0 sur son complémentaire $\mathcal{F}^c$.

L'optimisation de 5.4 est effectuée par la méthode Adam, pour Adaptive Moment Estimation (Kingma et Ba, 2014) qui constitue un raffinement de la descente de gradient stochastique classique, en introduisant une inertie dans le modèle de sorte à guider la solution plus rapidement vers la convergence (Ruder, 2016), en particulier dans les configurations pathologiques du paysage de la fonction de coût, telles que la structure en vallée illustree sur la figure 5.7. Dans notre cas d'étude, nous avons effectivement constaté une réduction du temps nécessaire à la convergence avec l'utilisation d'Adam. À chaque itération, la nouvelle valeur du gradient est estimée de manière approchée sur un *batch* de 100 vignettes tirées aléatoirement. Le nombre total d'époques dans le processus d'entraînement a été empiriquement fixé à 130. Enfin, pour contre-balancer le déséquilibre des données en faveur des pixels négatifs, nous avons ajouté un poids de 1 :1200 sur les instances négatives dans la minimisation de 5.4, correspondant à $3/60 \times 60$, soit une moyenne d'environ 3 feux tricolores par vignette.

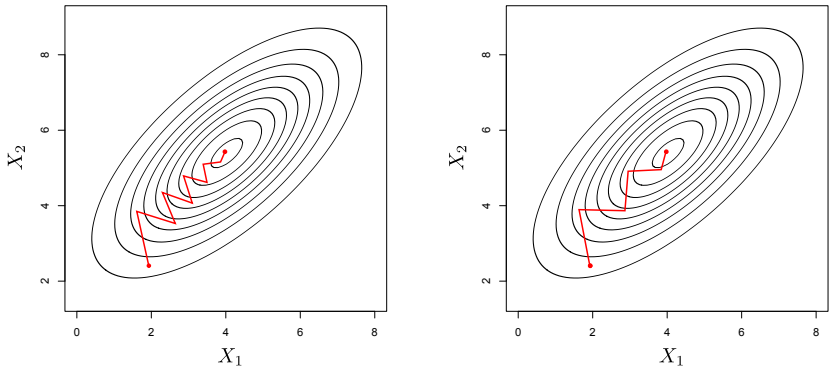


FIGURE 5.7 – Itérations (en rouge) d'un algorithme d'optimisation sur une fonction de coût dont les isolignes ont été représentées en noir. À gauche : descente de gradient classique. À droite : accélération par la prise en compte des moments.

L'évaluation du modèle est effectuée par validation croisée, en partitionnant la zone d'étude en 8 régions, contenant de l'ordre de 90 feux tricolores chacune. L'entraînement est effectué

sur les données issues de 7 zones puis validé sur la zone restante et l'opération est répétée 8 fois au total.

5.2.3 Résultats

Le modèle de réseau a été implémenté en Python, à l'aide du framework Keras [2], qui surcharge la librairieTensorFlow [3]. L'expérimentation a été lancée sur une machine muni d'un processeur 4 cœurs tournant à 3.30 GHz avec 8 Go de RAM. La phase d'entraînement a recquis 70 minutes pour 5490 images. Lors de l'étape de validation, nous avons également pu mesurer à 1.136 secondes le temps nécessaire à l'inférence de 60 images. Cette estimation suggère que le temps nécessaire au traitement de la ville de Tokyo à titre d'exemple (2190 km^2) reste inférieur à 7 minutes, ce qui souligne la capacité de montée en charge de notre approche sur des problèmes opérationnels concrets.

La figure 5.8 illustre un exemple d'inférence de type *dense prediction*.

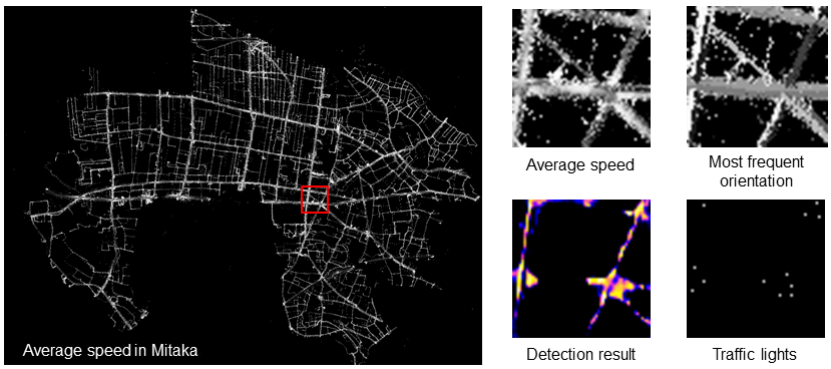


FIGURE 5.8 – À gauche : le champ scalaire de vitesse moyenne des traces GPS sur l'intégralité de la ville de Mitaka. À droite : (dans l'ordre de lecture) la vitesse moyenne, le cap dominant, l'inférence (en valeur de probabilité du sombre vers le clair) et la carte binaire de vérité terrain, découpés sur une vignette de 300×300 m.

5.2.3.1 Résultats

Dans un premier temps, la validation a été effectuée manuellement, en identifiant les proportions de vrais positifs dans la vérité terrain (le rappel) et les cartes inférées (la précision). L'étiquettage manuel a été effectué suivant deux modes :

- Un mode de **tolérance forte**, noté M_0 ci-après, dans lequel les détections mêmes mineures sont prises en compte.

- Un mode de **tolérance faible**, noté M_1, dans lequel seules les détections composées de pics clairement marqués et isolés, sont prises en compte .

2. Keras : `https://keras.io/`
3. TensorFlow : `https://www.tensorflow.org/`

Dans le tableau ci-dessous, on note R_i, P_i et F_i, respectivement, les indicateurs de rappel, précision et mesure F_1, pour le mode M_i ($i \in \{0, 1\}$). Voir section 3.4, page 160 pour plus de détails sur ces indicateurs.

Zone	R_0	R_1	P_0	P_1	F_0	F_1
1	59 %	54 %	54 %	70 %	56 %	61 %
2	85 %	68 %	58 %	57 %	69 %	63 %
3	98 %	97 %	57 %	59 %	72 %	73 %
4	92 %	85 %	53 %	56 %	67 %	67 %
5	100 %	79 %	36 %	44 %	53 %	57 %
6	87 %	70 %	60 %	74 %	71 %	72 %
7	66 %	56 %	67 %	71 %	66 %	63 %
8	73 %	53 %	66 %	75 %	70 %	68 %
Moy.	**82 %**	**71 %**	**56 %**	**63 %**	**65 %**	**65 %**

TABLE 5.1 – Indicateurs de performances R_i, P_i et F_i de notre modèle de réseau profond *U-Net-Like*, pour le mode de validation M_i (cf texte principal pour plus de détails sur les modes de tolérance dans la validation).

On observe que les indicateurs varient signifcativement sur certaines zones (en particulier sur la zone 5 par exemple) ce qui souligne un manque de données d'entraînement, ainsi qu'une difficulté pour le modèle à généraliser sur l'ensemble des types de tissus urbains de la ville.

Notons que si le recours à la segmentation permet de retirer le contrainte de taille maximale sur les fenêtres glissantes (la présence de multiples feux à détecter dans la même fenêtre n'est à présent plus un problème), la continuité de l'inférence entre deux fenêtres voisines n'est pas garantie. Pour obtenir une carte de probabilité totale sur l'ensemble de la zone, nous utilisons le mécanisme de recouvrement et moyennage présenté ci-dessous.

Dans un premier temps, on échantillonne aléatoirement la zone à traiter avec un nombre de vignettes. En pratique, ce nombre pourra être déterminé en fonction du recouvrment souhaité entre les instances, comme illustré sur le graphique 5.9.

Sur les points de recouvrement (*i.e.* ceux sur lesquels le nombre de cellules couvrantes est supérieure ou égale à 2), la valeur finale de probabilité estimée est calculée en moyennant les valeurs inférées sur les différentes vignettes, comme illustrée sur la figure 5.10. Nous avons remarqué que le le moyennage brut ne supprime pas totalement les discontinuités entre les vignettes. Pour résoudre ce problème, nous avons utilisé un moyennage par noyau gaussien (centré sur chaque vignette) avec un écart-type (dans chaque direction) fixé empiriquement à 15 pixels, soit 75 m sur le terrain. La figure 5.10 représente une comparaison des résultats obtenus avec ces deux méthodes de moyennage.

Pour extraire les positions estimées des différents feux tricolores à partir de la carte de probabilité, nous utilisons une démarche inspirée des travaux de Vallet (2013) pour la détection de changement dans les modèles numériques de terrain. La carte estimée et moyennée sur

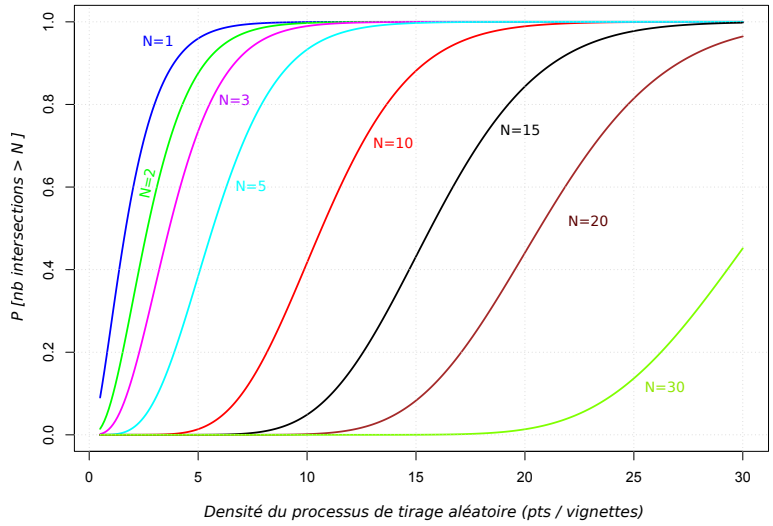


FIGURE 5.9 – Probabilité (en ordonnée) d'obtenir au moins N vignettes recouvrant un point $x \in \mathbb{R}^2$ de la zone à traiter, en fonction de la densité d'échantillonnage (en abscisse) exprimée en nombre de points tirés par unité de superficie d'une vignette. Les courbes ont été obtenues par calcul théorique sur un processus de Poisson en considérant l'emprise de la zone infinie devant la taille des cellules.

l'ensemble de la zone est seuillée avec une valeur $h \in [0, 1]$, comme illustré sur la figure 5.12. On extrait la suite les composantes connexes de la carte binaire obtenue par seuillage (Hernandez-Belmonte et al., 2011). Le problème se résume alors à trouver le seuillage h optimal. À cette fin, on utilise une approche énergétique, en notant $\mathcal{C}_h$ une configuration de feux tricolores donnée (*i.e.* un ensemble de n coordonnées du plan, chaque couple de coordonnées dénotant la position d'un centoïde de composante connexe) pour un niveau de seuillage h, et en définissant 4 contribution indépendantes :

- **Nombre de voisins :** pour chaque feu tricolore x_i, on calcule le nombre $q_v(x_i)$ de voisins, *i.e.* le nombre de feux (distincts de x_i) situés à une distance de x_i inférieur à 50 m (valeur définie empiriquement). On définit alors la composante ψ_1 :

$$\psi_1(\mathcal{C}_h) = -\sum_{i=1}^{n} f(q_v(x_i)), \tag{5.5}$$

  où f est la fonction de coût unitaire représentée sur la figure 5.11, et définie de sorte à maximiser les configurations contenant des groupes de 4 feux tricolores.

- **Attache aux données :** on favorise les configurations dont les comosantes connexes sont piquées dans la carte de probabilité.

$$\psi_2(\mathcal{C}_h) = -\sum_{i=1}^{n} \frac{\widehat{y}(x_i) - h}{1 - h}. \tag{5.6}$$

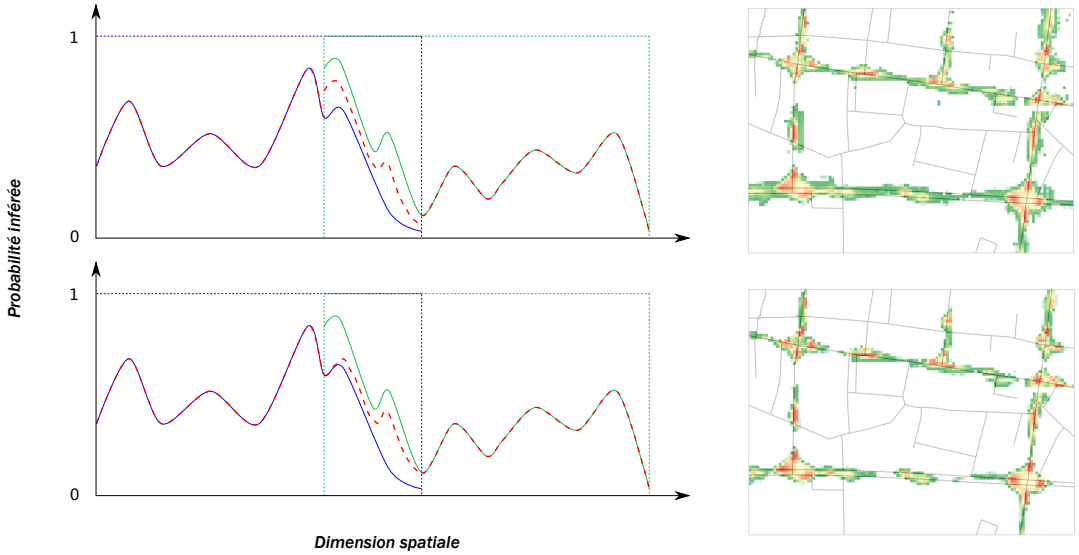


FIGURE 5.10 – Moyennage des vignettes sur les zones redondante : avec un noyau uniforme (en haut) et gaussien (en bas). Les emprises de vignettes et les carte de probabilité estimées individuellements sur chaque vignette, sont représentées en bleu (à gauche) et en vert (à droite), la zone centrale correspondant au chevauchement.

- **Séparabilité des pics :** on favorise les configurations dont les comosantes connexes sont bien marquées et unimodales.

$$\psi_3(\mathcal{C}_h) = \sum_{i=1}^{n} \frac{\widehat{y}(x_i) - \widehat{y}(x_i')}{1-h}. \tag{5.7}$$

  Ce terme pénalise les configurations dans lesquelles les composantes connexes sont constitués de multiples pics de valeurs de probabilité assez semblables.

- **Extension spatiale des composantes :** on favorise les configurations dont les composantes sont de faible emprise au sol.

$$\psi_4(\mathcal{C}_h) = \sum_{i=1}^{n} \mathcal{A}(x_i), \tag{5.8}$$

  où $\mathcal{A}(x_i)$ désigne le nombre de pixels occupés par la composante connexe i.

L'énergie totale est alors exprimée par sommation des expressions 5.5 à 5.8 :

$$E(\mathcal{C}_h) = \sum_{i=1}^{4} a_i \psi_i(\mathcal{C}_h), \tag{5.9}$$

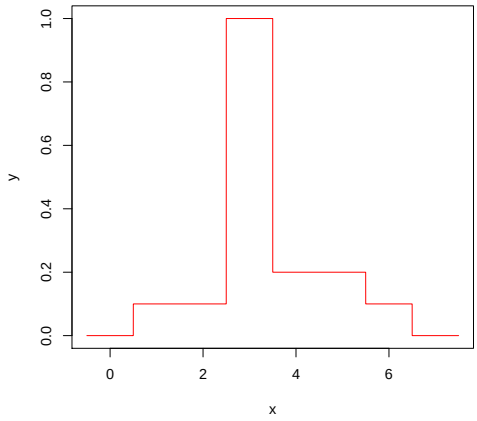


FIGURE 5.11 – Energie ψ_1 en fonction du nombre de voisins pour chaque feu tricolore.

où les réels a_i sont choisis de manière à attribuer plus ou moins de poids aux différentes contributions. Dans notre expérimentation, nous avons fixé $a_i = 1$. Le seuil optimal h est alors détérminé par minimisation de l'énergie :

$$h^* = \underset{h \in [0,1]}{\operatorname{argmin}} \; E(\mathcal{C}_h). \tag{5.10}$$

La recherche de h^* se fait donc itérativement en testant toutes les valeurs de h entre 0 et 1, par pas $\Delta h = 0.01$. Chaque nouveau seuillage exige de recalculer les composantes connexes et d'évaluer les 4 contributions ψ_i.

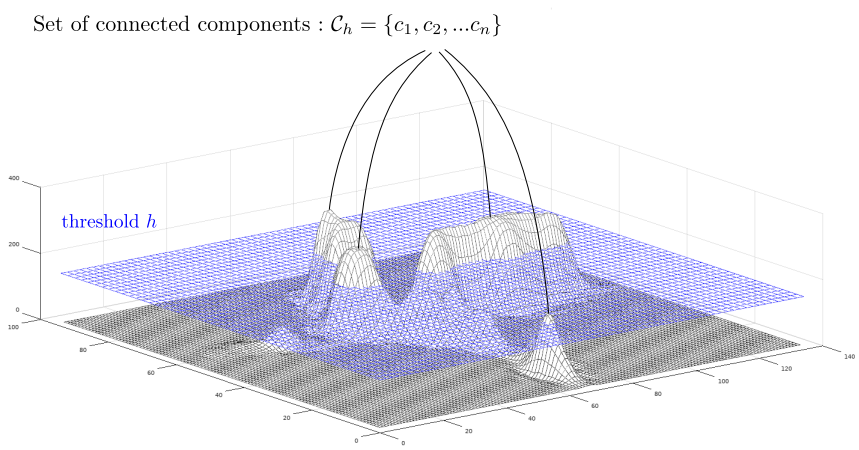


FIGURE 5.12 – Seuillage binaire de la carte de probabilité estimée par le réseau CNN et moyennée, et extraction des composantes connexes.

La figure 5.13 (à droite) illustre le processus d'extraction des composantes connexes (et donc des positions individuelles des feux tricolores) à partir de la carte de probabilité.

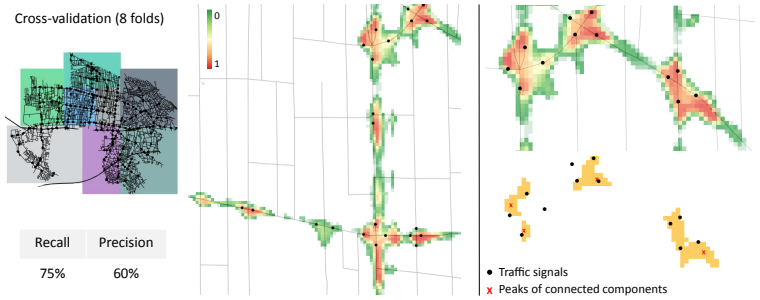


FIGURE 5.13 – À gauche : zones de validation croisée de Mitaka. Au centre : moyennage des vignettes individuelles par convolution gaussienne d'écart-type 15 m. À droite : processus d'extraction des composantes connexes après détermination d'un seuil optimal h^*.

Pour évaluer la précision de la carte de probabilité, nous avons calculé la différence L_1 (*i.e.* la moyenne des valeurs absolues des différences sur tous les pixels) entre la carte inférée $\widehat{y}$ et une carte de vérité terrain indiquant la valeur sur tous les pixels situés dans un rayon de 5 m d'un feu tricolore et 0 sinon. De plus, pour éviter de biaiser les scores vers le haut en effectuant la comparaison sur des zones vides de traces (par exemple dans les parcs ou les grands ensembles bâtis), la comparaison a été effectuée uniquement sur l'intersection de l'emprise de la zone avec un buffer (de rayon 10 m) pris autour des axes du réseau routier, comme illustré sur la figure 5.14.

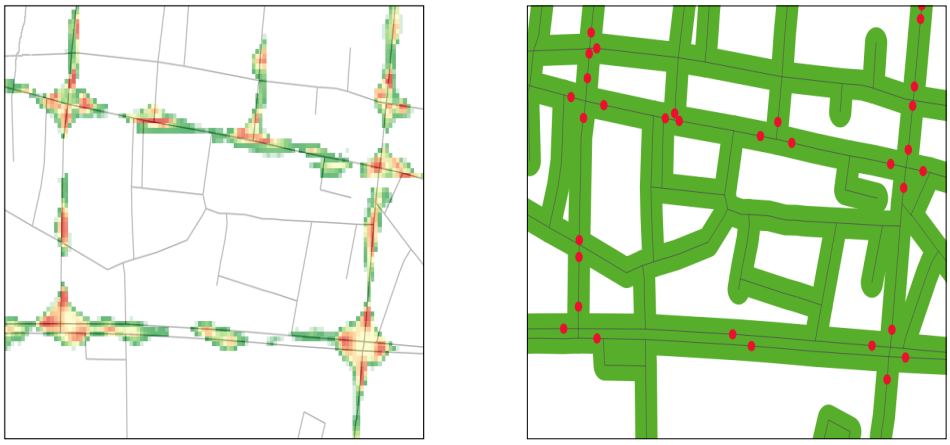


FIGURE 5.14 – Comparaison entre la carte inférée (à gauche) et la vérité terrain (0 sur les pixels verts et 1 sur les pixels rouge). La comparaison est effectuée uniquement sur les zones colorées.

