Potential and prediction of waiting times for carpooling in a territory

Panayotis Papoutsis

THÈSE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

École Doctorale N°601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Mathématiques et leurs interactions*

Par

Panayotis PAPOUTSIS

Potentiel et prévision des temps d'attente pour le covoiturage sur un territoire

Thèse présentée et soutenue à NANTES , le 17 décembre 2021
Unité de recherche : UMR 6629, Laboratoire de Mathématiques Jean Leray (LMJL)

Rapporteurs avant soutenance :

Pierre LATOUCHE	**Professeur des universités**	Université de Paris Descartes
Guillaume SAINT PIERRE	**Chargé de recherche HDR**	Cerema Sud-Ouest, Site de Toulouse

Composition du jury :

Présidente :	**Anne PHILIPPE**	**Professeure des universités**	Université de Nantes
Examinatrice :	**Sophie ANCELET**	**Chercheur**	Université Paris Saclay
Dir. de thèse :	**Bertrand MICHEL**	**Professeur des universités**	École Centrale de Nantes
Co-dir. de thèse :	**Gérard BIAU**	**Professeur des universités**	Sorbonne Université

Invité

Thomas MATAGNE	**Co-fondateur d'Ecov**	Ecov, Nantes

REMERCIEMENTS

Mes premiers remerciements vont à mes deux directeurs de thèse Bertrand et Gérard. Vous avez été toujours disponibles et vous m'avez soutenu dès le début de ma thèse voire même avant son démarrage officiel. J'ai énormément appris grâce à nos échanges et vos encouragements ont toujours été précieux pour moi. J'ai eu l'immense honneur d'avoir été encadré par vous deux et j'espère que nos chemins vont se recroiser dans le futur.

Ensuite, je tiens à remercier Thomas et Arnaud les deux fondateurs d'Ecov, qui ont directement eu confiance en moi. Ce fut une expérience très instructive de travailler au sein de votre société. Merci de m'avoir donné cette chance.

Je remercie également Guillaume Saint-Pierre et Pierre Latouche pour m'avoir fait l'honneur de rapporter ma thèse. Merci pour vos remarques pertinentes et constructives ainsi que l'intérêt que vous avez porté à mes travaux. Je remercie également Anne Philippe et Sophie Ancelet pour avoir accepté le rôle d'examinatrices. Merci pour vos retours et nos discussions constructives. Merci bien sûr à Arnaud G. et Anthony pour avoir accepté d'être membre de mon comité de suivi individuel. Nos discussions m'ont permis d'améliorer constamment les conditions de déroulement de ma thèse.

J'adresse également mes remerciements à tous les membres d'Ecov qui ont de prêt ou de loin contribué à la réalisation de cette thèse. Merci Laurie, Delphine, Manon, Maxime, Clément, Ugo, Pauline, Thomas, Nathalie, Aymeric, Dianzhuo, Karim, Victorien, Moustapha, Mark, Chloé, Youssouph, Alexandrine, Maden, Wafaa, Jean-Baptiste, Laure, Marie, Elsa, Rose, Romain, Harald, Julie, Brice, Rémi, Nadia, Etienne, Léa, Loïs, Sophie et bien d'autres bien sûr (je m'excuse si certaines personnes ne sont pas listées). Je tiens aussi à remercier plus particulièrement Claire et Antoine collègues et membres du CSE avec qui nous pouvons être fiers du travail accompli. Je tiens aussi à remercier Philippe G., Pierre, Didier, Jean-Marc, Jérémie, Julien, Viatcheslav, Romain S. d'Epango qui ont partagé le même incubateur à Saint-Denis. Nos pauses quotidiennes et nos discussions m'ont toujours remonté le moral pendant ma thèse.

Je remercie mes collègues de l'équipe Data/SIG d'Ecov. J'ai eu la chance de travailler avec une superbe équipe. Merci Tarn, Xi, Safa, Taha, Constant, Madeleine et Flavien. Je tiens à remercier en particulier Tarn qui m'a encadré et épaulé depuis le début de ma thèse au quotidien et enseigné la recherche en entreprise, la thèse n'aurait pas pu avancer sans ton aide.

Je remercie aussi mes collègues de thèse au LPSM et au LMJL qui m'ont accompagné durant cette thèse. Merci Nazih, Yohann, Robin, Adeline, Ugo, Nicolas, Joseph, Clément, Sebastien, Karzan. Merci bien sûr à Louise, Corinne

et Bénédicte pour tous les sujets administratifs entre Paris et Nantes.

Je remercie bien évidemment mes amis, votre soutien m'a toujours donné la force à accomplir ce travail. Merci Mous, Reda, Mounir, Andréas, Anne-Cécile, Selena, Laura, Alexis, Tassos, Aris, Dorothée, Jean-Christophe, Philippe A., Michel, Dimitri, Vago, Daphni, Nestor, Venia, Demba, Danie et Michel.

Enfin, je remercie surtout ma famille. Merci Papa et Maman pour tout ce que vous avez fait et continuez à faire pour moi. Votre soutien et vos conseils ont toujours été les meilleurs. Merci aussi à mes deux grandes soeurs qui m'ont toujours supporté et conseillé de la meilleure manière. Vous avez toujours réussi à me réconforter pendant cette thèse (et même plus). Pour finir, je tiens à remercier ma conjointe Ioanna, merci de m'avoir soutenu pendant la thèse et bien avant. Tu as toujours su comment m'aider et je t'en remercie du fond de mon coeur.

Table of Contents

1 Introduction **8**
- 1.1 Positionnement d'Ecov dans le marché du covoiturage 8
- 1.2 Besoin industriel et contexte statistique 12
- 1.3 Organisation du manuscrit et principales contributions 13

2 Quantile Regression for waiting time prediction in a stochastic carpooling network **35**
- 2.1 Introduction 35
- 2.2 Quantile Regression 37
  - 2.2.1 Linear Quantile Regression 40
  - 2.2.2 Quantile Regression with Random Forests 42
  - 2.2.3 Quantile Regression with Generalized Random Forests . 44
  - 2.2.4 Quantile Regression with Gradient Boosting 48
  - 2.2.5 Accuracy measures for Quantile Regression models . . . 49
- 2.3 Comparison of Regression models for carpooling request waiting times 52
  - 2.3.1 Response variable: Waiting times 53
  - 2.3.2 Explanatory variables 54
  - 2.3.3 Quantile Regression models comparison 57
- 2.4 Conclusion 59
- 2.5 Appendix : Details of the supporting mathematical results . . . 60
  - 2.5.1 Quantiles as minimisers of the Pinball loss 60
  - 2.5.2 Conditional quantiles as minimisers of a Pinball loss . . 61
  - 2.5.3 Pinball loss and its scoring function 63

3 Relaxing door-to-door matching reduces passenger waiting times: a workflow for the analysis of driver GPS traces in a stochastic carpooling service **65**
- 3.1 Introduction 66
- 3.2 Door-to-door matching is an obstacle to mass carpooling 68
- 3.3 Data science-GIS workflow for the analysis of GPS traces 72
  - 3.3.1 Data sources 72
  - 3.3.2 Data wrangling/geoprocessing 73
  - 3.3.3 Outputs 74
- 3.4 Case study of an operational stochastic carpooling service . . . 76
  - 3.4.1 Topological simplification of GPS traces on a carpooling line 77

3.4.2 Driver flow estimation . . . 78
3.4.3 Waiting time prediction . . . 80
3.4.4 Driver participation rate estimation . . . 83
3.5 Conclusions and future work . . . 85
3.6 Acknowledgements . . . 86

4 Bayesian hierarchical models for the prediction of the driver flow and passenger waiting times in a stochastic carpooling service 87
4.1 Introduction . . . 87
4.2 Bayesian hierarchical modelling of driver flow and passenger waiting times . . . 90
4.2.1 Multi-level moving average for driver flows . . . 91
4.2.2 Gamma regression for passenger waiting times . . . 93
4.3 Model validation with simulated pseudo waiting times . . . 96
4.4 Model validation with the Lane carpooling service . . . 99
4.4.1 Daily driver flows . . . 99
4.4.2 Temporal profiles of passenger pseudo waiting times . . . 103
4.4.3 Temporal profiles for passenger perceived waiting times 107
4.5 Conclusions . . . 109
4.6 Acknowledgements . . . 109
4.7 Appendix . . . 109
4.7.1 GPS traces pre-processing . . . 109
4.7.2 Simulation algorithms . . . 110
4.7.3 Competing models for driver flows . . . 112

5 Construction d'un prior informatif par transfert bayésien : application au covoiturage 115
5.1 Introduction . . . 115
5.2 Méthodologie: construction du prior informatif . . . 116
5.3 Description du modèle de prédiction des flux et temps d'attente 119
5.4 Résultats . . . 120
5.4.1 Détails de la procédure d'évaluation . . . 120
5.4.2 Jeu de données simulées . . . 121
5.4.3 Jeu de données d'Ecov . . . 123
5.5 Conclusions . . . 131

6 Valorisation Industrielle : Mise en production du modèle hiérarchique bayésien 133
6.1 Introduction . . . 133
6.2 Description des données et du modèle à industrialiser . . . 134
6.3 Industrialisation du modèle hiérarchique bayésien . . . 135
6.3.1 Les logiciels et outils informatiques . . . 135
6.3.2 La structure du paquet . . . 136

6.3.3 Utilisation du paquet *bayesian-hierarchical-model* 138
6.4 Automatisation du processus 143
6.5 Conclusions 145

Conclusion **147**

Bibliography **155**

CHAPTER 1

INTRODUCTION

1.1 Positionnement d'Ecov dans le marché du covoiturage

La pratique du covoiturage date des années 1950. En France, comme dans le monde entier, cette pratique s'est développée en parallèle avec l'internet. Ce n'est que plus récemment que les opérateurs de la mobilité ont commencé à collecter et exploiter les données sur la thématique du covoiturage. Un exemple important est le Registre National de Preuve de Covoiturage[1] qui centralise les données de plusieurs opérateurs de covoiturage en France. Tandis que le covoiturage de longues distances a largement fait ses preuves, avec BlaBlaCar (https://www.blablacar.fr/) comme acteur principal en France, le covoiturage du quotidien implique plusieurs acteurs et approches différentes. La Figure 1.1 ci-dessous illustre les différentes catégories actuellement présentes en France. Dans ce sens, nous pouvons définir le covoiturage du quotidien comme un covoiturage correspondant à des trajets de courtes distances (moins de 200 km).

1. https://covoiturage.beta.gouv.fr/

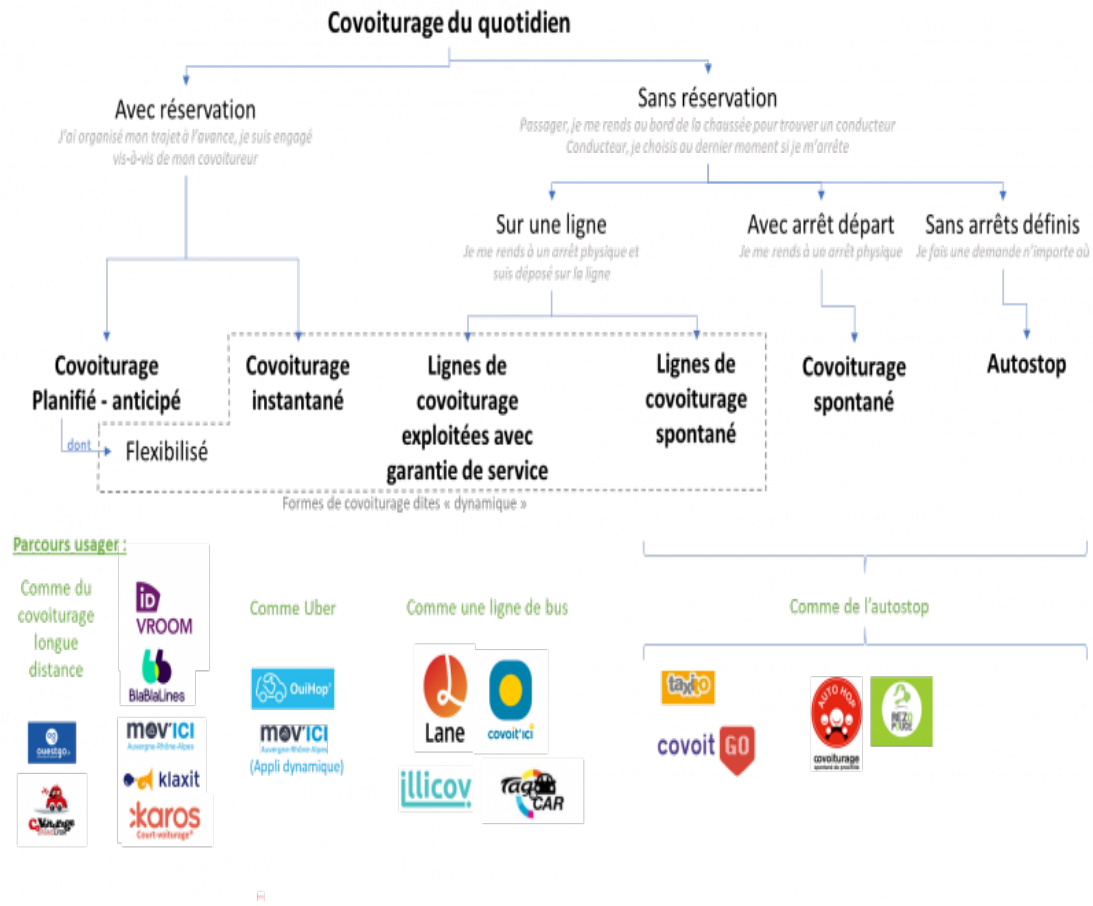


Figure 1.1 – Les acteurs industriels du covoiturage du quotidien, image provenant du site de la Fabrique des Mobilités https://wiki.lafabriquedesmobilites.fr/wiki/Le_covoiturage

Le document présent s'intéresse donc à l'approche d'un covoiturage de courte distance du quotidien, et plus précisément d'un covoiturage "dynamique" sans réservation. La société Ecov s'inscrit dans plusieurs catégories, comme illustré dans la Figure 1.1, via la commercialisation de plusieurs produits comme Ouihop', Covoit'Go et Covoit'Ici. Nos travaux sont basés sur les données issues du produit Covoit'Ici dont nous détaillons ci-dessous le fonctionnement et l'évolution.

Il s'agit tout d'abord d'un système de covoiturage de courte distance, destiné aux zones peri-urbaines et rurales, qui est majoritairement utilisé pour les trajets domicile-travail. De ce fait, nous observons un fort phénomène de migration pendulaire dans les données. Ce phénomène est clairement visible à la Figure 1.2 ci-dessous qui représente les flux moyens des conducteurs par intervalles de 15 minutes pour chaque jour de la semaine.

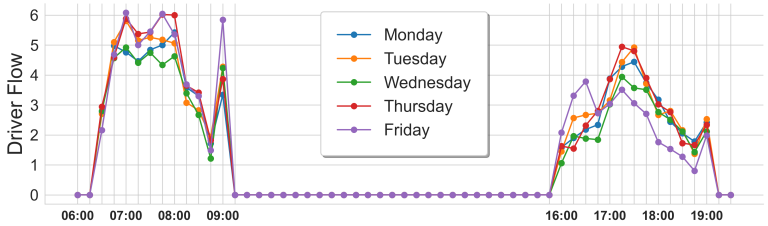


Figure 1.2 – Flux moyens des conducteurs par intervalles de 15 minutes pour chaque jour de la semaine sur la période du 2018-05-15 au 2019-05-31. Le lundi est en bleu, le mardi en orange, le mercredi en vert, le jeudi en rose et le vendredi en violet. Pour mieux voir le phénomène de migration pendulaire, les deux sens sont représentés sur cette figure. Le sens aller correspond aux heures d'ouverture du matin (6h00 à 9h00) et le sens retour aux heures d'ouverture du soir (16h00 à 19h00).

Par ailleurs, la majorité des acteurs proposent une plateforme permettant de mettre en contact les conducteurs avec les passagers de manière déterministe, à savoir un rendez-vous spatio-temporel fixé à l'avance. Cette mise en relation, qui s'appelle le *matching déterministe*, a fait ses preuves dans le covoiturage de longue distance. À l'inverse, Ecov propose un matching alternatif dit *matching stochastique*, à savoir mettre en relation un passager avec un flux de conducteurs. Pour cela, Ecov simplifie cette mise en relation en installant sur le territoire des arrêts de covoiturage connectés, comme illustré sur la Figure 1.3.

Figure 1.3 – Configuration d'un arrêt de covoiturage connecté. La structure orange ressemble à un abribus. Le passager informe les conducteurs potentiels de sa demande de covoiturage à l'aide de la console, demande qui est ensuite affichée sur le panneau électronique lumineux au bord de la voirie. Un conducteur peut récupérer le passager en toute sécurité sur la place de stationnement réservée à cette occasion. Photographie reproduite avec l'autorisation d'Ecov.

Les arrêts connectés permettent aux passagers de faire leurs demandes de covoiturage via une console. Il est donc possible mais pas obligatoire d'utiliser un smartphone. Les conducteurs sont ensuite informés de cette demande grâce au panneau lumineux positionné sur la voirie et aussi via l'application mobile dédiée à cela. Cette approche est proche du système de bus et permet de faire le lien entre les conducteurs et les passagers.
Le réseau de covoiturage a ensuite évolué en proposant des lignes de covoiturage. Les passagers choisissent alors leur ligne de covoiturage déterminée par leur arrêts de départ et de destination. Ce changement de point de vue se reflète dans le Chapitre 2 et 3. Par exemple, la Figure 1.4 illustre le réseau de covoiturage du "Vexin", premier réseau implanté en Île-de-France, où les passagers pouvaient faire une demande de covoiturage sans se soucier de ligne de covoiturage avec arrêts déterminés.

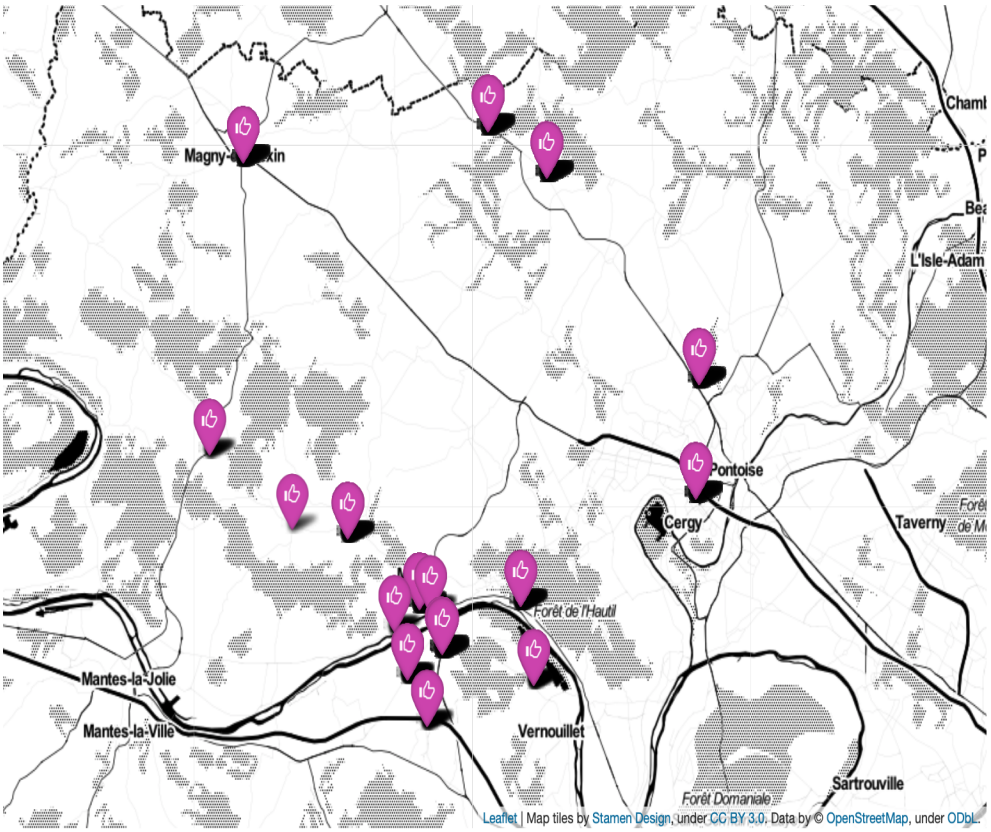


Figure 1.4 – Carte du réseau de covoiturage du "Vexin". Les arrêts connectés où les passagers peuvent faire une demande de covoiturage sont indiqués par les cercles roses avec le logo du pouce.

La nouvelle structure, c'est à dire les lignes de covoiturage, est semblable aux lignes de bus des transports en commun. La Figure 1.5 représente le schéma

des lignes de covoiturage qui sont développées dans la région de Lyon.

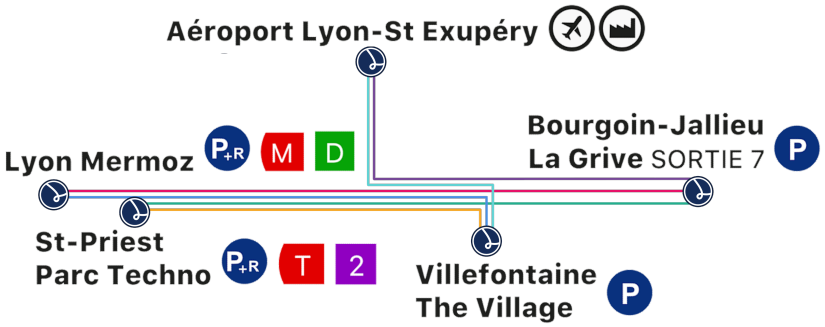


Figure 1.5 – Plan des lignes de covoiturage Lane. Schéma reproduit avec la permission d'Ecov.

Le passager utilisant une ligne de covoiturage a un choix précis de destinations parmi les arrêts desservis de cette ligne. Ainsi, cette dernière approche se démarque des concurrents, étant donné qu'il ne s'agit pas seulement d'une plateforme et/ou application mobile mais d'une solution complète combinant digital et infrastructure physique sur le territoire.

Il est intéressant, afin d'avoir une image plus globale du marché, d'énumérer quelques acteurs du covoiturage qui travaillent sur la même problématique, à savoir le covoiturage de courte distance, cependant avec des approches différentes. Par exemple Klaxit (https://www.klaxit.com/), Karos (https://www.karos.fr/) et Mobicoop (https://www.mobicoop.fr/) proposent des plateformes 100% digitales destinées aux trajets domicile-travail. Ensuite, Citygo (https://www.citygo.io/) et La Roue Verte (https://www.laroueverte.com/) se concentrent sur le covoiturage domicile-loisirs. La différence avec Ecov réside du fait qu'Ecov propose une solution qui s'adapte aussi bien au domicile-travail qu'au domicile-loisirs tout en combinant une infrastructure physique et digitale.

1.2 Besoin industriel et contexte statistique

Le présent travail de thèse est motivé par le besoin, d'une part de fiabiliser les informations disponibles pour les passagers (temps d'attente, flux disponible) au moment d'un covoiturage et d'autre part, de comprendre le comportement des conducteurs (heures de passage, arrêts desservis). L'enjeu majeur pour Ecov est d'être capable de faire une bonne prévision des temps d'attente. Comme nous l'avons vu précédemment, les passagers sont confrontés à un flux de conducteurs (non professionnels) qui sont sur leurs trajets du quotidien. On voit donc qu'il ne s'agit pas d'un simple calcul de temps d'attente grâce à une API ("Application Programming Interface"). Afin de répondre aux besoins industriels, cette thèse s'articule autour de cinq thématiques principales:

1. Les techniques de régression quantile pour la prédiction des temps d'attente.

2. La construction d'un processus de travail en vue d'utiliser les données issues du covoiturage.

3. La construction d'un modèle hiérarchique bayésien pour la prédiction des flux de trafic et des temps d'attente.

4. La construction d'une loi a priori pour améliorer la prédiction des temps d'attente en situation de jeu de données court.

5. La mise en production et l'exploitation industrielle du modèle hiérarchique bayésien.

Chacune des thématiques précédentes correspondent à un chapitre de la thèse. L'ordre des chapitres suit la temporalité de l'évolution du produit et des travaux effectués au sein d'Ecov.

1.3 Organisation du manuscrit et principales contributions

Chapitre 2 : La régression quantile pour la prédiction des temps d'attente dans un réseau de covoiturage stochastique

Dans ce chapitre, nous travaillons sur les données provenant de la structure initiale du service, à savoir le réseau de covoiturage (voir Figure 1.4). Les données disponibles sont majoritairement celles des passagers, de la géographie du territoire et de la temporalité des demandes de covoiturage. On note $\mathcal{D}_n = \{(\boldsymbol{X}_i, Y_i) : 1 \leq i \leq n\}$ l'échantillon de n observations issues du couple $(\boldsymbol{X}, Y)$. La variable cible Y représente les temps d'attente lors d'une demande de covoiturage et $\boldsymbol{X}$ les variables explicatives disponibles. La Figure 1.6 représente les temps d'attente observés sur le réseau de covoiturage que nous allons étudier.

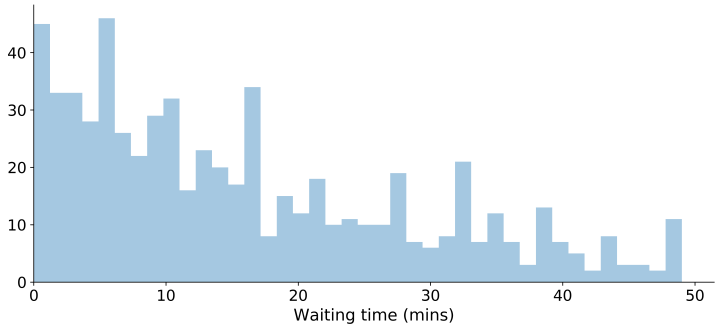


Figure 1.6 – Histogramme des temps d'attente observés sur le réseau de covoiturage du "Vexin" pendant la période du 2016-01-18 au 2018-10-02.

Dans le cadre du covoiturage, la prédiction d'un temps d'attente moyen n'est pas satisfaisante. En effet, un temps d'attente moyen décrit la tendance moyenne de la distribution des temps d'attente. Or, pour le covoiturage nous avons besoin d'avoir une information plus fine et plus précise. La question qui se pose pour un passager lors d'un covoiturage est de savoir quel est son temps d'attente (voire un temps d'attente maximal) et quelle est la fiabilité de cette information. Nous reformulons cette question par *"Quelle est la probabilité qu'un passager attende moins de y minutes avec $\alpha\%$ de chance?"*. Les méthodes de régression standard s'intéressent à la moyenne de la variable cible et ne répondent pas à la question. Afin d'y répondre de la manière la plus complète possible, nous avons retenu la technique de régression quantile. On note $\alpha \in (0, 1)$ le niveau du quantile de Y et le α-quantile conditionnel de $Y|\boldsymbol{X}$, défini par

$$q_\alpha(Y|\boldsymbol{X}) = \inf_{t\in\mathbb{R}}\{F_{Y|\boldsymbol{X}}(t) \geq \alpha\}, \tag{1.1}$$

avec $F_{Y|\boldsymbol{X}}$ la fonction de répartition de Y conditionnellement à $\boldsymbol{X}$. Contrairement aux techniques de régression standard, où nous utilisons la méthode des moindres carrés, la régression quantile utilise une autre méthode de minimisation du quantile. Le problème de minimisation du quantile est défini par

$$q_\alpha = \underset{a\in\mathbb{R}}{\operatorname{argmin}}\, \mathbb{E}[\rho_\alpha(Y - a)],$$

dans le cas non-conditionnel et

$$q_\alpha(Y|\boldsymbol{X}) = \underset{f\in L^1(\boldsymbol{X})}{\operatorname{argmin}}\, \mathbb{E}[\rho_\alpha(Y - f)],$$

dans le cas conditionnel, où $\rho_\alpha(y) = y(\alpha - \mathbf{1}_{[y\leq 0]})$ est la fonction de perte "Pinball" et $L^1(\boldsymbol{X}) = \{f : \mathbb{E}[|f(\boldsymbol{X})|] < \infty\}$. La Figure 1.7 représente la fonction de perte "Pinball" pour différentes valeurs de α. Nous proposons une démonstration du

problème de minimisation du quantile à la fois dans le cas non-conditionnel et conditionnel.

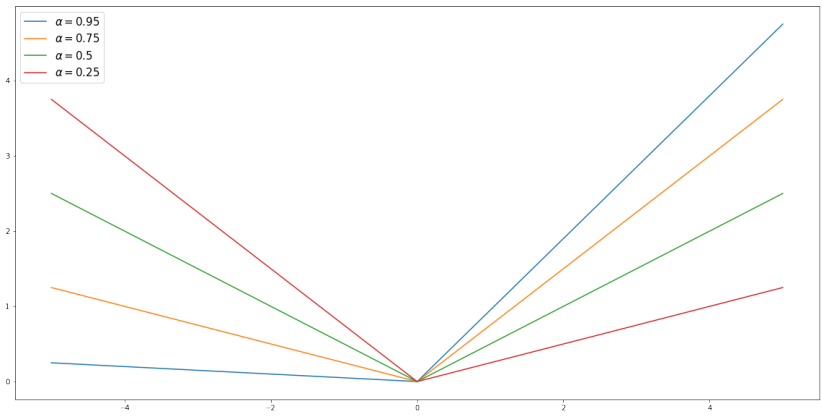


Figure 1.7 – Fonction de perte "Pinball" ρ_α pour différentes valeurs de α. Bleu : $\alpha = 0,95$, orange $\alpha = 0,75$, vert $\alpha = 0,5$, rouge $\alpha = 0,25$.

Après avoir présenté le principe de régression quantile, nous nous arrêtons sur quatre approches possibles. La première et la plus ancienne est celle décrite dans Koenker & Hallock, 2001. Le quantile conditionnel est supposé linéaire tel que $q_\alpha(Y|\boldsymbol{X}) = f(\boldsymbol{X}, \beta_\alpha)$ avec $f(\boldsymbol{X}, \beta_\alpha) = \boldsymbol{X}'\beta_\alpha$. La deuxième est nommée "Quantile Regression Forest". Elle est introduite dans Meinshausen, 2006a et se base sur les Forêts Aléatoires de Breiman, 2001. La différence essentielle entre cette approche et les Forêts aléatoires classiques est la pondération finale des arbres composant la forêt quantile. Dans cette même catégorie, nous présentons aussi la régression quantile par Forêts Aléatoires Généralisées dû à Athey et al., 2019. Cette fois-ci une toute nouvelle manière de construire les arbres aléatoires est développée et se base sur un critère de découpe prenant en compte la cible, le α-quantile de Y. Enfin, la dernière approche est le Gradient Boosting adapté à la régression quantile. Le principe consiste à appliquer la méthode classique de descente de gradient sur la fonction de perte "Pinball". De ce fait, on obtient en effet un estimateur du α-quantile.

Une fois ces notions introduites, nous détaillons les variables explicatives utilisées. Essentiellement, ces variables proviennent des demandes de covoiturages des passagers. Elles sont classées en trois catégories:

1. Les informations géographiques de la demande de covoiturage.

2. Les informations temporelles de la demande de covoiturage.

3. Les informations concernant les habitudes des passagers.

Par exemple, les couples origine-destination des demandes de passagers sont nombreuses pour la structure du réseau de covoiturage. Afin d'en tirer une information permettant d'améliorer les prédictions des modèles nous avons procédé a un partitionnement spatial des données. La Figure 1.8 montre la carte avec les sept groupes qui représentent les sept quartiers de destinations possibles. Ce partitionnement réduit le nombre de paires origine-destination observées de 252 à 66.

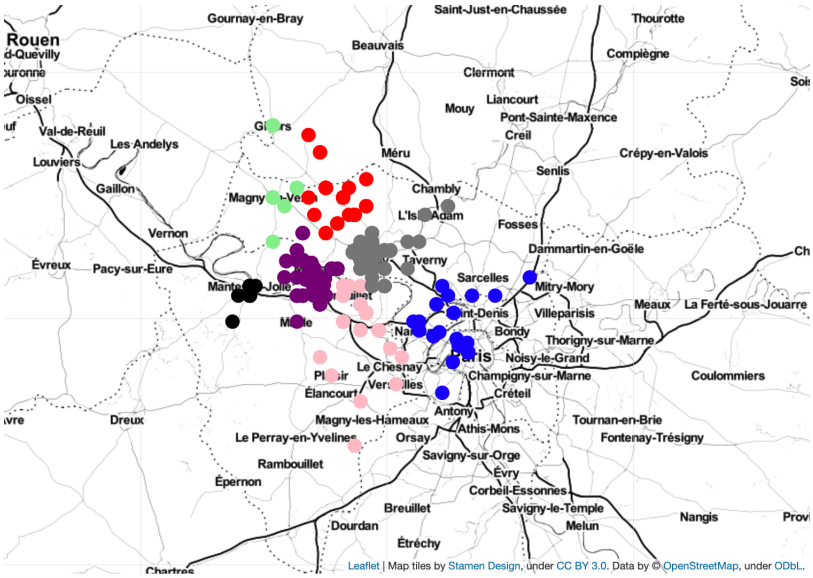


Figure 1.8 – Les quartiers de destination regroupés dans le réseau de covoiturage Vexin.

Nous comparons ensuite les performances de ces quatre algorithmes de régression quantile sur les données d'Ecov. En raison de la complexité du service de covoiturage tel qu'il est proposé par Ecov au cours de cette phase expérimentale et grâce à cette analyse de régression, Ecov a ensuite pivoté son service de covoiturage pour desservir un nombre limité de destinations, plus proches des lignes de bus. Ecov a également intégré une technologie permettant de collecter les traces GPS des itinéraires des conducteurs à partir de leur application mobile afin de mieux comprendre leurs comportements. Dans les chapitres suivants, nous analyserons les données de ce service amélioré et la manière dont il facilite une meilleure compréhension du potentiel de covoiturage.

Chapitre 3: Relaxing door-to-door matching reduces passenger waiting times: a workflow for the analysis of driver GPS traces in a stochastic carpooling service

Ce chapitre a fait l'objet d'une publication, dans le journal "Transportation Engineering", co-écrite avec Tarn Duong, Safa Fennia et Constant Bridon. Il s'agit de trois membres de l'équipe Data/SIG d'Ecov.

À présent, nous travaillons sur les données issues des lignes de covoiturage. Les nouveautés par rapport au chapitre précédent sont; la collecte des traces GPS des conducteurs et la structure améliorée du service. Les traces GPS permettent de compléter les informations concernant les passagers, la géographie du territoire et la temporalité des demandes de covoiturage. Une trace GPS est une information spatio-temporelle très riche et l'utilisation de celle-ci permet une meilleure compréhension des habitudes des conducteurs. Ainsi, le traitement et l'analyse de ces données nous ont permis d'obtenir deux résultats essentiels pour le covoiturage:

- La correspondance porte-à-porte de trajectoires complètes de l'origine à la destination est un obstacle structurel pour la transformation du covoiturage en un service de transport en commun.

- Les arrêts de covoiturage fixes favorisent la rencontre entre les conducteurs et les passagers et diminuent les temps d'attente.

Pour illustrer les difficultés de la correspondance passager-conducteur dans l'espace et dans le temps pour les trajectoires porte-à-porte, nous pouvons la représenter par la partition d'un cube 3D divisé en sous-cubes plus petits, où l'axe x est la longitude, l'axe y la latitude et l'axe z le temps, comme le montre la Figure 1.9. Ainsi, pour que deux trajectoires correspondent dans l'espace et dans le temps dans un sens porte-à-porte, elles doivent partager le même sous-cube pour l'origine, et de même pour la destination. Ainsi, une correspondance porte-à-porte plus stricte entraîne une diminution du nombre de conducteurs disponibles pour partager leurs trajectoires avec des passagers.

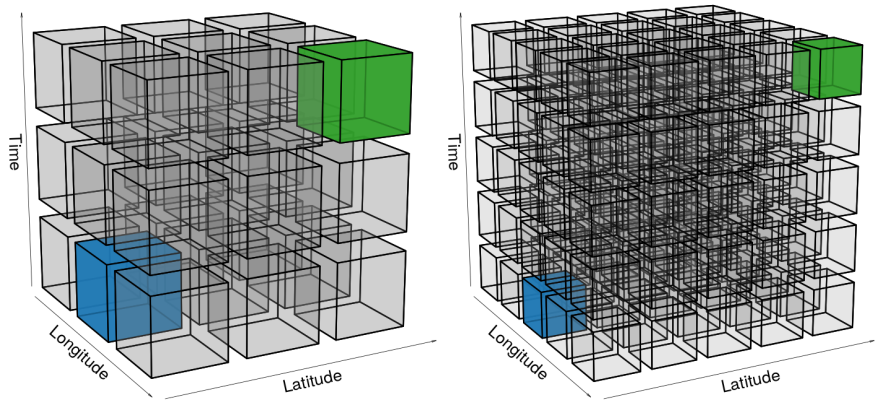


Figure 1.9 – Correspondance spatio-temporelle porte-à-porte, à gauche les conditions d'appariement sont moins strictes et à droite les conditions d'appariement sont restreintes. Le sous-cube bleu représente l'origine (quartier résidentiel), le vert la destination (lieu de travail), et les trajectoires qui partagent les mêmes sous-cubes d'origine et de destination sont considérées comme des correspondances porte-à-porte.

Ensuite, nous étudions les traces complètes des conducteurs et nous montrons l'intérêt des points de rencontre fixe. La Figure 1.10 illustre la classification hiérarchique spatiale faite sur 121 traces GPS de conducteurs. Nous voyons que le potentiel de covoiturage est bas lorsque nous nous basons sur la correspondance porte-à-porte entre les conducteurs et les passagers. Dans le cas de la correspondance porte-à-porte nous obtenons 9 clusters distincts dont certains représentent une trace unique. Au final, cette étude nous a permis de conclure que le potentiel de covoiturage, grâce aux arrêts de covoiturage fixe, a augmenté de 59%.

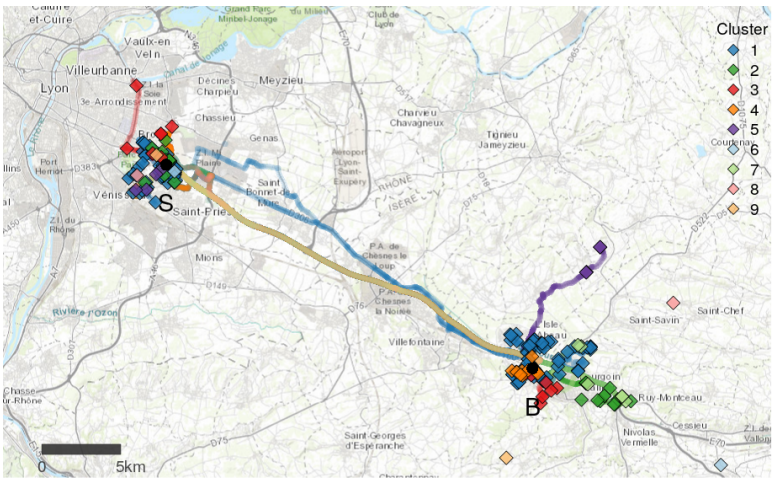


Figure 1.10 – Traces GPS de plusieurs conducteurs provenant du service de covoiturage stochastique opérationnel "Lane" à Lyon. Les différents clusters de traces GPS représentent les mises en relation de porte-à-porte. Les losanges pleins représentent les origines et destinations. Les points de rencontre sont les cercles noirs pleins, notés B = Bourgoin, S = St-Priest.

Les résultats précédents obtenus ont motivé la construction d'un processus de travail qui se trouve à la frontière de la Data Science et du SIG. Le but du processus, illustré par la Figure 1.11, est d'exploiter pleinement des données issues du covoiturage. Nous détaillons aussi dans le chapitre les traitements et les géotraitements appliqués sur les traces GPS.

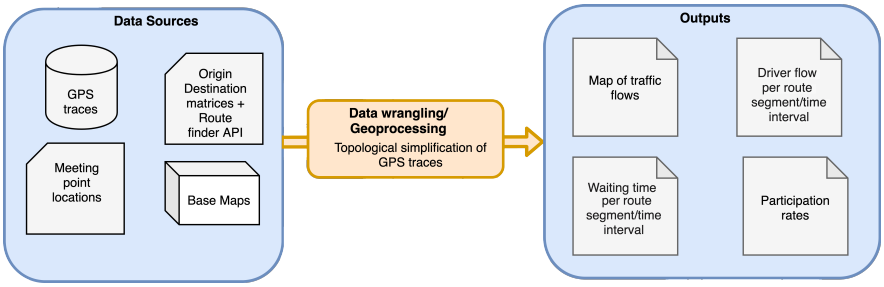


Figure 1.11 – Processus de travail Data Science-SIG pour l'analyse des traces GPS des conducteurs dans un service de covoiturage stochastique. (Gauche) Sources de données d'entrée spatio-temporelles. (Centre) Tâches de traitement des données et de géotraitement. (Droite) Résultats générés.

Le processus de travail mis en place génère quatre résultats:

1. La visualisation des traces GPS sur des fonds de cartes.
2. Une fiche horaire pour des intervalles de tailles différents (à l'heure, la demi-heure, le quart d'heure) des flux de conducteurs par ligne.

3. Une fiche horaire pour des intervalles de tailles différents (à l'heure, la demi-heure, le quart d'heure) de prédiction de temps d'attente.
4. L'évolution du temps d'attente en fonction du taux de participation des conducteurs sur les lignes de covoiturage.

Il est important de souligner que ce processus de travail proposé est la base de tous les travaux des chapitres suivants. Nous rappelons qu'en plus de la construction du processus de travail ce chapitre permet essentiellement de valider l'importance de pratiquer le covoiturage avec des points de rencontre fixes sur le territoire. Ceci permet effectivement de faire converger les conducteurs et les passagers afin de simplifier leur rencontre. En second lieu, nous avons aussi validé grâce aux données empiriques que ces points de rencontre fixes permettent de faire baisser les temps d'attente des passagers. Nous proposons aussi une estimation de l'ordre de grandeur de la baisse potentielle des temps d'attente en fonction du taux de participation des conducteurs. On donne une illustration de cette estimation à la Figure 1.12.

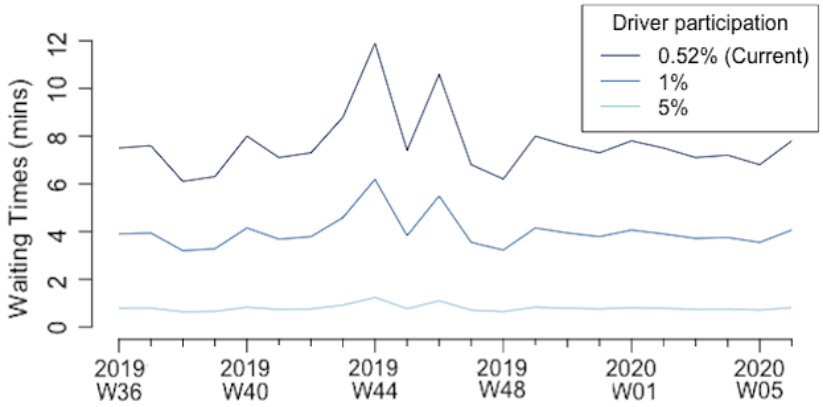


Figure 1.12 – Évolution du temps d'attente prédit des passagers en fonction du taux de participation des conducteurs sur la ligne de covoiturage Bourgoin > St-Priest, pendant les heures de fonctionnement du matin 06h30-09h00, du 2019-07-25 au 2020-02-17.

Chapitre 4 : Bayesian hierarchical models for the prediction of the driver flow and passenger waiting times in a stochastic carpooling service

Ce chapitre fait l'objet d'une publication en cours co-écrite avec Anne Philippe (Professeure à l'Université de Nantes), Bertrand Michel (Professeur à l'Université de Nantes et l'École Centrale de Nantes) et Tarn Duong (Docteur en Statistique et membre de l'équipe Data/SIG d'Ecov).

Nous souhaitons à présent construire un modèle pouvant apporter des informations plus précises concernant les flux de trafic sur une ligne de covoiturage ainsi que les temps d'attente qui en découlent. Les données disponibles sont de sources distinctes

- les traces GPS des conducteurs,
- les temps d'attente observés.

Les données concernant les conducteurs sont des séries temporelles représentant le flux de trafic quotidien sur la ligne de covoiturage. Nous représentons une telle série temporelle par la Figure 1.13, où un code couleur est utilisé pour différencier les différents types de jour. Cette classification des types de jours est importante. En effet, dans le domaine du covoiturage les habitudes des conducteurs sont influencées par le type de jour.

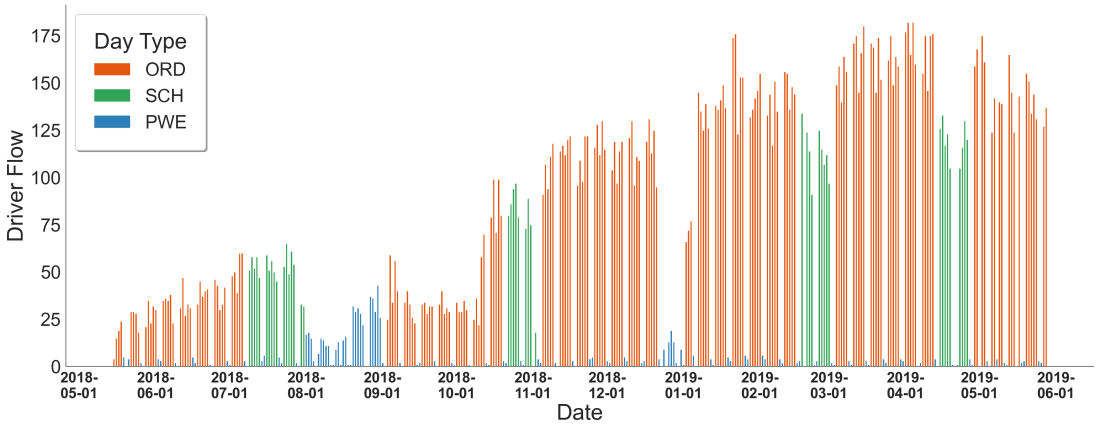


Figure 1.13 – Flux quotidiens de conducteurs au sein du service de covoiturage Lane, du 2018-05-15 au 2019-05-31. Les jours de semaine ordinaires (ORD) sont en orange, les vacances scolaires (SCH) en vert et les jours fériés/week-end (PWE) en bleu.

Les données provenant des passagers sont les temps d'attente observés et collectés lors d'une demande de covoiturage à différents moments de la journée. La Figure 1.14 représente les diagrammes en boîtes des temps d'attente des passagers pour chaque jour de la période du 2019-10-22 au 2020-01-15. Dès lors, nous avons d'une part une série temporelle des flux quotidiens et d'autre part plusieurs temps d'attente par jour. Nous proposons de combiner ces deux sources de données en construisant un modèle hiérarchique bayésien de deux étages.

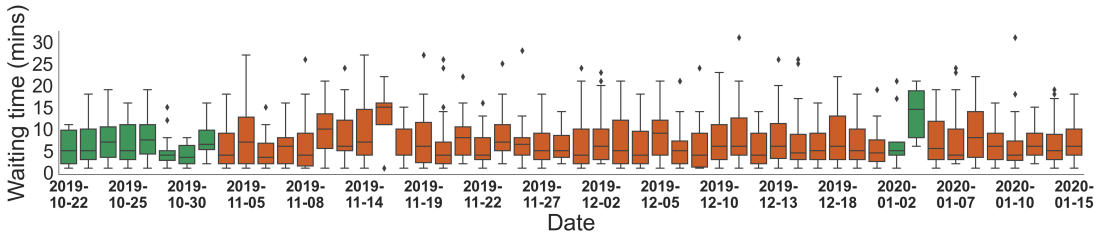


Figure 1.14 – Temps d'attente des passagers (en minutes) au sein du service de covoiturage Lane du 2019-10-22 au 2020-01-15. Les jours de semaines ordinaires (ORD) sont en orange, et les vacances scolaires (SCH) en vert.

Le premier étage s'intéresse à la modélisation des flux de trafic quotidien prenant en compte les différents types de jour. Il s'agit d'un modèle ayant une structure autorégressive d'ordre $K \geq 1$. Afin de le définir, nous notons y_i le flux de trafic quotidien pour le jour i. La modélisation du premier étage est définie par

$$y_i = \alpha_{\mathrm{DT}(i)} \sum_{k=1}^{K} \eta_{\mathrm{DT}(i-k)} y_{i-k} + \varepsilon_i, \tag{1.2}$$

avec ε_i indépendantes et identiquement distribuées selon une loi Gaussienne centrée de variance σ_ε^2 et DT une fonction qui caractérise les types de jours,

$$\mathrm{DT}(i) = \begin{cases} \mathrm{ORD} & \text{si le jour } i \text{ est ordinaire,} \\ \mathrm{SCH} & \text{si le jour } i \text{ est vacance scolaire,} \\ \mathrm{PWE} & \text{si le jour } i \text{ est férié ou weekend.} \end{cases} \tag{1.3}$$

Le paramètre α_{DT} est le coefficient pour le jour actuel i et η représente le coefficient de transition des types de jour qui précède le jour i sur le flux y_i. Le second étage concerne la modélisation des temps d'attente. Nous distinguons à partir de maintenant les temps d'attente perçus et les pseudos temps d'attente. Le pseudo temps d'attente est défini comme le temps d'attente d'un passager sans la contrainte de queue d'attente. Notons n_i le nombre de demandes des passagers pour le jour i et $t_{i,1} < \cdots < t_{i,n_i}$ les moments de la journée de ces demandes. Soit $t'_{i,j}$ l'heure d'arrivée du conducteur pour la demande du passager au moment $t_{i,j}, i = 1, \ldots, N$ et $j = 1, \ldots, n_i$. Le temps d'attente perçu $w^*_{i,j}$ et le pseudo temps d'attente $w_{i,j}$ pour la demande du passager au moment $t_{i,j}$ sont définis par

$$\begin{aligned} w^*_{i,j} &= t^{'}_{i,j} - t_{i,j}, \\ w_{i,j} &= t^{'}_{i,j} - \max(t_{i,j}, t^{'}_{i,j-1}). \end{aligned}$$

La modélisation retenue pour ce second étage est la régression Gamma. Cette dernière permet d'exprimer une relation d'inverse proportionnalité entre les

temps d'attente et les flux de trafic. En effet, cette relation est visible sur les données observées sur les différentes lignes. Soit la période de 24 heures d'une journée divisée en S intervalles égaux $I_1 < \cdots < I_S$, la fraction du flux quotidien de conducteurs y_i sur chaque intervalle est $y_i\beta_s$ avec $\beta_s \geq 0$ avec $\sum_{s=1}^{S} \beta_s = 1$. Conditionnellement au flux de trafic y_i, au temps de demande des passagers $t_{i,j} \in I_s$ et $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_S)$, nous supposons que les pseudos temps d'attente $w_{i,j}$ sont des variables aléatoires Gamma indépendantes de paramètres ν et β_s y_i, c'est-à-dire que

$$w_{i,j}|(y_i, \boldsymbol{\beta}, t_{i,j} \in I_s) \sim \Gamma(\nu, \beta_s y_i) \tag{1.4}$$

avec $i = 1, \ldots, N$ et $j = 1, \ldots, n_i$. Cette approche suppose que les pseudos temps d'attente dépendent de l'heure de la journée et du flux quotidien des conducteurs. Par définition, la moyenne conditionnelle du pseudo temps d'attente suivant notre modélisation est

$$\mathbb{E}[w_{i,j}|(y_i, \boldsymbol{\beta}, t_{i,j} \in I_s)] = \frac{\nu}{\beta_s y_i},$$

ce qui est cohérent avec notre intuition de la relation inverse entre le flux de conducteurs et le temps d'attente. Une illustration de ce modèle à deux étages est donnée en Figure 1.15.

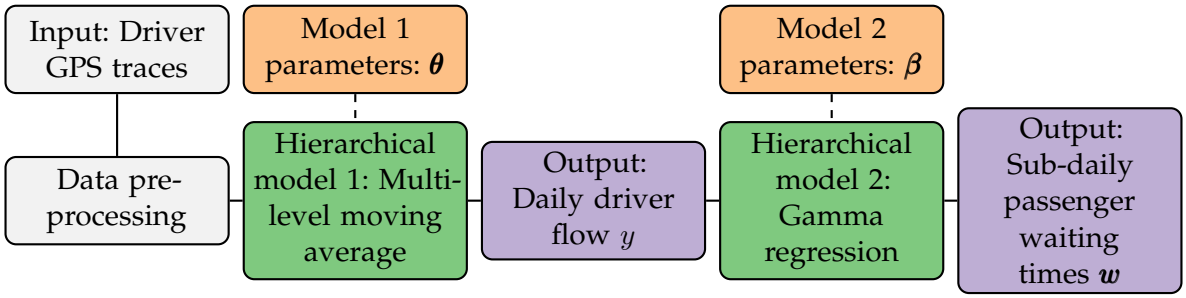


Figure 1.15 – Schéma du modèle hiérarchique bayésien pour la prédiction des flux de conducteurs et des temps d'attente des passagers. Les données d'entrée (traces GPS des conducteurs) sont en gris, les modèles hiérarchiques en vert, les paramètres du modèle en orange et les résultats du modèle en violet.

Ainsi, la validation du modèle est faite sur des données simulées qui sont générées en suivant le modèle génératif basé sur les données réelles. La Figure 1.16 représente les quantiles prédits par la loi prédictive a posteriori du modèle ainsi que les pseudos temps d'attente simulés.

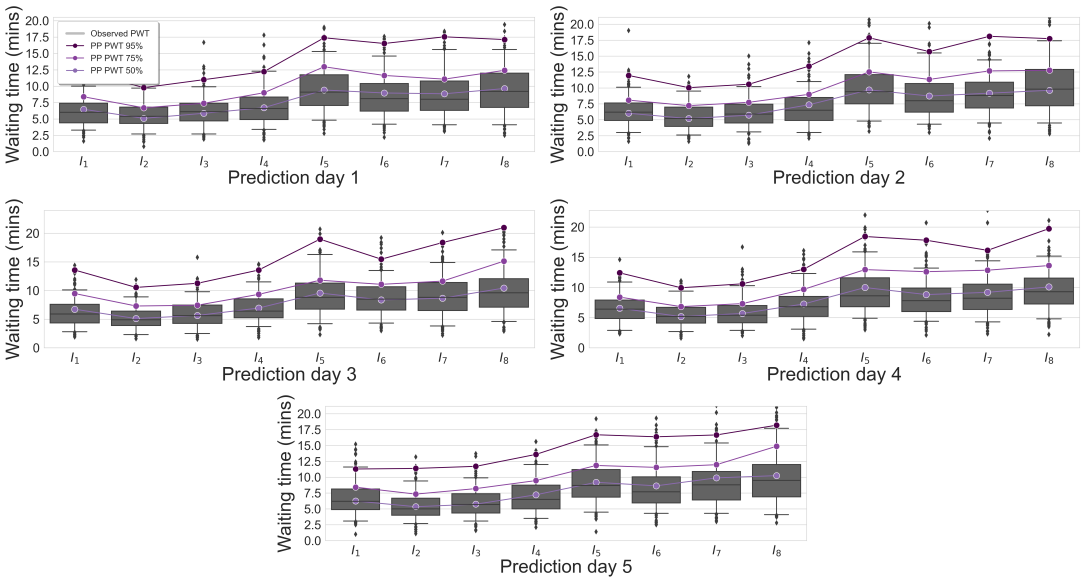


Figure 1.16 – Les box-plot gris sont les pseudos temps d'attente simulés pour des intervalles de 3 heures, pour les $\tilde{N}$ jours de prédiction. Les prédictions des quantiles des pseudos temps d'attente niveau 50%, 75% et 95% sont respectivement les cercles violet clair, violet moyen clair et violet foncé.

Par la suite, nous examinons les performances du modèle hiérarchique bayésien sur les données réelles d'Ecov. Les performances du premier étage du modèle, concernant les flux quotidiens, sont comparées à des modèles concurrents. Notons BHML le modèle présenté dans ce chapitre et BASE un modèle de moyenne coulissante fréquentiste ainsi que PROP le modèle nommé Prophet (modèle bayésien dédié aux séries temporelles). La Figure 1.17 représente l'erreur MSE de ces trois différents modèles pour des échantillons différents.

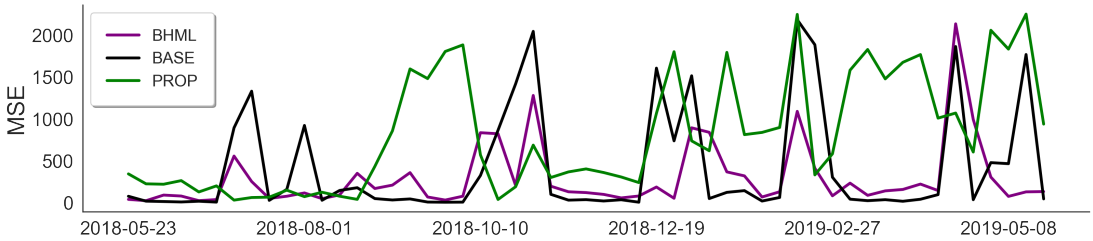


Figure 1.17 – Évolution de l'erreur MSE pour la prédiction du flux quotidien de conducteurs pour différentes périodes. Le modèle hiérarchique bayésien BHML est en violet, le modèle fréquentiste de base BASE en noir et le modèle bayésien Prophet PROP en vert.

Nous observons que le modèle BHML est plus performant que les autres pour quasiment tous les échantillons de tests. Les prédictions sont présentées par la

Figure 1.18. Le modèle PROP prédit généralement des quantités trop faibles pour les jours de semaine et trop élevées pour les week-ends par rapport aux flux de conducteurs observés, tandis que le modèle BASE semble avoir des difficultés à prédire le flux pour les jours de type PWE.

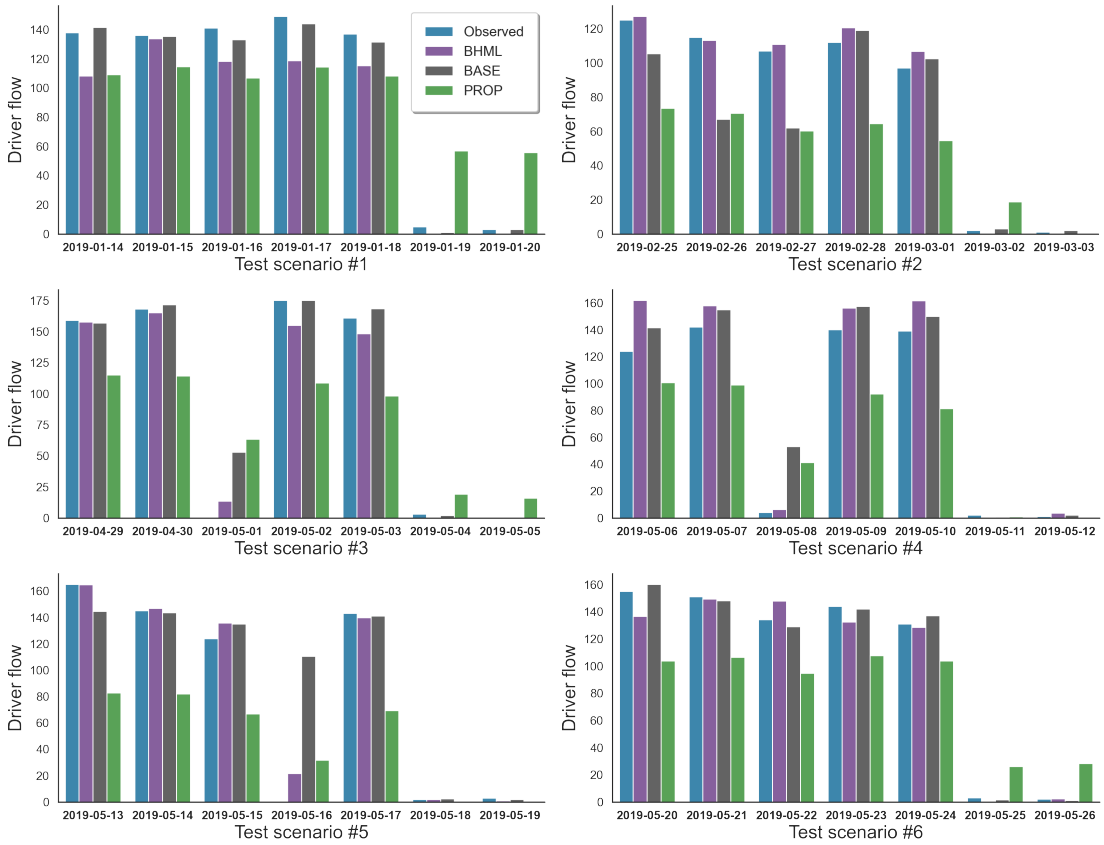


Figure 1.18 – Prédictions du flux quotidien de conducteurs pour six différentes périodes de test. Les flux journaliers de conducteurs observés sont en bleu. Le modèle hiérarchique bayésien BHML est en violet, le modèle fréquentiste de base BASE en noir et le modèle bayésien Prophet PROP en vert.

Enfin, les performances du deuxième étage du modèle hiérarchique bayésien sur les vrais pseudos temps d'attente sont illustrées par la Figure 1.19. L'avantage du modèle BHML est que nous disposons de toute la distribution prédictive a posteriori des temps d'attente, ce qui est plus complet par rapport aux prédictions ponctuelles ou par intervalles des modèles de régression classique. La médiane et le quartile supérieur des pseudos temps d'attente prédits ont tendance à suivre ceux des pseudos temps d'attente observés, en particulier pour les intervalles 06:00-07:00, 17:00-18:00 et 18:00-19:00. Avec les prédictions du BHML, nous pouvons affirmer que 95% des pseudos temps d'attente pour les demandes des passagers ne dépassent pas un seuil de 15 minutes pendant

la plupart des heures d'ouverture.

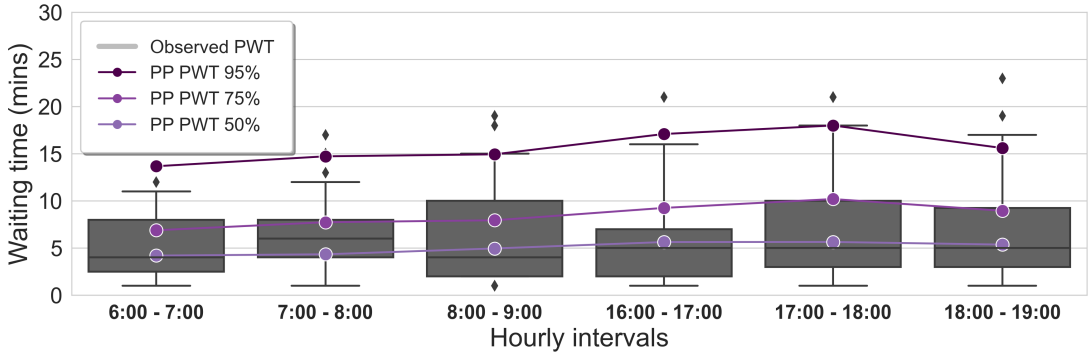


Figure 1.19 – Les box-plot gris sont les pseudos temps d'attente observés pour des intervalles d'une heure. Les prédictions des quantiles des pseudos temps d'attente niveau 50%, 75% et 95% sont respectivement les cercles violet clair, violet moyen clair et violet foncé.

Nous rappelons que nous nous sommes concentrés sur les pseudos temps d'attente $w_{i,j}$, bien que les temps d'attente perçus $w_{i,j}$ soient plus pertinents, puisque ces derniers sont les vrais temps d'attente du point de vue du passager. Pour cela, en plus du modèle hiérarchique bayésien, nous avons proposé un cadre pour l'analyse future des temps d'attente perçus en nous basant sur des simulations poissonniennes des arrivées des passagers. Puisque la distribution prédictive a posteriori complète est disponible, nous pouvons l'intégrer par rapport à la distribution des arrivées passagers pour obtenir les pseudos temps d'attente. Ensuite, nous sommes alors en mesure de reconstruire les temps d'attente perçus à l'aide de l'équation suivante

$$w^*_{i,j+1} = w_{i,j+1} + [w_{i,j} - \zeta_i | (w_{i,j} > \zeta_i)]$$

avec $\zeta_i = t_{i,j+1} - t_{i,j}$.

La contribution essentielle du chapitre est la prédiction des flux quotidiens des conducteurs et des pseudos temps d'attente des passagers à l'aide d'un modèle hiérarchique bayésien de deux étages. La principale difficulté était de combiner deux sources (traces GPS des conducteurs et temps d'attente des passagers) ayant une temporalité différente. Une deuxième contribution est celle concernant la reconstruction des temps d'attente perçus à l'aide du cadre bayésien qui nous donne accès à la totalité de la loi prédictive a posteriori.

Chapitre 5 : Construction d'un prior informatif par transfert bayésien: application au covoiturage

À présent, nous souhaitons transférer l'apprentissage du modèle bayésien hiérarchique présenté dans le Chapitre 4 d'une ligne de covoiturage vers une nouvelle ligne. Ce transfert bayésien va permettre d'améliorer les performances du prédicteur sur le nouveau jeu de données qui est de petite taille. Disposer d'un modèle prédictif performant dès l'ouverture d'une nouvelle ligne de covoiturage est un enjeu stratégique pour assurer dès son départ son bon fonctionnement.

Dans le cas des statistiques bayésiennes nous pouvons introduire de l'information sur les paramètres du modèle grâce à la construction d'un prior informatif. Dans ce chapitre nous développons un prior informatif afin d'améliorer les performances d'un modèle bayésien lors du lancement d'une nouvelle ligne de covoiturage. En notant $\mathcal{L}$, le jeu de données long qui va servir à la construction du prior, et $\mathcal{C}$ le jeu de données court, où on va appliquer le transfert bayésien. Le prior informatif qu'on note $\pi^{\mathcal{C}}$ pour le jeu de données $\mathcal{C}$ est construit à partir de la loi a posteriori $\pi^{\mathcal{L}}(\bullet|X^{\mathcal{L}})$ estimée sur le jeu de données $\mathcal{L}$. On suppose que les lois a posteriori convergent vers leurs lois limites dans le cas du jeu de données $\mathcal{C}$. En revanche, dans le cas de la situation d'historique court, les lois a posteriori n'ont pas eu le temps de converger. On suppose de ce fait que les données des nouvelles lignes (historique court) ressemblent aux données des lignes ayant un historique long. Les paramètres d'intérêts pour la construction sont respectivement $\theta^{\mathcal{L}}$ et $\theta^{\mathcal{C}}$. La construction du prior informatif s'effectue en deux temps:

- On commence par ajuster le modèle non-informatif sur le jeu de données $\mathcal{L}$.
- On construit ensuite le prior informatif qu'on va détailler ensuite pour le jeu de données $\mathcal{L}$.

La loi informative est construite à partir d'une loi a priori hiérarchique normale qui s'appuie sur la moyenne $\mu^{\mathcal{L}}$ et et la variance $\Sigma^{\mathcal{L}}$ de la loi a posteriori $\pi^{\mathcal{L}}(\theta|X^{\mathcal{L}})$. Sa forme est

$$\begin{aligned}\theta|k,r &\sim \mathcal{N}(k\mu^{\mathcal{L}}, r\Sigma^{\mathcal{L}}),\\ k|m,l &\sim \mathcal{N}(m(1,\dots,1)', l\,\mathrm{Id}).\end{aligned}$$

Les lois a priori des hyperparamètres r, m et l sont définies par

$$\begin{aligned}r &\sim \mathcal{N}(a_r, b_r),\\ m &\sim \mathcal{N}(1, \sigma_m^2),\\ l &\sim \mathcal{N}(1, \sigma_l^2),\end{aligned}$$

avec a_r et b_r des réels positifs fixés afin que la loi a priori de r soit non-informative. À terme, le prior informatif est défini par

$$\pi(\theta, k, r, m, l) \sim \pi(\theta|k, r)\pi(k|m, l)\pi(m)\pi(r)\pi(l)\pi(\sigma_m^2)\pi(\sigma_l^2) \tag{1.5}$$

avec les lois a priori des hyperparamètres σ_l^2 et σ_m^2 des priors non-informatifs de Jeffreys. Le schéma de la méthodologie du transfert bayésien par construction du prior est illustré à la Figure 1.20.

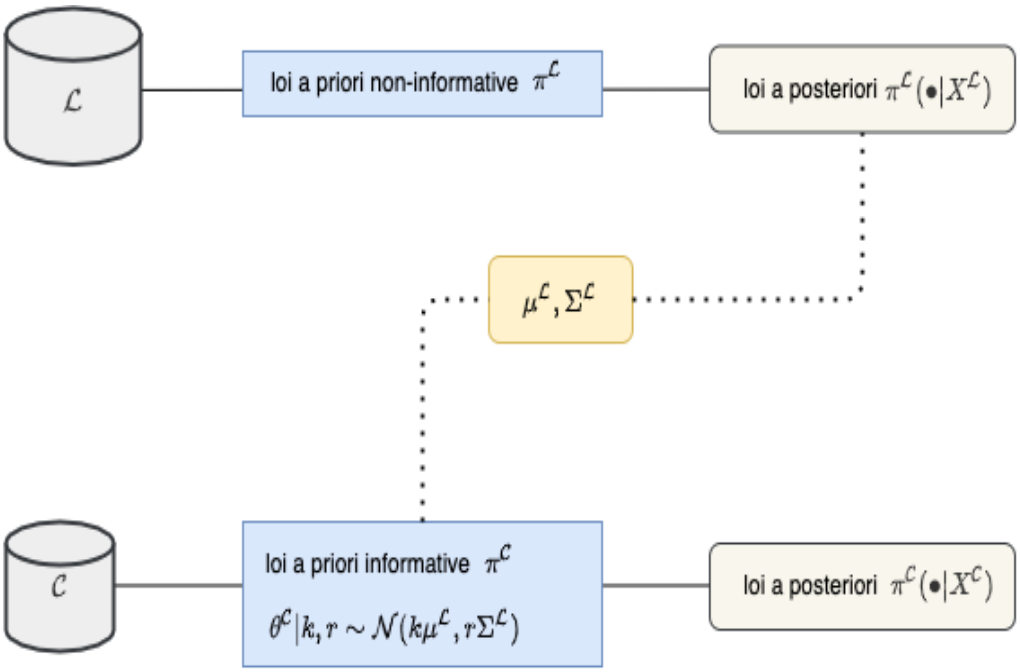


Figure 1.20 – Schéma de la méthode proposée pour la construction du prior informatif en situation d'historique court.

Une fois le modèle fait, nous cherchons à évaluer les résultats du modèle avec prior informatif et sans prior informatif, dans un premier temps sur des données jouets simulées, puis, dans un second temps sur les données réelles d'Ecov. Plus précisément, nous comparons la qualité de l'estimation des paramètres sur les données simulées pour lesquelles les vrais paramètres sont connus. En ce qui concerne les performances prédictives, nous évaluons l'apport du prior informatif sur les données réelles issues des lignes de covoiturage.

Les données simulées sont issues du protocole détaillé dans le Chapitre 4 qui permet de contrôler les valeurs qu'on se fixe pour $\theta^{\mathcal{L}}$ et $\theta^{\mathcal{C}}$. Ainsi, on simule les deux jeux de données en perturbant les paramètres d'un coefficient λ, i.e. $\theta^{\mathcal{L}} = \lambda\theta^{\mathcal{C}}$. Nous comparons ensuite les lois a posteriori de $\theta^{\mathcal{C}}$ en non-informatif et en informatif. La Figure 1.21 illustre les distributions des lois a posteriori des

paramètres du modèle informatif et non-informatif. Nous constatons que le prior informatif améliore les estimations des paramètres du modèle.

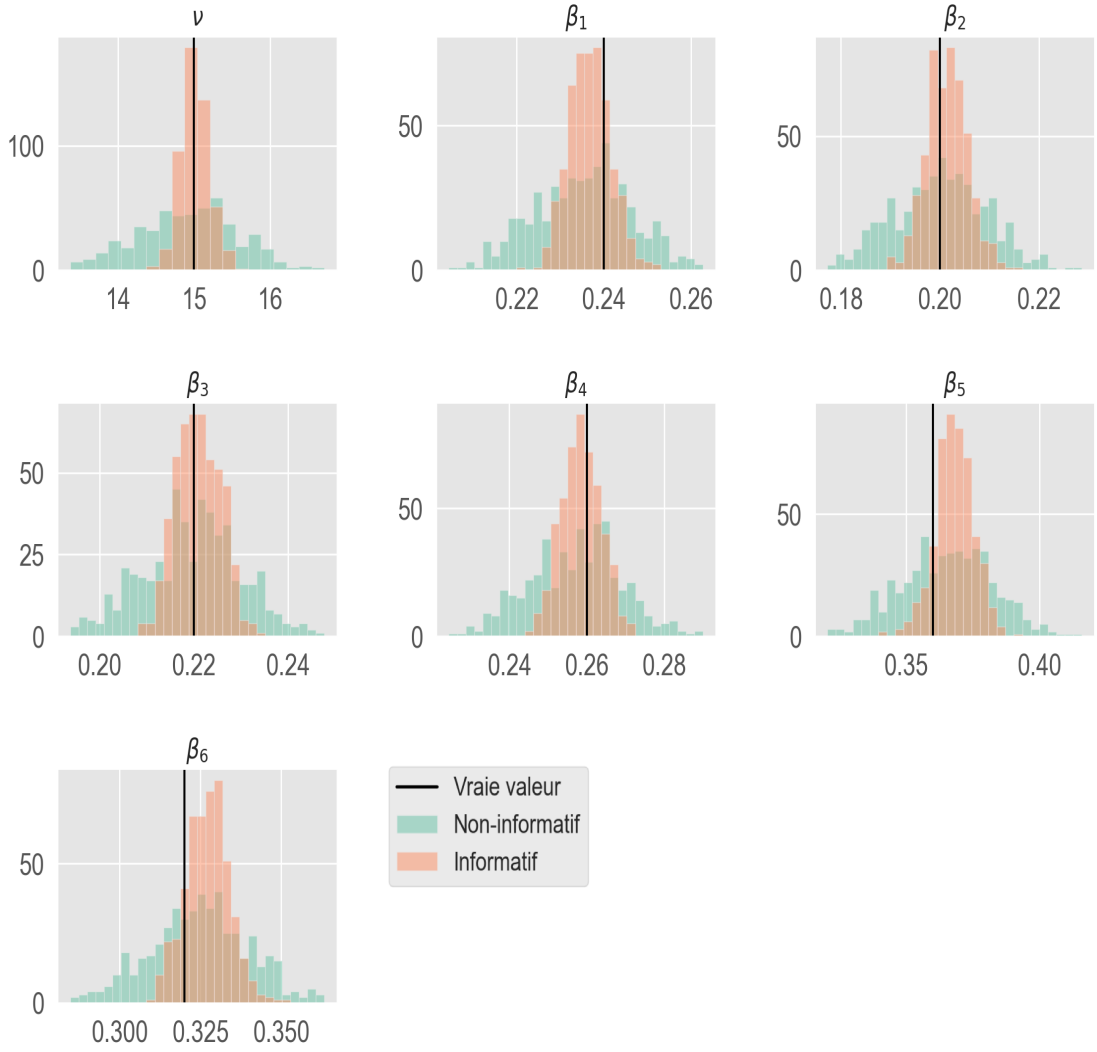


Figure 1.21 – Histogrammes des lois a posteriori des paramètres du modèle informatif en orange et du modèle non-informatif en vert.

En suivant la même méthodologie, nous avons construit le prior informatif sur les données réelles d'Ecov. Le jeu de données long est noté ici $\mathcal{A}$ et représente une ligne de covoiturage ayant le plus d'historique. Le jeu de données court, qu'on note ici $\mathcal{B}$, représente les données d'une ligne de covoiturage récente avec peu d'historique. L'apport du prior informatif est visible sur les lois a posteriori des paramètres du modèle aussi bien que sur les lois prédictives a posteriori des temps d'attente. La comparaison des lois a posteriori des paramètres du modèle en informatif et non-informatif pour le jeu de données $\mathcal{B}$ est illustrée

par la Figure 1.22. Comme dans le cas des données simulées nous observons une nette amélioration des estimations des paramètres dans le cas du modèle informatif.