Avec cette méthode de validation, nous obtenons un score L_1 *pixel-wise* de 9 % (± 0.5), indiquant ainsi un bon niveau de fidélité dans la carte générée. L'extraction des positions individuelles en revanche est plus problématique, puisque nous avons mesuré des scores de performance à 75% de rappel et 60% de précision, ce qui est clairement insuffisant, y compris pour un processus semi-automatique, et bien en deça des performances obtenues par les forêts aléatoires (chapitre 4).

En conclusion de cette section, nous noterons la simplicité de la méthode globale par réseaux de neurones, en particulier au niveau des pré-traitements nécessaires dans la phase de préparation des données. Le recours à la segmentation (à la place de la classification) permet de rendre moins crucial les paramètres de fenêtres (dimension, positionn, chevauchement...). Les premiers résultats sont encourageants, mais des travaux ultérieurs sont nécessaires pour définir une meilleure méthode d'extraction des feux tricolores à partir de la carte de probabilité.

5.3 Apprentissage structuré

5.3.1 Introduction

Dans cette seconde partie de chapitre, nous conclurons ce travail de thèse en explorant les capacités de l'apprentissage structuré pour la détection de la signalisation routière. À nouveau, par manque de données (de traces GPS mais aussi de la cartographie routière existante) nous nous limitons à une preuve de concept sur le cas des feux tricolores, et nous montrerons comment étendre naturellement la méthode au cas de la classification multinomiale.

Dans le chapitre 3, nous avons utilisé des instances présentant un recouvrement de 90%. Ce choix, motivé par la nécessité de garantir un balayage homogène de la position de l'élément recherché au sein de la fenêtre glissante, pose en retour des problèmes lors de la séparation du jeu de données en deux sous-ensembles : entraînement et validation. Par ailleurs, nous pourrions objecter que, les instances individuelles étant partiellement communes, elles sont nécessairement corrélées, et nous sortons donc du cadre classique de l'apprentissage, tel que formulé dans la section 1.3.1. De plus, quand bien même les instances auraient été formées disjointement, la proximité spatiale de l'emprise des fenêtres glissantes induit inévitablement une corrélation des variables cibles. Par exemple, sachant que la fenêtre glissante X_i est positive, la probabilité que la fenêtre suivante X_{i+1} soit également positive peut être considérée comme plus faible (en particulier dans le cas où la largeur des fenêtres est plutôt réduite, *i.e.* typiquement inférieure à une cinquantaine de mètres) ou à l'inverse plus élevée (en considérant que la positivité de X_i nous informe sur le fait que X_i, et par suite X_{i+1}, se situent probablement dans une zone urbaine de forte densité, en principe plus susceptible de comporter des feux tricolores). En conséquence, on a vraisemblablement $P(Y_i|X_i, Y_{i+1}) \neq P(Y_i|X_i)$, d'où la nécessité d'inférer les variables Y_i et Y_{i+1} simultanément, ou du moins, de manière globale et cohérente.

Les techniques d'apprentissage collectif, ou *relational learning* (Lu et Getoor, 2003; Dhurandhar et Dobra, 2010) offrent plusieurs stratégies pour répondre à cette problématique. La méthode la plus simple consiste à diviser l'entraînement en deux blocs distincts :

- *L'apprentissage statique* (ou intrinsèque), qui cherche à inférer les variables cibles à partir des données connues, *i.e.* à partir des variables explicatives propres à l'instance concernée, mais aussi celles des instances de son voisinage ainsi que les variables cibles *connues* dans ce même voisinage.

- *L'apprentissage dynamique*, qui cherche à inférer les variables cibles à partir des variables cibles *inconnues* des instances de son voisnage.

La méthode dite de classification itérative (Neville et Jensen, 2000) consiste à itérer entre ces deux types de classification (à l'aide de deux modèles de classifieurs distincts) jusqu'à convergence des étiquettes de toutes les instances de la zone d'étude. D'autres stratégies existent, notamment à base d'échantillonneur de Gibbs (Geman et Geman, 1987), dont le paradigme fondé sur la génération itérative de variables à partir de données incomplètes (à chaque itération), le rend particulièrement adapté à l'apprentissage dynamique. De nombreuses variantes en découlent, dont celle de Chakrabarti et al. (1998), qui à chaque itération affecte une distribution de probabilité à chaque étiquette du modèle, plutôt qu'une affectation en dur.

En réalité, les premiers travaux d'apprentissage structuré remontent aux années 50 (Macskassy et Provost, 2007), et sont issues de la physique statistique, avec notamment la modélisation des phénomènes globaux composés d'une multitude d'intéractions locales. On citera notamment les travaux d'Ising (1925) ainsi que l'extension proposée par Potts (1952), qui sont principalement utilisés pour dérire et trouver les configurations physiques d'énergie minimale de réseaux d'éléments à états discrets (comme par exemple les spins des électrons). Ces modèles appartiennent à la classe plus générale des champs de Markov (ou MRF pour *Markov Random Field*) et sont aujourd'hui intensivement utilisés dans tous les domaines où les prédictions doivent être effectuées sur un ensemble d'objets dont les positions spatiales (ou temporelles) entraînent des corrélations, comme par exemple dans le cas des pixels d'une image à segmenter (Kato, 1994), pour la classification de séries temporelles (Jebreen, 2017) ou encore pour la cartographie du risque en épidémiologie (Azizi, 2011). Les MRF sont eux-mêmes un cas particulier des modèles graphiques probabilistes (Koller et Friedman, 2009).

5.3.2 Les modèles graphiques probabilistes

Cette section constitue une brève présentation de la théorie générale des modèles graphiques. On pourra trouver plus de détails dans les ouvrages très complets de Wainwright et al. (2008), Koller et Friedman (2009) ou encore Sutton et al. (2012).

5.3.2.1 Les modèles dirigés

Définition 5.1. *On appelle réseau bayésien, ou modèle graphique dirigé (DAG pour Directed Acyclic Graph), un ensemble de variables aléatoires* $X = \{X_i\}_{i \in \mathcal{I}}$*, définies sur un espace* $\mathcal{X}$*, munit d'une structure de graphe* $G(X, E)$ *représentant les dépendances conditionnelles* $p(X_i|\Pi(X_i))$*, avec* $\Pi(X_i) \subseteq X$ *l'ensemble des nœuds antécédents de* X_i *dans* G*.*

Prenons un exemple de modèle simpliste, caractérisant une famille de lois sur 4 variables (à gauche sur la figure 5.15) : la présence de pluie (P), la présence d'un accident sur la chaussée (A), la présence d'un feu tricolore (F) et l'arrêt momentané du véhicule (S).

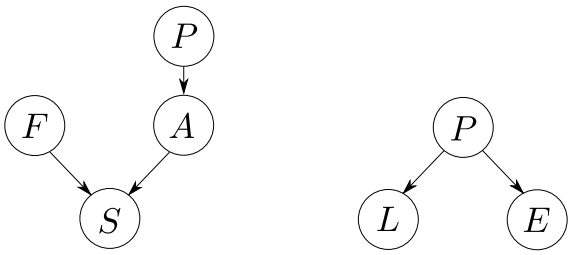


FIGURE 5.15 – Deux modèles graphiques probabilistes. P = Pluie, A = Accident, F = Feu tricolore, S = Arrêt du véhicule (stop), L = Allure lente, E = Essuie-glace.

Le modèle graphique indique les 2 points suivants :

- Les lois $p(A|P)$ (la pluie a une incidence sur la fréquence des accidents) et $p(S|A,F)$ (un arrêt du véhicule peut être dû, et de manière non exclusive, à la présence d'un accident sur la chaussée, ou à un feu tricolore) modélisent les causalités du phénomène étudié.

- La loi jointe sur les 4 variables s'exprime sous la forme factorisée :

$$p(X) = p(\{A,F,P,S\}) = p(P)p(F)p(A|P)p(S|A,F)$$

D'une manière plus générale, la probabilité jointe s'écrit :

$$p(X) = \prod_{i\in\mathcal{I}} p(X_i|\Pi(X_i)), \tag{5.11}$$

avec la convention $p(X_i|\varnothing) = p(X_i)$ lorsque X_i n'admet pas d'antécédent dans G.

On vérifie aisément que la loi p est normalisée, c'est-à-dire que (en notant n le nombre de nœuds du graphe) :

$$\sum_{X\in\mathcal{X}^n} p(X) = 1. \tag{5.12}$$

Le modèle graphique permet de formaliser les indépendances de la loi de probabilité. Par exemple, sur le modèle de gauche de la figure 5.15, on peut lire que la présence d'un feu tricolore est indépendante de la météo, ou formellement : $p(F|P) = p(F)$. En effet, à l'aide de la factorisation, on écrit :

$$p(F|P) = \frac{p(F,P)}{p(P)} = \frac{\sum_a \sum_s p(P)p(F)p(a|P)p(s|a,F)}{p(P)}$$
$$= p(F) \sum_a \sum_s p(a|P)p(s|A,F) = p(F).$$

De manière similaire, on peut facilement montrer des relations de dépendances conditionnelles. Par exemple, la probabilité de présence d'un feu tricolore sachant que le véhicule s'est arrêté est en général modifiée si on sait en plus qu'un accident est survenu sur la chaussée. Formellement : $p(F|S) \neq p(F|A,S)$. D'une certaine manière, la présence d'un accident diminue la probabilité de présence d'un feu, tandis que son abscence, au contraire, tend à l'augmenter. La connaissance de la conséquence commune S a donc brisé l'indépendance entre les causes potentielles F et A. On parle de mécanisme de type *explaining away*.

L'indépendance entre deux variables est donc une propriété qui dépend de l'observation d'autres variables. Par exemple, sur le second modèle de la figure 5.15, l'occurence de pluie est une cause probable commune à l'allure réduite des véhicules et à l'utilisation des essuie-glace. A l'aide de la forme factorisée 5.11, on montre facilement que nécessairement : $p(L|E,P) = p(L|P)$, tandis qu'en général $p(L|E) \neq p(L)$. Autrement dit, l'observation de la présence (ou abscence) de pluie, rend indépendantes l'utilisation des essuie-glace et l'adoption d'une allure réduite. En effet, n'ayant aucune information sur la météo, l'activation des essuie-glace augmente la probabilité de présence de pluie, qui en retour augmente celle de l'allure réduite du véhicule. En revanche, lorsque les conditions météorologiques sont connues, l'utilisation des essuie-glace n'apporte aucune information supplémentaie sur l'allure du véhicule. Nous nous trouvons donc ici dans une situation inverse à celle du paragraphe précédent, dans lequel une nouvelle observation tendait à renforcer les dépendences entre les variables du voisinage.

Nous avons vu 4 exemples de DAG dans les chapitres précédents : la chaîne de Markov cachée (au paragraphe 2.4.2), le classifieur bayésien naïf (figure 3.8), le modèle des *Random Ferns* (figure 3.11) et le mélange gaussien du paragraphe 4.4.2.3 dont les lois jointes associées s'expriment pour chacun d'eux sous la forme factorisée 5.11.

De par leur interprétabilité en termes de relations causales, les DAG sont particulièrement utilisés dans le cadre des diagnositics (par exemple en médecine, ou encore en maintenance préventive). Comme nous l'avons vu dans les expérimentations menées aux chapitres 2 et 3, un second avantage indéniable de ces modèles réside dans leur efficacité d'entraînement, réduite à l'inférence [4] des lois conditionnelles intervenant dans le produit 5.11.

5.3.2.2 Les modèles non-dirigés

Si les graphes dirigés sont parfaitement adaptés pour représenter les relations causales, ils sont insuffisants lorsque la direction des dépendances entre variables est plus floue, notamment lorsqu'elle est inconnue, symétrique ou fluctuante dans le temps. Koller et Friedman (2009) cite par exemple le cas de 4 étudiants (nommons les $(X_i)_{i=1..4}$) ayant préparé un examen en groupe, chacun d'eux travaillant avec exactement 2 amis, comme illustré sur la figure 5.16 (à gauche). Dans ce cas, il est difficile de définir un lien de causalité entre les

4. Par inférence paramétrique pour les variables continues, et par tableau de contingence pour les variables discrètes.

erreurs commises par les candidats à l'examen. D'autre part, notons que X_1 peut avoir une incompréhension d'un point quelconque du cours, qui se répercute sur X_2, qui le transmet à X_3, ce dernier le communiquant à X_4 qui à son tour la renforce chez X_1 et ainsi de suite. L'abscence de boucles dans les DAG (par définition, mais surtout pour que l'équation 5.11 comporte un nombre fini de termes), interdit ce genre de modélisation, d'où l'intérêt des modèles non-dirigés.

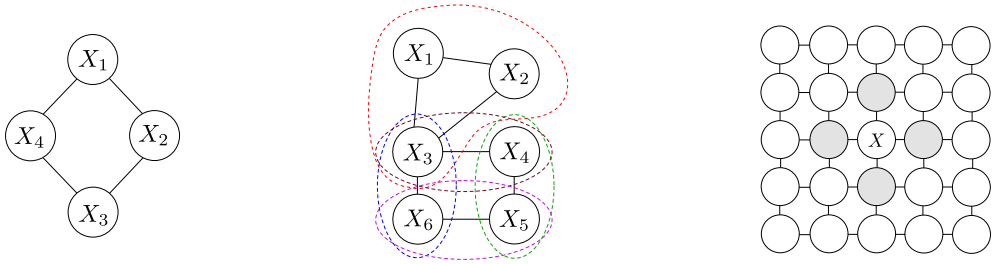


FIGURE 5.16 – Trois exemples de modèles graphiques non-dirigés. À gauche : le modèle des étudiants de Koller et Friedman (2009). Au centre : factorisation de la loi d'un champ de Markov en fonction des potentiels de clique. À droite : *couverture de Markov* d'un champ indiquant que la variable X est indépendante de toutes les autres variables du graphe dès lors que les cellules grisées ont été observées.

Ces modèles semblent également assez pertinents pour notre cadre de modélisation. La présence d'un feu tricolore sur un axe routier d'un carrefour n'implique pas (au sens causal du terme) la présence d'un feu sur les autres axes du carrefour. Mais la présence simultanée d'un feu sur tous les axes (ou sur aucun) est plus probable qu'une configuration mixte. À l'inverse, et comme illustré sur la figure 5.17, la corrélation peut être négative, par exemple lorsque deux arcs se suivent, la présence d'un feu sur les deux arcs simultanément paraît moins probable que toutes les autres configurations.

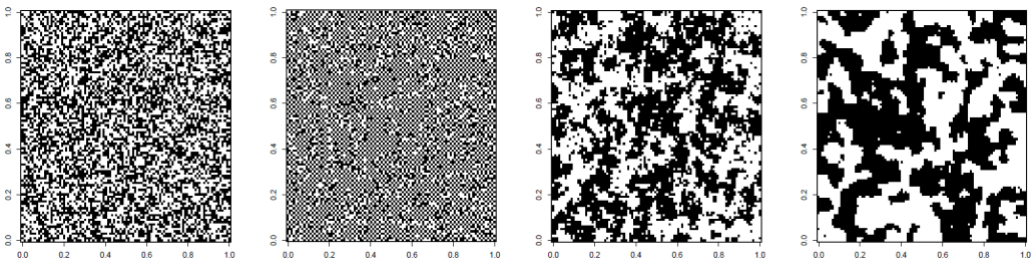


FIGURE 5.17 – Simulations d'un modèle d'Ising sur la grille $\mathbb{Z}^2$. À gauche : bruit blanc. Au centre gauche : champ négativement autocorrélé. Au centre droit : champ positivement et modérément autocorrélé. À droite : champ positivement autocorrélé.

Un autre intérêt fondamental de l'utilisation d'un modèle non-dirigé réside dans la lecture aisée des indépendances du modèle. Précisons un peu cette observation : nous avons vu dans le paragraphe précédent que les indépendances d'un DAG ne se déduisent pas immédiatement des voisinages du graphe. Certaines règles permettent de déterminer (à partir

de la réprésentation graphique) si deux variables quelconques sont indépendantes, sachant une liste de variables observées (voir le concept de *Bayes Ball* de Shachter (2013)).

En revanche, dans le cas des modèles non-dirigés, et sous certaines conditions, le théorème suivant, dû à Hammersley et Clifford (1971), établit un lien direct entre les indépendances du modèle graphique, et la forme factorisée de sa loi jointe. Introduisons d'abord une définition formelle de champ de Markov.

Définition 5.2. *On appelle champ de Markov (ou MRF pour Markov Random Field), un modèle graphique G non-dirigé sur un ensemble X de variables aléatoires, tel que tout nœud y est indépendant des autres nœuds du graphe, sachant ses voisins. Formellement :*

$$p(X_i|X_{j\neq i}) = p(X_i|\mathcal{V}(X_i)), \tag{5.13}$$

où $\mathcal{V} : X \to 2^X$ est la fonction multivoque qui à un nœud $X_i \in X$ associe l'ensemble de ses voisins dans G.

Théorème 5.1 (Hammersley-Clifford). *Soit p une distribution de probabilité sur un graphe G non-dirigé, telle que $p(X) > 0$ pour toute affectation $X \in \mathcal{X}^n$. Alors G est un champ de Markov si et seulement si p admet la factorisation :*

$$p(X) = \frac{1}{Z} \prod_{c \in \mathcal{C}_G} \psi_c(X_c) \tag{5.14}$$

avec $Z = \sum_{\mathcal{X}} \prod_c \psi_c(X_c)$ une constante de normalisation, $\mathcal{C}_G$ l'ensemble des cliques du graphe G, $X_c \subseteq X$ l'ensemble des nœuds de la clique c et ψ_c un ensemble de fonctions strictement positives, appelées potentiels de cliques.

Une loi se factorisant sous la forme 5.14 est appelée une *distribution de Gibbs*. Le théorème de Hammersley-Clifford nous dit que, sur le sous-ensemble des distributions strictement positives, les champ de Markov représentent exactement les distributions de Gibbs.

La preuve du sens réciproque découle directement de la substitution de la loi factorisée 5.14 dans le terme de gauche de l'égalité 5.13 à démontrer. La démonstration du sens directe est nettement plus délicate, et on pourra en trouver une version dans Cheung (2008).

Notons que la condition de stricte positivité de p est importante, ce qui exclut les relations déterministes entre les variables. Un contre-exemple célèbre pourra être trouvé dans Moussouris (1974).

La figure 5.16 (au centre) donne un exemple de factorisation sur un ensemble de cliques. D'après le théorème 5.1, si $p(X) > 0$ sur $\mathcal{X}^n$, alors le graphe G définit un champ de Markov si et seulement $p(X) \propto \psi_{123}(x_1, x_2, x_3)\psi_{34}(x_3, x_4)\psi_{45}(x_4, x_5)\psi_{56}(x_5, x_6)\psi_{36}(x_3, x_6)$.

Un avantage pratique des distributions de Gibbs par rapport aux réseaux bayésiens réside dans la facilité de lecture des zones d'indépendance. On peut démontrer que les propriétés

de Markov locale, globale et *pairwise* (Gandolfi et Lenarda, 2017) constituent trois notions équivalentes, et y sont donc nécessairement toutes vérifiées. Sur le modèle de droite de la figure 5.16, la propriété locale se traduit par l'indépendance conditionnelle de X à toutes les autres variables sachant les variables grisées (qui constituent la *couverture de Markov*, *i.e.* la zone dont l'observation permet une caractérisation probabiliste complète de la variable à inférer X, les autres variables plus *éloignées* étant alors rendues superflues).

Malgré tous ces avantages pratiques, contrairement aux réseaux bayésiens, les modèles dirigés sont clairement plus difficile à entraîner.

5.3.2.3 Les problèmes type

S'étant munit d'un modèle graphique (dirigé ou non) pré-entraîné (*i.e.* dont les lois conditionnelles ou les potentiels de cliques ont été inférés à l'aide d'un jeu d'entraînement), on dénombre trois problèmes principaux (Schmidt, 2007). Etant donné un ensemble (éventuellement vide) de nœuds observés dans, on peut chercher à :

- **Simuler** des réalisations de l'ensemble des variables aléatoires non observées.