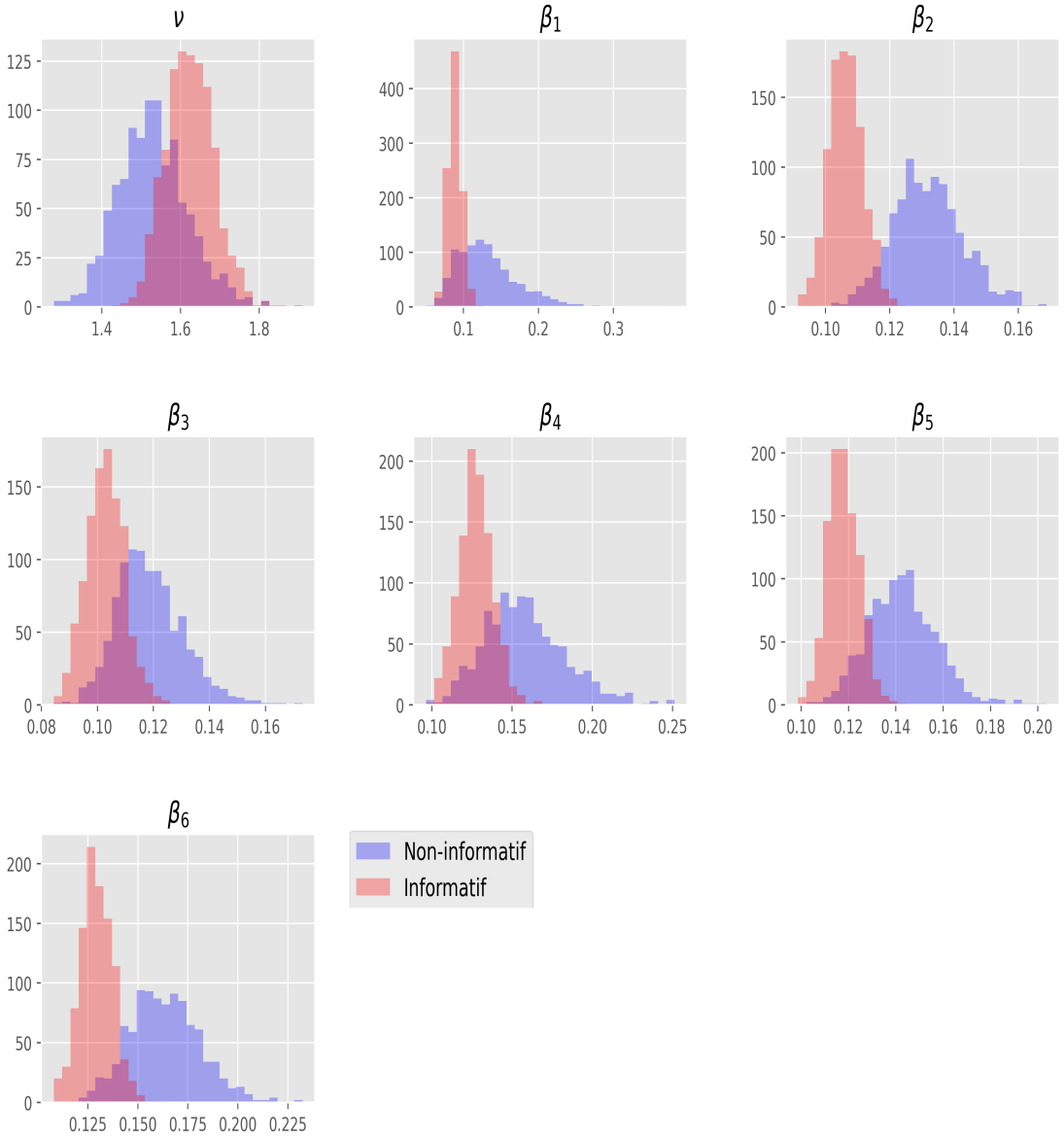


Figure 1.22 – Histogrammes des lois a posteriori des paramètres du modèle de temps d'attente pour le jeu de données $\mathcal{B}$. En bleu, les lois a posteriori avec un prior non-informatif et en rouge, les lois a posteriori avec un prior informatif.

Nous regardons aussi les performances prédictives des modèles informatifs et non-informatifs issues du jeu de données $\mathcal{B}$. La Figure 1.23 compare les intervalles de plus haute densité des lois prédictives du modèle informatif et

non-informatif sur le jeu de données $\mathcal{B}$. On obtient que les intervalles de plus haute densité des lois issues du modèle informatif sont plus courts et donc plus précis que le modèle non-informatif.

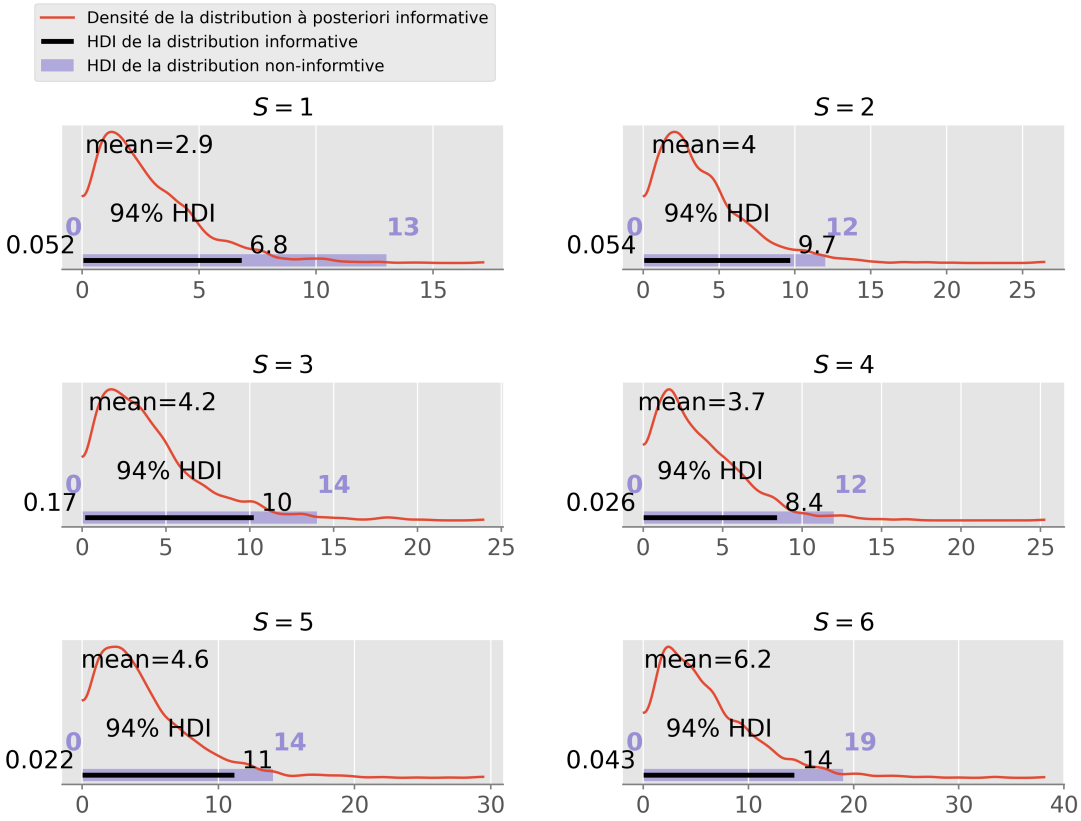


Figure 1.23 – Comparaison des lois a posteriori prédictives des temps d'attente pour le premier jour de prédiction pour les six créneaux du jeu de données $\mathcal{B}$. La courbe en rouge est la loi prédictive lorsque le prior est informatif et en noir sa région HDI (Highest Density Interval). En violet, la région HDI de la loi prédictive des temps d'attente en non-informatif.

Afin de s'assurer que ces lois prédictives sont en effet plus performantes nous devons aussi étudier le score PE. Nous rappelons que ce score mesure le pourcentage des bonnes prédictions en dessous d'un seuil δ (en minutes) donné. Son expression est

$$\mathrm{PE}(\delta) = \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} \mathbf{1}\{|q_\alpha(\tilde{w}_i) - w_i| < \delta\} \tag{1.6}$$

avec $q_\alpha(\tilde{w}_i)$ est le quantile au niveau α de la loi prédictive a posteriori de la $i^{ième}$ observation et w_i le $i^{ième}$ temps d'attente observé pour un échantillon de taille $\tilde{N}$.

Le quantile sélectionné pour comparer les scores PE est le quantile niveau 95% des lois a posteriori prédictives des temps d'attente. La Figure 1.24 compare les scores PE du quantile niveau 95% issues du modèle informatif et non-informatif pour le jeu de données $\mathcal{B}$. Nous constatons que le score du modèle informatif est meilleure que celle du modèle non-informatif. Ce résultat montre que la loi a priori informative apporte une réelle information supplémentaire.

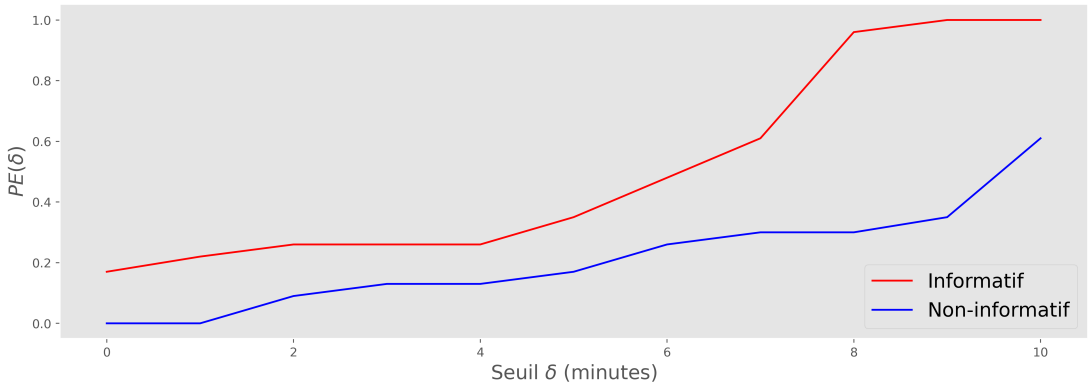
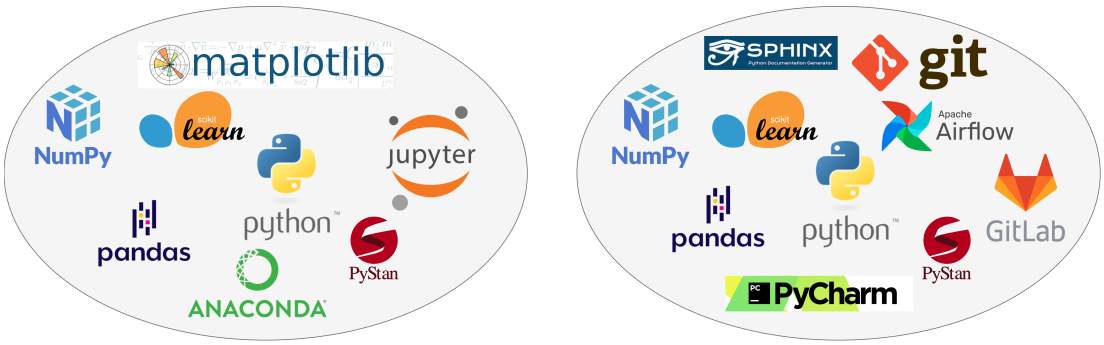


Figure 1.24 – Comparaison de le score PE du quantile posterior 95% pour le jeu de données $\mathcal{B}$. En bleu le score PE pour le modèle avec prior non-informatif et en rouge en avec prior informatif.

Chapitre 6 : Valorisation industrielle - Mise en production du modèle hiérarchique bayésien

Nous souhaitons à présent exploiter de manière industrielle les travaux exposés dans les chapitres précédents. En d'autres termes, l'objet de ce chapitre est de présenter les différentes étapes pour la mise en production du modèle hiérarchique bayésien introduit dans le Chapitre 4.

La première étape est la construction du paquet Python nommé *bayesian-hierarchical-model* reprenant les outils et les algorithmes développés dans ce manuscrit. Le développement de ce paquet a nécessité d'assurer le passage d'un modèle prédictif expérimental vers une version stable et industrielle de celui-ci. À ce titre, nous avons utilisé différents outils informatiques, certains d'entre eux sont dédiés pour la phase de prototypage du modèle et d'autres pour la phase d'industrialisation. Nous avons cartographié à la Figure 1.25 les outils informatiques choisis durant cette thèse pour les deux phases.



Prototypage du modèle — Industrialisation du modèle

Figure 1.25 – Outils principaux utilisés pour le prototypage du modèle (à gauche) et pour son industrialisation (à droite).

La structure du paquet développé suit les standards qu'on rencontre dans les paquets d'apprentissage statistique. Afin de donner une autonomie totale aux futurs utilisateurs du paquet nous avons implémenté dans le paquet les simulateurs de données introduit dans le Chapitre 4. Cela donne la possibilité d'utiliser le modèle hiérarchique bayésien sur des données simulées mais aussi sur des données réelles. Nous présentons en détail toutes les méthodes et les fonctions disponibles dans le paquet *bayesian-hierarchical-model*. Nous résumons ci-dessous les fonctionnalités principales:

- La récupération des données et leurs pré-traitements (dans le cas des données réelles) ou la simulation des données synthétiques (dans le cas des données simulées).
- L'entraînement du modèle hiérarchique bayésien sur les données récupérées ou simulées.
- La génération des lois a posteriori des paramètres du modèle ainsi que les lois prédictives a posteriori des temps d'attente.
- La génération des quantiles prédictifs a posteriori souhaités provenant de la loi prédictive a posteriori des temps d'attente.
- Le contrôle prédictif a posteriori du modèle hiérarchique bayésien.

Ensuite, nous nous intéressons à l'automatisation de l'utilisation du paquet *bayesian-hierarchical-model*. En effet, il est de grand intérêt de pouvoir automatiser le processus de prédiction des temps d'attente. Nous rappelons que les résultats obtenus par le modèle prédictif sont consommés par plusieurs services d'Ecov (site internet, application mobile, borne d'informations voyageurs, logiciel de tableau de bord). La solution retenue pour automatiser l'exécution du

modèle est Airflow[2]. Il s'agit d'une plateforme de planification de flux de travail. La Figure 1.26 illustre le schéma suivi pour l'intégration des données, leurs traitements, l'entraînement du modèle et l'exposition des résultats. Ce schéma fonctionne selon le concept d'ETL (Extract Load Transform). En particulier, nous avons segmenté les différentes tâches nécessaires en trois ETL dépendants les uns des autres.

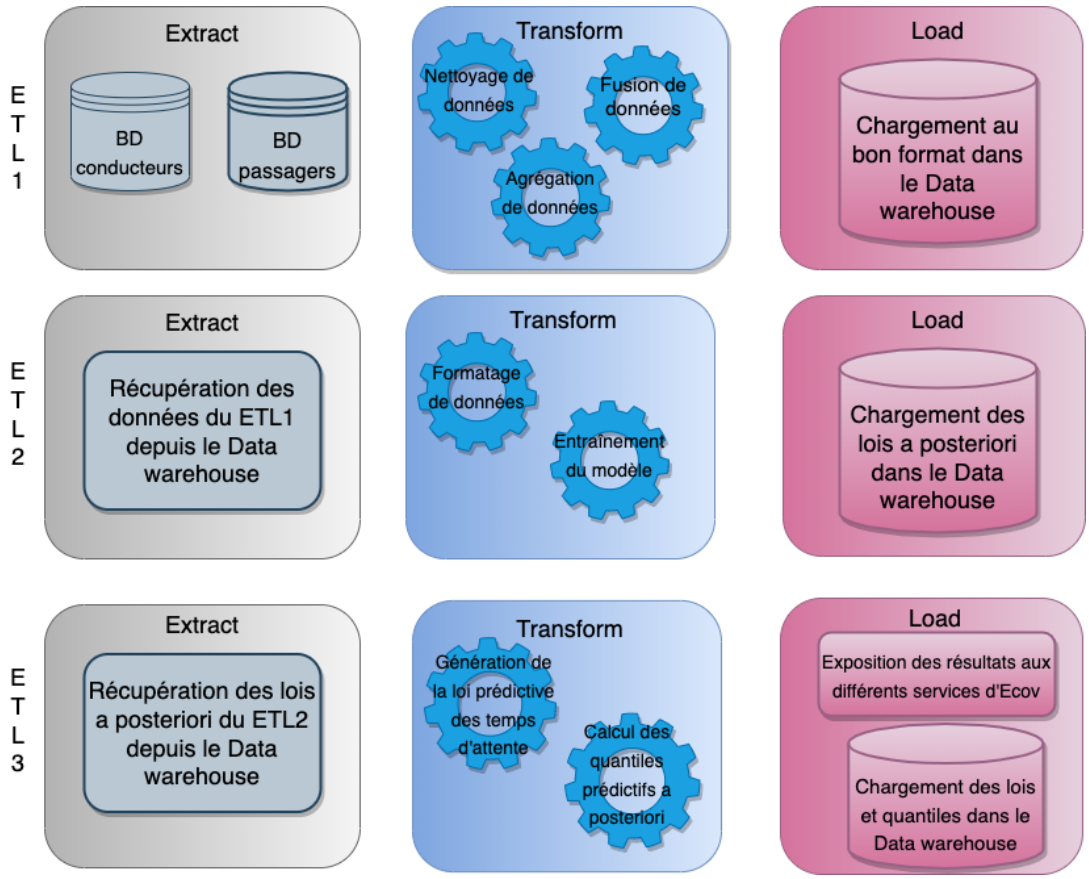


Figure 1.26 – Schéma de la structure des ETL mis en place afin d'exposer en production les prédictions des flux de trafic et des temps d'attente aux différents services d'Ecov.

2. https://airflow.apache.org/

CHAPTER 2

QUANTILE REGRESSION FOR WAITING TIME PREDICTION IN A STOCHASTIC CARPOOLING NETWORK

2.1 Introduction

The carpooling services provided by Ecov[1] differ from the usual carpooling services, since they are based on *stochastic matching* between driver(s) and passenger(s). To facilitate this matching, physical meeting points are constructed on the roadside. These physical meeting points are then equipped with electronic signs which display the carpooling requests made by the passengers to drivers on the road. The potential drivers thus can see a carpooling request in real-time and can decide to stop and embark the passenger(s) directly. This shifts away from a deterministic matching used by Uber-like and taxi-like services. The placement of these meeting points are decided with local government authorities in order to respond to the mobility requirements in each local area, and they constitute a carpooling network as illustrated in Figure 2.1.
The objective of Ecov is to design a carpooling service for frequent, short distance journeys (from 10km to 40km roughly) in sparsely populated peri-urban and rural areas. Its first carpooling network was located in the northwestern of France, a region called "Vexin" as shown in Figure 2.2. In the early experimental phase of the implementation of these carpooling services, the passengers are even allowed to request a carpooling trip to any destination within the service area, so the carpooling network is even more complicated than that represented in the schematic diagram in Figure 2.1 where the connected edges indicate carpooling was provided only between the meeting points. Whilst this is highly convenient for the passengers, the complete freedom in the choice of the destination leads to an explosion of the number of possible origin-destination pairs which a reliable carpooling service must be assured.

A key component in the reliability of a carpooling (or indeed an transportation service) is the accuracy of the predicted waiting times of passengers, since

1. www.ecov.fr

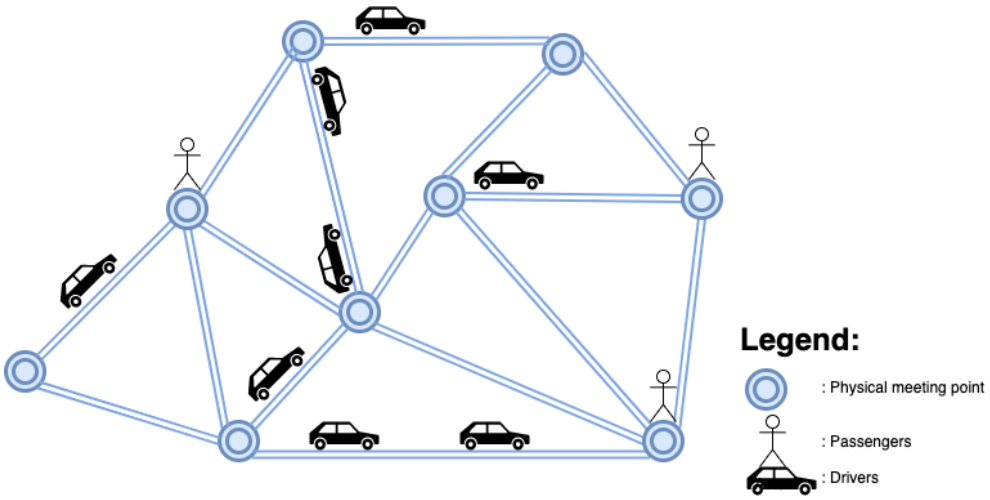


Figure 2.1 – Schematic of an Ecov carpooling network. The blue circles are the physical meeting points where the passenger can request a carpooling, and links between the circles indicate that a carpooling service is assured for this origin-destination pair.

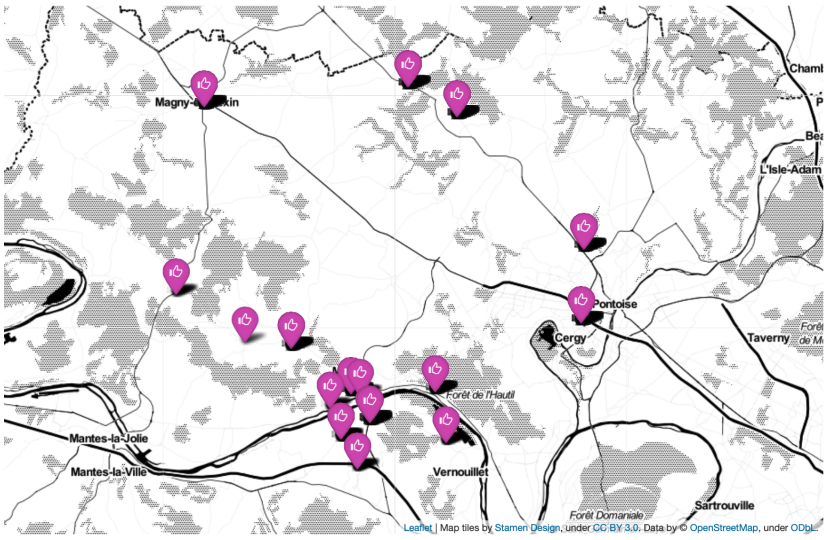


Figure 2.2 – Map of the "Vexin" carpooling network. The physical meeting points where passengers are able to make a carpooling request are indicated by the pink circles with a thumb logo.

the latter often prioritise the estimated time of arrival (ETA) when choosing between different modes of transport, along with the financial cost. For Ecov, as a transport provider, it is crucial to (i) quantify accurately the passenger waiting times and (ii) to reduce any excessive waiting times to a level suited to the market. These reliable waiting times are the foundation for building,

maintaining and expanding the carpooling service.

We focus on the first of these requirements. Therefore, we must be able to answer a passenger's question "What is the waiting time for my carpooling request to my destination if I make the request now from this location?" as accurately as possible. Whilst it appears that this can be answered by supplying the mean of the observed waiting times, this can hide the large variability that is inherent in predicting waiting times for the arrival of the suitable carpooling driver without any prior matching. To supply accurate information to the passenger based on highly variable stochastic phenomena, Ecov wishes to provide the response to the passenger's question above in the form of a probabilistic statement of the type that the

$$\text{passenger will wait less than } y \text{ minutes with } \alpha\% \text{ chance.}$$

Mathematically, this $\alpha \in (0,1)$ defines the α-quantile of Y, where Y is the random variable of the carpooling request waiting time. The quantile of Y is more precisely defined by

$$q_\alpha(Y) = \inf_{t\in\mathbb{R}}\{F_Y(t) \geq \alpha\}, \tag{2.1}$$

where F_Y is the cumulative distribution function of Y, i.e. $F_Y(y) = \mathbb{P}(Y \leq y)$. The availability of auxiliary information, e.g. about the geographical and temporal characteristics of the carpooling request, can greatly assist in the prediction of the quantiles of the waiting times. This auxiliary information can be incorporated into a single, unified framework of Quantile Regression. The waiting time predictions from a Quantile Regression are more detailed than simple mean predictions, which can be utilised to reassure passengers about the uncertainties in the stochastic matching in an Ecov carpooling service.

In Section 2.2 we focus on the definition of Quantile Regression as a minimisation problem, and then elaborate the solutions provided by the classical linear model approach, the Random Forests, the Generalised Random Forests approaches, and the gradient boosting approach. In Section 2.3 we describe in further detail the data collected from the Vexin carpooling network and then compare these different implementations of Quantile Regressions for these carpooling requests. We end with some concluding remarks and some details of the supporting mathematical results.

2.2 Quantile Regression

Classical regression relies on a well-known result that the mean of the response random variable Y is the minimiser of the expected squared error loss problem

$$\mathbb{E}[Y] = \operatorname*{argmin}_{a\in\mathbb{R}} \mathbb{E}[(Y-a)^2].$$

That is, $\mathbb{E}[Y]$ is the value that has smallest possible expected squared loss from the response Y. Then it is straightforward to pose the question of what the result would be if the squared distance were replaced by other distance measures. It is also well-known that for the absolute loss, then the result is the median of Y, i.e.

$$\text{med}[Y] = \underset{a \in \mathbb{R}}{\text{argmin}}\, \mathbb{E}[|Y - a|].$$

Since the median is the special case of the 0.5-quantile, then a further question to ask is which loss function would result in a generalisation of this median result to an arbitrary α-quantile, for any $0 \leq \alpha \leq 1$. This is achieved by the Pinball loss

$$\rho_\alpha(y) = y(\alpha - \mathbf{1}_{[y \leq 0]}), \tag{2.2}$$

where $\mathbf{1}_{[\cdot]}$ is the indicator function. The Pinball loss in Equation (2.2) is a convex loss function so it leads to a well-defined minimisation problem. In Figure 2.3 are illustrated the curves for the Pinball loss for $\alpha = 0.25, 0.5, 0.75, 0.95$. Apart from the green curve ($\alpha = 0.5$) which is symmetric, all the other Pinball losses are asymmetric about the origin (blue $\alpha = 0.95$, orange $\alpha = 0.75$, red $\alpha = 0.25$). The induced asymmetry allows for y and $-y$, whilst they have the same absolute loss $|y|$ and squared loss y^2, to have different Pinball losses. Since the curves of the Pinball loss consist of straight lines which change direction at 0, they resemble the trajectories of a ball in a Pinball arcade machine, hence their name. From a mathematical point of view, the sign of y in the Pinball loss plays an important role, unlike the more commonly considered squared and absolute losses which are non-negative loss functions.

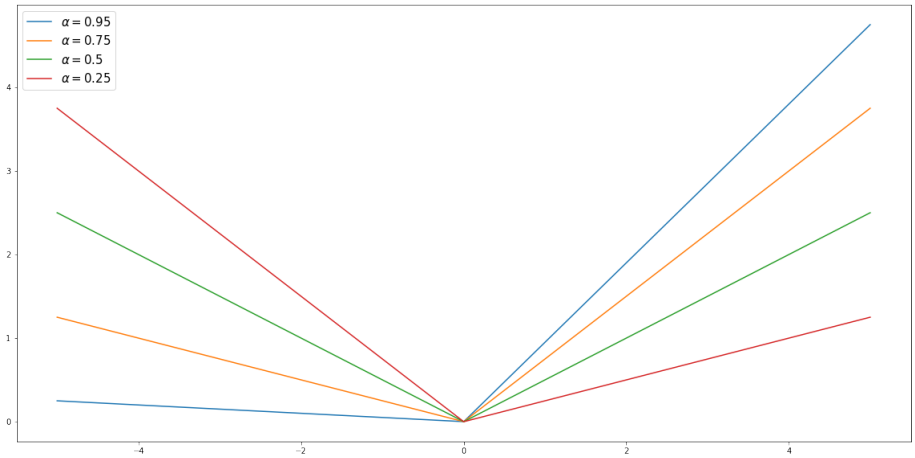


Figure 2.3 – Pinball loss functions ρ_α for different values of α. Blue: $\alpha = 0.95$, orange $\alpha = 0.75$, green $\alpha = 0.5$, red $\alpha = 0.25$.

An equivalent definition for the quantile given by Equation (2.1) is given by the

solution of the Pinball loss minimisation problem, as asserted in Lemma 1.

Lemma 1. *Let Y be an integrable real-valued random variable whose distribution function F_Y is strictly increasing. Let $\alpha \in (0,1)$. Then the quantile q_α is the unique value which satisfies*

$$q_\alpha = \underset{a\in\mathbb{R}}{\operatorname{argmin}}\, \mathbb{E}[\rho_\alpha(Y-a)].$$

Whilst this is an established result, we provide its proof in Appendix 2.5.1. In practice, the complete random variable Y is not known, and we only have observed instantiations of it, i.e. $Y_1, \ldots, Y_n$ which is a random sample from F_Y. As a result of Lemma 1, we can define an estimator of the quantile by a principle of M-estimation (empirical risk minimisation). Thus, the empirical estimator of the α-quantile is

$$\hat{q}_\alpha \in \underset{a\in\mathbb{R}}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \rho_\alpha(Y_i - a) \right\}.$$

Van der Vaart (2000, Chapter 21) establishes the weak consistency of this estimator under weaker conditions than those of Lemma 1. Koenker, Chesher, et al. (2005, Chapter 4) establishes the consistency of this estimator, i.e. $||\hat{q}_\alpha - q_\alpha|| \to 0$ as $n \to \infty$ under the conditions of Lemma 1.

For the regression context, the random vector $\boldsymbol{X}$, known as the explanatory variables or covariates, provides auxiliary information of the behaviour of the response variable Y. Analogous to the situation above where unconditional mean $\mathbb{E}[Y]$ is the minimiser of an expected squared loss problem, then conditional mean $\mathbb{E}[Y|\boldsymbol{X}]$ is the minimiser defined by

$$\mathbb{E}[Y|\boldsymbol{X}] = \underset{f\in L^2(\boldsymbol{X})}{\operatorname{argmin}}\, \mathbb{E}[(Y-f)^2],$$

where $L^2(\boldsymbol{X}) = \{f : \mathbb{E}[f(\boldsymbol{X})^2] < \infty\}$ is the space of all square integrable functions with respect to $\boldsymbol{X}$. It is straightforward to replace Y by $Y|\boldsymbol{X}$ in Equation (2.1) to obtain the conditional α-quantile as

$$q_\alpha(Y|\boldsymbol{X}) = \inf_{t\in\mathbb{R}}\{F_{Y|\boldsymbol{X}}(t) \geq \alpha\}, \tag{2.3}$$

where $F_{Y|\boldsymbol{X}}$ is the conditional distribution of $Y|\boldsymbol{X}$. However, this definition is not tractable for many reasons. Therefore we propose in Lemma 2 to provide an equivalent definition of Equation (2.3). As described in Armerin, 2014, it is a generalisation of the Pinball loss minimisation problem for the conditional α-quantile.

Lemma 2. *Let Y be an integrable real-valued random variable whose conditional distribution $F_{Y|\boldsymbol{X}}$ is almost surely increasing given a random vector $\boldsymbol{X}$. Let $\alpha \in (0,1)$. Then the conditional quantile $q_\alpha(Y|\boldsymbol{X})$ satisfies*

$$q_\alpha(Y|\boldsymbol{X}) = \underset{f\in L^1(\boldsymbol{X})}{\operatorname{argmin}}\, \mathbb{E}[\rho_\alpha(Y-f)],$$

where $L^1(\boldsymbol{X}) = \{f : \mathbb{E}[|f(\boldsymbol{X})|] < \infty\}$ *is the space of all integrable functions with respect to* $\boldsymbol{X}$.

The proof of Lemma 2 is given in Appendix 2.5.2. With this lemma, we are able to proceed to the definition of a Quantile Regression.

Quantile Regression was introduced in Koenker & Bassett Jr, 1978 and then deepened in Bassett Jr et al., 1982 and Koenker & Hallock, 2001. Just as classical regression methods are based on minimising the expected squared loss in order to estimate the conditional mean, the aim of Quantile Regression is to estimate conditional quantiles. Many authors have shown interest in Quantile Regression and have subsequently expanded on the standard techniques. For instance, the definition of the confidence intervals for quantiles regression estimates in Kocherginsky et al., 2005. Another is the adaptation for sequential time series forecasting proposed in Biau & Patra, 2011. More recently tree-based algorithms for Quantile Regression have been developed. For example, in Meinshausen, 2006b and in Athey et al., 2019 two different algorithms based on a Random Forests algorithm are proposed for Quantile Regression. Moreover, in Zheng, 2012 an adaptation of Gradient Boosting Machine for Quantile Regression is offered. We recall that in the regression context the aim is to give an estimate of $q_\alpha(Y|\boldsymbol{X})$, that we denote $\hat{q}_\alpha(Y|\boldsymbol{X})$, by using the observed responses Y_i given the explanatory variables $\boldsymbol{X}_i \in \mathbb{R}^p$ with $i = 1, \dots, n$. This scalar represents an estimation of the quantile of waiting times. For instance, if we set $\alpha = 0.9$ then $\hat{q}_{0.9}(Y|\boldsymbol{X})$ is the estimation of the 90% quantile. This value means that in 90% of the time, the observed waiting times does not exceed $\hat{q}_{0.9}(Y|\boldsymbol{X})$.

2.2.1 Linear Quantile Regression

The observed data set is denoted $\mathcal{D}_n = \{(\boldsymbol{X}_i, Y_i) : 1 \leq i \leq n\}$. Towards this goal of defining a regression model, we introduce a parametric definition of the quantile function, i.e. $q_\alpha(Y|\boldsymbol{X}) = f(\boldsymbol{X}, \beta_\alpha)$ where f is a known, non-random function satisfying the Lemma 2, and $\beta_\alpha \in \mathbb{R}^p$ is the regression parameter. The objective is to estimate the β_α parameter since it immediately yields an estimate of the quantile function as $\hat{q}_\alpha(Y|\boldsymbol{X}) = f(\boldsymbol{X}, \hat{\beta}_\alpha)$. The function f can take several forms, and we start with the historical approach, where it is a linear product of the covariate matrix and the regression parameter, that is $f(\boldsymbol{X}, \beta_\alpha) = \boldsymbol{X}'\beta_\alpha$. Following the Lemma 2, we obtain

the empirical estimator of the regression parameter is

$$\hat{\beta}_\alpha \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \rho_\alpha(Y_i - \boldsymbol{X}_i'\beta) \right\}.$$

Contrary to the classical regression case, here there is not an explicit expression for the solution for the quantile regression, as pointed out in Koenker &

Hallock (2001). To solve this problem, efficient numerical methods like linear programming are often deployed. For example, Portnoy et al. (1997) and Bosch et al. (1995) use an interior point method to compute $\hat{\beta}_\alpha$.

Despite that $\hat{\beta}_\alpha$ does not have an explicit expression, much is known about it. Its asymptotic normality and its rate of convergence is detailed in Koenker, Chesher, et al. (2005, Chapter 4) under the following conditions:

C1: The distribution function F_Y is absolutely continuous at Y_i, and its density is continuous and is uniformly bounded away from 0 and ∞ at the points $q_\alpha(Y_i|\boldsymbol{X}_i)$.

C2: There exist positive definite matrices D_0 and $D_1(\alpha)$ such that, as $n \to \infty$,

- $n^{-1} \sum_{i=1}^{n} \boldsymbol{X}_i \boldsymbol{X}_i' \to D_0$
- $n^{-1} \sum_{i=1}^{n} f_Y(q_\alpha(Y_i|\boldsymbol{X}_i)) \boldsymbol{X}_i \boldsymbol{X}_i' \to D_1(\alpha)$
- $\max_{i=1,\dots,n} n^{-1/2} \|\boldsymbol{X}_i\| \to 0$,

where f_Y is the density function of Y. When these two conditions are verified we obtain the asymptotic normality, i.e.

$$n^{1/2}(\hat{\beta}_\alpha - \beta_\alpha) \rightsquigarrow \mathcal{N}(0, \alpha(1-\alpha) D_1^{-1} D_0 D_1^{-1}).$$

The detailed proof of this result is given in Koenker, Chesher, et al. (2005, Chapter 4) and Van der Vaart (2000, Chapter 21). As for classical regression, Sherwood et al., 2017 proposed an adaptive Lasso penalisation for Quantile Regression. Let $\lambda \geq 0$ be the penalisation parameter, then the Lasso minimisation to obtain the regression parameter estimate is expressed as

$$\hat{\beta}_\alpha \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \rho_\alpha(Y_i - \boldsymbol{X}_i' \beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

Analogously Q. Li et al., 2010 introduced the Elastic-Net penalty in a Bayesian framework: let $\lambda_1, \lambda_2 \geq 0$, then

$$\hat{\beta}_\alpha \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \rho_\alpha(Y_i - \boldsymbol{X}_i' \beta) + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}.$$

Another Bayesian approach has been developed by Yu et al., 2001, and a Support Vector Machine algorithm by Crambes et al., 2013. We leave these penalisation methods and focus on Random Forest and Gradient Boosting methods in the following sections.

2.2.2 Quantile Regression with Random Forests

The construction of Quantile Regression Forests is introduced in Meinshausen, 2006b and is an extension of the Random Forests algorithm (Breiman, 2001). In comparison to linear Quantile Regression, this approach is a non-parametric method to estimate the conditional quantiles. Quantile Regression Forests is an ensemble method that aggregates a collection of decision trees. This aggregation differs from the classical approach of Random Forests which aim to estimate the conditional expectation $\mathbb{E}[Y|\boldsymbol{X}]$ via a weighted mean of the responses $Y_i, i = 1, \ldots, n$. Indeed, Quantile Regression Forests aim to estimate the full conditional distribution using this same weighted mean of the Y_i, and hence the conditional quantiles $q_\alpha(Y|\boldsymbol{X})$ follow as a corollary.