- **Décoder** : calculer la configuration de variables la plus probable et qui coïncide avec les valeurs prises par les variables observées.

$$\hat{x} = \underset{x \in \mathcal{X}^n}{\operatorname{argmax}} \prod_{c \in \mathcal{C}_G} \psi_c(x_c). \tag{5.15}$$

  Notons que la constante Z n'intervient pas explicitement dans ce problème.

- **Inférer** : estimer les probabilités marginales (conditionnelles aux observations) des variables (inconnues) du modèle.

$$p(X_s = x_s) = \frac{1}{Z} \sum_{\{x_t, t \neq s\}} \prod_{c \in \mathcal{C}_G} \psi_c(x_c). \tag{5.16}$$

  En pratique, cette tâche se résume souvent au calcul de la constante Z. De plus, pour deux sous-ensembles disjoints de nœuds S et T, la loi marginale conditionnelle $p(X_T|X_S)$ se calcule par la définition classique :

$$p(X_T|X_S) = \frac{p(X_{S \cup T})}{p(X_S)}, \tag{5.17}$$

  les deux membres de la fraction étant estimés à partir de l'équation 5.16.

Dans notre cadre d'étude, le décodage permet de se placer du point du vue du gestionnaire, qui souhaite obtenir une cartographie détaillée et cohérente sur l'ensemble du réseau routier (par exemple pour mener à bien des travaux d'aménagement). À l'inverse, l'inférence

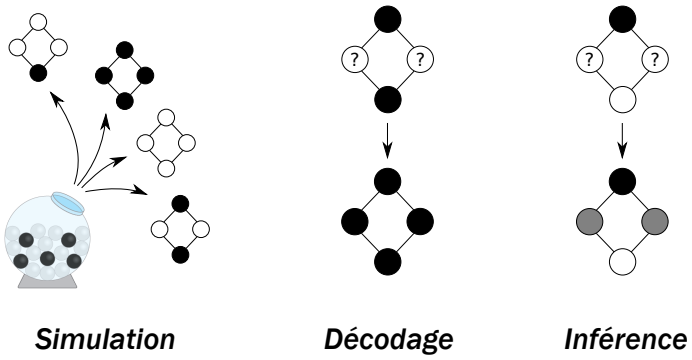


FIGURE 5.18 – Typologie des problèmes à résoudre avec un modèle graphique probabiliste (et étant donné éventuellement un sous-ensemble de variables observées). À gauche : simulation de réalisations distribuées suivant la loi du modèle. Au centre : calcul de la configuration la plus probable (*i.e.* le mode de la distribution). À droite : calcul des probabilités marginales. [Source du schéma de l'urne : Thomas Soell].

présente un intérêt du point de vue du véhicule individuel. En un point local donné, nous pensons que la probabilité marginale de présence d'un élément de signalisation est plus pertinente que le mode global. En particulier, on cherche à affecter en chaque point le mode de la loi marginale (qui est en général localement différent du mode global).

En toute généralité, les trois problèmes présentés ci-dessus sont NP-difficiles (Koller et Friedman, 2009), indiquant ainsi qu'il n'existe pas d'algorithme permettant, pour toute instance de problème posé, de trouver une solution en un temps acceptable en pratique. Dans certains cas particuliers cependant, des algorithmes efficaces existent. Dans les autres cas de figure, on a recours à des algorithmes d'approximation. Passons en revue les méthodes de résolution pour le problème de l'inférence.

Algorithmes de résolution

L'algorithme *Belief Propagation* (BP), aussi appelé *max-product* est une généralisation de l'algorithme de Viterbi présenté au paragraphe 2.4.2 (Yedidia et al., 2003). Le principe de l'algorithme consiste à transmettre à chaque nœud v un message contenant l'ensemble des informations nécesaires à la résolution du problème, collectées par les voisins de v sur chacun des sous-arbres partant de v, comme illustré sur la figure 5.19.

Il a été démontré que l'algorithme BP est exact pour des modèles de graphes en arbre (*i.e.* sans boucle). Notons que l'algorithme *max-product* permet de calculer le mode de la distribution, et donc de résoudre le problème de décodage. Il existe une version inférentielle de l'algorithme, appelée *sum-product*, dont la formulation est quasi-identique, pour calculer les lois marginales (Kschischang et al., 2001).

Pour des graphes avec boucles, plusieurs solutions alternatives ont été proposées :

- Lorsque la suppression d'un petit nombre de nœuds (dont les variables aléatoires associées sont à valeurs discrètes) permet d'obtenir une structure d'arbre, on peut

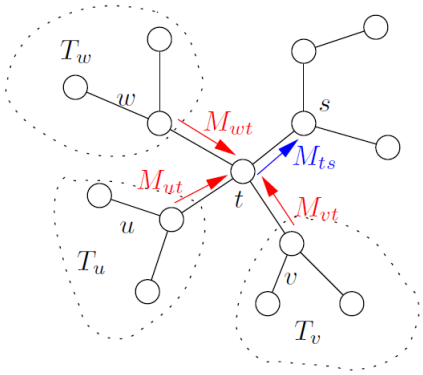


FIGURE 5.19 – Algorithme belief propagation. Source : Wainwright et al. (2008).

adopter la stratégie des *cutsets* (Suermondt et Cooper, 1990) : on crée un graphe conditionné pour chaque affectation possible des variables des nœuds concernés. Le problème peut alors être résolu par l'algorithme *sum-product* indépendamment sur chaque arbre, puis les solutions sont moyennées pour obtenir l'inférence finale. Le nombre d'arbre a créer étant une fonction exponentielle du nombre de nœuds contenus dans le *cutset*, cette solution devient vite impracticable lorsque le nombre de nœuds à éliminer augmente.

- Dans le cas particulier où les variables aléatoires sont binaires et où les potentiels de cliques vérifient des propriétés de sous-modularité, les techniques de flots maximums issus de la théorie des graphe permettent de résoudre le problème de manière exacte (Kolmogorov et Zabih, 2004).

- Lorsque le graphe possède des boucles, mais reste tout de même proche d'une structure d'arbre, ce qui est objectivement quantifié à l'aide de la notion de *largeur arborescente* (Kloks, 1994), on peut utiliser une technique de *Junction Tree* dans laquelle les variables individuelles sont groupées en *super-nœuds*, de sorte à ce que la structure globale du graphe soit un arbre. La technique d'inférence s'effectue alors par l'algorithme *sum-product*, en consdérant toutes les affectations possibles des variables dans les super-nœuds. La complexité de l'algorithme croît exponentiellement avec le nombre de variables par *super-nœuds* et n'est donc practicable que pour des graphes dont la largeur d'arborescente reste modérée. Voir Vats et Nowak (2014) par exemple, pour plus de détails.

Lorsque toutes ces solutions échouent, une alternative consiste à itérer l'algorithme BP, sans garanties de convergence, sur le graphe du modèle (Ihler et al., 2005). Cette méthode est appelée *Loopy Belief Propagation* (LBP).

D'autres algorithmes d'approximation peuvent être envisagés : méthodes empiriques itératives (Besag, 1986), méthodes de Monte-Carlo, méthodes variationnelles (Wainwright et al., 2008)...

5.3.3 Proposition d'un modèle

Dans notre cadre d'étude, le graphe routier constitue une bon support topologique pour modéliser les corrélations entre les variables à inférer. Cependant les instances du problème sont localisées au niveau des tronçons de route, et donc des arcs. On propose donc d'utiliser le graphe adjoint du réseau routier.

Définition 5.3 (graphe adjoint). *Soit $G(V, E)$ un graphe quelconque non-dirigé. On note G^* le graphe adjoint (ou le line graph) de G, défini par un ensemble W de sommets, de cardinal égal à celui de E, et tel que w_1 et w_2 sont reliés par un arc si et seulement si e_1 et e_2 partagent une extrémité commune dans G.*

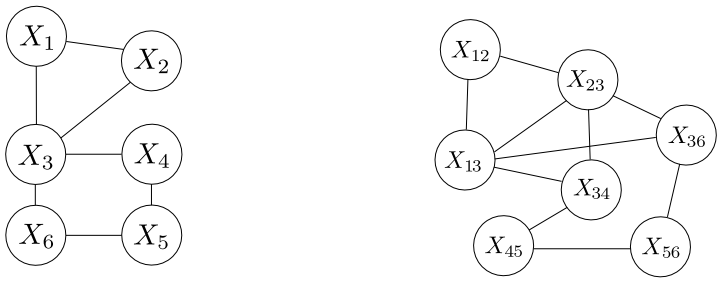


FIGURE 5.20 – Exemple d'un modèle de graphe G (à gauche) et de son graphe adjoint G^* (à droite). Notons que l'adjonction de graphe n'est pas involutive.

Dans un premier temps, nous avons découpé le réseau en tronçons individuels de longueur *préférentielle* l_0 égale à 40 m. Les arcs du réseau d'origine n'ayant pas nécessairement une longueur qui soit un multiple entier de 40 m, nous avons décidé de découper un arc de longueur l en sous-tronçons de longueur :

$$l_r = \frac{l}{p}(1+\varepsilon), \qquad \text{avec :} \qquad p = \frac{l}{\lfloor 2l_0 \rfloor} + 1, \tag{5.18}$$

où ε est un nombre proche de zéro (10^{-8} dans notre cas) permettant de se prémunir contre des erreurs d'arrondi qui seraient sinon susceptible d'entraîner la création de petits arcs résiduels indésirables. Notons H le graphe sur-échantillonné résultant. La longueur préférentielle $l_0 = 40$ m a été sélectionnée à partir de l'histogramme des longueurs des arcs du graphe routier (voir figure 5.24 en fin de section).

On construit alors le graphe adjoint H^*, dont on classifie les arrêtes en 3 catégories en fonction de la configuration du nœud associé dans le graphe H : le premier groupe (en vert sur la figure 5.21) dénote les liens entre les instances reliées par un nœud de degré 2 dans H et traduit ainsi les connexions hors carrefour physique. Le deuxième groupe (en bleu) rassemble les instances débouchant sur une même intersection physique v (avec $deg_H(v) > 2$) et telles que l'angle entre les deux axes routiers est inférieur à un angle $\alpha = 45°$. Ce groupe dénote les couples d'instances débouchant sur les côtés opposés d'un carrefour physique[5]. Enfin, le troisième groupe rassemble les arêtes restantes dans H^*,

5. Notons que les axes routiers sont des polylignes. L'angle α considéré est celui qui sépare les termi-

c'est-à-dire formellement les arêtes liant des arcs avec un angle α supérieur ou égal à 45° dans H. La figure 5.21 illustre le procédé de construction et de classification des arcs de H^*.

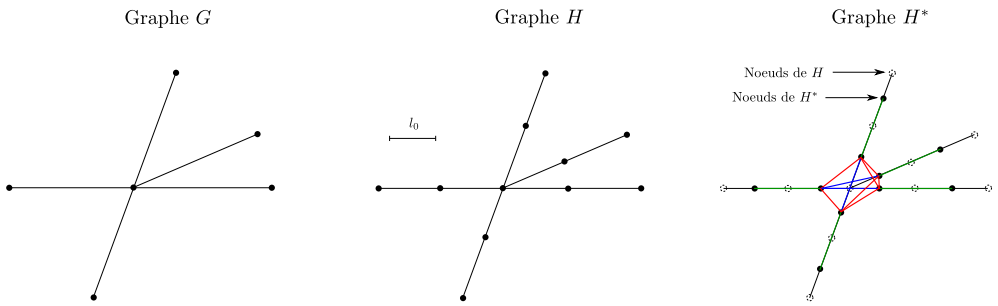


FIGURE 5.21 – Exemple de construction du modèle graphique. À gauche : graphe routier G d'origine. Au centre : construction du graphe H par sur-échantillonnage de G suivant une longueur préférentielle l_0. À droite : construction du graphe adjoint H^*, et classification des arcs en trois groupes en fonction du type de connectivité des instances dans H.

On utilise alors un modèle d'Ising (Ising, 1925), en attachant une fonction de potentiel binaire ψ à chacun des trois types de lien (tous les liens de type vert par exemple, partagent la même fonction ψ_v). On complète alors le modèle avec un ensemble de potentiel unaires permettant de relier les instances à l'estimation effectuée par la forêt aléatoire dans le chapitre 4, suivant un paradigme similaire à celui des champs de Markov conditionnels, ou *CRF* pour *Conditional Random Field* en anglais (Sutton et al., 2012). Cette approche combine donc une phase d'apprentissage classique avec une phase d'apprentissage structuré.

En notant V_k ($k \in \{1, 2, 3\}$) l'ensemble des arcs de H^* de type k, et ψ_k les potentiels associés, la loi jointe du modèle s'exprime :

$$p(x_1, x_2, ...x_n) = \frac{1}{Z} \prod_i \psi(x_i, y_i) \prod_{(i,j)\in V_1} \psi_1(x_i, x_j) \prod_{(i,j)\in V_2} \psi_2(x_i, x_j) \prod_{(v,w)\in V_3} \psi_3(x_v, x_w), \tag{5.19}$$

avec ψ le potentiel d'attache aux prédictions y_i effectuées par la forêt aléatoire. En pratique, et pour des raisons que nous ne détaillerons pas ici, on préfère représenter la loi 5.19 sous une forme exponentielle (Wainwright et al., 2008), facilitant ainsi la phase d'inférence.

On donne ci-dessous une représentation schématique du modèle (5.22) ainsi qu'une illustration du résultat de l'étape de construction sur une partie du réseau de Mitaka (5.23).

naisons des 2 polylignes du couple d'instances.

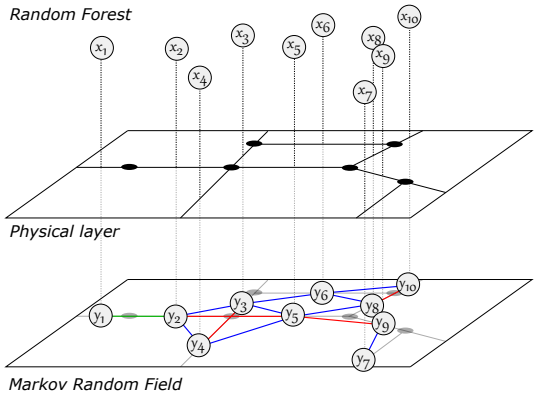


FIGURE 5.22 – Modèle graphique probabiliste (en bas) sur le graphe adjoint du réseau routier (au centre) conditionné sur les prédictions individuelles effectuées par les forêts aléatoires.

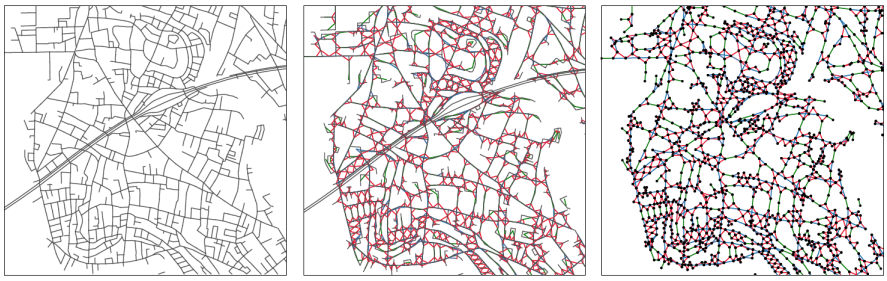


FIGURE 5.23 – Réseau routier (à gauche), construction du graphe adjoint (au centre) et graphe adjoint seul (à droite). le code couleur de la classification des arcs est identique à celui adopté sur les figures précédentes 5.21 et 5.22.

5.3.4 Apprentissage

L'apprentissage du modèle ainsi défini s'effectue par minimisation de la log-vraisemblance, en alternant une phase d'inférence et une phase de descente de gradient des paramètres θ du modèle. Un lemme bien pratique (voir Koller et Friedman (2009), p 947 par exemple) nous enseigne que la matrice hessienne de $\log Z(\theta)$ est égale à la matrice de covariance des statistiques exhaustives de la loi p. Par suite, elle est définie-positive, et $-\log Z(\theta)$ est donc une fonction concave, admettant un unique maximum, et garantissant la convergence de l'algorithme de descente de gradient vers une solution optimale.

Cependant, pour que la solution obtenue pendant la phase d'entraînemennt soit exacte, il est nécessaire que l'inférence effectuée à chaque étape de l'entraînement le soit aussi. Nous avons passé en revue les algorithmes d'inférence dans le paragraphe 5.3.2.3. Le graphe n'étant pas un arbre, une inférence exacte par l'algorithme *sum-product* n'est pas envisageable. Par ailleurs, il paraît difficile de trouver un sous-ensemble (de faible effectif) de nœuds permettant, moyennant leur conditionnement, de réduire le graphe à une structure d'arbre. Les variables aléatoires prennent leurs valeurs dans l'ensemble $\{0, 1\}$, mais les fonctions de potentiel n'ayant pas de raison d'être sous-modulaires, une résolution par l'algorithme des flots maximums paraît également exclue.

Intéressons-nous à la largeur arborescente $tw(H^*)$ du modèle. Son calcul est lui-même

NP-difficile, on cherche donc à le contraindre à l'aide d'une borne inférieure ou supérieure (Bodlaender et Koster, 2010). Considérons le lemme suivant, dont on pourra trouver une justification dans Harvey et Wood (2018) :

Lemme 5.1. *Soit G un graphe connexe et notons G^* son graphe adjoint. On a alors :*

$$tw(G^*) \geqslant \frac{tw(G)+1}{2} - 1, \tag{5.20}$$

où $tw(G) \in \mathbb{N}$ désigne la largeur arborescente de G.

Par ailleurs, à l'aide d'un théorème établi par Seymour et Thomas (1993), on peut facilement monter en corollaire que la largeur arborescente de la grille régulière $G_{m\times n}$ est supérieure à la plus petite de ses deux dimensions :

$$tw(G_{m\times n}) \geqslant \min(m,n) \tag{5.21}$$

De plus, une décomposition arborescente de G pouvant être restreinte pour former une décomposition arborescente de tout sous-graphe H de G, on a : $tw(H) \leqslant tw(G)$. De cette observation, du lemme 5.1 et de l'inégalité 5.21 découlent l'implication pratique suivante :

Tout graphe routier contenant une sous-partie de type réseau de Manhattan de n rues (i.e. un bloc régulier en grille de n colonnes et $m \leqslant n$ lignes), induit un modèle graphique probabiliste dont la largeur arborescente est au moins égale à $\lfloor n/2 \rfloor$.

Cette implication montre que la largeur arborescente du modèle proposé, est en général (pour des réseaux routiers urbains classiques) supérieure à la limite autorisée en pratique pour envisager d'appliquer l'algorithme *Junction Tree* en un temps raisonnable.

Nous choisissons donc d'effectuer l'inférence avec l'algorithme *Loopy Belief Propagation*.

5.3.5 Résultats

L'expérimentation a été réalisée sur le réseau routier de Mitaka comportant 669 feux tricolores. Les performances du modèle ont été évaluées par validation croisée, avec un découpage spatial est-ouest de la zone d'étude en deux secteurs, contenant respectivement 359 (54%) et 310 (46%) feux tricolores. La séparation a été effectuée sur un front rectiligne de 2000 m, au niveau d'une zone de faible densité du réseau routier, de sorte à limiter l'impact négatif du découpage.

Pour un réseau routier initial G contenant 6521 arcs (après suppression des nœuds de degré 2, *cf* paragraphe 4.1.1), la préparation du modèle a requis un temps total 17.48 sec, réparti comme suit : 12.73 sec pour le sur-échantillonnage du réseau et 4.75 sec pour le calcul du graphe adjoint et la classification des liaisons. Le graphe sur-échantillonné H contient 8455 arcs. Le graphe adjoint H^* contient 8455 nœuds et 30140 arcs. Parmi ces arcs, 13% sont de type 1 (liaisons de degré 2), 30% sont de type 2 (axes *opposés* sur un carrefour) et 57% sont de type 3 (axes *transversaux* sur un carrefour). Le faible nombre d'arcs de type 1

valide *a posteriori* le choix de la longueur préférentielle $l_0 = 40$ m.

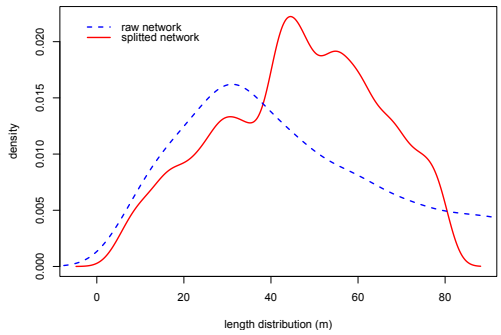


FIGURE 5.24 – Distribution des longueurs des arcs sur le réseau initial G (en pointillé bleu) et sur le réseau sur-échantillonné H (en rouge).

L'entraînement du modèle a été effectué à l'aide de la librairie Matlab UGM (Schmidt, 2007). On donne ci-dessus en figure 5.25 la courbe ROC de la classification, que l'on compare à celle obtenue par l'apprentissage non-structuré. Les résultats numériques sont reportés dans le tableau 5.2.

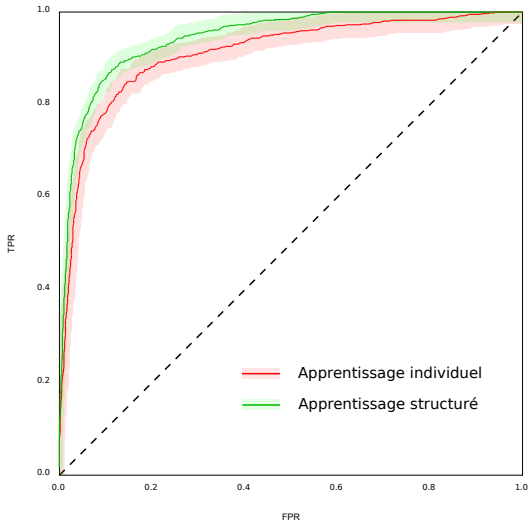


FIGURE 5.25 – Courbes ROC et bandes de confiance à 95% (calculées par bootstrap sur l'échantillon d'instances) de l'apprentissage individuel effectué au chapitre 4 (en rouge) et de l'apprentissage structuré (en vert).

On observe que la courbe ROC de l'apprentissage structuré est située au dessus de celle de l'apprentissage classique sur tout la plage $[0, 1]$ du FPR.

Méthode	AUC
Apprentissage individuel	90.97 (±0.52) %
Apprentissage structuré	94.42 (±0.44) %
Structuré - individuel	3.45 (±0.68) %

TABLE 5.2 – Aires sous la courbe ROC (AUC) pour le modèle d'apprentissage individuel (chapitre 4) et pour l'apprentissage structuré.

L'utilisation de l'apprentissage structuré permet donc d'améliorer (+3.45%) les résultats de la classification.

5.4 Conclusions du chapitre

Dans ce chapitre, nous avons expérimenté deux approches globales. La première, orientée segmentation d'images, possède l'avantage de pouvoir s'abstraire de la définition spatiale des instances, chaque vignette pouvant ainsi contenir un nombre quelconque d'éléments à détecter. Les limites de l'expérimentation résident principalement dans le manque de données d'entraînement pour pouvoir sélectionner et calibrer de manière rigoureuse un modèle de réseau de neurones. Les résultats ont cependant mis en évidence le potentiel de la méthode, et surtout sa simplicité en termes de pré-traitements, en comparaison des approches mises en œuvre dans les chapitres 3 et 4. Dans un seconde approche, nous avons utilisé un graphe probabiliste non-dirigé pour modéliser et ainsi exploiter l'autocorrélations des variables cibles sur le réseau routier. Les limitations dues à l'effectif réduit du jeu de données semblent moins pregnantes dans ce cadre d'application, ce qui nous a permis de montrer l'amélioration apportée par l'apprentissage structuré. Trois perspectives d'amélioration peuvent être envisagées :

- En premier lieu, on peut envisager d'utiliser un nombre plus élevé de types de fonction de potentiel ψ_k pour modéliser les différents types de liens.

- Deuxièmement, notons que le modèle de graphe a été défini de manière arbitraire, en considérant l'adjoint du graphe routier sur-échantillonné comme un bon candidat pour le problème à résoudre. Il pourrait être intéressant d'expérimenter une construction automatique du modèle, par exemple à l'aide des techniques décrites dans Friedman et al. (2008).

- Enfin, le modèle d'Ising peut être naturellement étendu en modélisant les variables cibles sous forme de vecteurs *one-hot*[6], permettant de gérer la classification multinomiale, peut se révéler particulièrement adapté à la détection de tous les types d'élément de la signalisation routière. On pourra se référer à Landrieu (2016) pour plus de détails sur ce modèle.

6. Par exemple, dans un problème à 4 modalités, la valeur 3 devient le vecteur $(0, 0, 1, 0)$

Conclusion générale

Résumé des travaux

La prolifération de terminaux mobiles connectés équipés d'un récepteur GPS offre la possibilité d'acquérir de grandes quantités de trajectoires de conduite sur un même itinéraire. Récemment, les techniques dites de *map inference* ont mis à profit cette nouvelle source de données pour construire une cartographie numérique de l'infrastructure routière. Si les premiers travaux visaient essentiellement à détecter la géométrie du réseau de routes, l'augmentation progressive du nombre de traces disponibles, ainsi que l'amélioration des performances de la technologie GPS ont permis d'enrichir les réseaux existants, notamment avec la détection des éléments de signalisation. La connaissance fine et exhaustive de cette signalisation est indispensable au bon fonctionnement du véhicule autonome, qui nécessite une cartographie de référence pour pallier d'éventuelles défaillances des capteurs embarqués. À terme, le potentiel des techniques de map inference laisse entrevoir la possibilité de construire un système en boucle autonome : les systèmes de conduite des véhicules autonomes utilisant une cartographie construite et mise à jour par ces même véhicules.

Parallèlement, la popularité et les performances croissantes des algorithmes d'apprentissage statistique incitent à examiner les capacités de cette classe de méthode pour la détection de la signalisation verticale. Si quelques travaux récents ont mis en évidence leur adéquation à la détection des feux tricolores à partir de trajectoires individuelles, aucune approche n'a encore été envisagée pour détecter la signalisation par apprentisage statistique sur un ensemble de traces GPS. C'est la question à laquelle nous avons cherché à apporter des éléments de réponse dans cette thèse.

Dans un premier temps, dans le chapitre 2, nous avons passé en revue et étendu les algorithmes de pré-traitement de données, à savoir, dans l'ordre de la séquence des opérations : le recalage (ou map-matching) des données sur un réseau routier de référence, le calcul, l'interpolation et le lissage des profils spatiaux de vitesse. Nous y avons mené une étude théorique et expérimentale sur l'impact de la qualité géométrique du réseau de référence sur le gain de précision apporté par la procédure de map-matching. Nous y avons également présenté un algorithme de reconstruction de trajectoires GPS à partir de mesures auxiliaires de cap et d'accélération, dans le cadre de la détection de carrefours giratoires.

Le chapitre 3, qui constitue la partie principale du travail de thèse, cherche à évaluer différents algorithmes d'apprentissage et différentes représentations mathématiques des instances, pour le problème de la détection de feux tricolores sur des données expérimentales. Notre méthodologie a consisté à découper l'ensemble des profils de vitesse en fenêtres glissantes de 100 m de longueur. Trois types de description ont alors été proposés : dans une premier mode, qui compose notre approche de base pour les comparaisons, les mesures

brutes de vitesse (puis ultérieurement d'accélération) sont concaténées dans un vecteur de descripteur ; le deuxième mode cherche à expérimenter une approche image adaptée depuis la littérature ; enfin, nous testons une approche fonctionnelle inspirée des travaux de Andrieu (2013) et Gregorutti (2015), en agrégant les différents profils de vitesse collectés dans la fenêtre, et en les projetant sur une base d'ondelettes de Haar. Ces trois approches ont été évaluées à l'aide d'un panel représentatif d'algorithmes d'apprentissage. Les expérimentations ont confirmé la supériorité de l'approche fonctionnelle, à la fois en termes de précision de détection, mais aussi en temps de calcul et en pouvoir de discrimination feux/stops, en particulier lorsque combinée avec une forêt d'arbres aléatoires. Plusieurs expérimentations sont menées à la suite pour étudier le comportement de l'algorithme (importance des descripteurs, convergence avec le nombre de profils, sensibilité à la précision...). La méthode est alors appliquée au cas des passages piétons, sur lequel nous avons obtenus de bons résultats.

Si les données utilisées dans le chapitre 3 étaient de nature expérimentale, avec en particulier une mesure très précise de la vitesse par effet Doppler, il est apparu important d'évaluer les performances de nos algorithmes sur les cas de données réelles. C'est précisément l'objectif du chapitre 4, qui concerne des travaux effectués au laboratoire CSIS de l'Université de Tokyo, grâce à une bourse de mobilité financée par l'École Doctorale. Les résultats expérimentaux ont mis en évidence : (1) la difficulté d'étendre l'approche fondée sur un système de fenêtres glissantes au cas d'un réseau routier complet et (2) la perte significative de fiabilité de détection associée à l'utilisation de données acquises en contexte opérationnel. Nous tentons également, avec un succès mitigé, d'étendre l'utilisation de l'apprentissage à l'inférence de la position des feux tricolores. Les résultats expérimentaux montrent toutefois la supériorité de cette approche lorsque nous l'avons comparé à un algorithme de la littérature. Dans ce chapitre, nous proposons et comparons un ensemble de méthodes de débiaisage des trajectoires GPS à partir du map-matching.

Dans le dernier chapitre, à vocation essentiellement exploratoire, nous tentons deux nouvelles approches issues du domaine de l'apprentissage pour répondre aux difficultés soulevées dans les chapitres précédents. Dans un premier temps, nous étudions la possibilité d'utiliser les techniques d'apprentissage profond (notamment via une architecture de réseau de neurones convolutionnel) pour segmenter une image globale dont chaque canal représente un aspect physique du jeu de trajectoires sur une zone donnée (cap, temps d'arrêt, accélération, vitesse, etc.). Le modèle, dont la sortie se présente sous la forme d'une carte de probabilité, permet ainsi de s'abstraire de la définition de l'emprise géométrique des instances individuelles sur le réseau routier. Les résultats ont montré le potentiel de la méthode, mais celle-ci trouve ses limites dans le faible effectif du jeu d'entraînement. Dans la seconde partie du chapitre, nous cherchons à exploiter l'auto-corrélation spatiale des instances individuelles pour améliorer les prédictions. Nous modélisons le problème sous forme d'un champ de Markov conditionné aux prévisions a priori effectuées par l'algorithme de base présenté au chapitre 4, et dont les potentiels d'interaction sont définis sur un graphe topologique dérivé du réseau routier. La méthode proposée présente une possibilité d'extension naturelle au cas de la classification multinomiale, dans laquelle on chercherait à détecter simultanément plusieurs types d'éléments de la signalisation routière. Par ailleurs, les résultats préliminaires montrent une amélioration significative des performances par rapport au cas de l'apprentissage non-structuré.

Perspectives

À l'aune de ces conclusions, nous voyons trois axes principaux d'amélioration.

Dans un premier temps, il nous paraît primordial de travailler sur des jeux de données beaucoup plus volumineux. Si les données FCD sont d'ores et déjà bien présentes sur le marché, et sont actuellement collectées, échangées et analysées par les constructeurs automobiles, les équipementiers, et les gestionnaires de flottes et de réseaux, leur accès par les laboratoires de recherche de taille humaine ne semble pas chose aisée. Les méthodes d'apprentissage nécessitent cependant de grosses masses de données pour entraîner mais aussi et surtout pour calibrer et valider les modèles. Nos expérimentations ont montré que la convergence des algorithmes avec le nombre de traces est atteinte très rapidement (une trentaine de passages seulement sont nécessaire pour atteindre les capacités optimales). Le volume des données à utiliser dans les travaux ultérieurs devront donc en premier lieu concerner l'emprise spatiale de la zone de collecte, et non le nombre de trajectoires effectivement collectées. Un jeu de données plus extensif nous permettra d'améliorer les performances de nos algorithmes tout en se prémunissant du risque de sur-apprentissage, et nous permettrait également d'évaluer les potentialités de méthodes plus ardues à entraîner, telles que les réseaux de neurones profonds.

Après la question du jeu de données, un deuxième axe de recherche se situerait très certainement dans l'extension des modèles proposés au cas de la classification multinomiale, avec à terme l'objectif de disposer d'un programme autonome de reconnaissance de tous les éléments de la signalisation (marquages au sol, cassis, non-communications, etc.). En ce sens, nous avons montré dans le chapitre 5, l'adéquation de l'apprentissage structurée, notamment avec le modèle de Potts (extension du modèle d'Ising proposé). Parallèlement, une alternative peut consister à entraîner l'algorithme exposé au chapitre 3 suivant un mode *one against all*, dans lequelle on construit une famille de classifieurs, chacun d'eux étant chargé de détecter un type précis d'élément de signalisation. L'avantage de cette approche par rapport à l'apprentissage structuré tient dans le fait qu'il s'agit d'une application potentiellement directe des méthodes étudiées dans cette thèse. Par ailleurs, notons que la modélisation à base de profils spatiaux de vitesse rend l'approche très générique et en principe directement transposable aux autres éléments à détecter, comme nous l'avons montré expérimentalement avec le cas des passages piétons. De plus, remarquons que certains éléments de l'infrastructure routière (en particulier les éléments non ponctuels, *e.g.* limitations de vitesse, marquages au sol, nombre de voies...) peuvent nécessiter le développement de méthodes alternatives de représentation des instances.

Enfin, l'accessibilité des données xFCD (en plus de la trajectoire brute) rend intéressante l'étude de l'ajout de ces variables auxiliaires dans le processus de détection de la signalisation routière, constituant de ce fait un troisième axe important dans les travaux futurs. Dans cette thèse, les données xFCD ont été brièvement utilisées dans le chapitre 2 pour lisser les trajectoires, mais ne sont pas intervenues explicitement dans la phase d'apprentissage. Pourtant, des données comme la pression sur les pédales, l'embrayage ou encore les valeurs d'accélérations ou de jerks, peuvent fournir des informations précieuses dans le cadre de notre problématique, mais soulèvent en même temps de nombreux défis en termes de *big data* et de protection de la vie privée.

Annexe A : Positional Accuracy Control in Dense Urban Environment with Low-Cost Receiver and Multi-Constellation GNSS

Dans cet annexe, nous reproduisons un article publié dans les actes de la conférence Multi-GNSS Asia 2017 (Méneroux et al., 2017).

Abstract : In this work, we investigate the effectiveness of a method leveraging multi-GNSS positioning with low-cost receiver to control the accuracy of a set of predefined points with short time acquisition and reduced sky view factor. Final accuracy is required to be sub-meter level. 20 positions located in Tokyo (Japan) have been observed for 6 minutes with a u-blox NEO M8T receiver and a L1-band antenna. Solutions have been computed with the free software RTKlib, for different combinations of satellite constellations. Eventually, we derived a probabilistic upper bound on the controlled points root mean square error (rmse) based on Rayleigh distribution and the central limit theorem. Results highlighted that completing GPS with one additional constellation may markedly reduce the predicted rmse as well as the convergence time. The gain in accuracy has been found to be more moderated when a maximal number of constellations is added, despite solution is reached much faster. We shall notice however that these findings may be dependent upon the selected mask angle for the computation. Some further analysis is also required to assess more precisely the respective contributions of GLONASS and Beidou systems in the overall enhancement.

Keywords : Positional Accuracy, Spatial Data Quality Control, Multi-GNSS

INTRODUCTION

In geospatial sciences, it is often required to collect ground truth data in order to validate the efficiency of the proposed methods. Most of the time, this is done with on-the-field survey, or by digitizing shapes on orthoimages. Though the latter approach is much faster, it is heavily relying on the assumption that the available aerial views have been orthorectified to a satisfactory level of precision. This is obviously not always a reasonable assumption, and it is not uncommon to work on orthoimages whose planimetric accuracy is believed to be up to 20 meters [1], especially in remote locations where the existing geodetic network may not be precise enough to ensure optimal correction of image geometry. This might prove to be critical in applications, where no margin is allowed for positional inaccuracy in the reference ground truth and metric or sub-metric precision is sought. To face this issue, a common approach consists in controlling the ground truth by surveying

a small (but sufficient) number of positions (or shapes) randomly sampled in the digitized dataset, in order to assess statistically their positional accuracy [1]. For most purposes, a total of 30 control points might be enough.

The most direct and efficient way to survey independent positions, is to use a GPS receiver, and observe each point during a predefinite time. When the positional accuracy is not a critical parameter of the study, it might be enough to use a common GPS receiver, similar to those which are embedded in cell phones or used for hiking purpose, with a standard deviation around 5 to 7 m in general [2]. When higher precision is required, until a few years ago, it was necessary to use expensive professional carrier phase measurement receivers (observing both L_1 and L_2 frequencies), while final positions are computed through post-processing in static differential mode with a base station in the vicinity.

However, recently some intermediary solution may be found in single phase GPS receiver modules, connected to a low-cost antenna, for a total price below two hundred dollars. These receivers are usually light, simple to use, reliable, with low power consumption and ability to track concurrently multiple constellations, making them especially adequate to control survey. This study aims at assessing the capabilities of this type of receiver, for fast and accurate positional accuracy control of ground truth positions in urban environment.

As a more general context, this analysis has been conducted as part of a *map inference* research work in Tokyo (Japan). As opposed to traditional cartographic techniques, map inference aims at deriving geographic information from GPS probe vehicles [3]. Though initially restricted at constructing roads geometry, these techniques are now extended to networks enhancement (e.g. speed limitations, road infrastructure, pedestrian crosswalks...). More recently, machine learning is used to statistically infer traffic signal positions, which constitutes a generic approach, adaptable to every kind of environment, while in turns it requires to get accurate ground truth data to train the algorithms. However, as it is also the case here, control points to survey are often located in dense urban area, with poor sky view factor [4], hence reducing the potential number of available satellites. Fortunately, East Asia is covered by both global and regional systems, making it possible to view significantly more satellites at any time, compared to any other part of the world [5].

As a specific application case of this problem, we investigate a method using low-cost equipment and relying on multiple GNSS constellations, for a fast, easy to carry out and accurate (enough) ground truth precision control in Tokyo dense urban environment.

The remaining of the paper is structured as follows : second part introduces experimentation objectives and methodology while results are detailed and analyzed in section 3. Eventually section 4 concludes the paper.

METHODOLOGY

The ground truth dataset contains a total of 669 positions of traffic signal stop lines in Mitaka (Greater Tokyo Area), digitized from Google Maps orthoimages and whose coordinates have been converted in UTM (zone 54) projection system. Throughout all this paper, a *stop line* is defined as the position along the road, where the front vehicle in queue is expected to stop while waiting for the signal to turn green. In Tokyo (and more generally

in Japanese cities), there is no ambiguity regarding the position of this line, since it is almost systematically painted on the surface of the road. Most of the positions to survey are featured by very low sky-view factor (similarly to figure 1).

FIGURE 26 – Example of a stop line position to survey, with reduced sky view

The front left corner of each line are then recorded in the ground truth dataset, whose positional accuracy is completely uncertain, since aerial pictures distortion might have resulted in large error on digitized point coordinates.

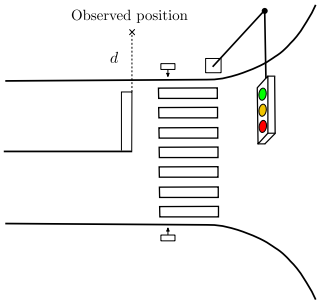


FIGURE 27 – Stop line survey protocol. The safety distance d is measured parallely to the line.

Note that, due to the fact that it was not practically feasible to survey directly the stop line, we observed a surrogate point at a distance d in orthogonal direction respectively to the road axis (figure 2). The numerical value of d is then used to correct the coordinates to the original position.

For the remaining of the project, it was necessary to get a sub-meter accuracy, so we performed a GPS survey experimentation to control the collected positions. Since, in most cases, it is not realistic to survey all positions included in the dataset, we randomly sampled 20 positions (without replacement), and surveyed each of them successively for 6 minutes. Observation was performed with a u-blox NEO M8T receiver and a TW4721 antenna (both operating only on $L1$ frequency) for a total cost below 150 USD. The choice of the antenna was motivated by the fact that it contains two orthogonally oriented dipoles, resulting in

an improved multipath signal rejection, which is of utmost importance in urban survey. Recording of observables in ubx format is carried out with a classical laptop and u-center software.

After conversion to rinex format, data are processed with RTKlib software [6]. Computation has been done with the following combinations of GNSS [7].

n°	GPS	GLONASS	Galileo	QZSS	Beidou
1	x				
2	x			x	
3	x	x			
4	x				x
5	x	x	x	x	
6	x		x	x	x

TABLE 3 – Combinations of constellations for positioning

Note that since concurrent reception of GLONASS and Beidou is not allowed on the receiver, 15 points have been observed with all systems except Beidou, while on the 5 remaining points, GLONASS observations were excluded.

Each position is computed in differential static mode, with a base station located on the roof of *Kaiyō University* in the western part of Tokyo, with a 10.4 km-long baseline. Computed solutions are then compared to original ground truth coordinates.