The mechanism of Quantile Regression Forests is borrowed from the Random Forests algorithm (Breiman, 2001). A Quantile Regression Forest is composed of several trees where each decision tree is trained on $\mathcal{D}_n^*$ which is re-sampled uniformly (with or without replacement) from the observed data $\mathcal{D}_n$. Each tree is a sequence of decisions which determines a partition of the re-sampled data $\mathcal{D}_n^*$. Let denote this partition $\mathcal{C}_1^b \cup \cdots \cup \mathcal{C}_M^b$ for the b^{th} tree. Each decision is a binary split that takes places in the middle of two consecutive data points in order to avoid possible ties. The splits are made recursively by maximising a CART-decision rule. For more details concerning the CART-decision rule and variable selection, see Biau & Scornet, 2016. The splitting stops when the current cell contains fewer points than a threshold called the **nodesize** $\in \{1, \ldots, a_n\}$ with a_n the number of re-sampled data points in each tree. Each partition class $\mathcal{C}_\ell^b$ is also called a leaf of the tree, also denoted ℓ^b for the ℓ^{th} leaf of the b^{th} tree with $b = 1, \ldots, B$. Every leaf is a partition class that can also be interpreted as a neighbourhood of a subset of the observations.

For a classical Random Forests Regression, for tree b, for a new data point $\boldsymbol{x}$, a set of weights $w_i(\boldsymbol{x}, b)$ are defined by

$$w_i(\boldsymbol{x}, b) = \frac{\mathbf{1}_{\{\boldsymbol{X}_i \in \ell^b(\boldsymbol{x})\}}}{|\ell^b(\boldsymbol{x})|}. \tag{2.4}$$

For the forest composed of B trees, the corresponding weights are obtained by summing over all individual trees

$$w_i(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^{B} w_i(\boldsymbol{x}, b), \tag{2.5}$$

with $\sum_{i=1}^n w_i(\boldsymbol{x}) = 1$. Then the prediction by the forest (i.e. an estimate of the conditional expectation $\mathbb{E}[Y|\boldsymbol{X}]$) is the weighted sum over all observations of $\mathcal{D}_n^*$. Let denote $\hat{Y}(\boldsymbol{x})$ the prediction of the new data point $\boldsymbol{x}$ and is obtained by

$$\hat{Y}(\boldsymbol{x}) = \sum_{i=1}^{n} w_i(\boldsymbol{x}) Y_i.$$

Figure 2.4 is an illustration of the two partitions $\mathcal{C}_1^1 \cup \cdots \cup \mathcal{C}_8^1$ and $\mathcal{C}_1^2 \cup \cdots \cup \mathcal{C}_8^2$ made from two decision trees trained on a subset $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{19}\}$, and where the remaining observed data $\{\boldsymbol{X}_{20}, \ldots, \boldsymbol{X}_{23}\}$ are excluded from the construction of these trees. For simplicity we consider the subset to be 2-dimensional and denote by $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}$ the two variables used for the splitting. In the first tree on the left of Figure 2.4, the observations $\boldsymbol{X}_4, \boldsymbol{X}_{10}, \boldsymbol{X}_{12}$ share the same neighbourhood $\mathcal{C}_6^1$. The new point $\boldsymbol{x}$ falls within this neighbourhood. According to Equation (2.4), with $b = 1$, the weights are $w_i(\boldsymbol{x}, 1) = \frac{1}{3}$ for $i = 4, 10, 12$ and 0 for observations outside the neighbourhood $\mathcal{C}_6^1$. In the same way, for the second decision tree on the right of Figure 2.4, the new observation $\boldsymbol{x}$ falls within the neighbourhood $\mathcal{C}_7^2$ which contains $\boldsymbol{X}_4$ and $\boldsymbol{X}_{18}$. Then the weights are $w_i(\boldsymbol{x}, 2) = \frac{1}{2}$ for $i = 4, 18$ and 0 for observations outside the neighbourhood $\mathcal{C}_7^2$. Consider a forest composed by these two decision trees, then the weights at $\boldsymbol{x}$ according Equation (2.5) are $w_4(\boldsymbol{x}) = \frac{5}{12}$, $w_{10}(\boldsymbol{x}) = \frac{1}{6}$, $w_{12}(\boldsymbol{x}) = \frac{1}{6}$ and $w_{18}(\boldsymbol{x}) = \frac{1}{4}$. Then the final estimate of the conditional expectation at $\boldsymbol{x}$ by this forest is $\hat{Y}(\boldsymbol{x}) = \frac{5}{12}Y_4 + \frac{1}{6}Y_{10} + \frac{1}{6}Y_{12} + \frac{1}{4}Y_{18}$.

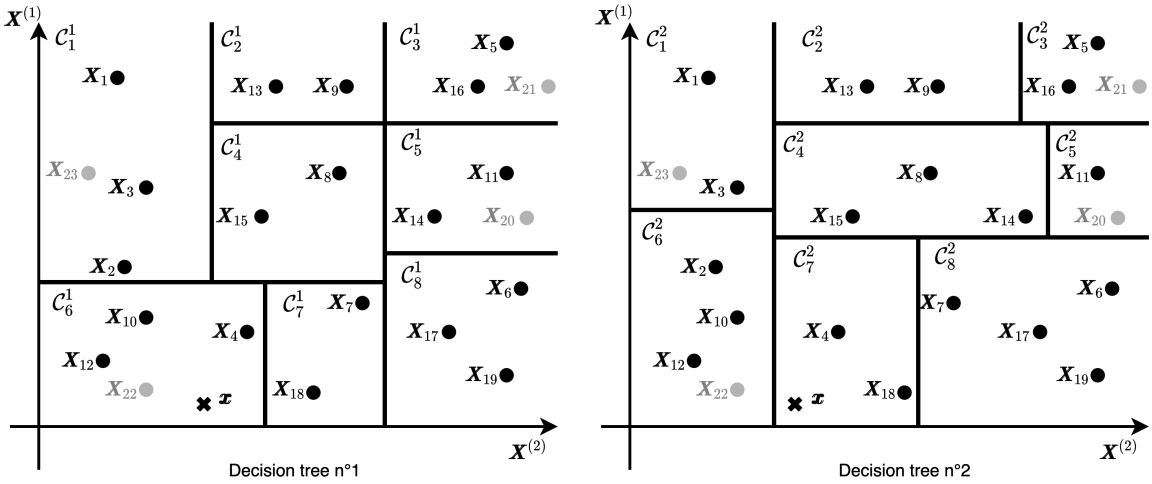


Figure 2.4 – Classical Random Forest with two decision trees trained on the same 2-dimensional re-sampled subset $\mathcal{D}_{19}^* = \{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_{19}\}$ (black circles). The remaining observed data $\{\boldsymbol{X}_{20}, \ldots, \boldsymbol{X}_{23}\}$ (gray circles) are excluded from their construction. The resulting partition classes are $\mathcal{C}_1^1 \cup \cdots \cup \mathcal{C}_8^1$ and $\mathcal{C}_1^2 \cup \cdots \cup \mathcal{C}_8^2$. A new data point is $\boldsymbol{x}$ (black cross).

In the context of Quantile Regression Forests, we must estimate the full conditional distribution $F_{Y|\boldsymbol{X}}$ and then the conditional quantile. Through the definition of the conditional distribution $Y|\boldsymbol{X}$ we obtain

$$F_{Y|\boldsymbol{X}}(y) = \mathbb{P}(Y \leq y|\boldsymbol{X}) = \mathbb{E}[\mathbf{1}_{\{Y \leq y\}}|\boldsymbol{X}].$$

Since this is a conditional expectation of an indicator function, the connection with classical Random Forests is more apparent. The estimation of $F_{Y|\boldsymbol{X}}$ given $\boldsymbol{x}$

is obtained from the weighted mean over the observations of $\mathbf{1}_{\{Y \leq y\}}$ instead of Y, that is

$$\hat{F}_{Y|\boldsymbol{x}}(y) = \sum_{i=1}^{n} w_i(\boldsymbol{x}) \mathbf{1}_{\{Y_i \leq y\}}. \tag{2.6}$$

The last step is to plug this estimator of the conditional distribution in Equation (2.3) to obtain the estimator of the conditional α-quantile as

$$\hat{q}_\alpha(Y|\boldsymbol{x}) = \inf_{t \in \mathbb{R}} \{\hat{F}_{Y|\boldsymbol{x}}(y) \geq \alpha\}. \tag{2.7}$$

Continuing with the same example in Figure 2.4 for a classical Random Forest, the estimate of the conditional distribution at the new data point $\boldsymbol{x}$ is $\hat{F}_{Y|\boldsymbol{x}}(y) = \frac{5}{12}\mathbf{1}_{\{Y_4 \leq y\}} + \frac{1}{6}\mathbf{1}_{\{Y_{10} \leq y\}} + \frac{1}{6}\mathbf{1}_{\{Y_{12} \leq y\}} + \frac{1}{4}\mathbf{1}_{\{Y_{18} \leq y\}})$, since the weights remain the same between the classical Random Forest and the Quantile Regression Forest. Moreover, under this approach, the prediction intervals for the new data point $\boldsymbol{x}$ are straightforward to compute. Denote $\alpha_{\min}$ for the lower level and $\alpha_{\max}$ the upper level, then we have directly

$$\begin{aligned}
\hat{q}_{\alpha_{\min}}(Y|\boldsymbol{x}) &= \inf_{t \in \mathbb{R}} \{\hat{F}_{Y|\boldsymbol{x}}(y) \geq \alpha_{\min}\} \\
\hat{q}_{\alpha_{\max}}(Y|\boldsymbol{x}) &= \inf_{t \in \mathbb{R}} \{\hat{F}_{Y|\boldsymbol{x}}(y) \geq \alpha_{\max}\}.
\end{aligned}$$

The prediction interval $\hat{\mathrm{I}}(\boldsymbol{x}) = [\hat{q}_{\alpha_{\min}}(Y|\boldsymbol{x}), \hat{q}_{\alpha_{\max}}(Y|\boldsymbol{x})]$ is well-defined. Indeed, since the distribution $\hat{F}_{Y|\boldsymbol{X}}$ is a continuous piecewise and increasing function, it is clear that $\hat{q}_{\alpha_{\min}}(Y|\boldsymbol{x}) \leq \hat{q}_{\alpha_{\max}}(Y|\boldsymbol{x})$ whenever $\alpha_{\min} \leq \alpha_{\max}$.

2.2.3 Quantile Regression with Generalized Random Forests

Generalized Random Forests, introduced in Athey et al., 2019, attempt to generalise Random Forests to several tasks: a more sophisticated non-parametric Quantile Regression, a conditional average partial effect estimation and a heterogeneous treatment effect estimation via instrumental variables. The general aim of Generalized Random Forests is to estimate the parameter function $\theta(\cdot)$ by solving a local moment equation of the form

$$\mathbb{E}[\psi_{\theta(\boldsymbol{X}_i),\nu(\boldsymbol{X}_i)}(Y_i)|\boldsymbol{X}_i = \boldsymbol{x}] = 0, \tag{2.8}$$

where $\boldsymbol{x}$ is a test point and $\psi_{\theta,\nu}$ is the scoring function which is indexed by θ and ν (an optional nuisance parameter). Since we are focused on generalising the form of the non-parametric Quantile Regression, the optional nuisance parameter ν is not required, and in the sequel, we refer to

$$\mathbb{E}[\psi_{\theta(\boldsymbol{X}_i)}(Y_i)|\boldsymbol{X}_i = \boldsymbol{x}] = 0$$

as the local estimation problem. The required scoring function for a non-parametric α-Quantile Regression is $\psi_{\theta(\boldsymbol{X}_i)}(Y_i) = \alpha - \mathbf{1}_{\{Y_i \leq \theta(\boldsymbol{X}_i)\}}$. We demonstrate in Appendix 2.5.3 that the scoring function $\psi_{\theta(\cdot)}(\cdot)$ is indeed well adapted for an α-Quantile Regression. This demonstration is based on the fact that the scoring function $\psi_{\theta(\cdot)}(\cdot)$ is intimately related to the Pinball loss $\rho_\alpha(\cdot)$. So the local estimation problem becomes

$$\mathbb{E}[\alpha - \mathbf{1}_{\{Y_i \leq \theta(\boldsymbol{X}_i)\}} | \boldsymbol{X}_i = \boldsymbol{x}] = 0. \tag{2.9}$$

Figure 2.5 illustrates the curves for the scoring function $\psi_0(y)$ for $\alpha = 0.25, 0.5, 0.75, 0.95$.

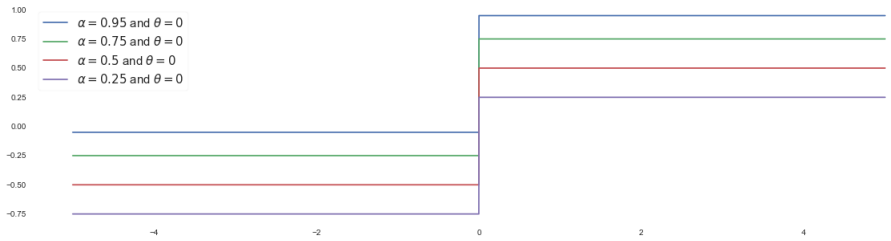


Figure 2.5 – Scoring function $\psi_0(y)$ for different values of α. Blue: $\alpha = 0.95$, green $\alpha = 0.75$, orange $\alpha = 0.5$, purple $\alpha = 0.25$.

In practice, the objective of a Generalized Random Forest is to solve an empirical version of the local estimation problem defined by Equation (2.9). Similarly to Quantile Regression Forests, we use the weights w_i obtained according to Equation (2.5) for the final aggregation of the forest. Thus the estimate of $\theta(\cdot)$ is the solution of the empirical local estimation problem

$$\hat{\theta}(\boldsymbol{x}) \in \underset{\theta(\cdot)}{\operatorname{argmin}} \left\| \sum_{i=1}^{n} w_i(\boldsymbol{x}) \psi_{\theta(\boldsymbol{x})}(Y_i) \right\|_2, \tag{2.10}$$

with $\|\cdot\|_2$ the Euclidean norm and the argmin is taken over the set of all measurable real-valued functions. Recall that these weights are a measure of the proximity between the re-sampled training data $\mathcal{D}_n^*$ and the new data point $\boldsymbol{x}$. It is important to note here that the trees where these weights are computed for Generalized Random Forests are not built following the same process as Quantile Regression Forests. Indeed, a new tree-building procedure is detailed below, and the resulting trees are aggregated following Equation (2.10).

Generalized Random Forests offer a more convenient way to estimate quantiles based on the partitioning of the trees of a forest. A new splitting rule is proposed which is adapted to Quantile Regression, rather than the usual variance-based splitting rule, to minimise the heterogeneity within each child

node, by maximising the heterogeneity between each child node. Each split starts at the parent node, denoted P, given the available data in this node $\mathcal{D}_P = \{(\boldsymbol{X}_i, Y_i) : i = 1, \ldots, n_P\}$ with n_P the number of available data points. We define $\hat{\theta}_P$ the solution of the local estimation problem

$$\hat{\theta}_P \in \underset{\theta(\cdot)}{\operatorname{argmin}} \left\| \sum_{\{\boldsymbol{X}_i \in \mathcal{D}_P\}} \psi_{\theta(\boldsymbol{x})}(Y_i) \right\|_2 . \tag{2.11}$$

Since that the splits are made in order to improve the quality of the estimate of $\theta(\cdot)$, then this means that the splits should take into account the regression objective i.e. the α-quantile of Y. So we split the parent node P into two child nodes C_1, C_2 which minimises the following error

$$\operatorname{err}(C_1, C_2) = \sum_{j=1}^{2} \mathbb{P}[\boldsymbol{X} \in C_j | \boldsymbol{X} \in P] \mathbb{E}[(\hat{\theta}_{C_j} - \theta(\boldsymbol{X}))^2 | \boldsymbol{X} \in C_j],$$

where the $\hat{\theta}_{C_j}$ are the fit over the child nodes C_j following Equation (2.11). The direct minimisation of this error is not possible since $\theta(\boldsymbol{x})$ is unknown. The alternative is to maximise the heterogeneity between the child nodes C_1 and C_2. The appropriate heterogeneity measure is the following delta criterion

$$\Delta(C_1, C_2) = \frac{n_{C_1} n_{C_2}}{n_P^2} (\hat{\theta}_{C_1} - \hat{\theta}_{C_2})^2. \tag{2.12}$$

Athey et al. (2019, Section 3, Proposition 1) asserts that maximising $\Delta(C_1, C_2)$ is equivalent to minimising $\operatorname{err}(C_1, C_2)$.

This duality is illustrated in Figure 2.6, where we have two examples of splits on the parent nodes P and $P^{'}$ containing the subset $\mathcal{D}_P = \mathcal{D}_{P'} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_9\}$ (for simplicity the parent nodes are the same). The subset represented by the blue circles $\{\boldsymbol{X}_2, \boldsymbol{X}_3, \boldsymbol{X}_6, \boldsymbol{X}_8\}$ should ideally be found after the split in a different child node to the black circles to achieve maximum heterogeneity between the child nodes. The split on the left, made on the parent node P, has a better separation than the split represented on the right on the parent node $P^{'}$. This leads to $\operatorname{err}(C_1, C_2) < \operatorname{err}(C_1^{'}, C_2^{'})$ and also its dual problem i.e. $\Delta(C_1, C_2) > \Delta(C_1^{'}, C_2^{'})$.

Whilst for Quantile Regression Forests, the heterogeneity between C_1 and C_2 is generally measured as the difference in the group variances, for the Generalized Random Forests this is measured by Equation (2.12). However, maximising this delta criterion $\Delta(C_1, C_2)$ by brute force requires numerating all possible splits on parent node data $\mathcal{D}_P$. This is computationally infeasible since it also requires estimating $\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$ according Equation (2.11) for each split. A computationally feasible alternative is based on approximating this delta criterion $\tilde{\Delta}(C_1, C_2)$, which in turn is based on approximations $\tilde{\theta}_{C_1}$ and $\tilde{\theta}_{C_2}$ of θ

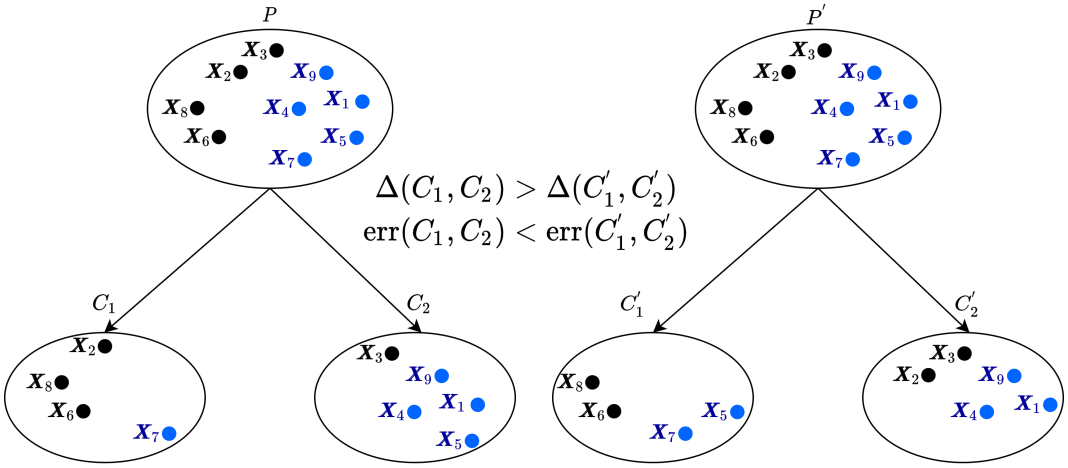


Figure 2.6 – Illustration of two splits on parents nodes P and P' applied in the available data $\mathcal{D}_P = \mathcal{D}_{P'} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_9\}$. The blue circles should ideally be separated in one child node while the black circles in the other child node to achieve maximum heterogeneity between the child nodes.

in the child nodes. These last two quantities are the gradient approximations of $\hat{\theta}_{C_1}$ and $\hat{\theta}_{C_2}$ obtained by

$$\tilde{\theta}_{C_j} = \hat{\theta}_P - \frac{1}{|\{i : \boldsymbol{X}_i \in C_j\}|} \sum_{\{i:\boldsymbol{X}_i \in C_j\}} A_P^{-1} \psi_{\hat{\theta}_P}(Y_i),$$

where A_P is a consistent estimate of the expectation of the gradient of ψ_θ such that

$$\mathbb{E}[A_P] = \nabla \mathbb{E}[\psi_{\theta(\boldsymbol{X}_i)}(Y_i) | \boldsymbol{X}_i \in P] = \nabla \mathbb{E}[\alpha - \mathbf{1}_{\{Y_i \leq \theta(\boldsymbol{X}_i)\}} | \boldsymbol{X}_i \in P].$$

The first (labelling) step consists of computing $\hat{\theta}_P$ and the matrix A_P^{-1} on the available data of the parent node $\mathcal{D}_P$ following Equation (2.11), and from which we obtain $A_P^{-1}\psi_{\hat{\theta}_P(\boldsymbol{X}_i)} = \mathbf{1}_{\{Y_i > \hat{\theta}_{P(\boldsymbol{X}_i)}\}}$. The second step is a standard CART variance-based split which is applied to $A_P^{-1}\psi_{\hat{\theta}_P(\boldsymbol{X}_i)}$. By construction, this split maximises

$$\tilde{\Delta}(C_1, C_2) = \sum_{j=1}^{2} \frac{1}{|\{i : \boldsymbol{X}_i \in C_j\}|} \left[\sum_{\{i:\boldsymbol{X}_i \in C_j\}} A_P^{-1} \psi_{\hat{\theta}_P(\boldsymbol{X}_i)} \right]^2.$$

Athey et al. (2019, Proposition 2, p. 10) demonstrate that $\tilde{\Delta}(C_1, C_2)$ and $\Delta(C_1, C_2)$ are approximately equivalent under suitable conditions.

So we are able to grow B individual trees according to maximisation of the approximate child heterogeneity outlined above. The final forest can now be built using the weights $w_i(\boldsymbol{x})$ that are computed as for classical Random

Forest Regression, by applying Equations (2.4) and (2.5). The final estimate of the conditional α-quantile by a Generalized Random Forest at at new data point $\boldsymbol{x}$ is

$$\hat{q}_\alpha(Y|\boldsymbol{x}) = \sum_{i=1}^{n} w_i(\boldsymbol{x})\psi_{\hat{\theta}(\boldsymbol{x})}(Y_i), \tag{2.13}$$

with $\psi_{\hat{\theta}(\boldsymbol{x})}(Y_i) = \alpha - \mathbf{1}_{\{Y_i \leq \hat{\theta}(\boldsymbol{X}_i)\}}$.

2.2.4 Quantile Regression with Gradient Boosting

The general aim of any boosting algorithm is to form an ensemble learner which is a linear combination of boosted weak learners. "Boosting" in this sense is to improve the accuracy of a weak learner such as a shallow decision tree. The boosted learner is built after a series of M iterations of an initial weak learner where it is improved at each iteration. Gradient Boosting is thus named because each iteration attempts to improve the learning process as guided by the gradient direction. It is a popular algorithm with several variants, all of which can be incorporated into Quantile Regression.

Gradient Boosting is based on the work originally conducted on Adaboost (Schapire, 1990; Freund, 1995; Freund & Schapire, 1997; Schapire, 1996). Since Breiman in 1997 (Breiman, 1997) made the fundamental observation that this procedure is a gradient-descent-type algorithm, the works of J. Friedman et al., 2000; J. H. Friedman, 2001; J. H. Friedman, 2002 formalised and baptised it with its current name of Gradient Boosting. These were refined by Bühlmann et al., 2007, and a new version named Extreme Gradient Boosting (XGBoost) was developed in Chen et al., 2016. More recent works have focused on the computational efficiency, where Biau, Cadre & Rouvière, 2019 proposed Accelerated Gradient Boosting (AGB), which in turn is based on Nesterov's accelerated gradient descent (Nesterov, 1983). These computational efficiencies are not limited to AGB, and can also be applied to XGBoost. The incorporation of Gradient Boosting into Quantile Regression relies on the approach of Zheng, 2012 who replaced the traditional squared cost function, as for a classical regression, with the Pinball function $\rho_\alpha(\cdot)$. Thus the weak learners are boosted according the gradient of the Pinball function. Since Biau & Cadre (2020, Section 3) proved the convergence and consistency of Gradient Boosting algorithms for any well-defined cost function, then Quantile Regression with Gradient Boosting is guaranteed to inherit these desirable convergence properties.

Maintaining the previous notation, where $\mathcal{D}_n = \{(\boldsymbol{X}_i, Y_i) : i = 1, \ldots, n\}$ is the sample of observed data and f is the target function that we wish to approximate in the minimisation problem described in Lemma 2, the Gradient Boosting algorithm can be summarised in four steps. Firstly, at iteration m, for $m \in 1, \ldots, M$, the steepest-descent step consists of computing the so-called "pseudo residuals" $\boldsymbol{r}^{[m]} \in \mathbb{R}^p$. The components of $\boldsymbol{r}^{[m]}$ are the negative gradient

of the Pinball function of the value of f obtained at iteration $m-1$, i.e.

$$\begin{aligned} r_i^{[m]} &= -\frac{\partial \rho_\alpha(y_i - f(\boldsymbol{X}_i))}{\partial f(\boldsymbol{X}_i)}\bigg|_{f=f^{[m-1]}} \\ &= [\alpha \mathbf{1}_{\{y_i \geq f^{[m-1]}(\boldsymbol{X}_i)} + (\alpha - 1)\mathbf{1}_{\{Y_i < f^{[m-1]}(\boldsymbol{X}_i)\}}. \end{aligned}$$

This steepest-descent is a greedy strategy since $\boldsymbol{r}^{[m]}$ is the local direction in $\mathbb{R}^p$ for which the Pinball loss is the most rapidly decreasing at $f^{[m-1]}$. Secondly, a regression tree is fitted to the $r_i^{[m]}$. The terminal leaves of this regression tree are denoted $\ell_m^j, j = 1, \ldots, j_m$, where j_m is the number of leaves. Thirdly, the step size factor (or step length) η_m is obtained as the solution to

$$\eta_m = \underset{\eta \in \mathbb{R}}{\operatorname{argmin}}\, \rho_\alpha(Y - (f^{[m-1]}(\boldsymbol{X}) - \eta)).$$

Fourthly, $f^{[m]}$ is updated following the rule $f^{[m]} = f^{[m-1]} + \eta_m \boldsymbol{r}^{[m]}$.

That is, in iteration 1, we obtain $f^{[1]}$, the first boosted approximation of the function f, and in iteration 2, the boosted approximation $f^{[2]}$ is obtained from $\boldsymbol{r}_1$, etc. If these four steps are iterated M times, then the resulting conditional quantile via this Gradient Boosting has an additive form, since we obtain

$$\hat{q}_\alpha(Y|\boldsymbol{X}) = f^{[M]}(\boldsymbol{X}) = f^{[0]} + \sum_{m=1}^{M} \eta_m \boldsymbol{r}^{[m]}.$$

These steps are summarised in Algorithm 2.1.

2.2.5 Accuracy measures for Quantile Regression models

The output of a Quantile Regression is the prediction of an α-quantile. So the measure of the accuracy of a regression model is the based on the comparison of the empirical waiting times with the predicted α-quantile. The first metric we consider is the Pinball loss defined in Equation (2.2). For a Quantile Regression, the Pinball loss should be a small value for an accurate model. Let $\hat{q}(Y)$ be the prediction for the α-quantile of an observation Y. The corresponding Pinball loss is

$$\rho_\alpha(Y - \hat{q}(Y)) = (Y - \hat{q}(Y))(\alpha - \mathbf{1}_{\{Y \leq \hat{q}(Y)\}}). \tag{2.14}$$

The interpretation of this Pinball loss is somewhat non-intuitive because it consists of a quantile-weighted difference between the predicted quantile $\hat{q}(Y)$ and the observed value Y. Moreover, this weighting is different depending on whether the predicted quantile is above or below the observed value. For instance, if we perform a regression for the 75% quantile then if the predicted quantile is above the observed value, the weighting assigned by the Pinball loss is 0.25 and conversely, 0.75 if it is below. In other words, two models

Algorithm 2.1: Quantile Regression with Gradient Boosting

Input : Subset of observed data $\mathcal{D}_n^* = \{(\boldsymbol{X}_i, Y_i) : 1 \leq i \leq n\}$, desired quantile level α, number of iterations M

Initialisation : Set $f^{[0]}(\cdot) = 0$ or α-quantile of $\mathcal{D}_n^*$

for m=1 **to** M **do**

(1) Compute pseudo-residuals

$$\begin{aligned} r_i^{[m]} &= -\frac{\partial \rho_\alpha(y_i - f(\boldsymbol{X}_i))}{\partial f(\boldsymbol{X}_i)}\bigg|_{f=f^{[m-1]}} \\ &= [\alpha \mathbf{1}_{\{Y_i \geq f^{[m-1]}(\boldsymbol{X}_i)} + (\alpha - 1)\mathbf{1}_{\{Y_i < f^{[m-1]}(\boldsymbol{X}_i)\}} \end{aligned}$$

(2) Fit regression tree to pseudo-residuals $r_i^{[m]}$, yielding terminal leaves ℓ_m^j with $j = 1, \ldots, j_m$

(3) **for** $j = 1$ **to** j_m **do**

compute step size factor

$$\eta_{j,m} = \underset{\eta \in \mathbb{R}}{\operatorname{argmin}} \left\{ \sum_{\boldsymbol{X}_i \in \ell_m^j} \rho_\alpha(Y_i - (f^{[m-1]}(\boldsymbol{X}_i) - \eta)) \right\}$$

end

(4) Update $f^{[m]}(\boldsymbol{X}_i) = f^{[m-1]}(\boldsymbol{X}_i) + \sum_{j=1}^{j_m} \eta_{j,m} \mathbf{1}_{\{\boldsymbol{X}_i \in \ell_m^j\}}$

end

Output : Estimated α-quantile function $\hat{q}_\alpha(Y|\boldsymbol{X}) = f^{[M]}(\boldsymbol{X})$.

with different mean absolute errors may have the same Pinball loss. This is illustrated in subplots 1 and 4 in the toy example in Figure 2.7 where both of these models have the same Pinball loss. Whilst model 1 has larger distances between predicted and observed quantiles than model 4, the weight of the pinball losses in the model 4 are higher than those in model 1 because the former consistently underestimates.

An alternative accuracy measure for Quantile Regression is called the Ramp loss and is defined by

$$\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \mathbf{1}_{\{Y_i \leq \hat{q}(Y_i)\}}$$

which represents the empirical fraction of the quantile estimates which exceed the observed values Y_i in the test dataset of size n_{test}. This loss is introduced Biau & Patra, 2011 and used for sequential quantile prediction of time series. The Ramp loss for an α-Quantile Regression should be as close as possible to $1 - \alpha$ to be an accurate model. For the regressions for the 75% quantile illustrated

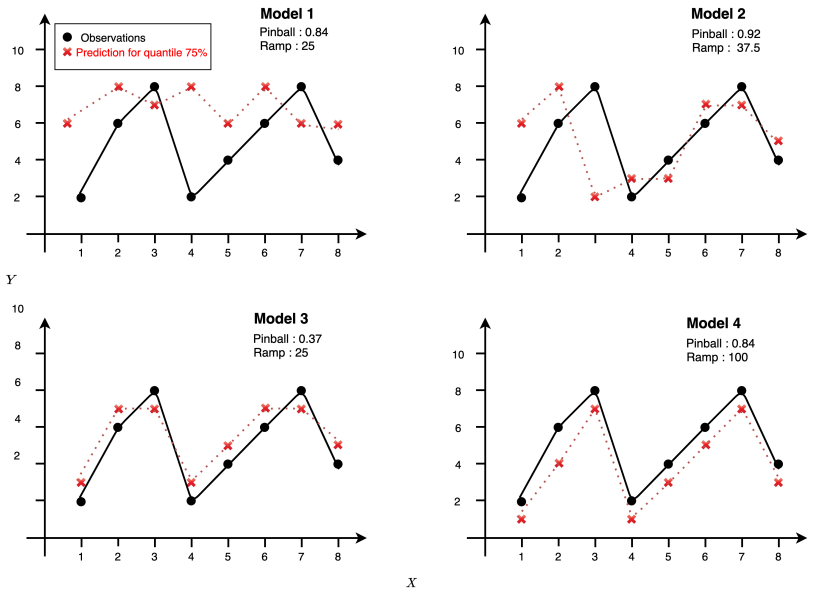


Figure 2.7 – Illustrative examples of loss metrics of Quantile Regressions on a test sample of size 8. The observed values are the black dots and the predicted 75% quantiles are the red crosses.

in models 1 and 3 of Figure 2.7, we have two predicted quantiles below their observed values. Given that $n_{\text{test}} = 8$, then the Ramp losses are $2/8 = 0.25$ in both cases. So based solely on the Ramp loss, models 1 and 3 have the same predictive performance. However a visual inspection reveals that this is not the interpretation that we wish to derive. In order to mitigate this, as well as the short-comings of the Pinball loss, we propose to examine both of these loss functions to inform our decisions. Consider again the example of the comparison of the models 1 and 4 in Figure 2.7. By considering the Ramp loss, we obtain directly that the better performing model is model 1. Likewise for the comparison between models 1 and 3. By additionally considering the Pinball loss, we can conclude that the model 3 is the most accurate model out of these four models. To compare the regression models, we use a k-fold cross validation procedure, which approximates the generalised error, i.e. the error that will be observed when the model is used in production. In summary this procedure consists in training the model on a systematic random sample of 90% of the observed data, and the predictions are carried out on the remaining 10% of the data (known as the test or prediction fold). These two steps are iterated for each 10% systematic sample as the test fold. For the kth test fold, the Pinball $\rho_{\alpha,k}$ and Ramp errors $R_{\alpha,k}$ are computed as the prediction errors. The generalised error is then obtained by averaging all the errors committed at each test fold. In addition to obtaining the generalised error we can also follow the evolution of the errors at each prediction fold and obtain information about the stability of

the models with respect to their predictive performance. Figure 2.8 illustrates this cross-validation procedure.

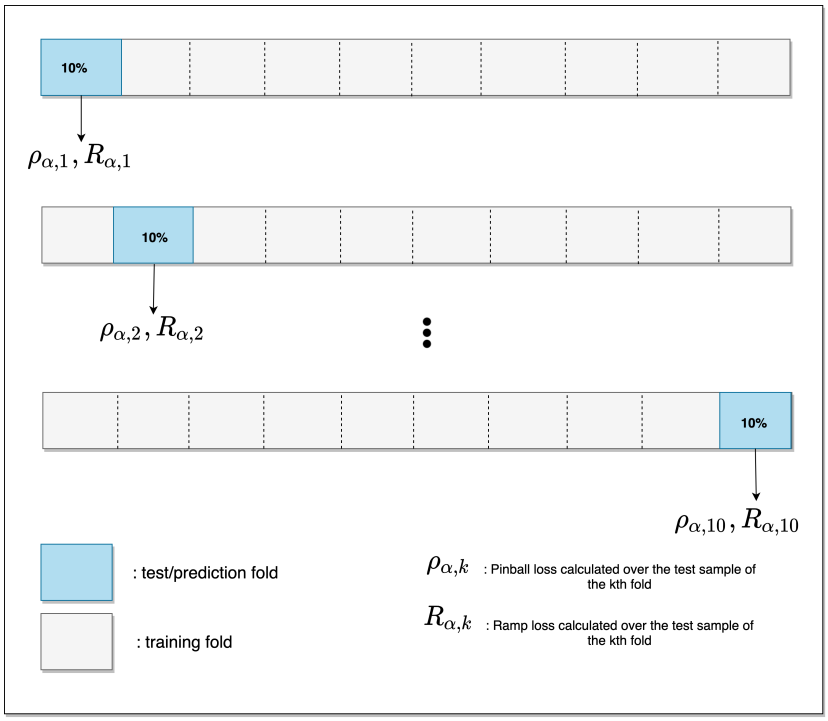


Figure 2.8 – The k-fold cross validation procedure to calculate the generalised errors, Pinball and Ramp, for the competing models.

2.3 Comparison of Regression models for carpooling request waiting times

The main dataset is drawn from approximately 700 carpooling trips carried out in the Vexin carpooling network from 2016-01-18 to 2018-10-02, whose map is shown in Figure 2.2. The first data collection date corresponds to the service launch date. This service was in a highly experimental phase during this period in at least two aspects. Firstly because there was scarce existing data about the carpooling in this area. Secondly because Ecov's real-time carpooling service was at the time the world's first of its type. Recall that the carpooling network is composed of physical meeting points where the passengers can make their carpooling requests. These physical meeting points are accompanied of electronic panels placed in the roadside. When a passenger makes a request at the ticket machine, it is displayed on the panels. Drivers passing by are thus informed and can potentially respond in real-time. In this experimental

phase, the carpooling service allowed the passenger choose freely the desired destination from any location within the service area. Whilst this freedom makes the service highly convenient for passengers, it renders the problem of waiting time prediction extremely difficult because the combinatorial explosion in the number of possible origin-destination pairs.

2.3.1 Response variable: Waiting times

After a successful carpooling trip, i.e. a passenger is picked up and then dropped off by a driver, then we can define the waiting time as the difference between the time that the passenger embarks in the driver's car and the time of the initial carpooling request. The departure time is collected by the central data server from a manual notification sent by the passenger via the mobile phone application, whereas the carpooling request time is automatically sent from the ticket machine. A histogram of the waiting times is plot in Figure 2.9.

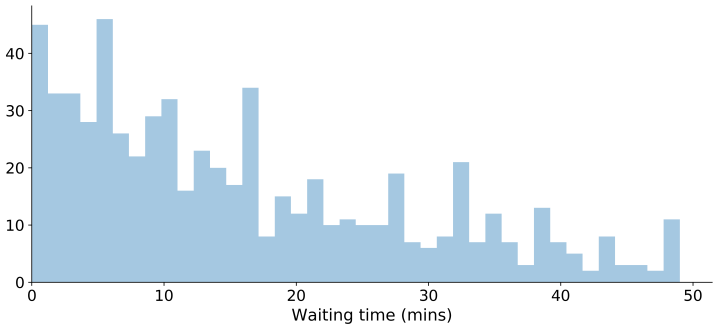


Figure 2.9 – Histogram of the waiting times in the Vexin carpooling network during the period 2016-01-18 to 2018-10-02.