It is important to keep in mind that surveyed positions are also uncertain (due to multipath effects, it is impossible to guarantee that survey error is small in front of the error on digitized positions). Consequently, instead of getting a fixed accuracy estimate, we can only provide a probabilistic upper bound on this quantity. More formally, assuming root mean square distance ρ between digitized and surveyed positions is known (after experimentation), with a confidence level c, the actual unknown root mean square error of digitized positions r is constrained by the following inequality :

$$r \leqslant \rho\Big(1 + \frac{\Phi^{-1}(c)}{\sqrt{n}}\Big)^{1/2} \tag{22}$$

where Φ is the normal cumulative distribution function, n is the number of controlled points and $0 \leqslant c \leqslant 1$. This inequality holds under the assumption that central limit theorem conditions are valid [8] (the mathematical proof is provided in annex).

For each combination of constellations, the results given in the following section are the upper bounds on r at 95% confidence level.

RESULTS AND DISCUSSION

Table 2 contains all results for a minimal ratio to fix ambiguity set equal to 3.0 (as suggested in the default parameters) with a 25° elevation mask, which seems to be a reasonable value in urban environment. Ambiguity resolution mode has been set to *instantaneous*.

n°	Accuracy (m)	# satellites	F/F (%)
1	1.77	6.18	64.7
2	1.22	7.00	76.5
3	1.69	11.00	60.0
4	0.95	11.33	33.3
5	1.34	13.64	28.6
6	1.39	14.00	50.0

TABLE 4 – Accuracy, average number of satellites and percentage of fixed ambiguities for each combination of constellations

We also tried to compute solutions in floating mode, but this option was eventually rejected since it resulted in practically imprecise positions.

Figure 3 depicts convergence time to reach a level of accuracy below 1 m, for each combination (excluding those containing Beidou for which we could not get enough data to provide statistically significant values).

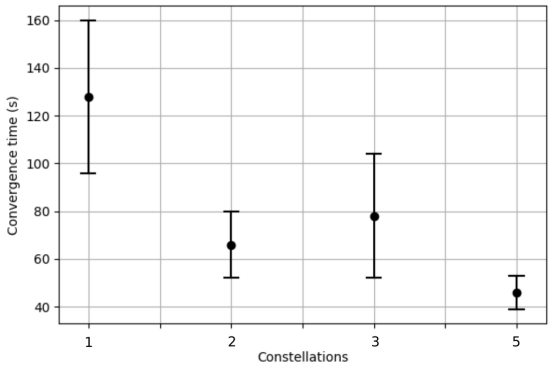


FIGURE 28 – Time to reach 1-m accuracy level for each combination of constellation with 95% confidence interval

Results demonstrate that the positional accuracy of the control survey is markedly improved when combining GPS and Beidou constellations, with almost twice as many visible satellites on average, resulting in a 46% decrease in the rmse. We shall notice that with this configuration, only 33% of ambiguities are fixed, which is one of the lowest scores among all combinations. It is surprising however that, with about the same number of tracked satellites, GLONASS enhancement provides only slight improvement on the GPS-alone accuracy.

With an average of about one additional satellite, QZSS system has a very significant

and positive impact both on the final accuracy (best combination after Beidou) and the rate of fixed ambiguities (76.5%).

Yet, relying on the maximal number of constellations does not improve solution as much as we might have expected, with a rmse between 1.3 and 1.4 m, despite as many as 14 satellites were visible on average. However, we shall notice that the dispersion of errors (on the 20 positions) seems to be more moderated with this maximal combination, compared to the single enhancement of Beidou or QZSS, meaning that cumulating the signal of all satellites provides more robust estimate, albeit slightly less accurate. Some more refined analysis are required to confirm this observation.

On the other side, while relying on too many constellations does not seem to improve significantly the position accuracy, final solution is unarguably reached much faster, with a convergence time up to 3 times shorter than GPS-alone configuration, and 1.5 times shorter than both QZSS and GLONASS augmentation. Further work will try to use these findings to derive a practical time of survey, related to the final expected precision.

It is important to note that in all these results, we kept a constant mask, while this parameter may significantly impact the quality of the positioning, especially in urban environment. However, our methodology is developed for fast and inexpensive control of positional accuracy, which implies that we want to spend minimal time for tuning the model, hence we selected a reasonable value for the mask elevation, and we kept it constant for all the survey, regardless of the configuration of each point environment.

CONCLUSION

In this study, we proposed a methodology to control the positional accuracy of a ground truth dataset, with low-cost equipment and fast survey. Since, both original points and GNSS solutions have unknown precision, we derived a probabilistic upper bound on the final root mean square error of the dataset to control. Experimental results analysis revealed that when enhancing GPS, both Beidou or QZSS (separately), seem to improve significantly the final accuracy with a moderate reduction of the convergence time. On the contrary, adding the maximal number of constellations result in a slightly less accurate position, while solution is on average reached much faster.

Next research steps will aim at further investigating the exact contributions of Beidou and GLONASS in the enhancement. We would like as well to perform more experiments (including different antenna and baseline lengths), in order to derive a practical time of survey recommendation for this methodology, apportioned to the number of controlled points, the expected accuracy and the sky view factor.

APPENDIX : Proof of equation 22

In this section, we provide a mathematical proof of the probabilistic upper bound on the root mean square error of digitized coordinates after survey, as provided in equation 22.

Let us denote p_1 and $p_2 \in \mathbb{R}^2$, respectively the digitized and surveyed observations of a given unknown point p. Furthermore, we will assume that each observation is expressed as $p_i = p + \varepsilon_i$ where ε_i is sampled from a two-dimensional normal distribution whose covariance matrix is equal to $\sigma_i I_2$ (standard deviations are equal in both x and y directions). Hence, the distance $\Delta^2 = ||p_1 p_2||^2$ is expressed as : $\Delta_X^2 + \Delta_Y^2$ with both Δ_X and Δ_Y varying according to the normal distribution $\mathcal{N}(0, \sigma_1^2 + \sigma_2^2)$. As a consequence, Δ follows a Rayleigh distribution :

$$R(x) = \frac{x}{\sigma_1^2 + \sigma_2^2} \exp\Big[-\frac{1}{2} \frac{x^2}{\sigma_1^2 + \sigma_2^2} \Big]$$

It may be easily demonstrated that the maximum likelihood estimator of the quantity $\sigma^2 = \sigma_1^2 + \sigma_2^2$ is :

$$\widehat{\sigma^2} = \frac{1}{2n} \sum_{i=1}^{n} \Delta_i^2$$

whose the variance computation is straightforward :

$$Var(\widehat{\sigma^2}) = \frac{1}{4n^2} \sum_{i=1}^{n} Var(\Delta^2) = \frac{\sigma^4}{n}$$

since the quantity Δ_i^2/σ^2 is following a χ^2 distribution with $k = 2$ degrees of freedom (with variance equal to $2k$), implying as a direct consequence that the variance of Δ_i^2 equals $4\sigma^4$.

Under the hypothesis that n is large enough to assume convergence of the results provided by the central limit theorem (typically $n \geqslant 30$), we may express the difference between the estimated and the actual (unknown) variance parameter in terms of probability.

$$\forall\ t \in \mathbb{R}^{+*} \quad \mathbb{P}\Big(|\sigma^2 - \widehat{\sigma^2}| \leqslant t \frac{\widehat{\sigma^2}}{\sqrt{n}} \Big) = \frac{1 + \Phi(t)}{2}$$

where Φ is the normal cumulative distribution function.

$$\mathbb{P}\Big(-t \frac{\widehat{\sigma^2}}{\sqrt{n}} \leqslant \sigma^2 - \widehat{\sigma^2} \leqslant t \frac{\widehat{\sigma^2}}{\sqrt{n}} \Big) = \frac{1 + \Phi(t)}{2}$$

$$\mathbb{P}\Big(\sigma^2 \leqslant \widehat{\sigma^2}(1 + t n^{-\frac{1}{2}}) \Big) = \Phi(t)$$

Then, recalling that $\widehat{\sigma^2}$ is equal to $\rho^2/2$ and through the substitution $t = \Phi^{-1}(c)$, with a confidence level $c \in [0, 1]$, the sum $2\sigma_1^2 + 2\sigma_2^2$ (factor 2 is to take into account the error in both directions x and y) is lower than :

$$T_n(c) = \rho^2\Big(1 + \frac{\Phi^{-1}(c)}{\sqrt{n}}\Big)$$

A fortiori, the inequality holds for each separate term and this concludes the proof. As a byproduct, we have the following asymptotic equality :

$$\lim_{n\to\infty} T_n(c) = \rho^2$$

which is independent of the confidence level c, or in other words, as we might have expected, the mean square error of the digitized positions is asymptotically smaller than the observed quadratic distance between digitized and surveyed points.

[1] D. Potere. "Horizontal Positional Accuracy of Google Earth's High-Resolution Imagery Archive". Sensors. 8(12), 7973-7981 July 2008.

[2] E.D. Kaplan, J.C. Hegarty. "Understanding GPS : Principles and Applications". Second Edition, Artech House, 2006

[3] J. Biagioni, J. Eriksson. "Inferring road maps from global positioning system traces : Survey and comparative evaluation". Transportation Research Record : Journal of the Transportation Research Board (2291). pp 61–71. 2012.

[4] T. Suzuki, N. Kubo, "Simulation of GNSS Satellite Availability in Urban Environments Using Google Earth", Proceedings of the ION 2015 Pacific PNT Meeting, Honolulu, Hawaii, April 2015, pp. 1069-1079.

[5] B. Li, S. Zhang, A.G. Dempster, C. Rizos. "Impact of RNSS on positioning in the Asia-Oceania region", Journal of Global Positioning, Vol 10. No. 2 : 114-124, 2011.

[6] Takasu, T., Kubo, N. and Yasuda, A., RTK-GPS, GPS/GNSS symposium 2007, Tokyo, Japan, 20-22 November (2007).

[7] X. Li, M. Ge, X. Dai, M. Fritsche, J. Wickert, H. Schuh. "Accuracy and reliability of multi-GNSS real-time precise positioning : GPS, GLONASS, BeiDou, and Galileo". Journal of Geodesy. Vol 89, 6, pp 607-635. June 2015.

[8] P. Brown. "Measure Theory and the Central Limit Theorem". August 24, 2011.

Annexe B

Dans cette section, nous détaillons les développements menés dans la section 4.2.

On note $K : \mathbb{R} \to \mathbb{R}^+$ une fonction noyau, c'est-à-dire symétrique, à valeurs positives et d'intégrale 1 sur l'ensemble des réels. De plus, on impose que K soit du second ordre, *i.e.* $\int_{\mathbb{R}} u^2 K(u)du < \infty$, et on note κ la quantité correspondante.

Soient $x_1, x_2, ...x_n \in [0, L]$ un ensemble de n points d'arrêt, associés à des temps d'arrêt $t_1, t_2, ...t_n \in \mathbb{R}^+$ (pour des raisons d'efficacité numérique, nous supposerons plus loin que les timestamps sont précis à la seconde près, permettant ainsi de considérer que $t_i \in \mathbb{N}$). Pour un facteur d'échelle $h \in \mathbb{R}^+$, on note K_h la transformée de la fonction K par une homothétie de rapport h :

$$K_h(u) = \frac{1}{h} K\Big(\frac{u}{h}\Big) \tag{23}$$

Soit $(X, T) \in \mathbb{R} \times \mathbb{R}^+$ un couple de variables aléatoires de loi jointe $\pi(x, t)$, représentant respectivement la position et le temps d'un arrêt. On note $\pi(x) = \int_{\mathbb{R}^+} \pi(x, t)dt$ la distribution marginale des positions d'arrêt et $\pi(t|x) = \pi(x, t)/\pi(x)$ la distribution conditionnelle des temps d'arrêt sachant la position. On définit alors l'estimation de densité par noyaux *pondérée* par :

$$\forall\, x \in [0, L] : \qquad \hat{f}_h(x) = \frac{1}{N} \sum_{i=1}^{N} t_i K_h(x - x_i) \tag{24}$$

D'autre part, on pose :

$$f(x) = \mathbb{E}[T|x]\pi(x) \tag{25}$$

Calculons l'espérance de l'estimateur, en un point x et pour une largeur de bande h donnés :

$$\begin{aligned}
\mathbb{E}[\hat{f}_h(x)] &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[T.K_h(x - X)] = \mathbb{E}[T.K_h(x - X)] \\
&= \int_{x'=-\infty}^{+\infty} \int_{t'=0}^{+\infty} t' K_h(x - x')\pi(t'|x')\pi(x')dt'dx' \\
&= \int_{x'=-\infty}^{+\infty} K_h(x - x') \Big[\int_{t'=0}^{+\infty} t'\pi(t'|x')dt' \Big] \pi(x')dx'
\end{aligned}$$

$$= \int_{x'=-\infty}^{+\infty} K_h(x-x')\mathbb{E}[T|X=x']\pi(x')dx'$$

$$= \int_{x'=-\infty}^{+\infty} \frac{1}{h}K\Big(\frac{x-x'}{h}\Big)f(x')dx'$$

En posant $x' = x + uh$, et par symétrie du noyau K, on transforme l'intégrale en :

$$\mathbb{E}[\hat{f}_h(x)] = \int_{u=-\infty}^{+\infty} K(u)f(x+uh)du$$

D'autre part, on peut exprimer la valeur prise par f en $x + uh$ à l'aide de son développement limité à l'ordre 2 :

$$f(x+uh) = f(x) + uhf'(x) + \frac{u^2h^2}{2}f''(x) + o(h^2)$$

D'où, par linéarité de l'intégrale, puis en utilisant les propriétés intégrales du noyau (normalisation, symétrie et variance finie) :

$$\mathbb{E}[\hat{f}_h(x)] \approx \int_{u=-\infty}^{+\infty} K(u)\big(f(x) + uhf'(x) + \frac{u^2h^2}{2}f''(x)\big)du$$

$$= f(x)\int_{u=-\infty}^{+\infty} K(u)du + hf'(x)\int_{u=-\infty}^{+\infty} uK(u)du + \frac{h^2}{2}f''(x)\int_{u=-\infty}^{+\infty} u^2K(u)du$$

On obtient un approximation de l'espérance de notre estimateur.

$$\mathbb{E}[\hat{f}_h(x)] \approx f(x) + \frac{1}{2}h^2f''(x)\kappa$$

Cette dernière équation montre bien que $\hat{f}_h(x)$ est un estimateur de la quantité $f(x) = \mathbb{E}[T|x]\pi(x)$.

La quantité résiduelle $\frac{1}{2}h^2f''(x)\kappa$ traduit donc le biais de l'estimateur en x. Pour obtenir l'erreur globale due au bias, on intègre son carré sur le domaine de x :

$$B(h) = \int_{\mathbb{R}} \frac{1}{2}h^2f''(x)\kappa = \frac{h^4\kappa^2R(f'')}{4}$$

avec : $R : L^2(\mathbb{R}) \to \mathbb{R}^+$ l'opérateur qui à une fonction f associe la valeur $\int_{\mathbb{R}}[f''(x)]^2dx$. $R(f'')$ traduit donc l'irrégularité de la fonction, et le biais total $B(h)$ croît à mesure que $R(f'')$ augmente. Inversement B tend vers 0 quand h tend vers 0. L'estimateur est asymptotiquement non-biasé.

Annexe C : Produits informatiques

Dans cette annexe, nous exposons les principaux développements informaatiques opérationnels (et dont la plupart sont en accès livre sur le dépôt de l'IGN), effectués au cours de la thèse.

.1 Roc4j : une librairie Java dédiée aux courbes ROC

Le package roc4j est conçu pour estimer et traiter les courbes ROC (Receiver Operating Characteristics) des classifieurs binaires en Java.

Parmi les principales caractéristiques de roc4j :

- **Calcul de la courbe ROC :** à partir des probabilités inférées sur un jeu de validation et à la résolution souhaitée.

- **Filtrage et lissage de la courbe ROC :** ces fonctionnalités de lissage sont particulièrement utiles lorsque l'effectif du jeu de données de validation est trop faible pour garantir une estimation robuste des courbes ROC. Trois algorithmes principaux sont disponibles dans le package pour lisser les courbes :

  — La *régression binormale*, qui suppose que les distributions des instances positives et négatives dans l'espace de notation sont normales, ce qui permet d'écrire la courbe ROC sous une forme paramétrique, facilitant ainsi l'inférence, même avec un jeu réduit.

  — La *convexification*, qui élimine l'ordre d'arrivée arbitraire des instances dans le processus de calcul des courbes en calculant par défaut son enveloppe convexe.

  — L'*estimation par noyaux*, qui constitue un estimateur statistique non-paramétrique de la courbe ROC, permettant ainsi un contrôle renforcé de la part de l'utilisateur sur le processus de calcul.

- **Calcul des bandes de confiance :** avec 4 méthodes d'estimation différentes, pour la plupart décrites dans Macskassy et Provost (2004) :

  — Vertical averaging

  — Threshold averaging

  — Kolmogorov-Smirnov

— Fixed-width band

- **Tracés graphiques :** exportables en png, svg, jpg...
- **Gestion des processus de validation :** implémentant directement les principaux processus de validation croisée (k-fold cross validation, leave-one-out...), rendant ainsi le calcul de la courbe ROC complètement intégré au modèle de classifieur.
- **Calcul optimal des points de fonctionnement :** entièrement paramétrable en fonction du contexte (préférence pour les faux positifs, faux négatifs...).
- **Calcul des aires sous la courbe :** avec bandes de confiance associées.
- **Bootstrap des instances :** pour étudier la variabilité de l'estimateur de la courbe ROC.

Le code complet, ainsi que la documentation en ligne et en pdf et plusieurs exemples de base sur des jeux de données simples et originaux, sont à disposition à l'adresse suivante :

`http://recherche.ign.fr/labos/cogit/demo/roc4j-doc/index.html`

On donne à suivre quelques exemples graphiques de résultats retournés par roc4j.

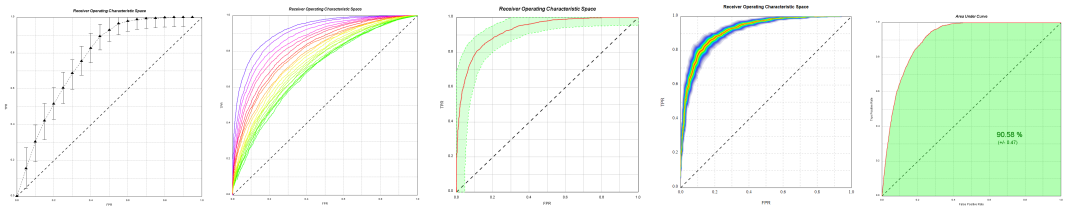


FIGURE 29 – Plusieurs exemples de courbes ROC avec bandes de confiance.

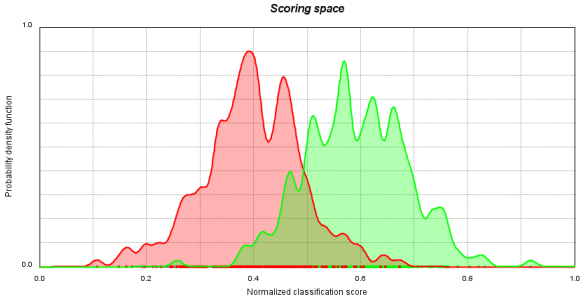


FIGURE 30 – Probabilités conditionnelles des instances de validation.

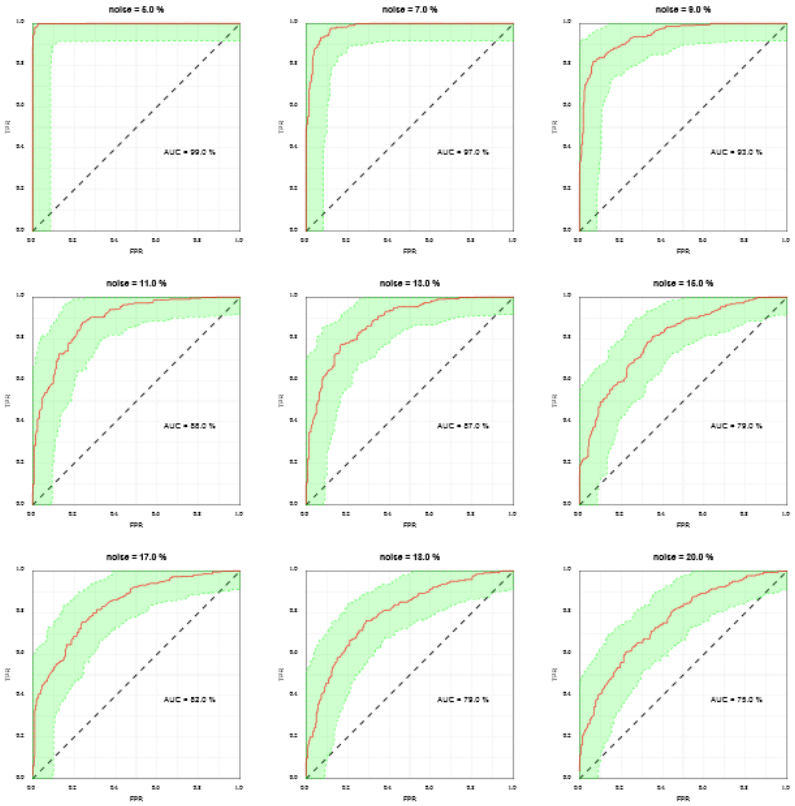


FIGURE 31 – Plusieurs exemples de courbes ROC générées sur des données synthétiques.