These waiting times are skewed right in Figure 2.9 due the long right tail. Most of the waiting times are between 0 and 10 minutes, with a smaller peak around 15 minutes. The range is around 50 minutes which indicates a high variability in waiting times. For operational reasons, we consider that any values greater than 40 minutes to be outliers since they exceed $Q_3 + 1.5 \cdot \text{IQR}$ where Q_3 is the third quartile and IQR is the inter-quartile range. After removing these outlier values, the range reduces to 40 minutes, which is remains stubbornly wide for a daily transportation service. In the box plot of waiting times per day in Figure 2.10, we observe that weekdays and Saturdays are similar with a median around 11 minutes, with Fridays being slightly lower. In France, Wednesday is a common day for working from home since there are also no school classes on Wednesday afternoon, this decrease in available drivers does not appear to lead to an increase in waiting times for passengers compared to other weekdays.

Also, Saturday mornings have school classes. Sundays, on the contrary, have significantly longer waiting times. This can be explained by the fact that on Sunday there are far fewer drivers than on other day of the week.

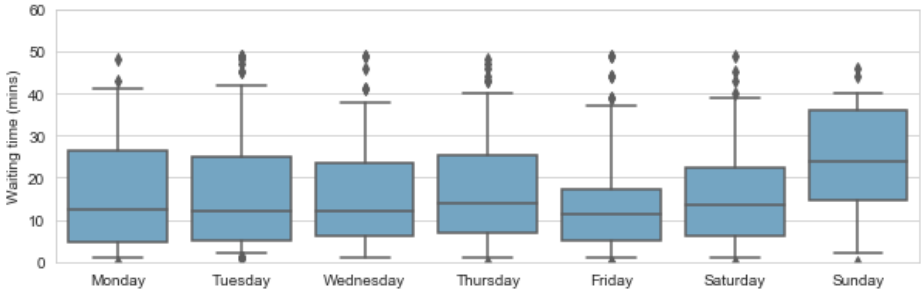


Figure 2.10 – Box plot of the waiting times per day of the week in the Vexin carpooling network during the period 2016-01-18 to 2018-10-02.

For comparison, the average waiting time for a bus in the same region is 12 minutes according to the Global Transit Usage Report 2016 compiled by Moovit.[2] The comparison with public bus waiting times is more pertinent than with than other on-demand carpooling services such as Uber, since Ecov's carpooling service more closely resembles the former than the latter.

2.3.2 Explanatory variables

The explanatory variables or covariates employed to predict the waiting times are divided into three categories: (i) the geographical information, (ii) the temporal information, and (iii) the passenger information associated with the carpooling request. This information is necessary to match a passenger to a driver within spatio-temporal constraints. At this early experimental phase of the carpooling service, no direct information from the drivers is collected (we will see in the following chapters the effect of the availability this fourth category of auxiliary information from the drivers). Since the passenger has the freedom to indicate any location in the service area as a carpooling destination, there are 252 separate named destinations observed from the approximately 700 trips. The origins are always one of the 17 physical carpooling meeting points, as shown in Figure 2.2. We reduce the number of possible origin-destination pairs by performing a spatial k-means clustering based on the latitude and longitude of the destinations. Figure 2.11 shows the map with these seven clusters which represent the seven possible destination neighbourhoods. This clustering reduces the number of observed origin-destination pairs from 252 to 66.

2. https://moovit.com/blog/50-million-global-transit-report/

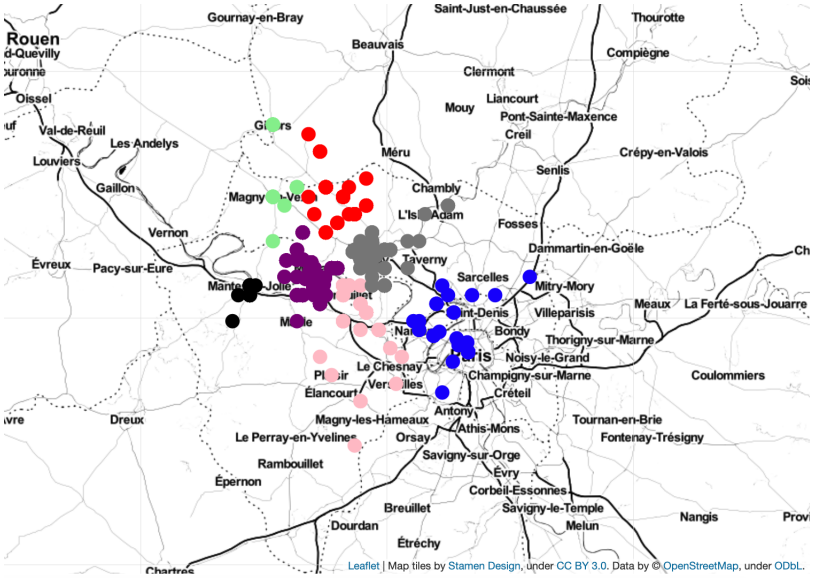


Figure 2.11 – Clustered destination neighbourhoods in the Vexin carpooling network.

The reduction in the number of origin-destination pairs increases the number of carpooling trips within each of these pairs. Figure 2.12 shows the number of observations for the five most frequented origin-destination pairs before (left) and after (right) clustering. Whilst these are not the same five origin-destination pairs, we are still able to observe that, after clustering, we have around twice as many carpooling trips per origin-destination pair which will assist in the robustifying the regression predictions. Since driver information is not collected directly, there is no observed information about the distance and the duration of the carpooling trip. Since these values may have an important influence as explanatory variables in the regression models, then we approximate them using route-finding APIs (Application Programming Interfaces) via the latitude and longitude of the origins and the destinations.

The temporal information is derived from the timestamps with the carpooling request. From this information we are able to derive new explanatory variables, namely the hour of the day and the day of the week, and subsequently, the total number of carpooling requests per hour and per day of the week. These aggregated values add information about the overall trends in the driver population into the regression models, as shown in Figure 2.13. From the left plot, we see that the distribution of carpooling requests within a day is a bi-modal distribution with peaks in the morning and evening peak hours. So we add an indicator variable for the morning (6h00 – 9h00) and evening peak hours (16h00 - 19h00). From the right plot, there are fewer carpooling requests on Sundays and this leads us to add an indicator variable for holiday days

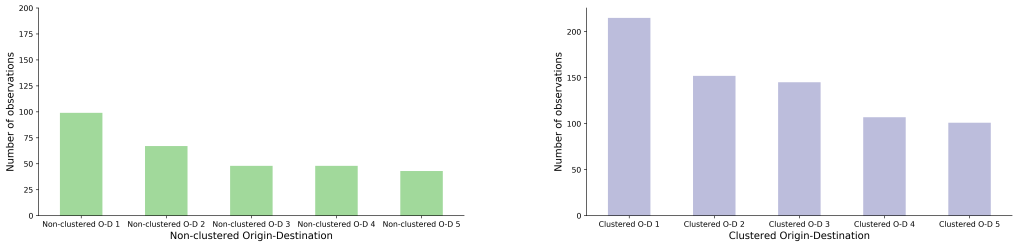


Figure 2.12 – Number of carpooling trips for the five most frequented origin-destination pairs in the Vexin carpooling network. (Left) Non-clustered origin-destinations. (Right) Clustered origin-destinations.

which tend to behave like Sundays.

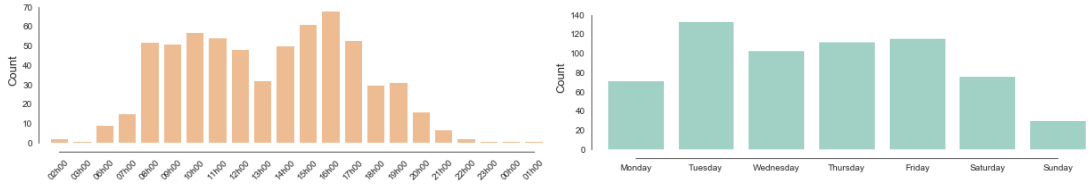


Figure 2.13 – Aggregated counts of the number of carpooling requests in the Vexin carpooling network for 2016-01-18 to 2018-10-02. (Right) Per day of week. (Left) Per hour.

The passenger information is collected by Ecov via a standard passenger registration form. As a passenger uses the service over time, we obtain more information about their carpooling habits, such as the duration of a passenger's registration and the number of carpooling requests made. These measure the familiarity of the passenger with Ecov's carpooling services: internal studies conducted by Ecov have shown that a passenger who is familiar with the Ecov system is likely to depart more quickly than a first-time passenger. Whilst it is also possible to analyse more of the passenger's personal data (gender, age, etc), in order to comply with the privacy concerns in force in the European Union, namely the General Data Protection Regulation (GDPR), we chose not to do so.

Table 2.1 is a summary of all the geographical, temporal and passenger exploratory variables in the regression models to predict waiting times.

Explanatory Variable	Description	Category
origin_id	Identifier of physical meeting point (origin)	Geographical
destination_cluster	Identifier of destination cluster	Geographical
origin_latitude	Latitude of origin	Geographical
origin_longitude	Longitude of origin	Geographical
destination_latitude	Latitude of destination	Geographical
destination_longitude	Longitude of destination	Geographical
travel_time	Duration of carpooling trip	Geographical
travel_distance	Distance of carpooling trip	Geographical
day_of_week	Day of week	Temporal
time_of_day	Time of day (continuous)	Temporal
is_peak_hour	Peak hour indicator	Temporal
is_vacation_period	Holiday indicator	Temporal
service_seniority	# days since registration	Passenger
number_of_requests	# carpooling requests since registration	Passenger
validated_requests	# completed carpooling trips since registration	Passenger

Table 2.1 – Summary table of the explanatory variables in the regression models. The first column is the name of the variable, the second is a description and the third is the category (geographical, temporal or passenger).

2.3.3 Quantile Regression models comparison

We provide a comparison of the performance of the four regression algorithms presented in the previous sections, namely the linear Quantile Regression (**QuantReg**), the Quantile Regression Forests (**QRF**), the Generalized Random Forests (**GRF**) and Quantile Regression with Gradient Boosting (**GBM**), on these carpooling trips made in the network. The quantiles chosen for the comparison are the lower quartile 25% and the upper quartile 75%. We seek to test the models on quantiles of the distribution which are different from the median, though we do not focus on extreme quantiles, e.g. 5% or 95%. These extreme quantiles are difficult to estimate and require large amounts of data for robust prediction, which we do not have for the Ecov carpooling trips. See De Haan et al., 2007 for a complete overview of the analysis of extreme quantiles. Table 2.2 shows the Pinball losses for the four competing models for all test folds of the 10-fold cross validation. The upper rows corresponds to the accuracy of the Quantile Regressions for the 75% quantile and the lower rows for the 25% quantile. Overall, the **GRF** model has uniformly the lowest Pinball losses (around 3–4) for all test folds, compared to the three other models (around 5–9)

for both quantiles. This is confirmed when considering the overall generalised pinball loss, with the performance gain of the **GRF** more prominent for the 75% quantile than with the 25% quantile. In addition, the variability of all loss functions over all test folds is low, indicating that the predictions of the waiting times for all the regressions models are stable.

$\alpha = 0.75$	kth test fold										All
Model	1	2	3	4	5	6	7	8	9	10	
QuantReg	8.17	7.97	7.43	12.05	8.70	10.46	9.84	8.20	10.07	8.98	9.19
GBM	7.17	8.44	9.11	7.36	7.79	8.26	8.38	9.37	9.29	9.36	8.45
QRF	8.29	6.25	7.54	8.80	8.93	9.145	7.10	9.17	9.45	8.75	8.34
GRF	4.38	4.32	4.62	4.70	4.53	4.02	5.40	4.54	4.16	3.87	**4.45**
$\alpha = 0.25$											
QuantReg	4.43	5.40	4.79	4.69	4.38	5.35	4.02	5.43	4.87	4.64	4.80
GBM	4.45	4.62	5.22	5.13	4.88	4.15	4.24	5.23	4.12	4.85	4.69
QRF	5.11	4.61	4.73	5.21	5.45	4.56	4.04	5.89	4.99	4.45	4.90
GRF	3.21	3.26	3.85	3.38	3.63	2.78	3.55	3.50	3.43	2.94	**3.35**

Table 2.2 – Pinball losses for all 10 test folds of the 10-fold cross validation for the 75% and 25% Quantile Regressions. The first column is the Regression model names, the middle columns are for the 10 test folds, and the last column is the overall generalised pinball loss with the lowest value in bold.

Table 2.3 shows the Ramp losses for the four competing models for all test folds of the 10-fold cross validation. The upper rows corresponds to the accuracy of the Quantile Regressions for the 75% quantile and the lower rows for the 25% quantile. In comparison to the Pinball losses, the Ramp losses have larger variability between each of the 10 test folds within the same Regression model. So there is no uniformly best model in this respect. If we examine the overall generalised loss, then all models seem to be centred towards the optimal value $1 - \alpha$, indicating that all models overall are predicting the target quantile accurately, even though there is a large variability for the individual test folds. The model with the loss closest to $1 - \alpha$ is **QuantReg** for $\alpha = 0.25$, and **GRF** for $\alpha = 0.75$.

Taking into account both the Pinball and Ramp losses, the **GRF** model seems to have most stable and accurate results of these considered Regression models. However this is only a relative comparison, since if we visually inspect the predicted quantiles of the waiting times for $\alpha = 0.25, 0.75$ in Figure 2.14, not even the **GRF** has a sufficient accuracy in these waiting time predictions to warrant an operational deployment.

$\alpha = 0.75$					kth test fold						All
Model	1	2	3	4	5	6	7	8	9	10	
QuantReg	29.23	35.38	35.94	12.50	23.44	23.44	12.50	37.50	21.87	26.56	**25.84**
GBM	30.77	23.08	29.69	32.81	35.94	23.44	23.44	26.56	23.44	18.75	26.79
QRF	32.31	38.46	29.69	26.56	29.69	26.56	29.69	29.69	25.00	28.12	29.58
GRF	23.33	21.67	36.67	25.00	31.67	13.33	23.33	21.67	21.67	20.00	23.83
$\alpha = 0.25$											
QuantReg	75.38	76.92	81.25	64.06	75.00	67.19	68.75	79.69	71.87	76.56	73.67
GBM	86.15	66.15	62.50	89.06	71.87	73.44	75.00	62.50	71.87	73.44	73.20
QRF	69.23	72.31	67.19	68.75	62.50	68.75	71.87	60.94	75.00	71.87	68.84
GRF	85.00	66.67	88.33	70.00	76.67	66.67	68.33	68.33	78.33	75.00	**74.33**

Table 2.3 – Ramp losses for all 10 test folds of the 10-fold cross validation for the 75% and 25% Quantile Regressions. The first column is the Regression model names, the middle columns are for the 10 test folds, and the last column is the overall generalised ramp loss with the closest value to $1 - \alpha$ in bold.

2.4 Conclusion

Quantile Regression is an interesting choice for predicting waiting times in a carpooling service, as it provides more detailed information than the mean waiting times provided by classical regression. The business objective was to harness this more detailed information to tailor the carpooling service to the customer behaviour and expectations. However, due to the complexity of the carpooling service as offered by Ecov during this early experimental phase, this business objective was not realised with this regression analysis, despite the strong mathematical framework underlying the quantile regression models. The operational complexity induced by allowing the passengers to select any arbitrary location in the service as a destination for a carpooling was not able to provide a reliable supply of carpooling drivers to their passengers. This creates a negative feedback where fewer passengers then continue to utilise the carpooling service. This ultimately leads to an insufficient number of completed carpooling trips (out of the total number of car trips made in the service area) required, as well as a lack of directly observed data of driver behaviour, for robust quantile regression analysis of the carpooling origin-destination pairs. From these lessons learnt in this early experimental phase, Ecov then pivoted their carpooling service to service a limited number of destinations, more similar to bus routes rather than Uber-like services, to aggregate sufficiently the passenger demand and driver supply to ensure a reliable matching between these two groups. Ecov also subsequently incorporated technology to collect the GPS traces of driver itineraries from their mobile phone application to better understand empirical driver behaviour. In the following chapters we

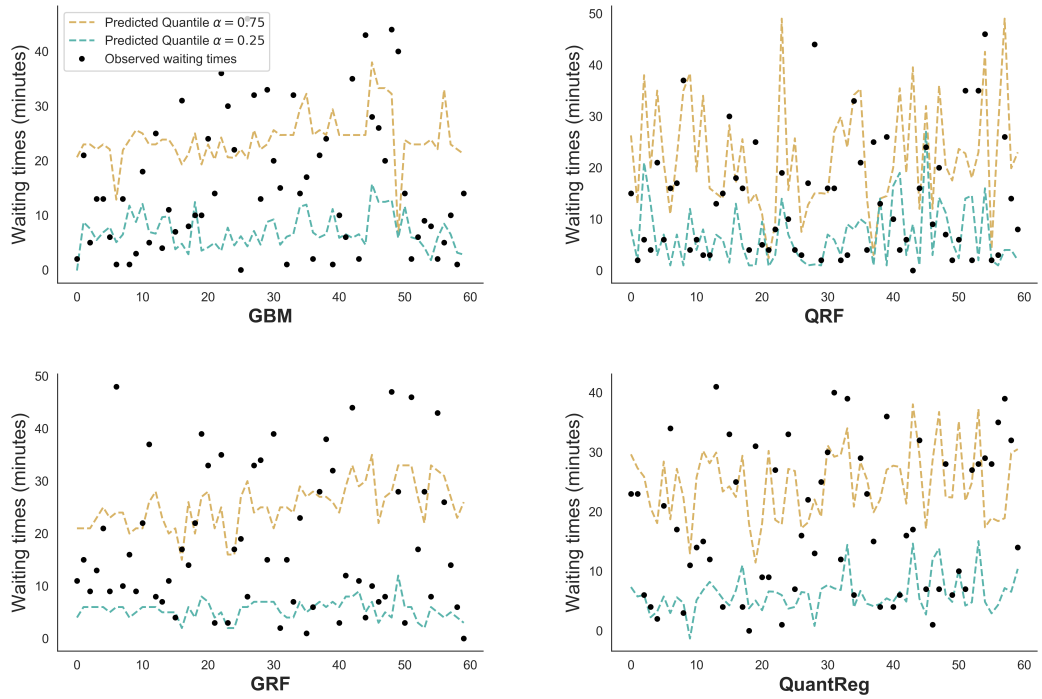


Figure 2.14 – Predicted quantiles from the Regression models for the waiting times in the Vexin carpooling network. The predictions for $\alpha = 0.25$ are in green, and for $\alpha = 0.75$ in brown, for a random test fold from the cross validation for the models **GBM** (top left), **QRF** (top right), **GRF** (bottom left) and **QuantReg** (bottom right). The black circles are the observed waiting times.

will analyse the data from these improved carpooling services and how they facilitate a better understanding of the carpooling potential in the service areas.

2.5 Appendix : Details of the supporting mathematical results

2.5.1 Quantiles as minimisers of the Pinball loss

Lemma 1 is proved by asserting that is equivalent to the following definition.

Definition 1. *Let Y be a real-valued integrable random variable whose distribution F_Y is strictly increasing. Then the quantile q_α satisfies*

i. $\forall \alpha \in]0;1[, F_Y(q_\alpha) \geq \alpha$

ii. if $q < q_\alpha$ *then* $F_Y(q) < \alpha$.

Proof of Definition 1. This proof is also given in Biau & Patra, 2011. We proceed by separating two cases, when $q \geq q_\alpha$ and $q < q_\alpha$.
For $q \geq q_\alpha$,

$$\begin{aligned}\mathbb{E}[\rho_\alpha(Y-q)] - \mathbb{E}[\rho_\alpha(Y-q_\alpha)] &= \mathbb{E}[(Y-q)(\alpha - \mathbf{1}_{[Y\leq q]})] - \mathbb{E}[(Y-q_\alpha)(\alpha - \mathbf{1}_{[Y\leq q_\alpha]})] \\ &= \mathbb{E}[(Y-q)(\alpha - (\mathbf{1}_{[Y\leq q_\alpha]} + \mathbf{1}_{[q_\alpha<Y\leq q]})) - (Y-q_\alpha)(\alpha - \mathbf{1}_{[Y\leq q_\alpha]})] \\ &= \mathbb{E}[(q_\alpha - q)(\alpha - \mathbf{1}_{[Y\leq q_\alpha]})] - \mathbb{E}[(Y-q)\mathbf{1}_{[q_\alpha<Y\leq q]}.\end{aligned}$$

Thus, the first term of right part

$$\begin{aligned}\mathbb{E}[(q_\alpha - q)(\alpha - \mathbf{1}_{[Y\leq q_\alpha]})] &= (q_\alpha - q)\mathbb{E}[\alpha - \mathbf{1}_{[Y\leq q_\alpha]}] \\ &= (q_\alpha - q)(\alpha - \mathbb{P}[Y \leq q_\alpha]) \\ &= (q_\alpha - q)(\alpha - F_Y(q_\alpha)) \\ &\geq 0.\end{aligned}$$

Note that by definition $q_\alpha - q \leq 0$ and by the Property 1 we have $F_Y(q_\alpha) \geq \alpha$. For the second term of the right part, indicator $\mathbf{1}_{[q_\alpha<Y\leq q]}$ implies that $Y - q \leq 0$ and we obtain that $-\mathbb{E}[(Y-q)\mathbf{1}_{[q_\alpha<Y\leq q]} \geq 0$.
For $q < q_\alpha$,

$$\begin{aligned}\mathbb{E}[\rho_\alpha(Y-q)] - \mathbb{E}[\rho_\alpha(Y-q_\alpha)] &= \mathbb{E}[(Y-q)(\alpha - \mathbf{1}_{[Y\leq q]})] - \mathbb{E}[(Y-q_\alpha)(\alpha - \mathbf{1}_{[Y\leq q_\alpha]})] \\ &= \mathbb{E}[(Y-q)(\alpha - \mathbf{1}_{[Y\leq q]}) - (Y-q_\alpha)(\alpha - (\mathbf{1}_{[Y\leq q]} + \mathbf{1}_{[q<Y\leq q_\alpha]}))] \\ &= \mathbb{E}[(q_\alpha - q)(\alpha - \mathbf{1}_{[Y\leq q}_\alpha)] - \mathbb{E}[(Y-q_\alpha)(\alpha - \mathbf{1}_{[q<Y\leq q_\alpha]})] \\ &= (q_\alpha - q)(\alpha - \mathbb{P}[Y \leq q]) - \mathbb{E}[(Y-q_\alpha)(\alpha - \mathbf{1}_{[q<Y\leq q_\alpha]})]\end{aligned}$$

The first term of right part $(q_\alpha - q) > 0$ is strictly positive by definition and $\alpha - \mathbb{P}[Y \leq q] = \alpha - F_Y(q) > 0$. We obtain Property 1 that $F_Y(q) < \alpha$. For the second term of the right part we have directly that $-\mathbb{E}[(Y-q_\alpha)(\alpha - \mathbf{1}_{[q<Y\leq q_\alpha]})] \geq 0$. Thus, we obtain the desired result, i.e. for all $q \in \mathbb{R}$, we have $\mathbb{E}[\rho_\alpha(Y-q)] - \mathbb{E}[\rho_\alpha(Y-q_\alpha)] \geq 0$. □

2.5.2 Conditional quantiles as minimisers of a Pinball loss

Definition 2. *Let Y be real-valued random variable whose conditional distribution $F_{Y|\boldsymbol{X}}$ is almost surely increasing. Then the conditional quantile $q_\alpha(Y|\boldsymbol{X})$ is a random variable that satisfies*

i. $q_\alpha(Y|\boldsymbol{X})$ *is an element of* $L^1(\boldsymbol{X})$,

ii. $\mathbb{P}(Y < q_\alpha(Y|\boldsymbol{X})|\boldsymbol{X}) \leq \alpha \leq \mathbb{P}(Y \leq q_\alpha(Y|\boldsymbol{X})|\boldsymbol{X})$ *almost surely.*

Definition 3. *Let $f : \mathbb{R} \to \mathbb{R}$ be a convex function and let $x_1, x_2 \in \mathbb{R}$. Then*

$$f(x_2) = f(x_1) + \int_{x_1}^{x_2} D_+ f(t)dt,$$

where $D_+ f$ is the right-hand derivative of f.

With these two definitions, we are able to state the proof of Lemma 2.

Proof Lemma 2. A more complete proof is given in Armerin, 2014. Let $x_0 \in \mathbb{R}$, the Pinball function is convex, hence the right-hand derivative at point $(x - x_0)$ is

$$D_+ \rho_\alpha(x - x_0) = \alpha - \mathbf{1}_{\{x \leq x_0\}}.$$

Then the Definition 3 is applied to Pinball loss function

$$\rho_\alpha(x_2 - x_0) = \rho_\alpha(x_1 - x_0) + \int_{x_1}^{x_2} D_+ \rho_\alpha(t - x_0)dt.$$

Let $x_1 = q(\boldsymbol{X}) \in L^1(\boldsymbol{X})$, the space of all integrable functions with respect to $\boldsymbol{X}$, and so by construction $x_2 = q_\alpha(Y|\boldsymbol{X}) \in L^1(\boldsymbol{X})$ for $x_0 = Y$. Then

$$\rho_\alpha(q_\alpha(Y|\boldsymbol{X}) - Y) = \rho_\alpha(q(\boldsymbol{X}) - Y) + \int_{q(\boldsymbol{X})}^{q_\alpha(Y|\boldsymbol{X})} [\alpha - \mathbf{1}_{\{t \leq Y\}}]dt.$$

It follows that

$$\mathbb{E}[\rho_\alpha(q_\alpha(Y|\boldsymbol{X}) - Y)] = \mathbb{E}[\rho_\alpha(q(\boldsymbol{X}) - Y)] + \mathbb{E}[\int_{q(\boldsymbol{X})}^{q_\alpha(Y|\boldsymbol{X})} [\alpha - \mathbf{1}_{\{t \leq Y\}}]dt].$$

Write

$$\begin{aligned}\mathbb{E}[\int_{q(\boldsymbol{X})}^{q_\alpha(Y|\boldsymbol{X})} [\alpha - \mathbf{1}_{\{t \leq Y\}}]dt] &= \mathbb{E}[\int_{q(\boldsymbol{X})}^{q_\alpha(Y|\boldsymbol{X})} \mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) \geq q(\boldsymbol{X})\}} [\alpha - \mathbf{1}_{\{t \leq Y\}}]dt] \\ &\quad + \mathbb{E}[\int_{q(\boldsymbol{X})}^{q_\alpha(Y|\boldsymbol{X})} \mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) < q(\boldsymbol{X})\}} [\alpha - \mathbf{1}_{\{t \leq Y\}}]dt].\end{aligned}$$

By definition of the definite integrals which states that the integral is bounded by the supremum and infimum of the integrated function. Then

$$\begin{aligned}\mathbb{E}[\int_{q(\boldsymbol{X})}^{q_\alpha(Y|\boldsymbol{X})} [\alpha - \mathbf{1}_{\{t \leq Y\}}]dt] &\leq \mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) \geq q(\boldsymbol{X})\}} (q_\alpha(Y|\boldsymbol{X}) - q(\boldsymbol{X}))(\alpha - \mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) \leq Y\}})] \\ &\quad + \mathbb{E}[\int_{q(\boldsymbol{X})}^{q_\alpha(Y|\boldsymbol{X})} \mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) < q(\boldsymbol{X})\}} [\alpha - \mathbf{1}_{\{t \leq Y\}}]dt] \\ &\leq \mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) \geq q(\boldsymbol{X})\}} (q_\alpha(Y|\boldsymbol{X}) - q(\boldsymbol{X}))(\alpha - \mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) \leq Y\}})] \\ &\quad + \mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) < q(\boldsymbol{X})\}} (q(\boldsymbol{X}) - q_\alpha(Y|\boldsymbol{X}))(-\alpha + \mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X}) < Y\}})].\end{aligned}$$

Since $q(\boldsymbol{X}), q_\alpha(Y|\boldsymbol{X}) \in L^1(\boldsymbol{X})$, then by definition of the expectation of finite variables,

$$\begin{aligned}
&\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})\geq q(\boldsymbol{X})\}}(q_\alpha(Y|\boldsymbol{X})-q(\boldsymbol{X}))(\alpha-\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})\leq Y\}})]\\
&\quad=\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})\geq q(\boldsymbol{X})\}}(q_\alpha(Y|\boldsymbol{X})-q(\boldsymbol{X}))]\mathbb{E}[(\alpha-\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})\leq Y\}})|\boldsymbol{X}]]\\
&\quad=\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})\geq q(\boldsymbol{X})\}}(q_\alpha(Y|\boldsymbol{X})-q(\boldsymbol{X}))]\mathbb{E}[(\alpha-\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})\leq Y\}})|\boldsymbol{X}]]\\
&\quad=\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})\geq q(\boldsymbol{X})\}}(q_\alpha(Y|\boldsymbol{X})-q(\boldsymbol{X}))](\alpha-\mathbb{P}(q_\alpha(Y|\boldsymbol{X})\leq Y)|\boldsymbol{X})]\leq 0,
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})< q(\boldsymbol{X})\}}(q(\boldsymbol{X})-q_\alpha(Y|\boldsymbol{X}))(-\alpha+\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})< Y\}})]\\
&\quad=\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})< q(\boldsymbol{X})\}}(q(\boldsymbol{X})-q_\alpha(Y|\boldsymbol{X}))]\mathbb{E}[(-\alpha+\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})< Y\}})|\boldsymbol{X}]]\\
&\quad=\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})< q(\boldsymbol{X})\}}(q(\boldsymbol{X})-q_\alpha(Y|\boldsymbol{X}))]\mathbb{E}[(-\alpha+\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})< Y\}})|\boldsymbol{X}]]\\
&\quad=\mathbb{E}[\mathbf{1}_{\{q_\alpha(Y|\boldsymbol{X})< q(\boldsymbol{X})\}}(q(\boldsymbol{X})-q_\alpha(Y|\boldsymbol{X}))](-\alpha+\mathbb{P}(q_\alpha(Y|\boldsymbol{X})< Y|\boldsymbol{X})]\leq 0.
\end{aligned}$$

The previous two inequalities arise from Definition 2 of the conditional quantile. Thus the result is $\mathbb{E}[\rho_\alpha(q_\alpha(Y|\boldsymbol{X})-Y)] \leq \mathbb{E}[\rho_\alpha(q(\boldsymbol{X})-Y)]$, for any function $q(\boldsymbol{X}) \in L^1(\boldsymbol{X})$. □

2.5.3 Pinball loss and its scoring function

We demonstrate the intimate link between the scoring function ψ_θ and the Pinball loss ρ_α. This relationship will justify the choice to use the ψ_θ as scoring function in Equation (2.8) in order to estimate quantiles of the order α. Denote $D_-\rho_\alpha, D_+\rho_\alpha$ as the left and right derivatives of ρ_α. If $h > 0$ is a sequence of positive real numbers converging to 0, then the left derivative is

$$\begin{aligned}
D_-\rho_\alpha(y-\theta(\boldsymbol{x})) &= \lim_{h\to 0}\frac{\rho_\alpha(y-\theta(\boldsymbol{x})-h)-\rho_\alpha(y-\theta(\boldsymbol{x}))}{h}\\
&= \lim_{h\to 0}\frac{(y-\theta(\boldsymbol{x})-h)(\alpha-\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})+h\}})-(y-\theta(\boldsymbol{x}))(\alpha-\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})\}})}{h}\\
&= \lim_{h\to 0}\frac{(y-\theta(\boldsymbol{x}))(\alpha-\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})+h\}}-\alpha+\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})\}})-h(\alpha-\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})+h\}})}{h}\\
&= \lim_{h\to 0}\frac{(y-\theta(\boldsymbol{x}))(\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})+h\}}+\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})\}})}{h}-(\alpha-\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})+h\}})\\
&= \mathbf{1}_{\{y\leq\theta(\boldsymbol{x})\}})-\alpha.
\end{aligned}$$

By symmetry for the right derivative, we obtain

$$D_+\rho_\alpha(y-\theta(\boldsymbol{x})) = \lim_{h\to 0}\frac{\rho(y-\theta(\boldsymbol{x})+h)-\rho(y-\theta(\boldsymbol{x}))}{h} = \alpha-\mathbf{1}_{\{y\leq\theta(\boldsymbol{x})\}}.$$

We note directly here the connection between ψ_θ and ρ_α

$$\begin{aligned}
D_+\rho_\alpha(y-\theta(\boldsymbol{x})) &= \psi_{\theta(\boldsymbol{x})}(y)\\
D_-\rho_\alpha(y-\theta(\boldsymbol{x})) &= -\psi_{\theta(\boldsymbol{x})}(y).
\end{aligned}$$

Solving the local moment condition in Equation (2.8) with the scoring function $\psi_{\theta(\boldsymbol{x})}(y) = \alpha - \mathbf{1}_{\{y \leq \theta(\boldsymbol{x})\}}$ thus corresponds to finding the zeros of the derivatives of the Pinball loss. That is, we verify Lemma 2 that the quantile is a minimiser of Pinball loss.

CHAPTER 3

RELAXING DOOR-TO-DOOR MATCHING REDUCES PASSENGER WAITING TIMES: A WORKFLOW FOR THE ANALYSIS OF DRIVER GPS TRACES IN A STOCHASTIC CARPOOLING SERVICE

Carpooling has the potential to transform itself into a mass transportation mode by abandoning its adherence to deterministic passenger-driver matching for door-to-door journeys, and by adopting instead stochastic matching on a network of fixed meeting points. Stochastic matching is where a passenger sends out a carpooling request at a meeting point, and then waits for the arrival of a self-selected driver who is already travelling to the requested meeting point. Crucially there is no centrally dispatched driver. Moreover, the carpooling is assured only between the meeting points, so the onus is on the passengers to travel to/from them by their own means. Thus the success of a stochastic carpooling service relies on the convergence, with minimal perturbation to their existing travel patterns, to the meeting points which are highly frequented by both passengers and drivers. Due to the innovative nature of stochastic carpooling, existing off-the-shelf workflows are largely insufficient for this purpose. To fill the gap in the market, we introduce a novel workflow, comprising of a combination of data science and GIS (Geographic Information Systems), to analyse driver GPS traces. We implement it for an operational stochastic carpooling service in south-eastern France, and we demonstrate that relaxing door-to-door matching reduces passenger waiting times. Our workflow provides additional key operational indicators, namely the driver flow maps, the driver flow temporal profiles and the driver participation rates.
Keywords : Data science, Stochastic Matching, GIS, Meeting point, Network

3.1 Introduction

Carpooling has seen an explosion of utilisation in recent years (Furuhata et al., 2013). There are many underlying reasons for this, with concerns ranging from greenhouse gas emissions and air pollution to road congestion to land use, as well as economic costs (Shaheen, Chan, et al., 2016). It also attracts intense interest since carpooling is a crucial element of almost all development plans for smart cities (Ghoseiri, 2012). A broad definition of carpooling involves a driver sharing their journey with passengers. In this paper we employ a narrower definition. We additionally require that a non-professional driver would have undertaken their journey for their own reasons, regardless of whether the passengers would have been present or not. The driver may receive payment to offset the costs of the use of their vehicle, but the profit motive is non-existent or at least not their primary motivation (Zhu, 2020). Hence we do not consider taxi-like services (such as Uber, Lyft and Kapten etc.) to be carpooling services as they employ professional drivers who create a journey in response to a passenger request and are then paid the market rate for the service rendered.

Due to the altruistic nature of carpooling, service providers tend to be small, local, non-profit organisations. Though it does not preclude viable business models arising from carpooling with non-professional drivers: the market leader BlaBlaCar levies a commission fee for facilitating the matching of drivers and passengers (Shaheen, Stocker, et al., 2017). This matching is managed by a centralised platform, which we call *deterministic matching* since a known driver is assigned in advance to collect the passenger. This deterministic matching is highly successful for infrequent, long distance, pre-reserved carpooling journeys, as witnessed by BlaBlaCar's status as a unicorn start-up company (a market capitalisation of at least 1000 million USD). Despite the success of deterministic passenger-driver matching in this market, attempts to export it other carpooling markets have not resulted in the same level of market penetration. This is most notable for frequent, short distance journeys (from 10 to 40 km roughly), which comprise the bulk of daily home-work commutes, and so carpooling remains a marginal practice in this market.

This paper focuses on short-distance, non-reserved carpooling, and it is what we refer to when employing 'carpooling' without any qualifiers. The advent of mass carpooling depends crucially on incentivising drivers and passengers to converge onto highly frequented meeting points (hotspots) along their door-to-door journeys (Stiglic et al., 2015). This type of incentivisation is well-established for a bus network where passengers embark/disembark only at the fixed bus stops. Thus mass carpooling requires a paradigm shift from considering carpooling as an exclusively private means of transport to a closer alignment to public transport models (Cooper, 2007).

Continuing with the public transport model, the meeting points are not

defined informally between passengers and drivers, but are decided in consultation with local government authorities so that they respond to the mobility requirements in the local area, taking into account various factors such as aggregated traffic flow, socioeconomic characteristics, pedestrian accessibility, local government regulations, etc. For our purposes, we consider that the identification of the meeting hotspots has been carried out beforehand. These meeting points are then connected to each other to define carpooling lines, which have massification potential, like traditional bus lines (Stiglic et al., 2015; X. Li et al., 2018).

Like a bus service, no pre-reservations are required, as a passenger makes an ad hoc carpooling request at a meeting point, and this request for the desired destination is communicated to all passing drivers in real-time via an electronic sign on the side of a highly frequented main road. Unlike for deterministic passenger-driver matching mentioned above, a specific driver is not assigned to the passenger by a centralised platform, but the decision to collect a passenger at the meeting point is made spontaneously by a self-selected driver. Since the actual driver who collects the passenger is not known deterministically in advance, but is only known to be drawn from the population of drivers, this is known as *stochastic matching*. Due to the inherent variability of these driver arrivals, stochastic matching is only feasible when employed in conjunction with a network of highly frequented meeting points.