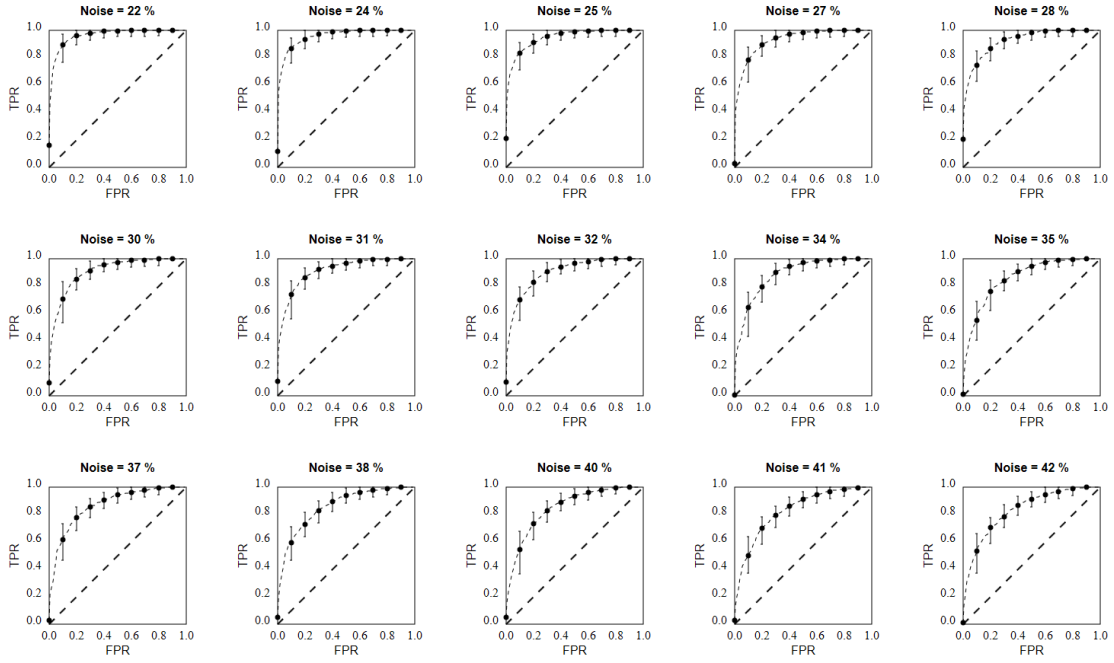


FIGURE 32 – Plusieurs exemples de courbes ROC générées sur des données synthétiques.

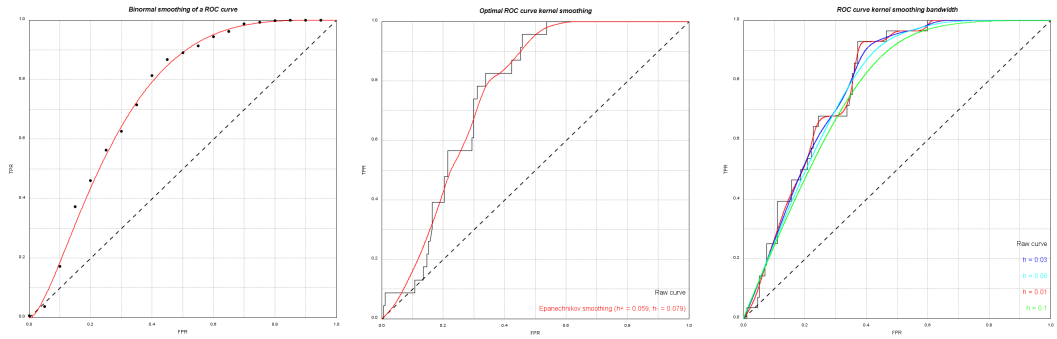


FIGURE 33 – Plusieurs exemples de regressions et de lissage de courbes ROC.

.2 Map-matcher : un programme pour recaler les traces GPS

Map-matcher implémente l'algorithme de recalage de traces de Newson et Krumm (2009), et contient également l'ensemble des améliorations décrites dans la section 2.4 du chapitere consacré 2 consacré aux pré-traitements. Il est particulièrement optimisé pour recaler un très grand nombre de traces (plusieurs centaines de milliers) sur un réseau de taille modérée (

Le logiciel comporte une version IHM (Interface Homme-Machine), ainsi qu'une version en ligne de commandes. Les deux versions nécessitent l'installation de la machine virtuelle Java.

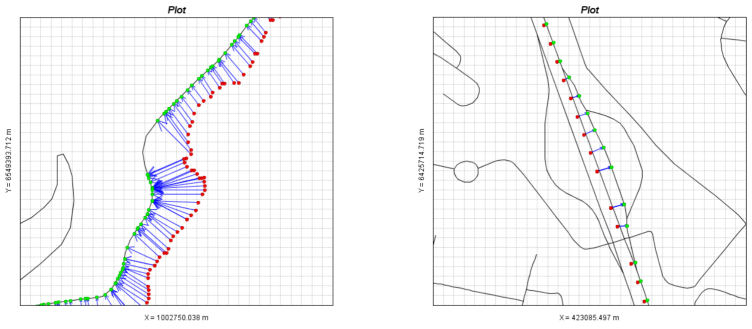


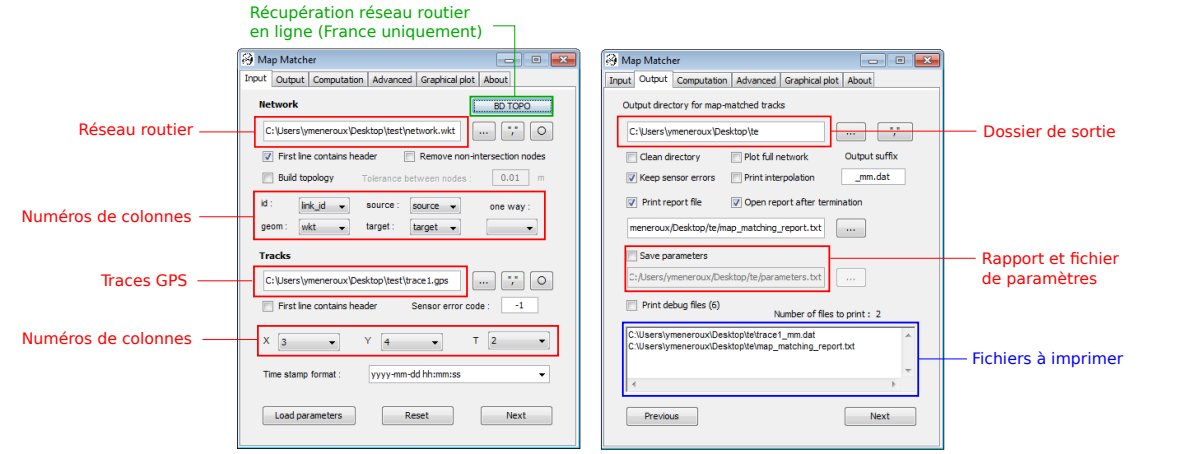
FIGURE 34 – Deux captures d'écrans du retour graphique de map-matcher. A gauche : effet d'*angle mort* dans le map-matching, nécessitant une correction longitudinale en post-traitement (filtrage de Kalman ou lissage du circuit de référence), tel que décrit dans la section 2.5. A droite : réponse du map-matching face à un tronçon routier marquer en contre-sens dans le réseau de référence.

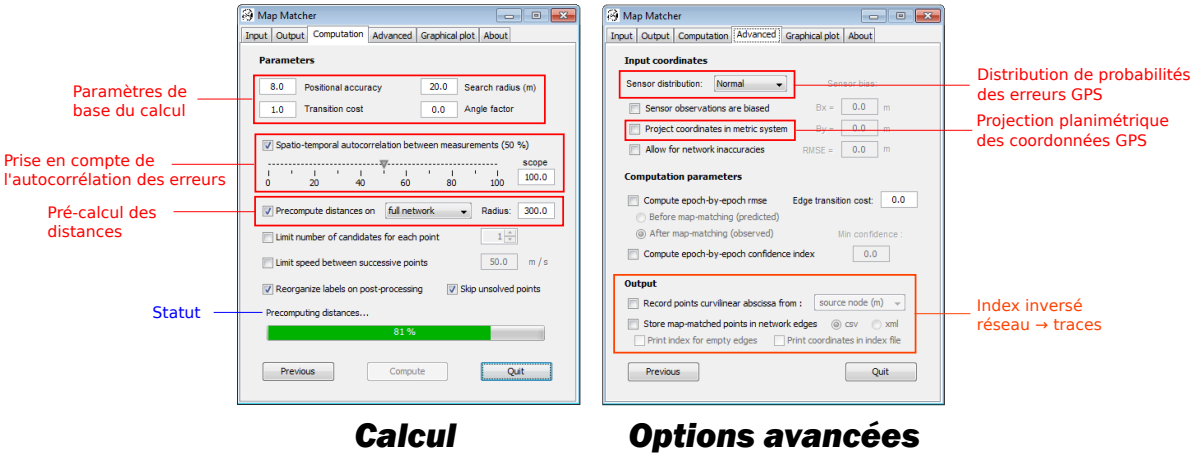
Le logiciel peut-être trouvé à l'adresse suivante :

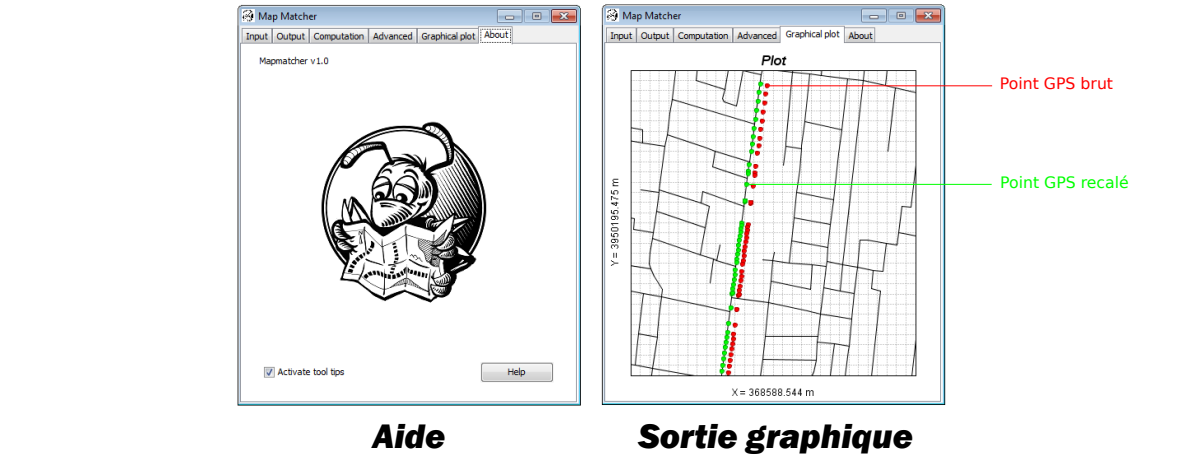
`https://github.com/IGNF/mapmatcher`

L'installation peut-être effectué via Maven.

La documentation n'étant pas encore disponible en ligne, on donne ci-dessous quelques captures d'écran explicatives :


Récupération réseau routier
en ligne (France uniquement)
Réseau routier
Numéros de colonnes
Traces GPS
Numéros de colonnes
Dossier de sortie
Rapport et fichier
de paramètres
Fichiers à imprimer
Entrées
Sorties


Paramètres de
base du calcul
Prise en compte de
l'autocorrélation des erreurs
Pré-calcul des
distances
Statut
Distribution de probabilités
des erreurs GPS
Projection planimétrique
des coordonnées GPS
Index inversé
réseau → traces
Calcul
Options avancées


Point GPS brut
Point GPS recalé
Aide
Sortie graphique

.3 PPED : un plugin d'acquisition de la vérité terrain

Ce programme a été développé en grande partie avec l'aide de M.D. Van Damme.

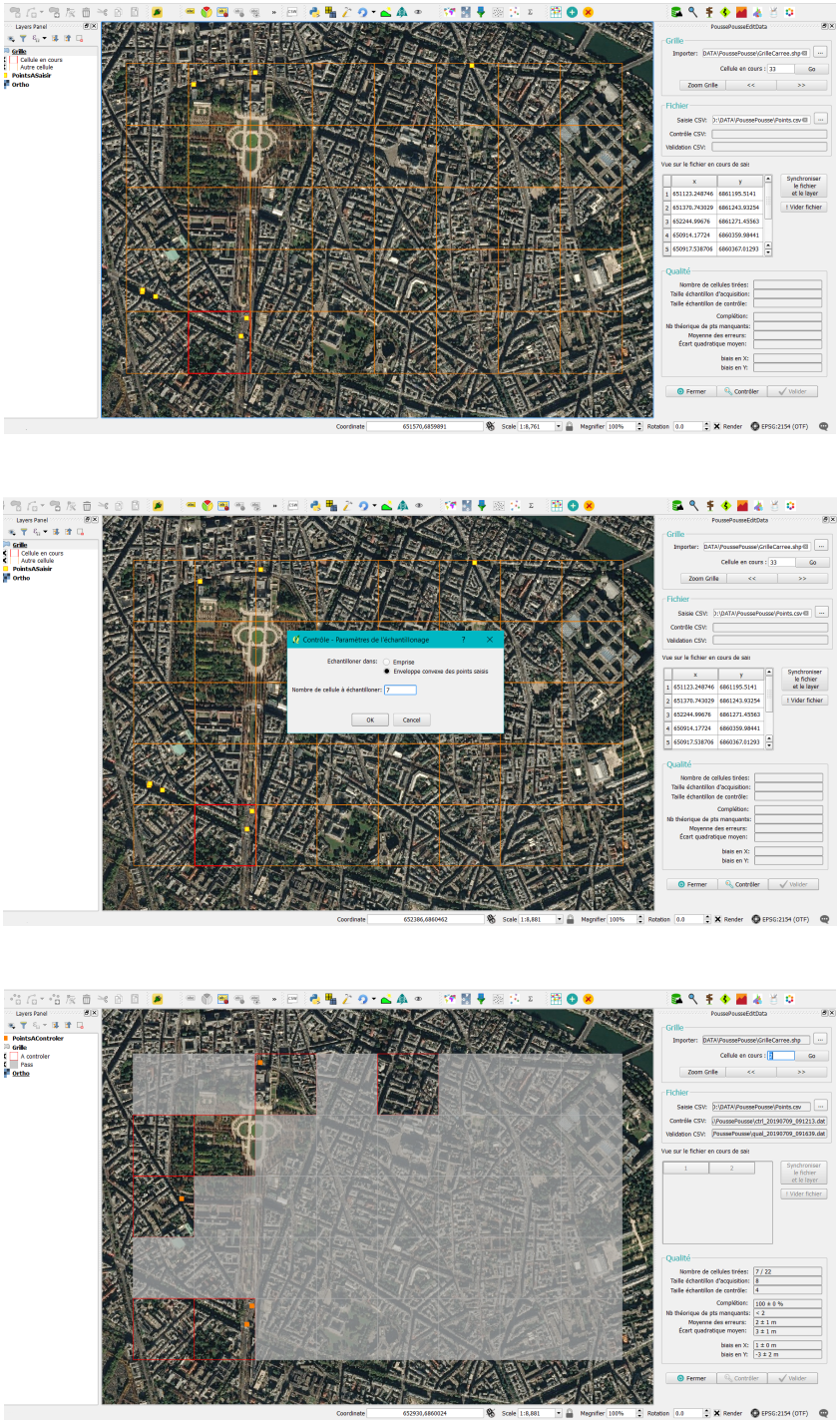


FIGURE 35 – De haut en bas : (1) Saisie des points dans une grille régulière. Le panneau de contrôle est représenté à droite de l'écran. (2) Sélection aléatoire d'un sous-ensemble de cellules pour contrôle de la qualité de saisie. (3) Résultat du contrôle de qualité (erreur quadratique moyenne, biais dans chaque direction, complétude...).

Bibliographie

Abraham, C., Cornillon, P.-A., Matzner-Løber, E., et Molinari, N. (2003). Unsupervised curve clustering using b-splines. *Scandinavian journal of statistics*, 30(3) :581–595.

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv :1803.08375*.

Ahres, Y., Janssen L., Kangaspunta, J., et Jambulapati, A. (2014). Real-time dense map matching with naive hidden markov models : Delay versus accuracy.

Aksoy, Y., Oh, T.-H., Paris, S., Pollefeys, M., et Matusik, W. (2018). Semantic soft segmentation. *ACM Transactions on Graphics (TOG)*, 37(4) :72.

Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., et Navab, N. (2016). Aggnet : deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging*, 35(5) :1313–1321.

Allen, D. M. (1971). *The prediction sum of squares as a criterion for selecting predictor variables*. University of Kentucky.

Alt, H., Knauer, C., et Wenk, C. (2001). Matching polygonal curves with respect to the fréchet distance. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 63–74. Springer.

Aminian, B., Renaudin, V., Borio, D., et Lachapelle, G. (2010). Indoor doppler measurement and velocity characterization using a reference-rover receiver. In *International Conference ION GNSS*, volume 4, pages 3069–3079.

Andrieu, C. (2013). *Modélisation fonctionnelle de profils de vitesse en lien avec l'infrastructure et méthodologie de construction d'un profil agrégé*. PhD thesis, Université Paul Sabatier-Toulouse III.

Andrieu, C., Pierre, G. S., et Bressaud, X. (2013a). A functional analysis of speed profiles : smoothing using derivative information, curve registration, and functional boxplot. *arXiv preprint arXiv :1312.2252*.

Andrieu, C., Saint Pierre, G., et Bressaud, X. (2013b). Estimation of space-speed profiles : A functional approach using smoothing splines. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 982–987. IEEE.

Aniruddha, A. K. et Babu, R. V. (2014). Visual object tracking via random ferns based classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6533–6537. IEEE.

Annunziata, C., D'APICE, C., Benedetto, P., et Luigi, R. (2007). Optimization of traffic on road networks. *Mathematical Models and Methods in Applied Sciences*, 17(10) :1587–1617.

Antoniou, C., Balakrishna, R., et Koutsopoulos, H. N. (2011). A synthesis of emerging data collection technologies and their impact on traffic management applications. *European Transport Research Review*, 3(3) :139–148.

Anuar, K., Habtemichael, F., et Cetin, M. (2015). Estimating traffic flow rate on freeways from probe vehicle data and fundamental diagram. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 2921–2926. IEEE.

Anwar, A., Zeng, W., et Arisona, S. (2014). Time-space diagram revisited. *Transportation Research Record : Journal of the Transportation Research Board*, (2442) :1–7.

Arai, A., Witayangkurn, A., Horanont, T., Shao, X., et Shibasaki, R. (2015). Understanding the unobservable population in call detail records through analysis of mobile phone user calling behavior : A case study of greater dhaka in bangladesh. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pages 207–214. IEEE.

Arnaud, M. et Emery, X. (2000). *Estimation et interpolation spatiale : méthodes déterministes et méthodes géostatistiques*. Hermès.

Attal, F. (2015). *Classification de situations de conduite et détection des événements critiques d'un deux roues motorisé*. PhD thesis, Université Paris-Est.

Auder, B. et Fischer, A. (2012). Projection-based curve clustering. *Journal of Statistical Computation and Simulation*, 82(8) :1145–1168.

Aydemir, O. et Kayikcioglu, T. (2011). Wavelet transform based classification of invasive brain computer interface data. *Radioengineering*, 20(1) :31–38.

Azizi, L. (2011). *Champs aléatoires de Markov cachés pour la cartographie du risque en épidémiologie*. PhD thesis, Grenoble.

Bahoken, F. et Olteanu-Raimond, A.-M. (2013). Designing origin-destination flow matrices from individual mobile phone paths : The effect of spatiotemporal filtering on flow measurement. In *ICC'13-26th International Cartographic Conference*, page 15p.

Bar-Gera, H. (2007). Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times : A case study from israel. *Transportation Research Part C : Emerging Technologies*, 15(6) :380–391.

Bardet, J.-B. (2006). Théorème de cochran et applications en statistiques.

Barret, M. (2009). *Traitement statistique du signal : Estimation, filtrage de Wiener, méthodes récursives, détection*. Ellipses.