The effects of the double innovations of fixed meeting points and stochastic matching are only sparsely covered by the recent comprehensive review of general carpooling and taxi-like services over the past two decades (H. Wang et al., 2019). So there are few off-the-shelf workflows which are suitable for the analysis of the data arising from a stochastic carpooling service. We introduce a data science-GIS workflow which fills this gap in the market. Its main data source is the GPS traces, and its secondary sources are the meeting point locations, the origin-destination matrices, the route finder API and the base maps. Data wrangling/geoprocessing are then applied to these data sources, with the critical geoprocessing step being the topological simplification of the GPS traces onto the carpooling network. This topological simplification is essential to be able to mutualise GPS traces which share common arrival times at the meeting points. From these simplified GPS traces, we can produce the waiting times. The latter allow us to assert that stochastic matching at meeting points leads to reduced passenger waiting times in comparison to door-to-door matching. In addition to the waiting times, other outputs from this workflow are the driver flow maps, the driver flow profiles, and the driver participation rates. These additional outputs are obtained at low marginal cost but which are important elements for evidence-based decision making in a stochastic carpooling service.

In Section 3.2 we present the theoretical reasons why door-to-door matching is insufficient to ensure a regular carpooling service. In Section 3.3 we detail

our data science-GIS workflow for the analysis of GPS traces. In Section 3.4, we apply this workflow to an operational stochastic carpooling service to produce the passenger waiting times and the other outputs. We end with some concluding remarks.

3.2 Door-to-door matching is an obstacle to mass carpooling

As alluded to in the introduction, door-to-door matching of complete trajectories from the origin to the destination is a structural obstacle to the transformation of carpooling to a mass transit service. To illustrate the difficulties of passenger-driver matching in space and in time for door-to-door trajectories, we can represent it with partition of a 3D cube divided into smaller sub-cubes, where the x-axis is the longitude, the y-axis the latitude and the z-axis the time, as shown in Figure 3.1. On the left, there are 9 sub-cubes, where each sub-cube represents the origin/destination of a door-to-door trajectory. The blue sub-cube in the lower left represents all the trajectories whose origins are, say, within a 5 km radius around a residential neighbourhood between 07:00 and 09:00 on Tuesday, and the green sub-cube the trajectories whose destination are within a 5 km radius of the workplace between 08:00 and 10:00 on Tuesday. So for two trajectories to match spatio-temporally in a door-to-door sense, they must share the same sub-cube for the origin, and similarly for the destination: this condition is met by the 1 pair of green and blue sub-cubes among all possible 27 pairs of sub-cubes. On the right, the conditions for a door-to-door matching are stricter, say the origin is 1 km within the residential neighbourhood during 07:00 to 07:30, and the destination is 1 km within the workplace during 08:30 to 09:00. This represents 1 pair out of 125 pairs of sub-cubes. Thus stricter door-to-door matching leads to fewer drivers being available to share their trajectories with passengers.

To supplement the heuristic observations for door-to-door matching in Figure 3.1, we demonstrate that the probability that two users (i.e. a driver and a passenger) share the same origin and destination at the same time decreases rapidly as the spatio-temporal matching conditions become more stringent. For the sake of simplicity, we suppose that the origin and destination for a driver and a passenger are both represented by independent random variables which are uniform over all sub-cubes in Figure 3.1. Let $U_{\mathrm{O}}^{\mathrm{d}}$ and $U_{\mathrm{D}}^{\mathrm{d}}$ be the origin and destination of a driver, and likewise $U_{\mathrm{O}}^{\mathrm{p}}, U_{\mathrm{D}}^{\mathrm{p}}$ for a passenger. These quantities are all uniform random variables $\mathcal{U}(\{1, \ldots, n\})$ where n is the number of sub-cubes in Figure 3.1. Then the probability of a door-to-door match between the driver and passenger is $p(n) = \mathbb{P}(U_{\mathrm{O}}^{\mathrm{p}} = U_{\mathrm{O}}^{\mathrm{d}}, U_{\mathrm{D}}^{\mathrm{p}} = U_{\mathrm{D}}^{\mathrm{d}})$. Since an exact formula for this probability is difficult to obtain, we approximate it by a Monte Carlo

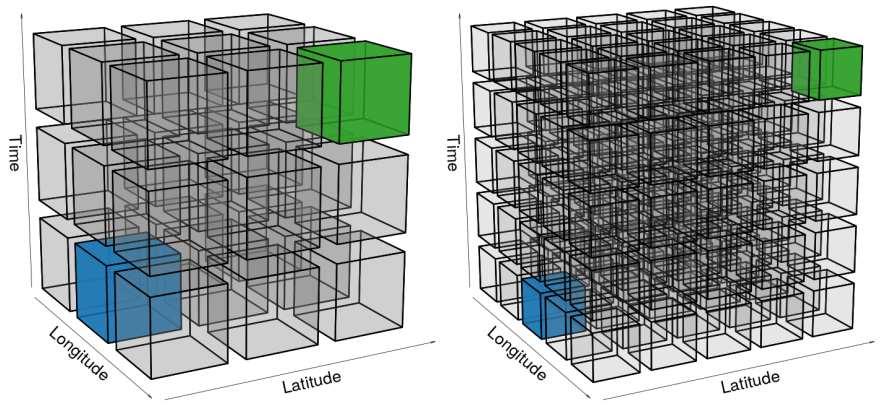


Figure 3.1 – Spatio-temporal door-to-door matching fragments the population of mutualisable trajectories. (Left) Relaxed matching conditions. (Right) Restricted matching conditions. Blue sub-cube represents the origin (residential neighbourhood), green the destination (workplace), and trajectories which share the same origin and destination sub-cubes are considered to be door-to-door matches.

re-sampling method. That is, we generate 1000 samples of $U^{\mathrm{p}}_{\mathrm{O}}, U^{\mathrm{d}}_{\mathrm{O}}, U^{\mathrm{p}}_{\mathrm{D}}, U^{\mathrm{d}}_{\mathrm{D}}$ and the probability of a door-to-door match is approximated as

$$\hat{p}(n) = \frac{1}{1000} \sum_{i=1}^{1000} \mathbf{1}\{U^{\mathrm{p}}_{\mathrm{O},i} = U^{\mathrm{d}}_{\mathrm{O},i}, U^{\mathrm{p}}_{\mathrm{D},i} = U^{\mathrm{d}}_{\mathrm{D},i}\}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. Figure 3.2 is the graph of the number of sub-cubes n versus the approximate probability of a door-to-door match $\hat{p}(n)$. If there is only 1 sub-cube (i.e. no spatio-temporal constraints) the probability of a match is 1. This probabilistic certainty decreases rapidly as the spatio-temporal constraints are added: for 27 sub-cubes, this probability is 0.6, and for 125 sub-cubes, it falls to 0.2.

The previous analysis was based on the synthetic uniformly distributed origins and destinations. For a more realistic example, we analyse some data generated by an operational carpooling service. Our example is the 'Lane' carpooling service (`lanemove.com`) operated by Ecov, in conjunction with Instant System (`instant-system.com`), since May 2018 in the peri-urban regions around Lyon in south-eastern France. Our main data source is the GPS traces of drivers, which can be considered to be a form of crowd-sourced data collection (Lee et al., 2011). Passenger GPS traces are more difficult to obtain, and as we are not able to replicate exactly the synthetic example of passenger-driver matching above, so we use door-to-door matching of driver GPS traces to illustrate the diminishing probabilities. Since these GPS traces provide highly detailed spatio-temporal information, we are able to determine the number of strict door-to-

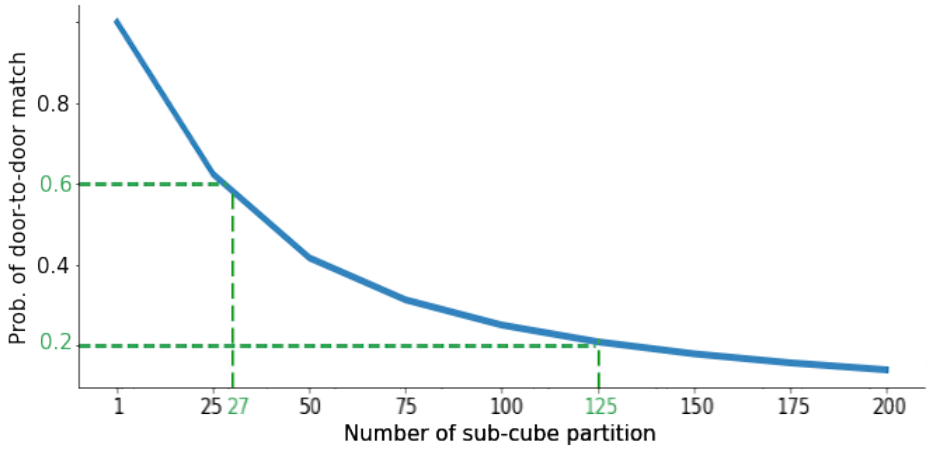


Figure 3.2 – Probability of door-to-door matches for uniformly distributed drivers and passengers, as a function of the number of sub-cube partition classes. Higher number of sub-cubes represent more stringent spatio-temporal matching conditions.

door matches which also pass by two meeting points, as well as the number of matches when door-to-door matching is relaxed. For an illustrative example in Figure 3.3, we analyse 121 GPS traces of drivers who travelled from the Bourgoin meeting point (solid black circle labelled B) to the St-Priest meeting point (solid black circle labelled S) in the Lane carpooling service during the morning operating hours (06:30 to 09:00) for the work week 2019-11-25 to 2019-12-01. A hierarchical clustering with complete linkage was carried out on the spatial locations of these origins and destinations. The dissimilarity matrix used for this hierarchical clustering is composed of the Euclidean distance between the 4-vector comprising the (origin longitude, origin latitude, destination longitude, destination latitude) of each trajectory. This dissimilarity takes into account both the origin and the destination, but not the intermediate GPS points as these actual route taken is not critical for our purposes. We cut the dendrogram at $h = 6000$ to yield 9 spatial clusters. These clusters are represented with the different colours. So GPS traces with the same colour can be considered as door-to-door matches with each other.

The number of GPS traces per cluster is given in Table 3.1: as cluster 1 contains 75% of the mutualisable traces, this leaves the other 25% spread sparsely over the other 8 clusters, fragmenting the supply of the carpooling trajectories to passengers.

To quantify the augmentation of the carpooling potential by relaxing door-to-door matching, we compare the door-to-door cluster with the largest cardinality (76 traces) from Table 3.1 to the number of the trajectories (121 traces) which coincide with this carpooling line regardless of their true origin and destination.

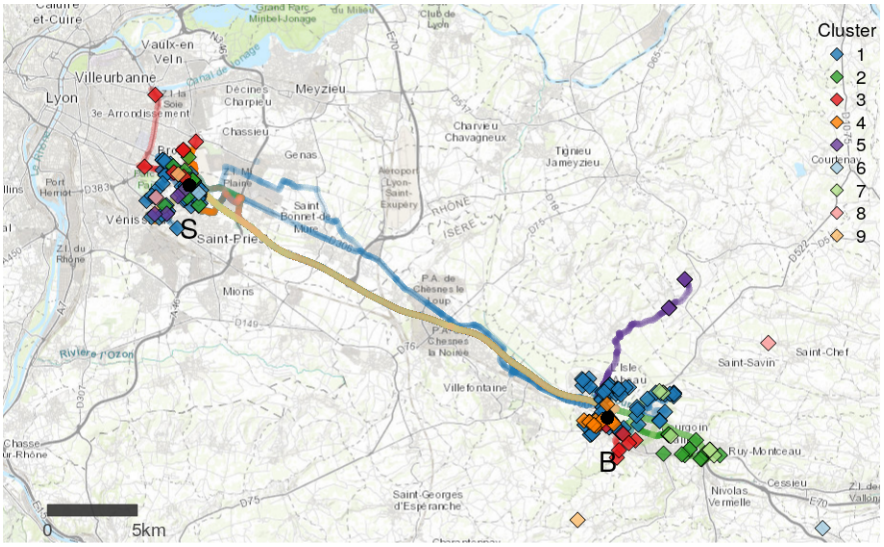


Figure 3.3 – Spatio-temporal door-to-door matching fragments the number of mutualisable trajectories in an operational stochastic carpooling service. The clusters of GPS traces of door-to-door matches are colour coded, with the GPS points as the solid circles, and the origins/destinations as the solid diamonds. The meeting points are the solid black circles, denoted B = Bourgoin, S = St-Priest.

Door-to-door cluster	1	2	3	4	5	6	7	8	9	Total
Number of GPS traces	76	15	7	9	4	1	7	1	1	121

Table 3.1 – Spatio-temporal door-to-door matching fragments the number of mutualisable trajectories in the Bourgoin > St-Priest carpooling line, during its morning operating hours 06:30–09:00, from 2018-11-25 to 2018-12-01. The first line is the door-to-door cluster label and the second line is the number of traces in each cluster.

These counts are an empirical equivalent of Figure 3.1: the left corresponds to the 121 meeting point (i.e. relaxed door-to-door) matches, whereas the right the 76 door-to-door matches. Thus meeting point matching represents an increase of 45 traces or 59% of the carpooling driver potential due to relaxing door-to-door matching.

Furthermore, Stiglic et al. (2015) and X. Li et al. (2018) provide more complex synthetic models to affirm that meeting points are essential to the feasibility of the mass carpooling services, and assert that it is almost impossible for a carpooling service to be based on door-to-door spatio-temporal matching.

Whilst these examples demonstrate that incentivising drivers to converge to meeting points, rather than relying on door-to-door matching, increases the potential pool of mutualisable journeys, we have not yet demonstrated that

this leads to reduced waiting times. This would be straightforward for the synthetic examples but this is not the case for empirically observed drivers and passengers journeys. In the next section we introduce a general workflow which indeed allows us to confirm these reduced waiting times for empirical GPS traces.

3.3 Data science-GIS workflow for the analysis of GPS traces

The GPS traces analysis workflow is illustrated in Figure 3.4. The left rectangle of Figure 3.4 contains the main data sources: the GPS traces, the meeting point locations, the origin-destination matrices, the route finder API and the base maps. The first two are supplied in-house by the carpooling service provider, the origin-destination matrices are usually supplied by a third party which has carried out a mobility survey (e.g. a national statistical agency INSEE (2018)), the route finder API is provided by a GPS navigation operator (e.g. TomTom (2019)), and the base maps are accessed from a cartography provider (e.g. OpenStreetMap contributors (2019)). There are specialised data wrangling techniques specific to spatial databases, known collectively as *geoprocessing*, and these are carried out, in conjunction with traditional data wrangling, in the central rectangle. The critical geoprocessing task concerns the topological simplification of the GPS traces onto the carpooling network. Whilst GPS traces are a rich source of information of driver behaviour, they are voluminous and complex. Our approach is based on network analysis tools (Guidotti et al., 2017) and complexity reduction/harmonisation algorithms (Douglas et al., 2011). This topological simplification is essential to be able to mutualise GPS traces which share common arrival times at the carpooling meeting points. Once these GPS traces are in a suitable format, we are able to produce the required outputs in the right rectangle, namely the predicted waiting times, the driver flow maps, the driver flow temporal profiles and the driver participation rates.

3.3.1 Data sources

Our primary data source are the driver GPS traces. A GPS trace is represented by an ℓ-sequence of triplets $\boldsymbol{X} = \{(X_i, Y_i, T_i)\}_{i=1}^{\ell}$ where (X_i, Y_i) are the longitude, latitude coordinates of the GPS sensor at the i^{th} timestamp T_i. We have n GPS traces $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ in the data collection period. The m meeting point locations are represented by their GPS coordinates $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m$. The origin-destination matrix is such that its $(j, k)^{th}$ entry is the number of journeys from j^{th} origin to the k^{th} destination. In addition to the origin-destination matrix, we have the GPS coordinates of the origins and the destinations. Whilst it is common

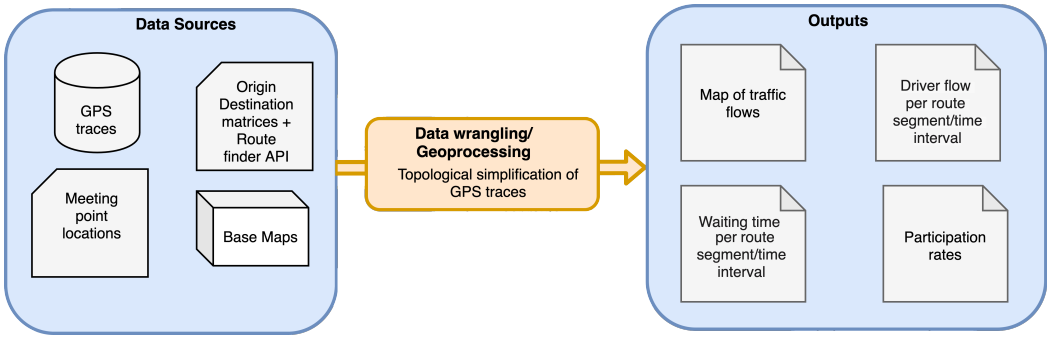


Figure 3.4 – Data science-GIS workflow for the analysis of driver GPS traces in stochastic carpooling service. (Left) Spatio-temporal input data sources. (Centre) Data wrangling and geoprocessing tasks. (Right) Generated outputs.

that they coincide, this is not required for our workflow. The base maps are graphics files of maps of the study area, which facilitate the fast and accurate map rendering at any desired scale.

3.3.2 Data wrangling/geoprocessing

From the m meeting points $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m$, a directed graph is constructed where the meeting points are the nodes, and an edge is drawn between the two nodes if carpooling between these two meeting points is guaranteed by the service provider. Thus a carpooling line is represented by an acyclic sub-graph with at least two nodes.

The crucial data wrangling/geoprocessing process applied to the GPS traces is the topological simplification of GPS traces on a carpooling line. Around each of the m meeting points, a buffer zone of 1 km radius is drawn to obtain $B(\boldsymbol{M}_1), \ldots, B(\boldsymbol{M}_m)$. The intersection of the buffer zones and the GPS trace, $\boldsymbol{X} \cap B(\boldsymbol{M}_1), \ldots, \boldsymbol{X} \cap B(\boldsymbol{M}_m)$, is m sub-sequences of the GPS points of $\boldsymbol{X}$. For those meeting points with non-empty intersections, we consider that the driver is able to collect a passenger at these points without onerous detours.

This spatial intersection only considers the spatial proximity of the driver to a passenger at a meeting point. For the carpooling to succeed, they also need to be in temporal proximity. Among the spatial intersections $\boldsymbol{X} \cap B(\boldsymbol{M}_1), \ldots, \boldsymbol{X} \cap B(\boldsymbol{M}_m)$, we examine the corresponding timestamps and retain only those in a suitably restrained time interval. If this reduced set of spatio-temporal intersections is non-empty then we proceed to the last data wrangling/geoprocessing step.

We compute the closest GPS points in $\boldsymbol{X}$ to the meeting points $\boldsymbol{M}_j$, as defined by $\boldsymbol{X}_{\boldsymbol{M}_j} = \{(X_k, Y_k, T_k) : k = \operatorname{argmin}_{1 \leq i \leq \ell} \|((X_i, Y_i) - \boldsymbol{M}_j\|\}, j = 1, \ldots, m$. From this closest point $\boldsymbol{X}_{\boldsymbol{M}_j}$, we can extract the corresponding timestamp to be an estimate of the arrival time at the meeting point $\boldsymbol{M}_j$. As an example, if the

meeting points $\boldsymbol{M}_1, \boldsymbol{M}_2$ form the carpooling line $\{\boldsymbol{M}_1 > \boldsymbol{M}_2\}$, and if the GPS trace $\boldsymbol{X}$ has well-defined estimated arrival times at $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$, then we are able to reduce the complexity of the GPS trace. That is the ℓ points of $\boldsymbol{X}$ can be reduced to the sequence of 4 points

$$\tilde{\boldsymbol{X}}(\boldsymbol{M}_1, \boldsymbol{M}_2) = \{(X_1, Y_1, T_1) > \boldsymbol{X}_{\boldsymbol{M}_1} > \boldsymbol{X}_{\boldsymbol{M}_2} > (X_\ell, Y_\ell, T_\ell)\}$$

where (X_1, Y_1, T_1) is the driver origin and (X_ℓ, Y_ℓ, T_ℓ) is the driver destination. With this simplified trace $\tilde{\boldsymbol{X}}(\boldsymbol{M}_1, \boldsymbol{M}_2)$, we are still able to determine if the driver can fulfil a passenger request at a given time on the carpooling line $\{\boldsymbol{M}_1 > \boldsymbol{M}_2\}$. The complex topology of $\boldsymbol{X}$ is thus simplified by retaining a small number of key derived indicators (Lee et al., 2011).

We repeat these data wrangling/geoprocessing steps for all n GPS traces. The result is a reduced set of $\tilde{n}$ GPS traces which correspond to the driver journeys which closely resemble the spatio-temporal characteristics of the likely passenger requests along the carpooling line.

3.3.3 Outputs

For the first output in the workflow in Figure 3.4, if we visualise the GPS traces of the reduced set of $\tilde{n}$ meeting point matches with the base maps, then we obtain a map of the driver flow that matches to the passengers in the carpooling line, as in Figure 3.3. For the second output in the workflow, suppose that the initial time interval of interest is divided into n_T sub-intervals $\tau_j, j = 1, \ldots, n_T$ since we wish to quantify the driver flow at a higher temporal resolution. Computing the driver flows $f(\tau_j), j = 1, \ldots, n_T$, is straightforward as it only requires an enumeration of the simplified GPS traces whose estimated arrival times fall within each sub-interval τ_j. That is, the driver flow for the carpooling line $\{\boldsymbol{M}_1 > \boldsymbol{M}_2\}$ during the time interval τ_j is

$$f(\tau_j, \boldsymbol{M}_1, \boldsymbol{M}_2) = \#\{i : \tilde{\boldsymbol{X}}_i(\boldsymbol{M}_1, \boldsymbol{M}_2) \in \tau_j, i = 1, \ldots, \tilde{n}\}. \tag{3.1}$$

For the third output in the workflow, let $W(t)$ be the waiting time until the driver arrival for a carpool request made at time t. For stochastic carpooling, since a specific driver is not dispatched to the given passenger, the problem is equivalent to the arrival time of the first driver from the population of available drivers. Assuming a Poissonian driver arrival process, the waiting time and the driver flow are inversely proportional to each other, $W(t) \propto \text{len}(\tau_j)/f_(\tau_j)$ where $t \in \tau_j$ and $\text{len}(\tau_j)$ is the length of the time interval τ_j. For simplicity, we set the constant of proportionality to 1 as this corresponds to the assumption that all geolocated drivers are willing to respond to a carpooling request. It is a reasonable assumption that the vast majority of geolocated drivers are willing to pick up a passenger, according to unpublished evidence supplied by Ecov.

Thus for the carpooling line $\{\boldsymbol{M}_1 > \boldsymbol{M}_2\}$, the passenger waiting times for the time interval $\tau_j, j = 1, \ldots, n_T$, are

$$W(\tau_j, \boldsymbol{M}_1, \boldsymbol{M}_2) = \text{len}(\tau_j)/f(\tau_j, \boldsymbol{M}_1, \boldsymbol{M}_2). \tag{3.2}$$

For the fourth output in the workflow, the driver participation rate is $P = n_1/n_0$ where n_1 is the total number of the drivers who are motivated to carpool in response to a passenger request, and n_0 is the total numbers of drivers who undertake journeys in the same geographical region as the carpooling service. Both n_1 and n_0 are difficult to define and to estimate precisely. We propose that $\tilde{n}$, calculated above as the number of drivers who share their geolocation, to be our proxy for n_1, as the vast majority of carpooling journeys are assured by drivers who are willing to share their geolocation.

To enumerate all n_0 drivers in the same geographical region as the carpooling network is difficult since the GPS traces for all drivers are not available. Our proxy ($\tilde{n}_0$) is derived from inferring likely trajectories from the reference origin-destination matrix. Usually this origin-destination matrix is provided at the county-level, but this is insufficiently detailed to decide if the drivers match with the meeting points on the carpooling lines. So we infer likely trajectories. These inferred likely trajectories are determined as the fastest route from the origins (county centroids) to the destinations (county centroids) by a route finder API. We employ a route finder API rather than an explicit model-based methodology, e.g. Tang et al. (2016), to infer these most likely routes. Model-based methods are the product of extensive theoretical and empirical work, and these tend to be difficult to access due to their proprietary nature. They also tend to be limited to dense urban regions, which are not the target regions for stochastic carpooling. Thus $\tilde{n}_0$ is the sum of the driver flow from all origin-destination pairs whose likely trajectories coincide with the carpooling lines. The driver participation rate for a carpooling line $\{\boldsymbol{M}_1 > \boldsymbol{M}_2\}$ is

$$\tilde{P} = \tilde{n}/\tilde{n}_0 \tag{3.3}$$

where $\tilde{n} = \sum_{j=1}^{n_T} f(\tau_j, \boldsymbol{M}_1, \boldsymbol{M}_2)$ from Equation (3.1).

Since there is no comparable door-to-door carpooling service operating concurrently with the meeting-point stochastic carpooling service, a direct comparison of empirical passenger waiting times is not possible. Instead, we propose an indirect comparison in three stages: (i) extract all driver GPS traces which connect two meeting points in a restrained time interval, as the meeting point matches, (ii) extract the largest hierarchical cluster of these GPS traces to serve as the door-to-door matches, and (iii) compute the driver flows using Equation (3.1) for both sets of matches, and then convert them using Equation (3.2) to passenger waiting time predictions.

In addition to the waiting times as an output, there are also the driver flow maps, the driver flow temporal profiles, and the driver participation rates. All

these outputs are useful in understanding the transport mix of the local area as well as the market penetration of the carpooling into the transport mix.

3.4 Case study of an operational stochastic carpooling service

Our case study focuses on the Lane carpooling service introduced earlier. Before we progress further into the data analysis of the driver GPS traces, we describe the operational details of this stochastic carpooling service. The physical meeting points require an integrated infrastructure to facilitate this real-time stochastic matching, as illustrated in Figure 3.5. The orange structure on the right functions like a bus shelter to provide protection from inclement weather whilst the passenger waits, and a prominent visual point of reference for drivers on the road. The passenger makes a carpooling request on the console (the green device with a horizontal yellow stripe). This request is displayed on the electronic sign on the roadside. In this configuration, the electronic sign is located close to the meeting point, but this can vary considerably according to the local geographical characteristics. A driver who wishes to embark the passenger in response to their request is able to do so safely in the reserved parking place.

Figure 3.5 – Configuration of a physical meeting point for the 'Lane' carpooling service. The orange structure is like a bus shelter. A passenger notifies potential drivers of their carpooling request using the console, which is then displayed on the roadside electronic sign. A driver can safely embark the passenger in the reserved parking place. Reproduced with permission from Ecov.

3.4.1 Topological simplification of GPS traces on a carpooling line

The schematic diagram of the carpooling lines in the Lane network is shown on the left of Figure 3.6. The visual similarities of the schematic of this carpooling service with those associated with bus or train services is deliberately designed to induce the perception of carpooling as a form of public transport. There are 5 physical meeting points (Lyon Mermoz, St-Priest Parc Techno, Aéroport Lyon-St Exupéry, Villefontaine The Village, and Bourgoin La Grive Sortie 7), denoted by the circles with the stylised $\mathcal{L}$, which function analogously to bus stops. According to mobility studies in this territory, the coloured lines connect the meeting points that have a sufficient driver flow between them to maintain a carpooling service with stochastic matching. These connected meeting points form a carpooling line, analogous to a bus line, where carpooling is only available between these meeting points.

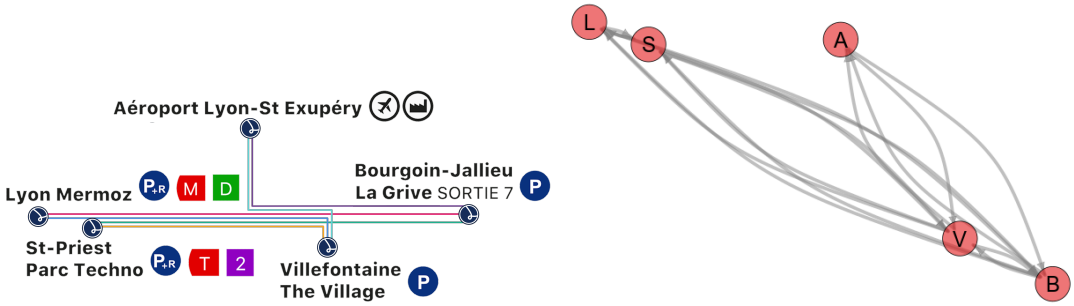


Figure 3.6 – (Left) Schematic diagram of the Lane carpooling network, which resembles the geographically restrained trajectories of a public transport service. Reproduced with permission from Ecov. (Right) Carpooling network represented as a directed graph. Nodes are the meeting points, edges connect meeting points whenever a carpooling service between them is assured.

This carpooling network is represented as a directed graph, as shown on the right of Figure 3.6, where each node is a meeting point and the edge connects two nodes if they form segment of a carpooling line. For brevity, the node labels are abbreviated to the first letter, i.e. L = Lyon Mermoz, S = St-Priest Parc Techno, A = Aéroport Lyon-St Exupéry, V = Villefontaine The Village, and B = Bourgoin La Grive Sortie 7. We focus on the most frequented carpooling line, that is, the Bourgoin > St-Priest line (green line in Figure 3.6). The topology of the road network ensures that most of the journeys from Bourgoin to St-Priest pass also by the Villefontaine meeting point, that is, the B > S carpooling line includes both sub-graphs B > S and B > V > S. The period of data collection is 2019-07-25 (service launch date) to 2020-02-17 (last date for which consistent driver GPS traces are available), during the most frequented time period (the

morning operating hours 06:30-09:00).

A complete GPS trace $\boldsymbol{X}$ is displayed as the sequence of 530 blue circles in Figure 3.7. This GPS trace passes within 1 km of the B, V and S meeting point nodes, so its simplified topology consists of the 5-node sequence {origin > B > V > S > destination}, shown as the orange arrows.

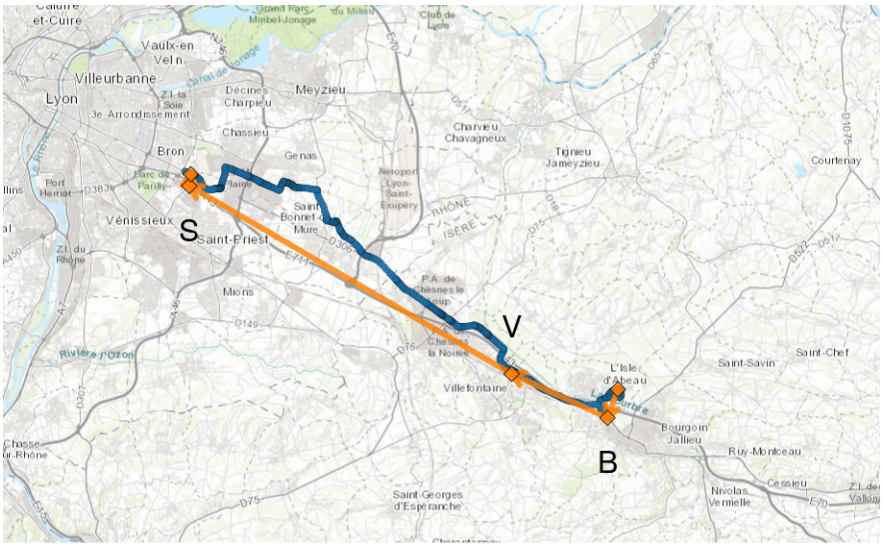


Figure 3.7 – Topological simplification of a GPS trace in the Bourgoin > St-Priest carpooling line, during its morning operating hours 06:30–09:00, on 2019-11-28. The complete GPS trace are the 530 blue circles; the sequence of five nodes, as its simplified topology, are the orange arrows, and the orange diamonds are the origin, carpooling meeting points, destination nodes. The meeting points are S = St-Priest, V = Villefontaine, and B = Bourgoin.

This simplified GPS trace represents a data compression rate of over 99% yet it retains the important information to decide the matching potential of this GPS driver trace with a passenger request on the Bourgoin > St-Priest carpooling line. In Table 3.2 is the average data compression for the $\tilde{n} = 121$ GPS traces on the Bourgoin > St-Priest line. The first column is the average number of GPS points in the complete driver traces $\#\boldsymbol{X}$, the second is the average number of GPS points of the simplified topologies $\#\tilde{\boldsymbol{X}}$, and the last column is the average data compression rate $(1 - \#\tilde{\boldsymbol{X}}/\#\boldsymbol{X})$.

3.4.2 Driver flow estimation

Of the $\tilde{n} = 121$ GPS traces that follow the Bourgoin > St-Priest carpooling line, 31 GPS traces have an arrival time at Bourgoin within 08:00 to 08:30, i.e., a close spatio-temporal match for a passenger request for a departure at the Bourgoin meeting point between 08:00 and 08:30 am, with a destination at the

Line	# points in complete GPS traces	# points in simplified GPS traces	% compression
B> S	313	5	98.3

Table 3.2 – Data compression rate on the Bourgoin > St-Priest carpooling line, during the morning operating hours 06:30–09:00, for all driver GPS traces from 2019-07-25 to 2020-02-17. The first column is the average number of GPS points in the complete driver traces, the second is the average number of points of the simplified GPS traces, and the third column is the average data compression rate.

St-Priest meeting point. Of these 31 GPS traces, 17 of them are door-to-door matches (as defined as belonging to the largest door-to-door cluster of GPS traces in Table 3.1). These 17 traces are both door-to-door and meeting point matches and their simplified traces are displayed in Figure 3.8 as the orange diamonds/arrows. The simplified traces of the remaining 14 meeting point but not door-to-door matches are the blue diamonds/arrows. The latter represent an 82% increase in the number of drivers (from 17 to 31) who can potentially respond to a passenger carpooling request on the Bourgoin > St-Priest line.

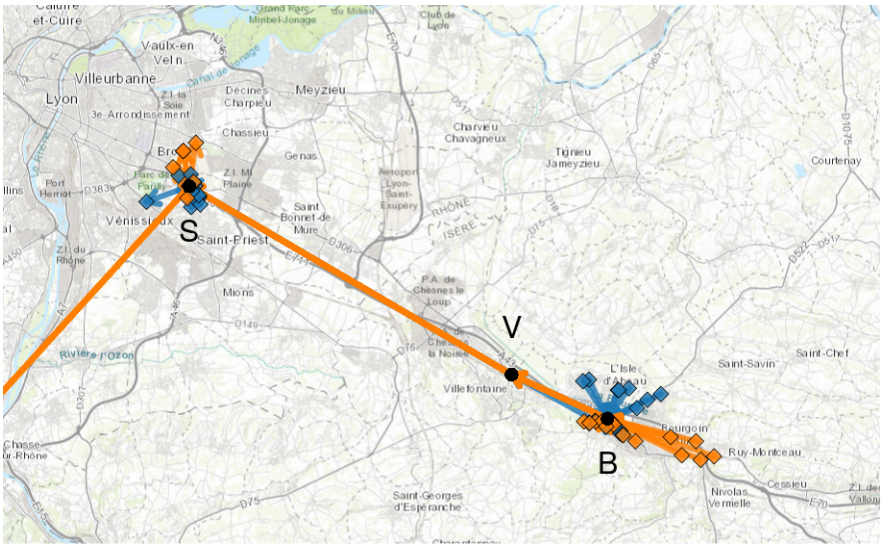


Figure 3.8 – Matching on meeting points increases the number of driver spatio-temporal matches in comparison to door-to-door matching, during a single 30 minute period (08:00-08:30), on 2019-11-28. The orange arrows are the 14 GPS traces which are both meeting point and door-to-door, and the blue arrows are the 17 GPS traces which are meeting point matches but not door-to-door match matches. The diamonds are the origin/destination points. The solid black circles are the meeting points: S = St-Priest, V = Villefontaine, and B = Bourgoin.

Table 3.3 summarises the weekly evolution of the impact of meeting point matching over door-to-door matching. The first set of three columns focus on the entire morning operating hours (06:30–09:00) whereas the second set on the single 30 minute period (08:00–08:30) as this latter restricted period is a more realistic time frame that potential passengers are willing to wait for a driver to arrive. The first column contains the weekly aggregate number of door-to-door matches, the second the number of meeting point matches, and the third is the percentage increase due to meeting point matches, i.e. $(\#\text{ meeting point matches} - \#\text{ door-to-door matches})/\#\text{ door-to-door matches}$. The number of door-to-door matches are enumerated from a similar hierarchical clustering to that in Table 3.1, and the number of meeting point matches are computed from Equation (3.1). This table demonstrates that the increase in the driver flow due to meeting point matching is maintained over the entire data collection period.

The simplified GPS traces also allow us to compute the driver flows for narrower time intervals than the 2.5 hour and 0.5 hour intervals in Table 3.3. Following Smith et al. (1997) and McShane et al. (1990) that 15 minutes intervals are a suitable choice because the variation in driver flows for shorter intervals is less stable, Table 3.4 displays the average driver flow for 15 minute intervals during 06:30 to 09:00. For robustness, we aggregate these driver flows over all weeks in the data collection period in applying Equation (3.1) since we have increased the intra-day temporal resolution.

3.4.3 Waiting time prediction

It is straightforward to convert these average driver flows in Table 3.4 into predicted waiting times using Equation (3.2). Suppose that a passenger makes a carpool request at 08:10 at the Bourgoin meeting point to travel to St-Priest. The expected waiting time is the length of the interval divided by the average driver flow in the interval 08:00–08:15, i.e. 7.5 minutes. Given that we have already established a highly detailed spatio-temporal profile of the average driver flow for a carpooling line in Table 3.4, then the predicted waiting times at the same temporal resolution are shown in Table 3.5.

For the Bourgoin > St-Priest carpooling line from 2019-07-25 to 2020-02-17, we observed roughly 1500 carpooling requests with a reliably recorded waiting time. Each box plot in Figure 3.9 displays the observed waiting times each 15 minute interval during the morning opening hours with at least one observed waiting time.

The accuracy of these predicted waiting times with respect to these observed ones is illustrated in Figure 3.10. Each box plot displays the RMSE (Root Mean Squared Error) between the observed and the predicted waiting times (from Table 3.5) for each 15 minute interval. For all 15 minute intervals, the median

	Driver flow (06:30–09:00)			Driver flow (08:00–08:30)		
Week n°	# Door-to-door	# Meeting point	% increase	# Door-to-door	# Meeting point	% increase
2019W36	54	100	85	18	24	33
2019W37	67	99	48	20	21	5
2019W38	81	122	51	20	27	35
2019W39	72	119	65	20	28	40
2019W40	43	94	119	9	19	111
2019W41	48	106	120	12	29	141
2019W42	50	103	106	18	33	83
2019W43	30	85	183	11	27	145
2019W44	43	63	47	9	14	56
2019W45	48	102	113	14	28	100
2019W46	41	71	73	12	22	83
2019W47	60	110	83	15	30	100
2019W48	76	121	59	17	31	82
2019W49	58	94	62	19	32	68
2019W50	47	99	111	5	22	340
2019W51	82	103	26	21	27	29
2020W01	12	23	92	4	6	50
2020W02	53	96	81	14	23	64
2020W03	63	100	59	14	27	93
2020W04	74	105	42	16	23	44
2020W05	55	104	89	16	23	44
2020W06	44	110	150	14	27	93
2020W07	57	96	68	13	23	77

Table 3.3 – Weekly aggregate driver flow increase of meeting point matching compared to door-to-door matching in the Bourgoin > St-Priest carpooling line, during the morning operating hours 06:30–09:00, and 08:00–08:30, from 2019-09-02 to 2020-02-17. The first columns contains the number of door-to-door matches, the second the number of meeting point matches, and the third the percentage increase due to the meeting point matches.

RMSE is around 2 to 4 minutes which implies that the predicted waiting times fairly closely track the observed waiting times. This gives us confidence in our method for predicting waiting times in a stochastic carpooling service.

Since there is no comparable operational door-to-door carpooling service to Bourgoin > St-Priest stochastic carpooling line, then it is not possible to compare empirical waiting times from each type of carpooling. Our proxy is to compare the predicted waiting times for the door-to-door matches and the

	Driver flow									
	06:30 – 06:45	06:45 –07:00	07:00 –07:15	07:15 –07:30	07:30 –07:45	07:45 –08:00	08:00 –08:15	08:15 –08:30	08:30 –08:45	08:45 –09:00
B > S	1	1.5	2.5	1.5	3	1.5	2	2	2	1

Table 3.4 – Daily average driver flow on the Bourgoin > St-Priest carpooling line, per 15 minute intervals, during the morning operating hours 06:30–09:00 for a typical day from 2019-09-02 to 2020-02-17

	Predicted waiting time (min)									
	06:30 – 06:45	06:45 –07:00	07:00 –07:15	07:15 –07:30	07:30 –07:45	07:45 –08:00	08:00 –08:15	08:15 –08:30	08:30 –08:45	08:45 –09:00
B > S	15.0	10.0	6.0	10	6.0	5.0	7.5	7.5	7.5	15

Table 3.5 – Waiting time predictions for a carpool request on the Bourgoin > St-Priest carpooling line, per 15 minute intervals, during the morning operating hours 06:30–09:00 for a typical day from 2019-09-02 to 2020-02-17.

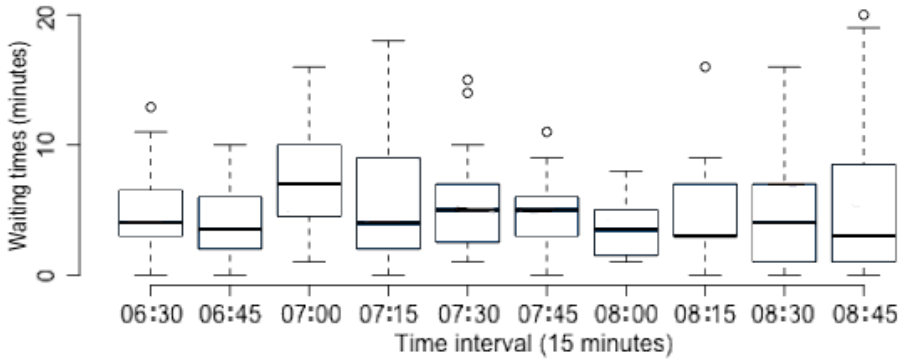


Figure 3.9 – Box plots of observed waiting times on the Bourgoin > St-Priest carpooling line, per 15 minute intervals during the morning operating hours 06:30–09:00, from 2019-09-02 to 2020-02-17.

meeting point matches. Since our method for waiting time prediction is fairly accurate for meeting point matches according to Figure 3.10, we reason that it will also yield accurate waiting times for door-to-door matches. In Table 3.6 are the predicted waiting times based on the weekly aggregate driver flows from Table 3.3 via Equation (3.2) for both door-to-door and meeting point matches. For all weeks, we observe a decrease in the predicted passenger waiting times. From anecdotal evidence from Ecov, 15 minutes corresponds roughly to the maximum time that passengers are willing to wait for a driver to arrive since a pre-arranged meeting time has not made. This 15 minute threshold is exceeded by the door-to-door waiting times for most weeks, whereas the meeting point

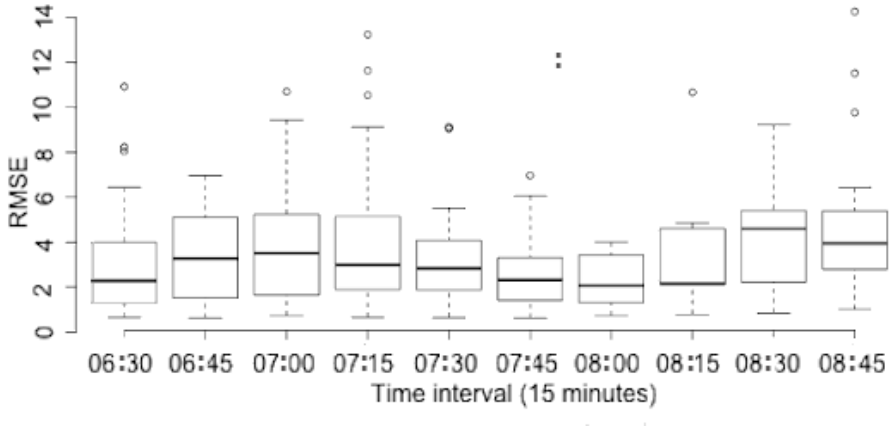


Figure 3.10 – RMSE between the observed and predicted waiting times on the Bourgoin > St-Priest carpooling line, per 15 minute intervals during the morning operating hours 06:30–09:00, from 2019-09-02 to 2020-02-17.

matched waiting times are lower than 15 minutes for most weeks.

3.4.4 Driver participation rate estimation

A key question for the service provider is what driver participation rate leads to passenger waiting times around 5 to 10 minutes, as observed in Figure 3.9? To respond to this question, we first need to enumerate the population of all drivers on a carpooling line. The county-level origin-destination matrix of home-work trajectories from the French official statistical agency (INSEE, 2018) is insufficiently detailed to decide if the drivers with these origins-destinations travel on the carpooling lines. So we infer likely trajectories, as determined as the fastest route by the TomTom route finder API (TomTom, 2019) starting on Tuesday 8am from the origins (county centroids) to the destinations (county centroids). A spatial intersection, similar to that carried out for the driver GPS traces, is computed to determine which trajectories pass within 1 km of the carpooling meeting points. These trajectories are shown in Figure 3.11. Note that there is no temporal information attached to this origin-destination matrix, but since they are home-work trajectories, we suppose they are effected in the morning peak hours which matches the time interval of the driver GPS traces.

If we then aggregate the corresponding driver flows in the origin-destination matrix, then we obtain $\tilde{n}_0 = 3821$ drivers whose likely trajectories match the Bourgoin > St-Priest carpooling line. From Table 3.4, there is a daily average of $\tilde{n} = 20$ driver GPS traces between 06:30 and 09:30. This yields an estimated driver participation rate of $\tilde{P} = \tilde{n}/\tilde{n}_0 = 0.52\%$ from Equation (3.3). Even with this low driver participation rate, average passenger waiting times of 5–10 minutes are observed in Figure 3.9.

Week n°	Predicted waiting time 06:30-09:00			Predicted waiting time 08:00-08:30		
	Door-to-door (min)	Meeting point (min)	% decrease	Door-to-door (min)	Meeting point (min)	% decrease
2019W36	13.9	7.5	-46	8.3	6.2	-25
2019W37	11.2	7.6	-32	7.5	7.1	-5
2019W38	9.3	6.1	-34	7.5	5.6	-26
2019W39	10.4	6.3	-39	7.5	5.4	-29
2019W40	17.4	8.0	-54	16.7	7.9	-53
2019W41	15.6	7.1	-55	12.5	5.2	-59
2019W42	15.0	7.3	-51	8.3	4.5	-45
2019W43	25.0	8.8	-65	13.6	5.6	-59
2019W44	17.4	11.9	-32	16.7	10.7	-36
2019W45	15.6	7.4	-53	10.7	5.4	-50
2019W46	18.3	10.6	-42	12.5	6.8	-45
2019W47	12.5	6.8	-45	10	5.0	-50
2019W48	9.9	6.2	-37	8.8	4.8	-45
2019W49	12.9	8.0	-38	7.9	4.7	-41
2019W50	16	7.6	-53	30.0	6.8	-77
2019W51	9.1	7.3	-20	7.1	5.6	-22
2020W01	14.2	7.8	-45	10.7	6.5	-39
2020W02	11.9	7.5	-37	10.7	5.6	-48
2020W03	10.1	7.1	-30	9.4	6.5	-30
2020W04	13.6	7.2	-47	9.4	6.5	-30
2020W05	17.0	6.8	-60	10.7	5.6	-48
2020W06	13.2	7.8	-41	11.5	6.5	-43
2020W07	53.6	41.7	-22	30	30	0

Table 3.6 – Weekly predicted passenger waiting times for door-to-door and meeting point matching in the Bourgoin > St-Priest carpooling line, during the morning operating hours 06:30–09:00, and 08:00–08:30, from 2019-09-02 to 2020-02-17. The first columns contains the predicted waiting times for door-to-door matches, the second for meeting point matches, and the third the percentage decrease due to the meeting point matches.

If we were able to increase this low driver participation rate even modestly (to 1% or 5%), then the predicted passenger waiting times would fall substantially, as illustrated in Figure 3.12. In this case, these waiting times would be lower than those of bus lines and approach those of high frequency metro/-subway train lines. The methods for increasing driver participation, as they lie largely outside of data science, are out of scope of this paper but are of intense interest to the service provider (Zhu, 2017; Zhu, 2018; Zhu, 2020).

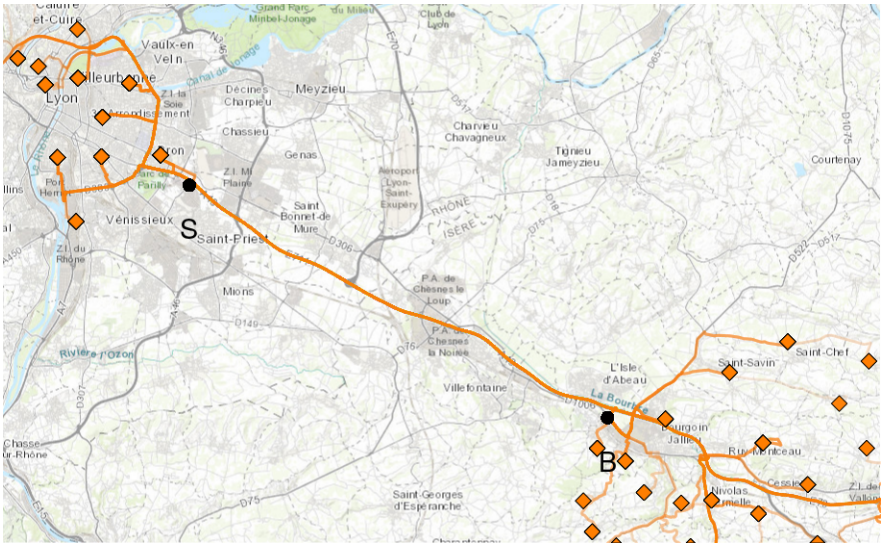


Figure 3.11 – Likely driver itineraries from the TomTom route finder API in the same geographical region as the Bourgoin > St-Priest carpooling line. The origins and destinations (county centroids) are the orange diamonds. The solid black circles are the meeting points: S = St-Priest, B = Bourgoin.

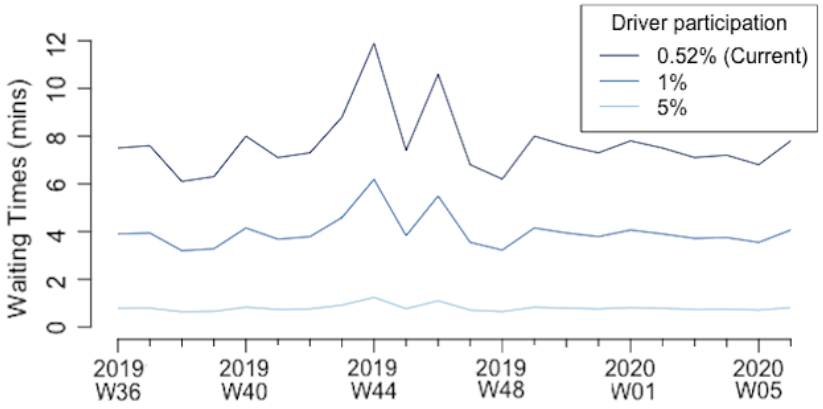


Figure 3.12 – Evolution of the predicted passenger waiting time as a function of the driver participation rate in the Bourgoin > St-Priest carpooling line, during the morning operating hours 06:30–09:00, from 2019-07-25 to 2020-02-17.

3.5 Conclusions and future work

Stochastic real-time carpooling services differ from competing services which offer deterministic door-to-door matching for complete trajectories. Whilst the latter offer a high level of personal convenience in highly urbanised regions, door-to-door matching structurally inhibits mass adoption of carpooling, especially in peri-urban regions. Relaxing the strict door-to-door matching allows,

and subsequently implementing stochastic meeting point matching, allows for the mutualisation of high throughput road segments, and thus removes this obstacle. We introduced a novel data science-GIS workflow for a stochastic carpooling service. The crucial mathematical abstraction in this workflow is to reduce the complexity of driver GPS traces to a graph-based topology of the carpooling network. We illustrated this workflow on an operational stochastic carpooling service in a peri-urban region in south-eastern France. We provided quantitative justifications that the physical meeting points, by facilitating a critical mass of drivers and passengers drawn from a much larger geographical area, leads to passenger waiting times which are lower than those achieved by door-to-door matching. Our workflow is novel combination of two closely related, but historically separate, disciplines of data science and GIS into a single workflow. In addition to predicting the passenger waiting times, it is able to deliver outputs for the driver flow maps, driver flow spatio-temporal profiles, and driver participation rates. This workflow forms a solid prototype for other workflows to accompany the expansion of stochastic carpooling services to address the mobility requirements in neglected peri-urban regions in the future.

3.6 Acknowledgements

The authors thank Ecov for providing the data sets of driver GPS traces and passenger waiting times. The authors also thank Bertrand Michel from the Central Engineering School of Nantes and Gérard Biau from Sorbonne University for their feedback.

CHAPTER 4

BAYESIAN HIERARCHICAL MODELS FOR THE PREDICTION OF THE DRIVER FLOW AND PASSENGER WAITING TIMES IN A STOCHASTIC CARPOOLING SERVICE

Carpooling is an integral component in smart carbon-neutral cities, in particular to facilitate home-work commuting. We study an innovative carpooling service which offers stochastic passenger-driver matching. Stochastic matching is when a passenger makes a carpooling request, and then waits for the first driver from a population of drivers who are already en route. Crucially a designated driver is not assigned as in a traditional carpooling service. For this new form of stochastic carpooling, we propose a two-stage Bayesian hierarchical model to predict the driver flow and the passenger waiting times. The first stage focuses on prediction of the aggregated daily driver flows, and the second stage processes these daily driver flow into hourly predictions of the passenger waiting times. We demonstrate, for an operational carpooling service, that the predictions from our Bayesian hierarchical model outperform the predictions from a frequentist model and a Bayesian non-hierarchical model. The inferences from our proposed model provide insights for the service operator in their evidence-based decision making.
Keywords : Hierarchical modelling, Gamma regression, GPS traces, MCMC, Multi-level moving average

4.1 Introduction

Providing ecologically sustainable transportation that is accessible for all is one of the major challenges in the transition to post-carbon societies. A key component of the solution is carpooling services which cater to the mobility

requirements in marginalised peri-urban regions with sparser population and physical/digital infrastructure. These carpooling lines, which closely resemble traditional bus lines, connect the physical meeting points for drivers and passengers. The meeting points are placed strategically in highly frequented areas, which take into account various factors such as aggregated traffic flow, socioeconomic characteristics, pedestrian accessibility, local government regulations, etc. This concentrates the demand and the supply of carpooling so that they can reach a critical mass more quickly and more sustainably. These meeting points are where the passenger makes a carpooling request on an electronic console. Since a driver is not allocated in advance, this request is then displayed on a electronic sign on the roadside which informs all passing drivers of a passenger request to the specified destination. This is a real-time, stochastic matching between a passenger and a flow of potential drivers. This new type of passenger-driver matching, along with the aggregating effects of the highly frequented physical meeting points, enables carpooling to reach economical feasibility in peri-urban regions.

From a mathematical and technological point-of-view, it is vastly more difficult to provide a reliable waiting time of a driver arrival in stochastic matching than in deterministic matching. In the latter, a reliable waiting time requires only the tracking of a single assigned driver, whereas stochastic matching requires a more comprehensive understanding of the general driver flow. To assist in the construction of this understanding, the service operator can incentivise drivers to share their GPS locations in real-time. We focus on how to predict the driver flows and passenger waiting times from these GPS traces, which can be considered to be a form of crowd-sourced data collection (Lee et al., 2011). Due to the novelty of stochastic carpooling, there is a scarcity of empirically verified predictive models, apart from simple frequentist approaches (Ray, 2014; Papoutsis et al., 2021). We propose a more sophisticated Bayesian hierarchical approach where we first build predictive models of the potential driver flow from the observed GPS driver traces, and which are subsequently employed to predict the passenger waiting times. At the time when a passenger request is made, we model this instantaneous driver flow as a moving average of previous driver flows. Then we model the passenger waiting time as a regression with covariates based on the driver traffic flow from the first stage. Our objective is to construct a Bayesian two-stage hierarchical model which is able to predict well the daily driver flows and the hourly passenger waiting times.

The empirical data in this paper are extracted from the 'Lane' stochastic carpooling service (`www.lanemove.com`), operated by the carpooling provider Ecov (`www.ecov.fr`), in conjunction with Instant System (`www.instant-system.com`), in a peri-urban region in south-eastern France. See Papoutsis et al., 2021 for more details on its set-up. The data collection period is the 382 days from 2018-05-15 (service launch) to 2019-05-31 (beginning of the following year's

summer holiday season in France). The daily driver flows in the Lane network are presented in Figure 4.1, where we enumerate each driver GPS trace, rather than each unique driver. The ordinary weekdays (ORD) are in orange, the school holidays (SCH) in green, and the public holidays/weekends (PWE) in blue. The classic temporal cycles of driver flow data indicate that a moving average is a relevant approach for prediction.

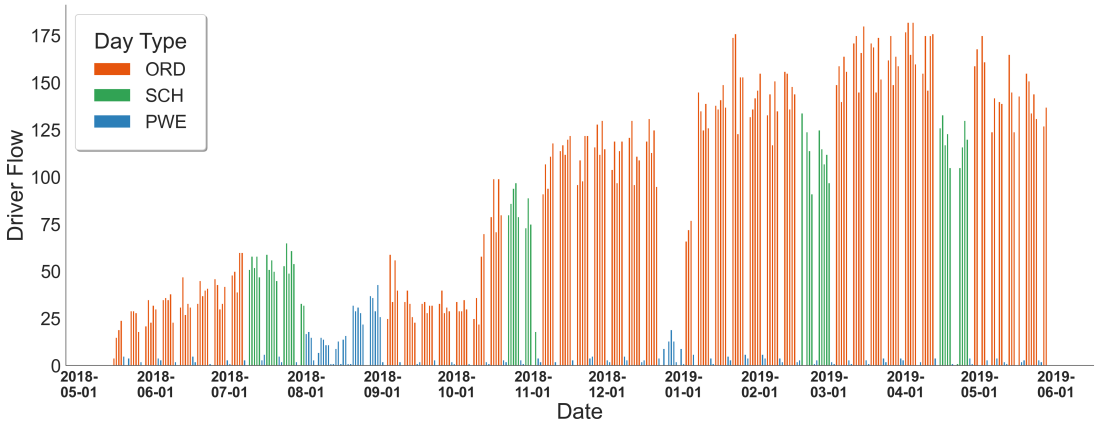


Figure 4.1 – Daily driver flow in the Lane carpooling service, from 2018-05-15 to 2019-05-31. The ordinary weekdays (ORD) are in orange, the school holidays (SCH) in green and the public holidays/weekend days (PWE) in blue.

The passenger waiting times cover the period from 2019-07-25 to 2020-02-17. This range of dates is different to those for the driver GPS traces above since, due to operational technical difficulties, the passenger waiting times were not reliably recorded from 2018-05-15 until 2019-10-21, so these dates are excluded from the analysis. In Figure 4.2 are the (approximately) 1500 observed passenger waiting times aggregated for each day. The Lane service is guaranteed only for ordinary weekdays, and whilst the passengers and drivers are not prevented from using the service on other days, there are far fewer carpooling requests on school holiday weekdays and no requests on public holidays/weekends.

In Section 4.2, we describe the two stages of the Bayesian hierarchical model for the daily driver traffic flow and the hourly passenger waiting times. In Section 4.3, we carry out a validation of the proposed model with simulated data. In Section 4.4 we apply it to empirical data drawn from an operational carpooling service, and compare the predictions of driver and passenger behaviour with those from frequentist and Bayesian non-hierarchical models. We end with some concluding remarks.

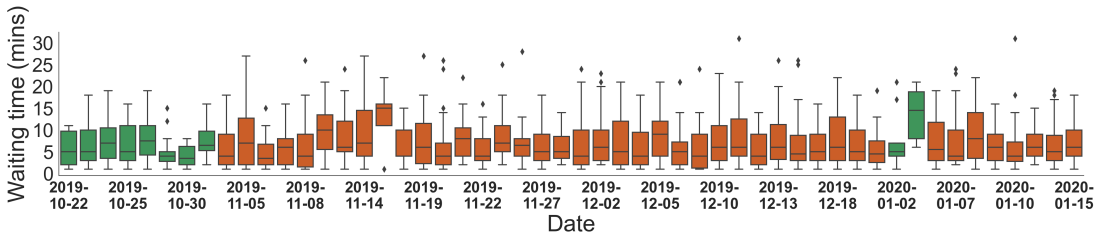


Figure 4.2 – Passenger waiting times (in minutes) in the Lane carpooling service from 2019-10-22 to 2020-01-15. The ordinary work days (ORD) are in orange, and the school holidays (SCH) in green.

4.2 Bayesian hierarchical modelling of driver flow and passenger waiting times

As the driver flow and the passenger waiting time are fundamental quantities in transportation research, their estimation and prediction are the subject of a vast field of active research so we cite only a few references. Historically the simplest models for the driver flow are the moving window averages (Stephanedes et al., 1980). More advanced methods draw from time series analysis, within a frequentist (Ding et al., 2002) or a Bayesian framework (Ghosh et al., 2007) have been posited. Established methods for waiting time prediction for stochastic carpooling tend to be frequentist approaches (Ray, 2014; Papoutsis et al., 2021).

Due to the hierarchical relationship between the driver flows and the passenger waiting times in a stochastic carpooling service, it is natural to consider nested hierarchical models. A general introduction to hierarchical models is provided in Gelman, 2006 and Gelman & Hill, 2006. Hierarchical models can be implemented with frequentist approaches, though we cite only industrial applications using Bayesian approaches here, e.g., image analysis learning (F.-F. Li et al., 2005), football results prediction (Baio et al., 2010) and electricity load forecasting (S. Wang et al., 2017). Within the transport sector, examples include traffic accident prediction (Deublein et al., 2013) and traffic flow modelling (Zammit et al., 2013). These latter approaches do not combine the driver traffic flow and the passenger waiting times and do not analyse data with differing time scales in the different stages in the hierarchical model, as we propose.

Our proposed Bayesian multi-level hierarchical model is composed of two nested stages, as illustrated in the flowchart in Figure 4.3. The input data (driver GPS traces) are preprocessed, as outlined in Appendix 4.7.1, so that they are suitable as input into the hierarchical models. The first model is a multi-level moving average model. It combines the robustness and simplicity of moving averages with the targeted adjustments of the multi-level coefficients (Stephanedes et al., 1980; Ghosh et al., 2007). The coefficient $\boldsymbol{\theta}$ in this moving average model depends on the day types (Kung et al., 2014; Bao et al., 2017),

and so the number of components of $\boldsymbol{\theta}$ depends on the number of different day types considered. The output from the first hierarchical model is the daily driver flow, which is the immediate input into the second hierarchical model. The latter is a Gamma regression, whose regression coefficient $\boldsymbol{\beta}$ has S components, with $\beta_s \in (0, 1)$ for $s = 1, \ldots, S$, for each of the S time intervals of a 24-hour period. The role of $\boldsymbol{\beta}$ is to assign the daily traffic flow to these sub-daily time intervals. The output of this second hierarchical model is the temporal profile of the passenger waiting times $\boldsymbol{w} \in \mathbb{R}_+^S$ for these sub-daily time intervals. The scarcity of the driver GPS traces allows us to model the driver flow robustly only at a daily level, whereas a higher temporal resolution of the output passenger waiting times is required for a carpooling service. Bayesian hierarchical models offer an intuitive treatment of these differing temporal resolutions within a single workflow.

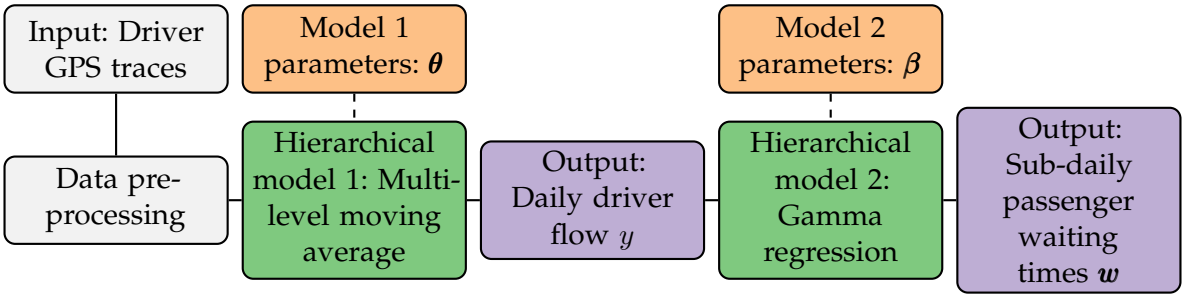


Figure 4.3 – Flowchart of Bayesian hierarchical model for driver flow and passenger waiting time prediction. The input data (driver GPS traces) are in grey, the hierarchical models in green, the model parameters in orange, and the model outputs in purple.

4.2.1 Multi-level moving average for driver flows

From a visual inspection of the daily driver flows in Figure 4.1, a standard moving average which ignores the day types would be unable to account for the abrupt differences in the driver flow when consecutive days are of different day types. Let the day type function of day i be

$$\mathrm{DT}(i) = \begin{cases} \mathrm{ORD} & \text{if day } i \text{ is an ordinary workday} \\ \mathrm{SCH} & \text{if day } i \text{ is a school holiday} \\ \mathrm{PWE} & \text{if day } i \text{ is a public holiday or a weekend} \end{cases} \tag{4.1}$$

where $i = 1, \ldots, N$. Thus a suitable K^{th} order recurrence relation of the daily driver flow y_i, for $i \geq K \geq 1$, satisfies

$$y_i = \alpha_{\mathrm{DT}(i)} \sum_{k=1}^{K} \eta_{\mathrm{DT}(i-k)} y_{i-k} + \varepsilon_i \tag{4.2}$$

where $\alpha_{\mathrm{DT}(\cdot)}$ is the coefficient for the current day i, $\eta_{\mathrm{DT}(\cdot)} = \mathbf{1}\{\mathrm{DT}(\cdot) = \mathrm{ORD}\} + \eta_{\mathrm{SCH}}\mathbf{1}\{\mathrm{DT}(\cdot) = \mathrm{SCH}\} + \eta_{\mathrm{PWE}}\mathbf{1}\{\mathrm{DT}(\cdot) = \mathrm{PWE}\}$ are the coefficients for the past K driver flows, and $\{\varepsilon_i\}$ are a sequence of independent normal random variables $\mathcal{N}(0, \sigma_\varepsilon^2)$. To ensure the identifiability of $\eta_{\mathrm{DT}(\cdot)}$, without loss of generality, we set $\eta_{\mathrm{ORD}} = 1$ for all days.

The model in Equation (4.2) has a moving average structure of order K, but with two additional multi-level coefficients that make the average adaptive to the day types for the current day i and the previous K days. The multi-level coefficients $\eta_{\mathrm{DT}(\cdot)}$ allows us to model the current driver conditioned on the previous day types, whereas the multi-level coefficients $\alpha_{\mathrm{DT}(\cdot)}$ re-scale these flows conditioned on the current day type. For example, if day i is a school holiday, then the right hand side of Equation (4.2) is

$$\alpha_{\mathrm{SCH}} \sum_{k=1}^{K} [\mathbf{1}\{\mathrm{DT}(i-k) = \mathrm{ORD}\} + \eta_{\mathrm{SCH}}\mathbf{1}\{\mathrm{DT}(i-k) = \mathrm{SCH}\} + \eta_{\mathrm{PWE}}\mathbf{1}\{\mathrm{DT}(i-k) = \mathrm{PWE}\}] y_{i-k}. \tag{4.3}$$

In the summand of Equation (4.3), the day type functions allow us to sum over the K previous days, even if they are of different types. If a previous day is a work day, then its contribution to the current driver flow is $\alpha_{\mathrm{SCH}} y_{i-k}$; if a previous day is a school holiday then it is $\alpha_{\mathrm{SCH}}\, \eta_{\mathrm{SCH}} y_{i-k}$; if a previous day is a public holiday/weekend then it is $\alpha_{\mathrm{SCH}} \eta_{\mathrm{PWE}} y_{i-k}$.

Our model in Equation (4.2) possesses a similar structure to an autoregressive model, though it does not strictly satisfy the definition of the latter. It cannot be defined with a back shift operator due to the action of the multi-level coefficients $\alpha_{\mathrm{DT}(\cdot)}$ and $\eta_{\mathrm{DT}(\cdot)}$, and the process $\{y_i \in [0, \infty), i = 1, 2, \ldots\}$ is non-stationary due to the drift in the driver participation rate after the launch of the carpooling service.

Let $\boldsymbol{\theta} = (\alpha_{\mathrm{ORD}}, \alpha_{\mathrm{SCH}}, \alpha_{\mathrm{PWE}}, \eta_{\mathrm{SCH}}, \eta_{\mathrm{PWE}}, \sigma_\varepsilon^2)$ and we fix $\eta_{\mathrm{ORD}} = 1$ identically, and have $\alpha_{\mathrm{ORD}}, \alpha_{\mathrm{SCH}}, \alpha_{\mathrm{PWE}}, \eta_{\mathrm{SCH}}, \eta_{\mathrm{PWE}} \in (0, 1)$ and $\sigma_\varepsilon^2 \in \mathbb{R}_+$. Let the N days of observed daily driver flows be $y_i, i = 1, \ldots, N$, where $N > K$. Since the error variables are independent Gaussian random variables, then the conditional likelihood of $\boldsymbol{y} = (y_K, y_{K+1}, \ldots, y_N)$ is

$$L(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma_\varepsilon^2)^{(N-K+1)/2}} \exp\left[-\frac{1}{2\sigma_\varepsilon^2} \sum_{i=K}^{N} (y_i - g_i(\boldsymbol{\theta}))^2\right]$$

where $g_i(\boldsymbol{\theta}) = \alpha_{\mathrm{DT}(i)} \sum_{k=1}^{K} \eta_{\mathrm{DT}(i-k)} y_{i-k}$. This conditional likelihood is formed by the product of the conditional densities of y_i, given $y_{i-K}, \ldots, y_{i-1}$, for $i = K+1, \ldots, N$.

In Bayesian analysis, the parameter of interest $\boldsymbol{\theta}$ is treated as a random variable, and its prior distribution π represents our belief in its uncertainty. The posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ represents an update of the prior distribution by taking into account the observed data. In our case, we do not have access to existing knowledge that would provide an informative prior and thus we form a non-informative prior on $\boldsymbol{\theta}$, i.e. $\pi(\boldsymbol{\theta}) \propto \sigma_\varepsilon^{-2}$ (Congdon, 2014, Chapter 1). This leads to the following posterior distribution

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{L(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})1(\boldsymbol{\theta})_\Theta}{\int_\Theta L(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\pi(\boldsymbol{\theta})} \propto \frac{1(\boldsymbol{\theta})_\Theta}{\sigma_\varepsilon^{N-K+1}} \exp\left[-\frac{1}{2\sigma_\varepsilon^2}\sum_{i=K}^{N}(y_i - g_i(\boldsymbol{\theta}))^2\right] \tag{4.4}$$

where $\Theta = (0,1)^5 \times \mathbb{R}_+$. For the inference on $\boldsymbol{\theta}$, Monte Carlo approximations are required since the posterior distribution (and its moments, quantiles etc.) cannot be calculated explicitly. The most widely used family of methods is the Monte Carlo Markov Chain (MCMC) which generates a Markov Chain $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots\}$ whose equilibrium distribution converges to the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y})$.

The next stage is to predict a driver flow $\tilde{y}$ in the future from the observed past driver flows $\boldsymbol{y}$. Bayesian prediction is based on the posterior predictive distribution of $\tilde{y}|\boldsymbol{y}$. Its density $p(\tilde{y}|\boldsymbol{y})$ is given by

$$p(\tilde{y}|\boldsymbol{y}) = \int_\Theta p(\tilde{y}|\boldsymbol{\theta}, \boldsymbol{y})\pi(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.5}$$

Since $p(\tilde{y}|\boldsymbol{y})$ is a compound probability distribution, we can simulate samples from this predictive distribution.

For the choice of an MCMC sampler, we use the NUT sampler (Hoffman et al., 2014), which is the default in the pyStan Python package (`https://pystan.readthedocs.io`). This package is an interface to the state-of-art platform for Bayesian computations Stan (`https://mc-stan.org`). To carry out the integration and then a random draw from the posterior predictive distribution of daily driver flows $p(\tilde{y}|\boldsymbol{y})$ in Equation (4.4), we are only required to input the prior $\pi(\boldsymbol{\theta})$, the likelihood $L(\boldsymbol{y}|\boldsymbol{\theta})$ and the recurrence relation which generates the vector of simulated driver flows $\boldsymbol{y}$ (Algorithm 4.2 in Appendix 4.7.2) into pyStan. The latter automatically simulates for the jth iteration, $j = 1, \ldots, J$,

$$\tilde{\boldsymbol{y}}^{(j)} = \begin{bmatrix} y^{(j,1)} \\ \vdots \\ y^{(j,N)} \end{bmatrix} \sim \begin{bmatrix} p(\tilde{y}^{(j,1)}|\boldsymbol{y}) \\ \vdots \\ p(\tilde{y}^{(j,N)}|\boldsymbol{y}) \end{bmatrix}, \tag{4.6}$$

and its final output is the sequence of posterior prediction vectors $\tilde{\mathbb{Y}} = \{\tilde{\boldsymbol{y}}^{(1)}, \ldots, \tilde{\boldsymbol{y}}^{(J)}\}$.

4.2.2 Gamma regression for passenger waiting times

For simplicity, we assume that a passenger can only make one request at a time for themselves only at a carpooling meeting point, and the drivers can embark

only one passenger in their vehicle in the order that the passenger requests are made. For day i, let y_i be the daily traffic flow, and that n_i passengers make carpooling requests at times $t_{i,1} < \cdots < t_{i,n_i}$. Let $t'_{i,j}$ be the driver arrival time for the passenger request at time $t_{i,j}, i = 1, \ldots, N$ and $j = 1, \ldots, n_i$. The perceived waiting time $w^*_{i,j}$ and the pseudo waiting time $w_{i,j}$ for the passenger request at time $t_{i,j}$ are

$$w^*_{i,j} = t^{'}_{i,j} - t_{i,j}$$
$$w_{i,j} = t^{'}_{i,j} - \max(t_{i,j}, t^{'}_{i,j-1})$$

with the convention $t^{'}_{i,0} = t_{i,1}$ for the first passenger on day i. Figure 4.4 illustrates the difference between the perceived and the pseudo waiting times for two passengers A, B who are both not the first passenger of the day. Passenger A arrives first and is the jth passenger of day i, and makes a carpooling request at time $t_{i,j}$. Passenger B arrives immediately afterwards and is the $(j+1)$th passenger with request time $t_{i,j+1}$. Suppose that there are at least two drivers en route to embark these passengers, and they have not received any passenger requests before passenger A's request. The first driver arrives at $t'_{i,j} > t_{i,j+1}$ (i.e. after passenger B's request time) and the second driver at $t'_{i,j+1} > t'_{i,j}$. The perceived waiting time for the passenger A is $w^*_{i,j} = t'_{i,j} - t_{i,j}$ (the blue brace in Figure 4.4) and for the passenger B it is $w^*_{i,j+1} = t'_{i,j+1} - t_{i,j+1}$ (the green brace). The pseudo waiting time for passenger A is $w_{i,j} = w^*_{i,j}$ (the blue brace) since they are at the front of the queue, and for passenger B it is $w_{i,j+1} = t'_{i,j+1} - t'_{i,j}$ (the grey brace). The pseudo waiting time $w_{i,j+1}$ for passenger B is the difference between their departure time and the departure time of the previous passenger A, and this is shorter than the perceived waiting time $w^*_{i,j+1}$.