Barreyre, C., Laurent, B., Loubes, J.-M., Cabon, B., et Toulouse, I. (2016). Détection d'événements atypiques dans des données fonctionnelles. *Les journées de la Statistique*.

Bartos, M., Park, H., Zhou, T., Kerkez, B., et Vasudevan, R. (2018). Vehicles as sensors : high-accuracy rainfall maps from windshield wiper measurements. *arXiv preprint arXiv :1806.10988*.

Batista, G. E. A. P. A., Prati, R. C., et Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1) :20–29.

Bejiga, M., Zeggada, A., Nouffidj, A., et Melgani, F. (2017). A convolutional neural network approach for assisting avalanche search and rescue operations with uav imagery. *Remote Sensing*, 9(2) :100.

Bel Hadj Ali, A. (2001). *Qualité géométrique des entités géographiques surfaciques : Application à l'appariement et définition d'une typologie des écarts géométriques.* PhD thesis, Université de Marne-la-Vallée.

Bendat, J. S. et Piersol, A. G. (2011). *Random data : analysis and measurement procedures*, volume 729. John Wiley & Sons.

Bentabet, L., Jodouin, S., Ziou, D., et Vaillancourt, J. (2003). Road vectors update using sar imagery : a snake-based method. *IEEE Transactions on Geoscience and Remote Sensing*, 41(8) :1785–1803.

Berlinet, A., Biau, G., et Rouviere, L. (2008). Functional supervised classification with wavelets. In *Annales de l'ISUP*, volume 52, page 19.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society : Series B (Methodological)*, 48(3) :259–279.

Besse, P. (1979). *Etude descriptive d'un processus : Approximation et interpolation.* PhD thesis.

Besse, P. et Thomas-Agnan, C. (1989). Le lissage par fonctions splines en statistique, revue bibliographique. *Statistique et Analyse des données*, 14(1) :55–84.

Besse, P. C., Guillouet, B., Loubes, J.-M., et Royer, F. (2016). Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 17(11) :3306–3317.

Bettini, C., Wang, X. S., et Jajodia, S. (2005). Protecting privacy against location-based personal identification. In *Workshop on Secure Data Management*, pages 185–199. Springer.

Biagioni, J. et Eriksson, J. (2012a). Inferring road maps from global positioning system traces : Survey and comparative evaluation. *Transportation Research Record : Journal of the Transportation Research Board*, (2291) :61–71.

Biagioni, J. et Eriksson, J. (2012b). Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 79–88. ACM.

Biau, G., Devroye, L., et Lugosi, G. (2008a). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(Sep) :2015–2033.

Biau, G., Devroye, L., et Lugosi, G. (2008b). On the performance of clustering in hilbert spaces. *IEEE Transactions on Information Theory*, 54(2) :781–790.

Bijleveld, F., Vasenev, A., Hartmann, T., et Doree, A. G. (2011). Real-time and post processing of gps data in the field of visualizing asphalt paving operations. *European Group for Intelligent Computing in Engineering (EG-ICE), Enschede, the Netherlands*, pages 1–8.

Biljecki, F., Ledoux, H., et Van Oosterom, P. (2013). Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, 27(2) :385–407.

Bodlaender, H. L. et Koster, A. M. (2010). Treewidth computations i. upper bounds. *Information and Computation*, 208(3) :259–275.

Bonin, O. (2002). Modèle d'erreurs dans une base de données géographiques et grandes déviations pour des sommes pondérées ; application à l'estimation d'erreurs sur un temps de parcours.

Bosser, P. (2011). Gnss : systèmes globaux de positionnement par satelitte, cours de l'ecole nationale des sciences géographiques.

Bostrom, H. (2007). Estimating class probabilities in random forests. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pages 211–216. IEEE.

Boureau, J.-G. (2008). *Manuel d'interprétation des photographies aériennes infrarouges : application aux milieux forestiers et naturels*. Inventaire forestier national.

Bouteloup, D. (2010). Calculs topométriques : cours de l'ecole nationale des sciences géographiques.

Bowyer, K. W., Chawla, N. V., Hall, L. O., et Kegelmeyer, W. P. (2011). SMOTE : synthetic minority over-sampling technique. *CoRR*, abs/1106.1813.

Boyer, F. (2012). Agrégation externe de mathématiques equations différentielles ordinaires.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2) :123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1) :5–32.

Breiman, L. (2004). Consistency for a simple model of random forests.

Breiman, L. et al. (1996). Heuristics of instability and stabilization in model selection. *The annals of statistics*, 24(6) :2350–2383.

Breiman, L., Friedman, J., Stone, C. J., et Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Buisson, C. (2017). Impact de la fraction de véhicules équipés de capteurs sondes et du nombre de données utilisées sur la précision d'une estimation de temps de parcours : Evaluation de la qualité métrologique par une simulation simplifiée.

Cao, L. et Krumm, J. (2009). From gps traces to a routable road map. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 3–12. ACM.

Chakrabarti, S., Dom, B., et Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD Record*, volume 27, pages 307–318. ACM.

Chalko, T. (2009). Estimating accuracy of gps doppler speed measurement using speed dilution of precision (sdop) parameter. *NU Journal of Discovery*, 6 :4–9.

Chambers, E. W. et Letscher, D. (2009). On the height of a homotopy. In *CCCG*, volume 9, pages 103–106.

Chambers, E. W., Letscher, D., Ju, T., et Liu, L. (2011). Isotopic fréchet distance. In *CCCG*.

Chen, A., Ramanandan, A., Farrell, J. A., et al. (2010). High-precision lane-level road map building for vehicle navigation. In *Position Location and Navigation Symposium (PLANS)*, pages 1035–1042.

Chen, C. et Cheng, Y. (2008). Roads digital map generation with multi-track gps data. In *Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop on*, volume 1, pages 508–511. IEEE.

Chen, Q., Song, X., Yamada, H., et Shibasaki, R. (2016a). Learning deep representation from big and heterogeneous data for traffic accident inference. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 338–344. AAAI Press.

Chen, Q., Song, X., Yamada, H., et Shibasaki, R. (2016b). Learning deep representation from big and heterogeneous data for traffic accident inference. In *AAAI*, pages 338–344.

Chen, Y. et Krumm, J. (2010). Probabilistic modeling of traffic lanes from gps traces. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 81–88. ACM.

Cheng, Q., Nouveliere, L., et Orfila, O. (2013). A piecewise vehicle fuel consumption model for eco-driving assistance system. In *International Symposium on Dynamics of Vehicles on Road and Tracks (IAVSD 2013)*, pages elec–proc.

Cheung, S. (2008). Proof of hammersley-clifford theorem. *Unpublished, February*.

Cheung, Y. K. et Daescu, O. (2009). Fréchet distance problems in weighted regions. In *International Symposium on Algorithms and Computation*, pages 97–111. Springer.

Choe, H. C., Karlsen, R. E., Gerhart, G. R., et Meitzler, T. J. (1996). Wavelet-based ground vehicle recognition using acoustic signals. In *Wavelet Applications III*, volume 2762, pages 434–446. International Society for Optics and Photonics.

Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*.

Conan-Guez, B. (2002). *Modélisation supervisée de données fonctionnelles par perceptron multi-couches*. PhD thesis, Université Paris Dauphine-Paris IX.

Cottet, F. (1997). Traitement des signaux et acquisition de données.

Criminisi, A., Shotton, J., et Konukoglu, E. (2011). Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114*, 5(6) :12.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4) :303–314.

Dabiri, S. et Heaslip, K. (2018). Inferring transportation modes from gps trajectories using a convolutional neural network. *Transportation research part C : emerging technologies*, 86 :360–371.

Dalenius, T. (1986). Finding a needle in a haystack or identifying anonymous census records. *Journal of official statistics*, 2(3) :329.

Daniels, M. (2010). Classification of percussive sounds using wavelet-based. *CCRMA, Stanford University thesis.*

Danish Ministry of Energy, U. et Climate (2017). Analysis of geospatial data requirement to support the operation of autonomous cars. Technical report.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7) :909–996.

Davies, J. J., Beresford, A. R., et Hopper, A. (2006). Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, 5(4) :47–54.

DeBarr, D. et Wechsler, H. (2009). Spam detection using clustering, random forests, and active learning. In *Sixth Conference on Email and Anti-Spam. Mountain View, California*, pages 1–6. Citeseer.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6) :141–142.

Després, B. (2010). *Lois de conservations eulériennes, lagrangiennes et méthodes numériques*, volume 68. Springer Science & Business Media.

Deville, J.-C. (1974). Méthodes statistiques et numériques de l'analyse harmonique. In *Annales de l'INSEE*, pages 3–101. JSTOR.

Dhurandhar, A. et Dobra, A. (2010). Collective vs independent classification in statistical relational learning. *Submitted for publication.*

Dietterich, T. G. et Bakiri, G. (1991). Error-correcting output codes : A general method for improving multiclass inductive learning programs. In *AAAI*, pages 572–577. Citeseer.

Dijkstra, E. W. (1959). "a note on two problems in connexion with graphs". *IEEE Transactions on Systems Science and Cybernetics SSC4*, 1 :269–271.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., Weingessel, A., et Leisch, M. F. (2009). Package 'e1071'. *R Software package, avaliable at http ://cran. rproject. org/web/packages/e1071/index. html.*

Dizier, V. et Margollé, M. (2018). Réseau de neurones convolutif pour la détection et la localisation de feux tricolores à partir de traces gps. stage de recherche des elèves-ingénieurs de l'ensg, 2eme année. document interne.

Do, T.-N., Lallich, S., Pham, N.-K., et Lenca, P. (2009). Un nouvel algorithme de forêts aléatoires d'arbres obliques particulièrement adapté à la classification de données en grandes dimensions. In *EGC*, pages 79–90.

Doche, A. (2018). Bmw, premier constructeur à intégrer les données gps partagées avec here, https ://www.caradisiac.com/bmw-premier-constructeur-a-integrer-les-donnees-gps-partagees-avec-here-165224.htm. Accessed : 2019-01-23.

Donoho, D. L. et Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3) :425–455.

Dupont, B. (2018). Haute-garonne : le département expérimente la voiture connectée avec météo france et continental, https ://www.francebleu.fr/infos/economie-social/haute-garonne-le-departement-experimente-la-voiture-connectee-avec-meteo-france-et-continental-1517931863. Accessed : 2019-01-23.

Dupré, X. (2013). Receiving operator characteristic (roc).

Dupuis, Y., Merriaux, P., Subirats, P., Boutteau, R., Savatier, X., et Vasseur, P. (2014). Gps-based preliminary map estimation for autonomous vehicle mission preparation. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4241–4246. IEEE.

Edelkamp, S. et Schrödl, S. (2003). Route planning and map inference with global positioning traces. In *Computer science in perspective*, pages 128–151. Springer.

Efron, B. (1992). Bootstrap methods : another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer.

Ehrlich, J. (2017). Quelle infrastructure pour le véhicule autonome. *Routes/Roads*, 373 :43–46.

Ehrlich, J., Bacelar, A., et S, B. (2014). Véhicules traceurs : une solution bas coût innovante et prometteuse pour la surveillance des réseaux routiers. In *Séminaire international "Technologies de prévention et de réduction des effets des catastrophes et apport des STI à l'exploitation des réseaux"*.

el Habib Boukhobza, M. et Mimi, M. (2016). Classification automatique de la densité des tissus mammaires. *Traitement du Signal*, 33 :441–460.

Ellison, A. M. (1987). Effect of seed dimorphism on the density-dependent dynamics of experimental populations of atriplex triangularis (chenopodiaceae). *American Journal of Botany*, 74(8) :1280–1288.

Eltchaninoff, N. (1996). Géoroute, une base de données routières pour la france. CFC N° 149 - septembre 1996.

Eubank, R. L. (1999). *Nonparametric regression and spline smoothing*. CRC press.

Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., Rahmati, A., et Song, D. (2017). Robust physical-world attacks on deep learning models. *arXiv preprint arXiv :1707.08945*, 1 :1.

Ferraty, F. et Vieu, P. (2006). *Nonparametric functional data analysis : theory and practice*. Springer Science & Business Media.

Flajolet, P. et Sedgewick, R. (2009). *Analytic combinatorics*. cambridge University press.

Flamary, R. (2011). *Apprentissage statistique pour le signal : applications aux interfaces cerveau-machine*. PhD thesis, Université de Rouen.

Floyd, R. W. (1962). "algorithm 97 : Shortest path". *Communications of the ACM*, 5(6) :345.

Friedman, J., Hastie, T., et Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA :.

Friedman, J., Hastie, T., et Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3) :432–441.

Fu, M.-y., Li, J., et Zhou, P.-d. (2003). Design and implementation of bidirectional dijkstra algorithm. *JOURNAL-BEIJING INSTITUTE OF TECHNOLOGY-ENGLISH EDITION-*, pages 366–370.

Gandolfi, A. et Lenarda, P. (2017). A note on gibbs and markov random fields with constraints and their moments. *Mathematics and Mechanics of Complex Systems*, 4(3) :407–422.

Gather, U. et Schultze, V. (1999). Robust estimation of scale of an exponential distribution. *Statistica Neerlandica*, 53(3) :327–341.

Geisberger, R., Sanders, P., Schultes, D., et Delling, D. (2008). Contraction hierarchies : Faster and simpler hierarchical routing in road networks. In *International Workshop on Experimental and Efficient Algorithms*, pages 319–333. Springer.

Geman, S. et Geman, D. (1987). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision*, pages 564–584. Elsevier.

Genuer, R., Poggi, J.-M., et Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14) :2225–2236.

Giambartolomei, G. (2015). *The Karhunen-Loeve Theorem*. PhD thesis.

Giannotti, F. et Pedreschi, D. (2008). *Mobility, data mining and privacy : Geographic knowledge discovery*. Springer Science & Business Media.

Gilliéron, P.-Y. et Peyret, F. (2018). Comment recueillir des informations de position provenant de véhicules traceurs.

Gilsinger, J.-M. et Jaï, M. (2010). *Éléments d'analyse fonctionnelle : fondements et applications aux sciences de l'ingénieur*. PPUR Presses polytechniques.

Giraud, C. (2014). *Introduction to high-dimensional statistics*. Chapman and Hall/CRC.

Girres, J.-F. (2012). *Modèle d'estimation de l'imprécision des mesures géométriques de données géographiques*. PhD thesis, Université Paris-Est.

Girres, J.-F. et Touya, G. (2010). Quality assessment of the french openstreetmap dataset. *Transactions in GIS*, 14(4) :435–459.

Goh, C. Y., Dauwels, J., Mitrovic, N., Asif, M. T., Oran, A., et Jaillet, P. (2012). Online map-matching based on hidden markov model for real-time traffic sensing applications. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 776–781. IEEE.

Gonçalves, L., Subtil, A., Oliveira, M. R., et Bermudez, P. (2014). Roc curve estimation : An overview. *REVSTAT–Statistical Journal*, 12(1) :1–20.

Gorinevsky, D. (2004). Monotonic regression filters for trending deterioration faults. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5394–5399. IEEE.

Gregorutti, B. (2015). *Forêts aléatoires et sélection de variables : analyse des données des enregistreurs de vol pour la sécurité aérienne*. PhD thesis, Université Pierre et Marie Curie-Paris VI.

Gregorutti, B., Michel, B., et Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing*, 27(3) :659–678.

Grejner-Brzezinska, D., Toth, C., et Yi, Y. (2005). On improving navigation accuracy of gps/ins systems. *Photogrammetric engineering & remote sensing*, 71(4) :377–389.

Grenapin, A. (2013). Pourquoi les embouteillages coûtent 677 euros aux foyers français. In *Le Point*.

Guichon, D. et Piel, F. (2013). Sources de données flottantes et mobiles liées au trafic : principales fonctionnaliées pour l'appui à la gestion du trafic. Rapport d'étude du Sétra.

Günther, F. et Fritsch, S. (2010). neuralnet : Training of neural networks. *The R journal*, 2(1) :30–38.

Guo, Z., Chen, Q., Wu, G., Xu, Y., Shibasaki, R., et Shao, X. (2017). Village building identification based on ensemble convolutional neural networks. *Sensors*, 17(11) :2487.

Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3) :331–371.

Haghani, A., Hamedi, M., Sadabadi, K., Young, S., et Tarnoff, P. (2010). Data collection of freeway travel time ground truth with bluetooth sensors. *Transportation Research Record : Journal of the Transportation Research Board*, (2160) :60–68.

Hammersley, J. M. et Clifford, P. (1971). Markov fields on finite graphs and lattices. *Unpublished manuscript*, 46.

Hanley, J. A. et McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1) :29–36.

Hansen, B. E. (2009). Lecture notes on nonparametrics. *Lecture notes*.

Hart, P. E., Nilsson, N. J., et Raphael, B. (1968a). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, 4(2) :100–104.

Hart, P. E., Nilsson, N. J., et Raphael, B. (1968b). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2) :100–107.

Harvey, D. J. et Wood, D. R. (2018). The treewidth of line graphs. *Journal of Combinatorial Theory, Series B*, 132 :157–179.

Hernandez-Belmonte, U. H., Ayala-Ramirez, V., et Sanchez-Yanez, R. E. (2011). A comparative review of two-pass connected component labeling algorithms. In *Mexican International Conference on Artificial Intelligence*, pages 452–462. Springer.

Hillnertz, F. (2014). Incremental self learning road map.

Hoang, G.-M., Denis, B., Härri, J., et Slock, D. T. (2016). Breaking the gridlock of spatial correlations in gps-aided ieee 802.11 p-based cooperative positioning. *IEEE Transactions on Vehicular Technology*, 65(12) :9554–9569.

Hornik, K., Stinchcombe, M., et White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5) :359–366.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6) :417.

Hua, J., Shen, Z., et Zhong, S. (2017). We can track you if you take the metro : Tracking metro riders using accelerometers on smartphones. *IEEE Transactions on Information Forensics and Security*, 12(2) :286–297.

Huber, W., Lädke, M., et Ogger, R. (1999). Extended floating-car data for the acquisition of traffic information.

Ihler, A. T., John III, W. F., et Willsky, A. S. (2005). Loopy belief propagation : Convergence and effects of message errors. *Journal of Machine Learning Research*, 6(May) :905–936.

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1) :253–258.

Islam, S. (2016). Estimation of annual average daily traffic (aadt) and missing hourly volume using artificial intelligence.

Jacovi, A., Shalom, O. S., et Goldberg, Y. (2018). Understanding convolutional neural networks for text classification. *arXiv preprint arXiv :1809.08037*.

Jebreen, K. (2017). *Modèles graphiques pour la classification et les séries temporelles*. PhD thesis, Aix-Marseille.

Kaplan, E. et Hegarty, C. (2005). *Understanding GPS : principles and applications*. Artech house.

Kato, J. Z. (1994). *Modelisations markoviennes multiresolutions en vision par ordinateur. Application a la segmentation d'images SPOT*. PhD thesis, Nice.

Kelley, H. J. (1960). Gradient theory of optimal flight paths. *Ars Journal*, 30(10) :947–954.

Kingma, D. P. et Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.

Kloks, T. (1994). *Treewidth : computations and approximations*, volume 842. Springer Science & Business Media.

Koita, A., Daucher, D., et Fogli, M. (2013). New probabilistic approach to estimate vehicle failure trajectories in curve driving. *Probabilistic Engineering Mechanics*, 34 :73–82.

Koller, D. et Friedman, N. (2009). *Probabilistic graphical models : principles and techniques*. MIT press.

Kolmogorov, V. et Zabih, R. (2004). What energy functions can be minimizedvia graph cuts ? *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2) :147–159.

Körding, K. P. et Wolpert, D. M. (2006). Bayesian decision theory in sensorimotor control. *Trends in cognitive sciences*, 10(7) :319–326.

Krishnamurthy, P., Tipper, D., et Joshi, J. (2013). Position location technologies for wireless systems.

Kschischang, F. R. (2017). The wiener-khinchin theorem. the edward s. rogers sr. department of electrical and computer engineering university of toronto.

Kschischang, F. R., Frey, B. J., Loeliger, H.-A., et al. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2) :498–519.

Kumar, S. K. (2017). On weight initialization in deep neural networks. *arXiv preprint arXiv :1704.08863*.

Kursa, M. B. (2012). rferns–random ferns method implementation for the general-purpose machine learning. *Journal of Statistical Software*.

Kuwata, K. et Shibasaki, R. (2015). Estimating crop yields with deep learning and remotely sensed data. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 858–861. IEEE.

Lamb, P. et Thiébaux, S. (1999). Avoiding explicit map-matching in vehicle location. In *6th World Conference on Intelligent Transportation Systems (ITS-99)*.

Landrieu, L. (2016). *Learning structured models on weighted graphs, with applications to spatial data analysis*. PhD thesis, Paris Sciences et Lettres.

Lannuzel, S. (2000). Référentiels géodésiques - coordonnées. note technique du centre d'hydrographie (shom).

Lathuilière, S., Mesejo, P., Alameda-Pineda, X., et Horaud, R. (2018). Deepgum : Learning deep robust regression with a gaussian-uniform mixture model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 202–217.

Laureshyn, A., Åström, K., et Brundell-Freij, K. (2009). From speed profile data to analysis of behaviour : classification by pattern recognition techniques. *IATSS research*, 33(2) :88–98.

Leduc, G. (2008). Road traffic data : Collection methods and applications. *Working Papers on Energy, Transport and Climate Change*, 1(55).

Lee, Y.-W., Suh, Y.-C., et Shibasaki, R. (2008). A simulation system for gnss multipath mitigation using spatial statistical methods. *Computers & Geosciences*, 34(11) :1597–1609.

Li, J., Qin, Q., Han, J., Tang, L.-A., et Lei, K. H. (2015). Mining trajectory data and geotagged data in social media for road map inference. *Transactions in GIS*, 19(1) :1–18.

Li, W., Jiang, M., Chen, Y., et Lin, M. C. (2018). Estimating urban traffic states using iterative refinement and wardrop equilibria. *IET Intelligent Transport Systems*, 12(8) :875–883.

Li, X., Zhang, G., Huang, H. H., Wang, Z., et Zheng, W. (2016). Performance analysis of gpu-based convolutional neural networks. In *2016 45th International Conference on Parallel Processing (ICPP)*, pages 67–76. IEEE.

Liaw, A. et Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3) :18–22.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3) :18–22.

Lindgren, G., Rootzén, H., et Sandsten, M. (2013). *Stationary stochastic processes for scientists and engineers*. Chapman and Hall/CRC.

Liu, C., Chan, Y., Alam Kazmi, S. H., et Fu, H. (2015). Financial fraud detection model : based on random forest. *International journal of economics and finance*, 7(7).

Liu, X., Biagioni, J., Eriksson, J., Wang, Y., Forman, G., et Zhu, Y. (2012). Mining large-scale, sparse gps traces for map inference : Comparison of approaches. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 669–677, New York, NY, USA. ACM.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2) :129–137.

Lopes, J., Bento, J., Huang, E., Antoniou, C., et Ben-Akiva, M. (2010). Traffic and mobility data collection for real-time applications. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 216–223. IEEE.

López-Pintado, S. et Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486) :718–734.

Lotfi, M., Solimani, A., Dargazany, A., Afzal, H., et Bandarabadi, M. (2009). Combining wavelet transforms and neural networks for image classification. In *System Theory, 2009. SSST 2009. 41st Southeastern Symposium on*, pages 44–48. IEEE.

Loubes, J.-M., Maza, É., Lavielle, M., et Rodriguez, L. (2006). Road trafficking description and short term travel time forecasting, with a classification method. *Canadian Journal of Statistics*, 34(3) :475–491.

Louppe, G. (2014). Understanding random forests : From theory to practice. *arXiv preprint arXiv :1407.7502*.

Lu, Q. et Getoor, L. (2003). Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 496–503.

Lv, F., Wang, W., Wei, Y., Sun, Y., Huang, J., et Wang, B. (2019). Detecting fraudulent bank account based on convolutional neural network with heterogeneous data. *Mathematical Problems in Engineering*, 2019.

Maboudi, M., Amini, J., Hahn, M., et Saati, M. (2016). Road network extraction from vhr satellite images using context aware object feature integration and tensor voting. *Remote Sensing*, 8(8) :637.

Macskassy, S. et Provost, F. (2004). Confidence bands for roc curves : Methods and an empirical study. Proceedings of the First Workshop on ROC Analysis in AI. August 2004.

Macskassy, S. A. et Provost, F. (2007). Classification in networked data : A toolkit and a univariate case study. *Journal of machine learning research*, 8(May) :935–983.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. National Institute of Science of India.

Mallik, S. (2014). Intelligent transportation system. *International Journal of Civil Engineering Research*, 5(4) :367–372.

Marcou, G. (2004). Décentralisation : approfondissement ou nouveau cycle ? *CAHIERS FRANCAIS-PARIS-*, pages 8–14.

Marin, J., Vázquez, D., López, A. M., Amores, J., et Leibe, B. (2013). Random forests of local experts for pedestrian detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2592–2599.

Massaro, E., Ahn, C., Ratti, C., Santi, P., Stahlmann, R., Lamprecht, A., Roehder, M., et Huber, M. (2017). The car as an ambient sensing platform [point of view]. *Proceedings of the IEEE*, 105(1) :3–7.

McMaster, R. B. (1986). A statistical analysis of mathematical measures for linear simplification. *The American Cartographer*, 13(2) :103–116.

Menardi, G. et Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, pages 1–31.

Méneroux, Y., Manandhar, D., Ranjit, S., Saint Pierre, G., et Shibasaki, R. (2017). Positional accuracy control in dense urban environment with low-cost receiver and multi-constellation gnss. In *Proc. 9th Multi-GNSS Asia–MGA Conference*.

Ministère de la transition écologique et solidaire, . (2018). Développement des véhicules autonomes : orientations stratégiques pour l'action publique. Technical report.

Mitchell, M. W. (2011). Bias of the random forest out-of-bag (oob) error for certain input parameters. *Open Journal of Statistics*, 1(03) :205.

Mitrović, D. (2004). Learning driving patterns to support navigation.

Miyazaki, H., Kuwata, K., Ohira, W., Guo, Z., Shao, X., Xu, Y., et Shibasaki, R. (2016). Development of an automated system for building detection from high-resolution satellite images. In *Earth Observation and Remote Sensing Applications (EORSA), 2016 4th International Workshop on*, pages 245–249. IEEE.

Mockus, J. (2012). *Bayesian approach to global optimization : theory and applications*, volume 37. Springer Science & Business Media.

Moreno, A. T. et García, A. (2013). Use of speed profile as surrogate measure : Effect of traffic calming devices on crosstown road safety performance. *Accident Analysis & Prevention*, 61 :23–32.

Morizet, N., Godin, N., Tang, J., Maillet, E., Fregonese, M., et Normand, B. (2016). Classification of acoustic emission signals using wavelets and random forests : Application to localized corrosion. *Mechanical Systems and Signal Processing*, 70 :1026–1037.

Moussouris, J. (1974). Gibbs and markov random systems with constraints. *Journal of statistical physics*, 10(1) :11–33.

Munoz-Organero, M., Ruiz-Blaquez, R., et Sánchez-Fernández, L. (2018). Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on gps traces while driving. *Computers, Environment and Urban Systems*, 68 :1 – 8.

Mustière, S. et Devogele, T. (2008). Matching networks with different levels of detail. *GeoInformatica*, 12(4) :435–453.

Myronenko, A. et Song, X. (2010). Point set registration : Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12) :2262–2275.

Nadaraya, E. (1965). On non-parametric estimates of density functions and regression curves. *Theory of Probability & Its Applications*, 10(1) :186–190.

Nason, G. (1995). Choice of the threshold parameter in wavelet function estimation. In *Wavelets and statistics*, pages 261–280. Springer.

Nason, G. et Maechler, M. M. (2006). The wavethresh package.

Negulescu, C. (2007). Interpolation : cours de prépération à l'agrégation, option calcul scientifique et modélisation.

Neville, J. et Jensen, D. (2000). Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20.

Newson, P. et Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM.

Nguyen, C., Wang, Y., et Nguyen, H. N. (2013). Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. *Journal of Biomedical Science and Engineering*, 6(05) :551.

Ozuysal, M., Fua, P., et Lepetit, V. (2007). Fast keypoint recognition in ten lines of code. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. Ieee.

Palomino-Garibay, A., Camacho-Gonzalez, A. T., Fierro-Villaneda, R. A., Hernandez-Farias, I., Buscaldi, D., Meza-Ruiz, I. V., et al. (2015). A random forest approach for authorship profiling. In *Proceedings of CLEF*.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3) :1065–1076.

Pearson, K. (1916). Ix. mathematical contributions to the theory of evolution.—xix. second supplement to a memoir on skew variation. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 216(538-548) :429–457.

Petovello, M. (2015). How does a gnss receiver estimate velocity ? *Inside GNSS*, pages 38–41.

Pfister, R., Schwarz, K. A., Janczyk, M., Dale, R., et Freeman, J. (2013). Good things peak in pairs : a note on the bimodality coefficient. *Frontiers in psychology*, 4 :700.

Phoon, K., Huang, S., et Quek, S. (2002). Simulation of second-order processes using karhunen–loeve expansion. *Computers & structures*, 80(12) :1049–1060.

Picinbono, B. (1998). *Signaux aléatoires - Tome 2 Fonctions aléatoires et modèles avec problèmes résolus*. Dunod.

Postadjian, T., Le Bris, A., Sahbi, H., et Mallet, C. (2017). Investigating the potential of deep neural networks for large-scale classification of very high resolution satellite images. *ISPRS Annals*, 4 :183–190.

Potts, R. B. (1952). Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press.

Protschky, V., Ruhhammer, C., et Feit, S. (2015). Learning traffic light parameters with floating car data. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2438–2443. IEEE.

Purson, E., Bonanaud, P., Levilly, B., Klein, E., et Bacelar, A. (2015). Evaluations simultanées de différentes technologies innovantes de recueil de données trafic pour le calcul de temps de parcours en temps réel. In *Congrès ATEC ITS France 2015 : Les Rencontres de la Mobilité Intelligente*.

Qiang, L., Qian, G., Lixin, M., et Mingyao, Q. (2012). Measuring variability of arterial road traffic condition using archived probe data. *Journal of Transportation Systems Engineering and Information Technology*, 12(2) :41–46.

Qiu, J. et Wang, R. (2016). Road map inference : A segmentation and grouping framework. *ISPRS International Journal of Geo-Information*, 5(8) :130.

Quddus, M. A., Ochieng, W. Y., et Noland, R. B. (2007). Current map-matching algorithms for transport applications : State-of-the art and future research directions. *Transportation research part c : Emerging technologies*, 15(5) :312–328.

R Development Core Team (2008). *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Raafat, M., Abdullah, B., Taher, M., et Moustafa, M. N. (2016). Towards privacy-preserving driver's drowsiness and distraction detection : A differential privacy approach. *International Journal of Computing and Digital Systems*, 5(05).

Ramsay, J., Hooker, G., et Graves, S. (2009). *Functional data analysis with R and MATLAB*. Springer Science & Business Media.

Ramsay, J. O. et Dalzell, C. (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 539–572.

Ramsay, J. O. et Silverman (2005). *Functional Data Analysis*. Springer series in statistics.

Ramsay, J. O. et Silverman, B. W. (2007). *Applied functional data analysis : methods and case studies*. Springer.

Ranacher, P., Brunauer, R., Trutschnig, W., Van der Spek, S., et Reich, S. (2016). Why gps makes distances bigger than they are. *International Journal of Geographical Information Science*, 30(2) :316–333.

Ranjit, S., Nagai, M., Rittaporn, I., Ajjanapanya, T., Hilding, F., Witayangkurn, A., et Shibasaki, R. (2014). Gps enabled taxi probe's big data processing for traffic evaluation of bangkok using apache hadoop distributed system. In *Asian Transportation Research Society Symposium*, pages 291–296.

Remondi, B. W. (2004). Computing satellite velocity using the broadcast ephemeris. *GPS solutions*, 8(3) :181–183.

Resconi, G. (2014). Conflict compensation, redundancy and similarity in databases federation. In *Transactions on Computational Collective Intelligence XIV*, pages 120–135. Springer.

Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.

Roberts, W. D. S. (1993). Gps time correlation and its implication for precise navigation.

Rogers, S. et Girolami, M. (2016). *A first course in machine learning*. CRC Press.

Ronneberger, O., Fischer, P., et Brox, T. (2015). U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv :1609.04747*.

Saarikivi, P. (2011). State of the art of floating car measurements. MOBI-ROMA project.

Saint Pierre, G. (2003). *Identification du nombre de composants d'un mélange gaussien par chaînes de Markov à sauts réversibles dans le cas multivarié ou par maximun de vraisemblance dans le cas univarié*. PhD thesis, Toulouse 3.

Saltelli, A., Chan, K., et Scott, E. (2000a). Wiley series in probability and statistics. In *Sensitivity analysis*. Wiley.

Saltelli, A., Chan, K., et Scott, E. M., editors (2000b). *Sensitivity analysis*. Wiley series in probability and statistics. J. Wiley & sons, New York, Chichester, Weinheim.

Saporta, G. (1981). Méthodes exploratoires d'analyse de données temporelles. *Cahiers du bureau universitaire de recherche opérationnelle*, (37-38).

Schliep, K., Hechenbichler, K., et Schliep, M. K. (2007). The kknn package. *Unknown*.

Schmidt, M. (2007). Ugm : A matlab toolbox for probabilistic undirected graphical models.

Schroedl, S., Wagstaff, K., Rogers, S., Langley, P., et Wilson, C. (2004). Mining GPS traces for map refinement. *Data mining and knowledge Discovery*, 9(1) :59–87.

Scornet, E., Biau, G., Vert, J.-P., et al. (2015). Consistency of random forests. *The Annals of Statistics*, 43(4) :1716–1741.

Seymour, P. D. et Thomas, R. (1993). Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58(1) :22–33.

Shachter, R. D. (2013). Bayes-ball : The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *arXiv preprint arXiv :1301.7412*.

Shafer, G. (1992). Dempster-shafer theory. *Encyclopedia of artificial intelligence*, 1 :330–331.

Shirato, R. (2016). Dynamic map development in sip-adus. In *SIP-adus Workshop, Tokyo*.

Siddiqui, M. (1962). Some problems connected with rayleigh distributions. *Journal of Research of the National Bureau of standards*, 66(2) :167–174.

Sillard, P. (2001). *Estimation par moindres carrés*. Hermès Science Publications.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC press.

Simon, D. (2010). Kalman filtering with state constraints : a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8) :1303–1318.

Simonyan, K. et Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*.

Sirefelt, R. (2015). Road extraction from aerial images. Master's thesis in Complex Adaptive Systems.

Sohr, A., Brockfeld, E., et Krieg, S. (2010). Quality of floating car data. In *Conference Proceedings, paper*, number 02392, pages 11–15.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., et Salakhutdinov, R. (2014). Dropout : a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1) :1929–1958.

Stanković, R. S. et Falkowski, B. J. (2003). The haar wavelet transform : its status and achievements. *Computers & Electrical Engineering*, 29(1) :25–44.

Steiner, A. et Leonhardt, A. (2011). Map-generation algorithm using low-frequency vehicle position data. Technical report.

Suermondt, H. J. et Cooper, G. F. (1990). Probabilistic inference in multiply connected belief networks using loop cutsets. *International Journal of Approximate Reasoning*, 4(4) :283–306.

Sumathi, S., Beaulah, H. L., et Vanithamani, R. (2014). A wavelet transform based feature extraction and classification of cardiac disorder. *Journal of medical systems*, 38(9) :98.

Sun, Y. et Genton, M. G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2) :316–334.

Suquet, C. (2003). Lois des grands nombres.

Sutton, C., McCallum, A., et al. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4) :267–373.

Sweeney, L. (2000). Simple demographics often identify people uniquely. *Health (San Francisco)*, 671 :1–34.

Tesfamariam, S. et Liu, Z. (2010). Earthquake induced damage classification for reinforced concrete buildings. *Structural safety*, 32(2) :154–164.

Therneau, T. M., Atkinson, E. J., et al. (1997). An introduction to recursive partitioning using the rpart routines. Technical report, Technical report Mayo Foundation.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Torkkola, K., Venkatesan, S., et Liu, H. (2004). Sensor selection for maneuver classification. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 636–641. IEEE.

Truong, Q., Touya, G., et De Runz, C. (2017). Towards vandalism detection in openstreetmap through a data driven approach. In *GIScience 2018*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Tully, S., Kantor, G., et Choset, H. (2011). Inequality constrained kalman filtering for the localization and registration of a surgical robot. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 5147–5152. IEEE.

Vallet, B. (2013). Homological persistence for shape based change detection between digital elevation models. *ISPRS Annals*, 3 :W2.

Van Rijsbergen, C. J. (1974). Foundation of evaluation. *Journal of documentation*, 30(4) :365–373.

Van Winden, K. (2014). Automatically deriving and updating attribute road data from movement trajectories.

Van Winden, K., Biljecki, F., et Van der Spek, S. (2016). Automatic update of road attributes by mining gps tracks. *Transactions in GIS*, 20(5) :664–683.

Vats, D. et Nowak, R. D. (2014). A junction tree framework for undirected graphical model selection. *The Journal of Machine Learning Research*, 15(1) :147–191.

Vauglin, F. (1997). Modèles statistiques des imprécisions géométriques des objets géographiques linéaires.

Villamizar, M., Garrell, A., Sanfeliu, A., et Moreno-Noguer, F. (2012). Online human-assisted learning using random ferns. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2821–2824. IEEE.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, 13(13) :260–269.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2) :1–305.

Walter, E. (2016). *Méthodes numériques et optimisation, un guide du consommateur*.

Wang, C., Hao, P., Wu, G., Qi, X., Lyu, T., et Barth, M. (2017). Intersection and stop bar position extraction from crowdsourced gps trajectories. Technical report.

Wang, Y., Zheng, Y., et Xue, Y. (2014). Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 25–34. ACM.

Wang, Z., Lai, C., Chen, X., Yang, B., Zhao, S., et Bai, X. (2015). Flood hazard risk assessment model based on random forest. *Journal of Hydrology*, 527 :1130–1141.

Warnant, R., Van De Vyvere, L., et Warnant, Q. (2018). Positioning with single and dual frequency smartphones running android 7 or later.

Wei-lun, C. (2011). Gabor wavelet transform and its application. *R98942073*.

Wenk, C. et al. (2010). Geodesic fréchet distance inside a simple polygon. *ACM Transactions on Algorithms (TALG)*, 7(1) :9.

Wilson, C. K. H., Rogers, S., et Weisenburger, S. (1998). The potential of precision maps in intelligent vehicles. In *IEEE International Conference on Intelligent Vehicles*, pages 419–422. Citeseer.

Witayangkurn, A., Horanont, T., et Shibasaki, R. (2013). The design of large scale data management for spatial analysis on mobile phone dataset. *Asian Journal of Geoinformatics*, 13(3).

Wohlfarth, T. (2013). *Machine-learning pour la prédiction des prix dans le secteur du tourisme en ligne*. PhD thesis, Télécom ParisTech.

Wolfermann, A., Alhajyaseen, W., et Nakamura, H. (2011). Modeling speed profiles of turning vehicles at signalized intersections. In *3rd International Conference on Road Safety and Simulation RSS2011, Transportation Research Board TRB, Indianapolis*.

Worrall, S. et Nebot, E. (2007). Automated process for generating digitised maps through gps data compression. In *Australasian Conference on Robotics and Automation*, volume 6. Brisbane : ACRA.

Xie, X., Bing-YungWong, K., Aghajan, H., Veelaert, P., et Philips, W. (2015). Inferring directed road networks from gps traces by track alignment. *ISPRS International Journal of Geo-Information*, 4(4) :2446–2471.

Yedidia, J. S., Freeman, W. T., et Weiss, Y. (2003). Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8 :236–239.

Zaklouta, F., Stanciulescu, B., et Hamdoun, O. (2011). Traffic sign classification using kd trees and random forests. In *The 2011 International Joint Conference on Neural Networks*, pages 2151–2155. IEEE.

Zambom, A. Z. et Dias, R. (2012). A review of kernel density estimation with applications to econometrics. *arXiv preprint arXiv :1212.2812*.

Zhang, L., Thiemann, F., et Sester, M. (2010). Integration of gps traces with road map. In *Proceedings of the Third International Workshop on Computational Transportation Science*, pages 17–22. ACM.

Zhang, Q. et Couloigner, I. (2006). Automated road network extraction from high resolution multi-spectral imagery. In *Proceedings of ASPRS 2006 Annual Conference*, pages 01–05.

Zhao, H., Kumagai, J., Nakagawa, M., et Shibasaki, R. (2002). Semi-automatic road extraction from high-resolution satellite image. *International Archives of Photogrammetry Remote Sensing And Spatial Information Sciences*, 34(3/A) :406–411.

Zheng, Y. (2015). Trajectory data mining : an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3) :29.

Zivot, E. (2009). Maximum likelihood estimation. *Lecture Notes on course" Econometric Theory I : Estimation and Inference (first quarter, second year PhD)", University of Washington, Seattle, Washington, USA*.

Zverovich, V. et Avineri, E. (2015). Braess' paradox in a generalised traffic network. *Journal of Advanced Transportation*, 49(1) :114–138.