From Figure 4.4, we observe that the perceived waiting times $w^*_{i,j}$ and $w^*_{i,j+1}$ for passengers A and B overlap, whereas the pseudo waiting times $w_{i,j}$ and $w_{i,j+1}$ do not overlap by construction. The overlapping nature of the interval processes that determine the perceived waiting times renders the problem of their unconditional prediction to be non-identifiable. Thus we focus on the pseudo waiting times, and we wish to formulate sub-daily predictions of them. Let the 24 hour period of a day be divided into S equal intervals $I_1 < \cdots < I_S$. The fraction of the daily driver flow y_i on each interval $I_s, s = 1, \ldots, S$ is $y_i\beta_s$, where $\beta_s \geq 0$ and $\sum_{s=1}^{S} \beta_s = 1$. Conditional on the traffic flow y_i and the passenger request times $t_{i,j} \in I_s$, we suppose that the pseudo waiting times $w_{i,j}$ are independent Gamma random variables with parameters ν and $\beta_s y_i$, i.e.

$$w_{i,j}|(y_i, \boldsymbol{\beta}, t_{i,j} \in I_s) \sim \Gamma(\nu, \beta_s y_i) \tag{4.7}$$

for $i = 1 \ldots N$ and $j = 1, \ldots, n_i$. This Gamma regression model assumes that the pseudo waiting times depend on the time of day and on the daily driver flow. Furthermore, it ensures that the conditional mean pseudo waiting time is

$$\mathbb{E}[w_{i,j}|(y_i, \boldsymbol{\beta}, t_{i,j} \in I_s)] = \frac{\nu}{\beta_s y_i}$$

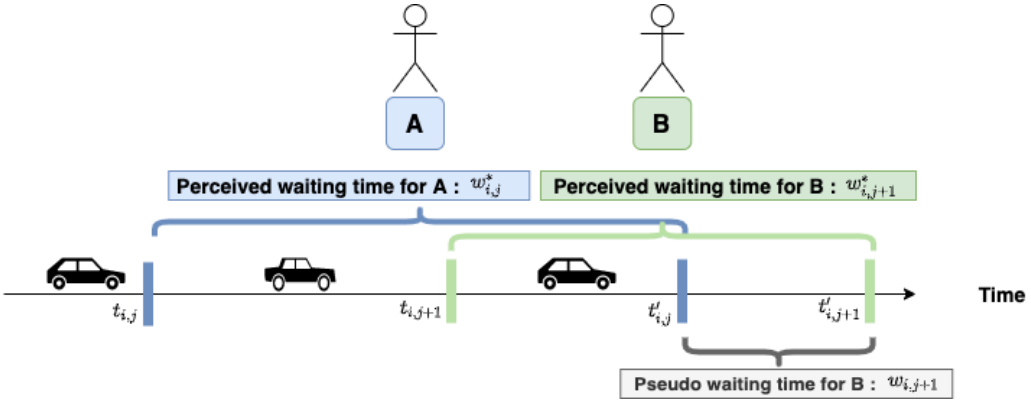


Figure 4.4 – Perceived and pseudo waiting times for the case of two passengers at a carpooling meeting point. Passenger A is at the head of the queue so their perceived waiting time (blue brace) coincides with their pseudo waiting time. For passenger B, their pseudo waiting time (grey brace) is the difference between their departure time and the departure time of passenger A, which is shorter than their perceived waiting time (green brace).

which is consistent with our intuition of the inverse relationship between the driver flow and the waiting time. Since $\boldsymbol{\beta}$ is constant for all i, then the model assumes that the relative proportions of the driver flow in the intervals $I_1, \ldots, I_S$ remain unchanged for all aggregate daily driver flows.

A Dirichlet distribution is a natural choice as a prior distribution on the coefficients $\boldsymbol{\beta}$: $\boldsymbol{\beta} \sim \mathrm{Dir}(S, \boldsymbol{\alpha})$ where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_S)$ are the concentration parameters, since it imposes the constraint $\sum_{s=1}^{S} \beta_s = 1$ on the coefficients. The corresponding Dirichlet density is $p(\boldsymbol{\beta}) = \left[\prod_{s=1}^{S} \beta_s^{\alpha_s - 1}\right] / \mathrm{B}(\boldsymbol{\alpha})$ where $B(\boldsymbol{\alpha}) = \prod_{s=1}^{S} \Gamma(\alpha_s) \Big/ \Gamma(\sum_{s=1}^{S} \alpha_s)$ and $\Gamma(x) = \int_0^\infty u^{x-1} e^{-u}\, du$. The $\boldsymbol{\beta}$ vector allows us to rebuild the temporal distribution of the sub-daily traffic flow from an aggregated daily driver flow.

Let $\boldsymbol{t}_i = (t_{i,1}, \ldots, t_{i,n_i})$ be the vector of the n_i observed passenger carpooling request times for the day $i \in \{1, \ldots, N\}$, and $\boldsymbol{t} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_N)$ be all observed passenger carpooling request times. Likewise for the passenger pseudo waiting times $\boldsymbol{w}_i$ for day i, and $\boldsymbol{w}$ for all days. Let $\boldsymbol{y} = (y_1, \ldots, y_N)$ be the observed driver flows for all days. It is reasonable to assume that the waiting times are mutually independent given $(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{t})$. The conditional likelihood of the pseudo waiting times is thus given by the joint density of $\boldsymbol{w}$ given $(\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{t})$

$$L(\boldsymbol{w}|\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{t}) = \prod_{i=1}^{N} p(\boldsymbol{w}_i|\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{t}) = \prod_{i=1}^{N} p(\boldsymbol{w}_i|\boldsymbol{\beta}, y_i, \boldsymbol{t}_i)$$

with $p(\boldsymbol{w}_i|\boldsymbol{\beta}, y_i, \boldsymbol{t}_i) = \prod_{s=1}^{S} \prod_{\{j:t_{i,j} \in I_s\}} (\beta_s y_i)^{\nu} w_{i,j}^{\nu-1} \exp(-\beta_s y_i w_{i,j}) / \Gamma(\nu)$. Then we

obtain the posterior density of $\boldsymbol{\beta}$, using a non-informative prior on $\boldsymbol{\beta}$, as

$$\pi(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{t},\boldsymbol{w}) \propto \prod_{i=1}^{N}\prod_{s=1}^{S}\prod_{\{j:t_{i,j}\in I_s\}} \frac{(\beta_s y_i)^{\nu}}{\Gamma(\nu)} w_{i,j}^{\nu-1}\exp(-\beta_s y_i w_{i,j})\mathbf{1}\{\boldsymbol{\beta}\in\mathbb{R}_+^S\}.$$

Let $\tilde{w}_s$ be the pseudo waiting time for a future day for a passenger who makes a carpooling request in the time interval $\tilde{t} \in I_s$. If we observe a new daily driver flow $\tilde{y}$ on this future day, then the posterior predictive distribution of the waiting time $\tilde{w}_s$ is

$$p(\tilde{w}_s|\tilde{y},\boldsymbol{y},\boldsymbol{w}) = \int_0^1 p(\tilde{w}_s|\tilde{y},\beta_s)\pi(\beta_s|\boldsymbol{y},\boldsymbol{w},\boldsymbol{t})\,d\beta_s, \tag{4.8}$$

which is then collated into an S-vector $(p(\tilde{w}_1|\tilde{y},\boldsymbol{y},\boldsymbol{w}),\ldots,p(\tilde{w}_S|\tilde{y},\boldsymbol{y},\boldsymbol{w}))$ for all time intervals $I_1,\ldots,I_S$.

To carry out the integration and then a random draw from the posterior predictive distribution of $p(\tilde{w}_s|\tilde{y},\boldsymbol{y},\boldsymbol{w})$ in Equation (4.8), we are only required to input the posterior predicted value of the driver flow $\tilde{y}$ (Equation (4.5)), the recurrence relation which generates the vector of simulated driver flows $\boldsymbol{y}$ (Algorithm 4.2 in Appendix 4.7.2), and the recurrence relation which generates the vector of simulated passenger pseudo waiting times $\boldsymbol{w}$ (Algorithm 4.3 in Appendix 4.7.2) into pyStan. The latter automatically simulates this $N \times S$ matrix distribution, for the jth iteration, $j = 1,\ldots,J$,

$$\tilde{\boldsymbol{W}}^{(j)} = \begin{bmatrix} \tilde{w}_{1,1}^{(j)} & \ldots & \tilde{w}_{1,S}^{(j)} \\ \vdots & & \vdots \\ \tilde{w}_{1,N}^{(j)} & \ldots & \tilde{w}_{N,S}^{(j)} \end{bmatrix} \sim \begin{bmatrix} p(\tilde{w}_1^{(j,1)}|\tilde{y},\boldsymbol{y},\boldsymbol{w}) & \ldots & p(\tilde{w}_S^{(j,1)}|\tilde{y},\boldsymbol{y},\boldsymbol{w}) \\ \vdots & & \vdots \\ p(\tilde{w}_1^{(j,N)}|\tilde{y},\boldsymbol{y},\boldsymbol{w}) & \ldots & p(\tilde{w}_S^{(j,N)}|\tilde{y},\boldsymbol{y},\boldsymbol{w}) \end{bmatrix}$$

and its final output is the sequence of posterior prediction matrices $\tilde{\mathbb{W}} = \{\tilde{\boldsymbol{W}}^{(1)},\ldots,\tilde{\boldsymbol{W}}^{(J)})\}$.

4.3 Model validation with simulated pseudo waiting times

We choose parameter values to produce simulated data which are comparable to those observed in the Lane carpooling service. We set the initial day $i = 1$ to be 2018-01-01, and the weekdays (ORD), school holidays (SCH) and public holidays/weekends (PWE) to be those observed in south-eastern France. For the simulation algorithms, the number of days is $N = 365$, the day types coefficients are $\boldsymbol{\theta} = (0.333, 0.33, 0.331, 1, 1, 1)$, the autoregression order is $K = 3$, the error variance is $\sigma_\varepsilon^2 = 5$, the 24 hour period is divided in $S = 8$ equal intervals of 3 hours, the first Gamma shape parameter is $\nu = 7$, the Gamma regression

parameters are $\boldsymbol{\beta} = (0.012, 0.01, 0.011, 0.013, 0.018, 0.016, 0.017, 0.019)$, and the number of replicates (waiting times per time interval) is $J = 10$.

We generate one simulated data set of $N = 365$ days, each with one daily driver flow $y_i, i = 1, \ldots, N$ (Algorithm 4.2), and $J = 10$ passenger pseudo waiting time $N \times S$ matrices $\mathbb{W} = \{\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(J)}\}$ (Algorithm 4.3), and the corresponding $N \times S$ posterior prediction matrices $\tilde{\mathbb{W}} = \{\tilde{\boldsymbol{W}}^{(1)}, \ldots, \tilde{\boldsymbol{W}}^{(J)}\}$. The data from these $N = 365$ days from 2018-01-01 to 2018-12-31 form the reference training data set. With the same parameters, we simulate a further $\tilde{N} = 5$ days (2019-01-01 to 2019-01-05) as the oracle test data set of $\tilde{N} \times S$ matrices $\mathbb{W}_{\text{test}} = \{\boldsymbol{W}^{(1)}_{\text{test}}, \ldots, \boldsymbol{W}^{(J)}_{\text{test}}\}$. Furthermore, from the training data only, for these same extra $\tilde{N}$ days, we generate the corresponding $\tilde{N} \times S$ posterior prediction matrices $\tilde{\mathbb{W}}_{\text{test}} = \{\tilde{\boldsymbol{W}}^{(1)}_{\text{test}}, \ldots, \tilde{\boldsymbol{W}}^{(J)}_{\text{test}}\}$. For brevity we have omitted the comparison of the driver flows and focus on the passenger waiting times for these simulated data: we make a more thorough comparison of both driver flows and passenger waiting times for the empirical data in the sequel.

From a passenger point of view, whilst the magnitude of the waiting times are important as a perception of the service quality, it is equally important that these posterior predicted waiting times be as close to the observed ones, whatever their magnitude. For example, suppose that a driver arrives 12 minutes after a passenger makes a carpooling request. In this case, a prediction of 15 minutes is better than 5 minutes since the former is closer to the observed waiting time than the latter (which is too optimistic). Therefore we propose the following metric to measure these discrepancies for a given threshold δ:

$$\text{PE}(\mathbb{W}, \tilde{\mathbb{W}}; \delta) = \frac{1}{J\tilde{N}S} \sum_{i=1}^{\tilde{N}} \sum_{s=1}^{S} \sum_{j=1}^{J} \mathbf{1}\{|\bar{\tilde{w}}_{i,s} - w^{(j)}_{i,s}| < \delta\} \tag{4.9}$$

where $\bar{\tilde{w}}_{i,s} = \frac{1}{J} \sum_{j=1}^{J} \tilde{w}^{(j,i)}_{s}$ is the mean of the posterior predicted waiting times distribution for day i, and time interval I_s.

We focus on the temporal profiles, over the $S = 8$ periods of a day, of the waiting times. In Figure 4.5 are the quantiles of the waiting times for all time intervals $I_s, s = 1, \ldots, 8$, for all days $i = 1, \ldots, \tilde{N}$ in the test phase. The grey box plots are of the observations $\mathbb{W}_{\text{test},i,s}$ and the light, medium and dark purple circles superimposed over the box plots are the 50%, 75%, 95% quantiles of the posterior predictions $\tilde{\mathbb{W}}_{\text{test},i,s}$. For operational purposes, short term prediction for the coming week is sufficient, and this is verified by the close agreement of the quantiles of the posterior predicted pseudo waiting times with their observed values for all $\tilde{N} = 5$ prediction days. The advantage of an MCMC approach here is that we are able to reproduce the entire sampling distribution of the predicted waiting times, which is more comprehensive than point or interval predictions.

The PE metric from Equation (4.9), as a function of δ, illustrated in Figure 4.6,

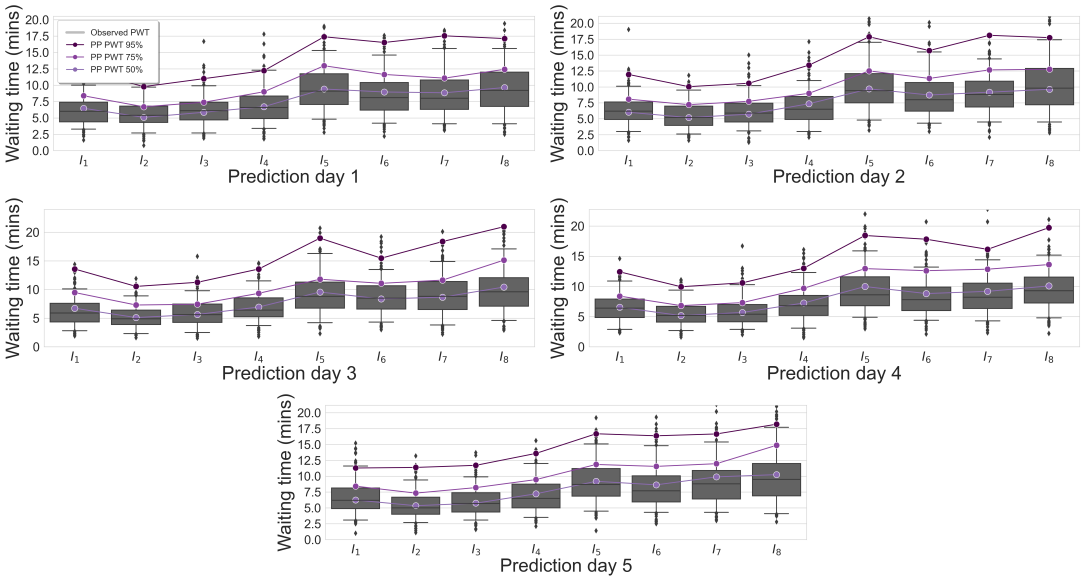


Figure 4.5 – Predictions of simulated pseudo waiting times for 3-hourly intervals, for all $\tilde{N}$ prediction days. The observed waiting times are the grey box plots, and the 50%, 75%, 95% quantiles of the posterior predicted waiting times are the light, medium and dark purple circles.

during both the training phase $\mathrm{PE}(\mathbb{W}, \tilde{\mathbb{W}}; \delta)$ (blue curve) and the test phase $\mathrm{PE}(\mathbb{W}_{\text{test}}, \tilde{\mathbb{W}}_{\text{test}}; \delta)$ (red curve). The test predictions are more accurate than the training predictions for small values of $\delta < 4$ minutes since red PE curve is below the blue PE curve in this interval. This reverses for δ greater than 4 minutes, and after 12 minutes, both curves level off at 1. Thus the posterior predictions from our proposed Bayesian hierarchical model can have robust prediction performance according to this PE metric.

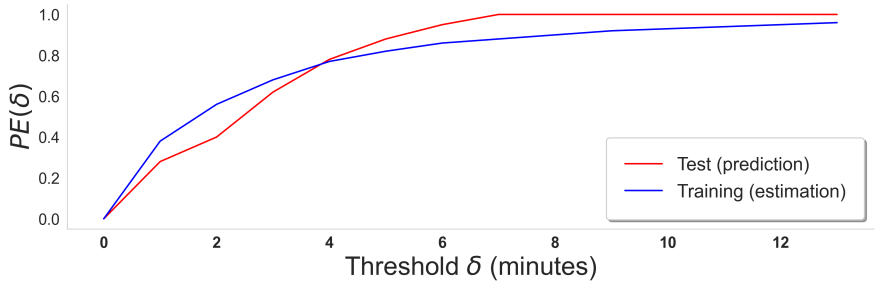


Figure 4.6 – Evolution of the PE metric of simulated and posterior predicted pseudo waiting times, as a function of the threshold δ. The blue curve is for the training phase, and the red curve for the test phase.

4.4 Model validation with the Lane carpooling service

Our objective is to employ the two-stage Bayesian hierarchical model to predict the daily driver flow distribution and the passenger pseudo waiting time distribution for the hourly intervals $I_s, s = 1, \ldots, S$, with $S = 24$ for the upcoming week. These are then compared to the observed driver flows and the pseudo waiting times from the same period.

4.4.1 Daily driver flows

We have approximately 5000 GPS traces for the 382 days from 2018-05-15 to 2019-05-31. We first apply the preprocessing, as outlined in Appendix 4.7.1, to convert the driver GPS traces into a format suitable for computing the daily driver flows y_i. For the driver flow moving average model in Equation (4.2), the $\boldsymbol{\theta}$ coefficient has a different value for each day type, since these day types are a key determinant of home-work daily commutes. This is verified empirically by the box plots of the daily driver flow in Figure 4.7. The daily driver flow for an ordinary weekday (ORD) approaches 150 trajectories, which is about double the driver flow on school holidays (SCH), and more than 20 times larger than on the public holiday/weekends (PWE).

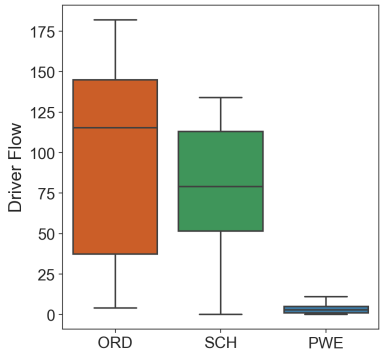


Figure 4.7 – Aggregate driver flow by day type, for the Lane carpooling service from 2018-05-15 to 2019-05-31. The ordinary weekdays (ORD) are in orange, the school holidays (SCH) in green and the public holidays/weekend days (PWE) in blue.

We divide the observed driver flows y_i into 6 different pairs of training phases, starting from 2018-05-15 and with varying N, and test phases with $\tilde{N} = 7$. In each case, we select a test week with certain characteristics as outlined in Table 4.1. The first column are the dates of the test week, the second column are the day types in the test week, the third column are the dates of the training

weeks, and the fourth column is the number of training days (N). For these training-test scenarios, in addition to our proposed Bayesian hierarchical multi-level (BHML) predictions, we compute predictions from a baseline frequentist model (BASE), and a Bayesian Prophet model (PROP). The details of these competing models are described in Appendix 4.7.3. We input the daily driver flows into the first hierarchical model from the Bayesian hierarchical multi-level model BHML to produce the posterior predicted daily driver flows $\tilde{y}_i$, as well the corresponding predictions/estimations from the frequentist baseline model BASE and the Bayesian Prophet model PROP.

	Test week	Test week day types	Training weeks	# training days (N)
#1	2019-01-14 – 2019-01-20	All ORD after holiday period (PWE/SCH)	2018-05-15 – 2019-01-13	244
#2	2019-02-25 – 2019-03-03	All SCH	2018-05-15 – 2019-02-24	286
#3	2019-04-29 – 2019-05-05	All ORD except 1 PWE (2019-05-01)	2018-05-15 – 2019-04-28	349
#4	2019-05-06 – 2019-05-12	All ORD except 1 PWE (2019-05-08)	2018-05-15 – 2019-05-05	356
#5	2019-05-13 – 2019-05-19	All ORD except 1 PWE (transport strike 2019-05-16)	2018-05-15 – 2019-05-12	363
#6	2019-05-20 – 2019-05-26	All ORD	2018-05-15 – 2019-05-19	370

Table 4.1 – Training-test scenarios for daily driver flows. The first column are the dates of the test week ($\tilde{N} = 7$), the second is the day types in the test week, the third are the dates of the training weeks and the fourth column is the number of training days N.

For the Test scenario #6, the training phase covers the dates 2018-05-15 to 2019-05-19. In Figure 4.8 is the evolution of the goodness-of-fit of the three different models for daily driver flow estimation (leaving out the first week 2018-05-15 to 2018-05-21 which serves as the 'burn-in' period). The goodness-of-fit is measured by the MSE of the estimated and the observed daily driver flows, aggregated per week. Visually the BHML tends to have the best goodness-of-fit (smallest MSE) for most weeks. The sum of these weekly MSEs are BASE: 421.9, PROP: 816.9, BHML: 297.2, which confirms our visual impression that the BHML achieves the best overall estimation accuracy.

Therefore we can be confident that the Bayesian hierarchical multi-level moving average model has good estimation accuracy/goodness-of-fit, but this good performance does not necessarily translate to prediction (Makridakis et al., 2020). So for each scenario described in Table 4.1, we compute the BHML, BASE and PROP models for the training phase, and then the days of the test phase are input into each these training models to yield the daily driver flow predic-

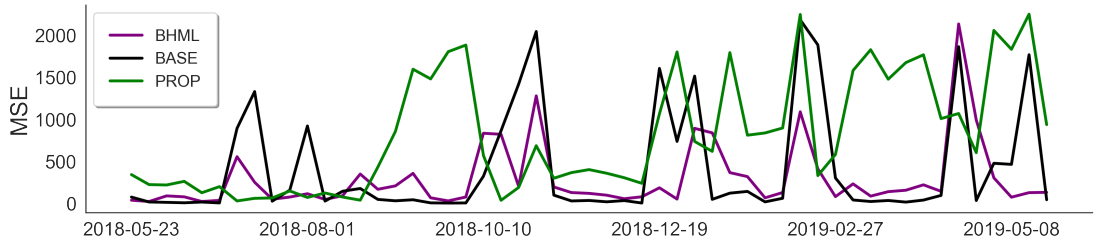


Figure 4.8 – Evolution of the goodness-of-fit of the daily driver flow estimations over the training period (2018-05-15 to 2019-05-19, test scenario # 6). Goodness-of-fit is measured by the weekly aggregated estimation MSE. Bayesian hierarchical multi-level BHML is in purple, frequentist baseline BASE in black, and Bayesian Prophet PROP in green.

tions. These predictions are presented in Figure 4.9: the Bayesian hierarchical multi-level BHML in purple, the frequentist baseline model BASE in black, the Bayesian Prophet PROP in green, and as well as the observed daily driver flows in blue. The PROP predictions are mostly too low on week days and too high on weekends for all six test weeks in comparison to the observed driver flows, whilst the BHML appears to have marginally better prediction performance than the BASE.

In Figure 4.10 are the MSEs between the observed and predicted daily driver flows: the frequentist baseline model BASE in black, the Bayesian Prophet PROP in green, and the Bayesian hierarchical multi-level BHML in purple. Overall the BHML has the best prediction accuracy for all test week scenarios. PROP is the uniformly the worst of these three models for all test weeks. BASE is the best for the test scenario #1 (all ORD after PWE/SCH period) and #6 (all ORD) with almost zero prediction MSE, though the difference with BHML is not so large. These two test scenarios are where all days in the test week are the same day type. For the other test week scenarios #1, #3, #4, #6, BHML has the smallest prediction MSE, some times by a large margin. These test week scenarios include a day which is a different day type to the other days within the test week, which the BHML handles the best.

For the service operator, the sharp differences in the driver flow for different day types within the same week has operational repercussions. For example, since the driver flow is consistently low for all pubic holidays, the service operator must communicate to passengers that the service quality on a public holiday is not the same as that for an ordinary weekday. This is analogous to a public holiday schedule in lieu of a usual weekday schedule provided by a bus operator. On a more positive note, since the driver flow for weekdays on either side of the public holiday is similar to other weekdays further away, then the service operator can also communicate that this temporary reduction in service quality is limited to the public holiday itself and the usual weekday service

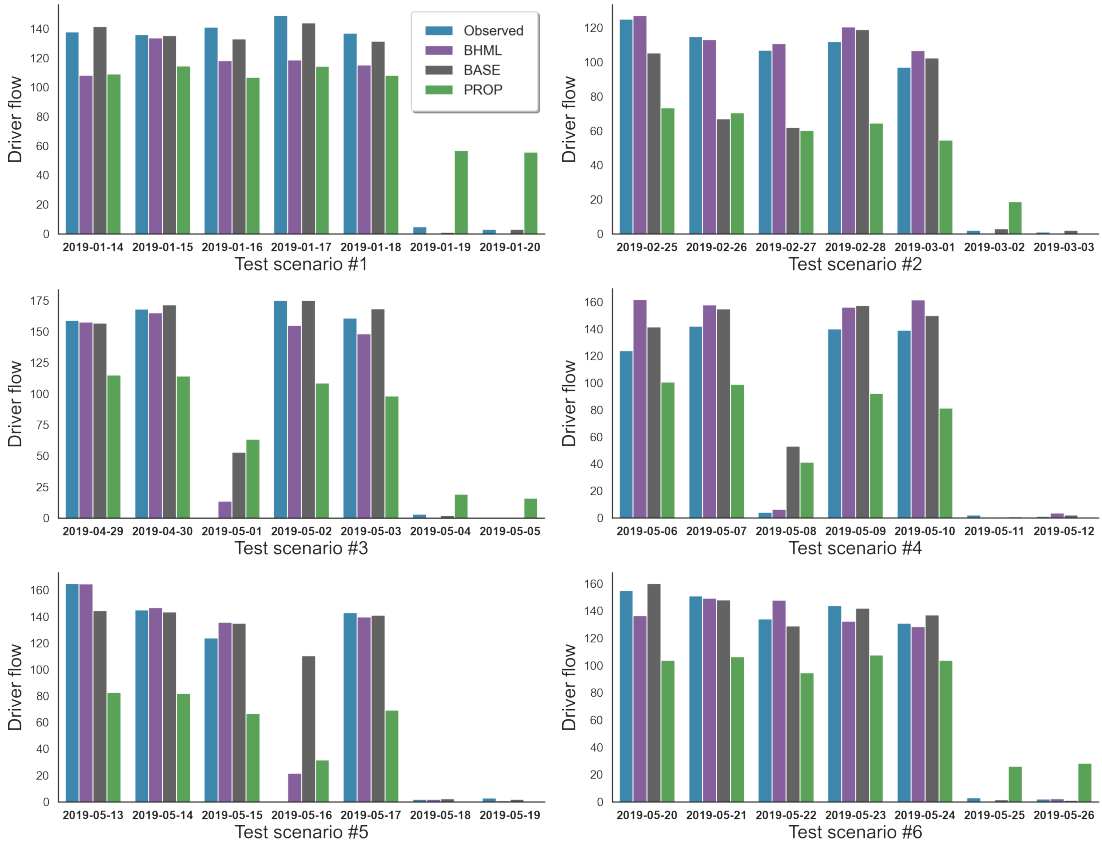


Figure 4.9 – Predictions of daily driver flows for the six test week scenarios. Observed daily driver flows are in blue. Bayesian hierarchical multi-level BHML are in purple, frequentist baseline BASE in black, and Bayesian Prophet PROP in green.

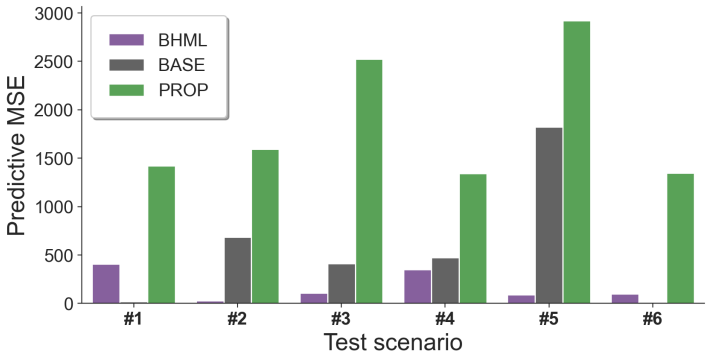


Figure 4.10 – Prediction MSE of the daily driver flow predictions for the six test week scenarios. Bayesian hierarchical multi-level BHML are in purple, frequentist baseline BASE in black, and Bayesian Prophet PROP in green.

level can be assured on the preceding and following weekdays.

4.4.2 Temporal profiles of passenger pseudo waiting times

For the passenger pseudo waiting time Gamma regression, the β coefficient, which determines in the intra-day distribution of the waiting times, is considered to be constant for all days. In Figure 4.11 are the mean observed daily traffic flows for each weekday from the Lane carpooling service, where the day is divided into 15 minute intervals. Since the service operating hours are 06:00–09:00 and 16:00–19:00, there are few drivers outside them. Each dot in the figure is the mean number of drivers for each 15 minute interval for each week day from 2018-05-15 to 2019-05-31. Each week day has a similar shape so this gives some empirical justification for supposing a constant β for all days. For the service operator, this means that it can treat all non-public holiday weekdays as similar to each other.

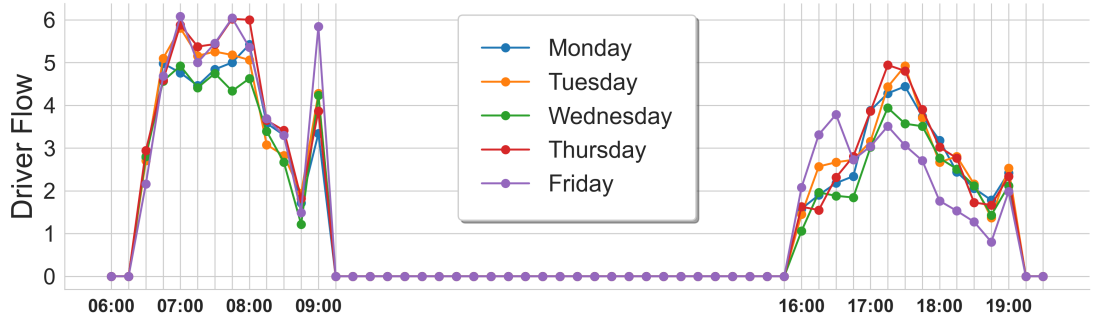


Figure 4.11 – Mean driver flows for 15 minute interval for each weekday for the Lane carpooling service, from 2018-05-15 to 2019-05-31. Monday is in blue, Tuesday in orange, Wednesday in green, Thursday in pink, Friday in violet.

Before we examine the predictions from this Gamma regression model, we provide some heuristic justification of the model itself, namely concerning the choice of the Gamma distribution and the conditioning of the waiting times with respect to the driver flow. As expressed in Equation (4.7), the pseudo waiting times are represented by different Gamma distributions for each hourly interval. In Figure 4.12, we observe that each of these empirical distributions (i) resemble Gamma distributions and (ii) have a different rate parameter β_s within each different hourly interval.

The other main assumption of the Gamma regression model in Equation (4.7) is that the mean pseudo waiting time is a decreasing function of the driver flow. In Figure 4.13, we have divided the observed daily driver flows into three categories: low (<100 vehicles per day), medium (100–200 vehicles per day), and high (>200 vehicles per day). Overall we observe that the mean waiting time is inversely proportional to the driver flow level.

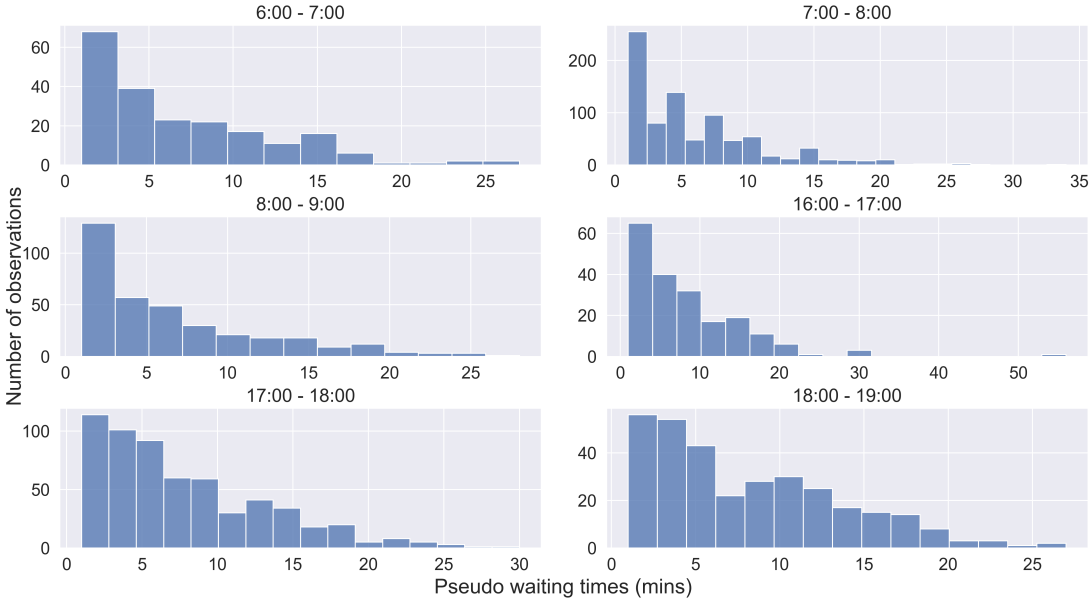


Figure 4.12 – Histograms of observed pseudo waiting times for each hourly interval for weekdays from 2019-07-25 to 2020-02-17.

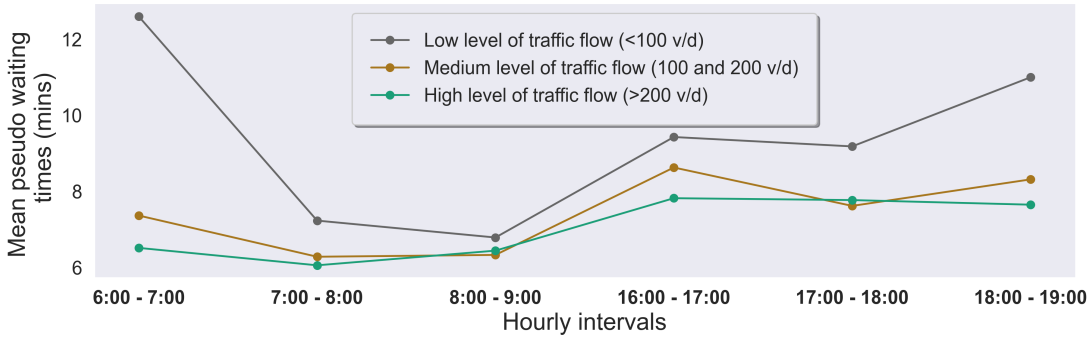


Figure 4.13 – Mean pseudo waiting times per hourly intervals as a function of driver flow from 2019-07-25 to 2020-02-17. Grey: low (<100 vehicles per day), brown: medium (100–200 vehicles per day), and green: high (>200 vehicles per day).

Now that we have verified that a Gamma regression model is suitable for the data observed in the Lane carpooling service, we proceed with the BHML to form predictions of the passenger pseudo waiting times. Since there are insufficient passenger carpooling requests to robustly compute observed hourly waiting time profiles over an entire day for the school holidays (SCH) and the public holidays/weekends (PWE), we restrict ourselves to forming predictions for the weekdays (ORD). In Figure 4.14 are the box plots of the weekly number of observed pseudo waiting times for each hourly interval for the weekdays

from 2019-07-25 to 2020-02-17. Although there are $S = 24$ hourly intervals, only those 6 which correspond to the service operating hours (06:00–09:00 and 16:00–19:00) contain any observed passenger waiting times.

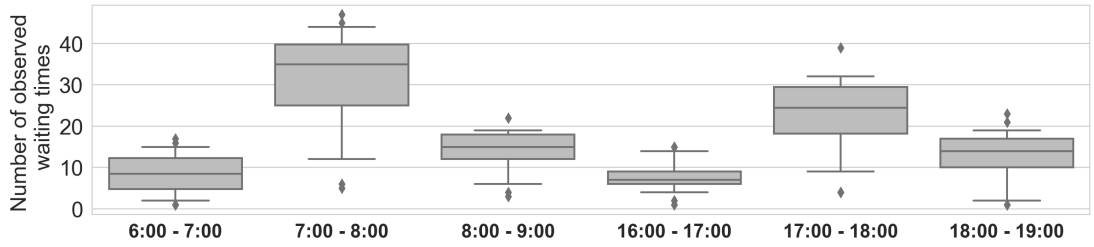


Figure 4.14 – Box plots of the weekly number of observed pseudo waiting times for each hourly interval for weekdays from 2019-07-25 to 2020-02-17.

There are a maximum of around 40 observed waiting times per hourly interval per week, which are not sufficient to infer robustly their distribution within each interval. To remedy this data sparsity, we aggregate a moving window of test data so for time interval I_s on day i, we combine its observed pseudo waiting times $w_{\text{test},i,s}$ with those for the same time interval from the previous 5 weeks with the same day of week and same day type, i.e. $\{w_{\text{test},i-k,s} : \text{DT}(i-k) = \text{DT}(i), \text{DN}(i-k) = \text{DN}(i), k = 1, \ldots, 35\}$. These days added to the test data are correspondingly removed from the training data. We aggregate the final 5 weeks to be a single test phase, so the Test scenario #7 is composed of training weeks (2019-07-25 – 2020-01-12) with 1289 observed training waiting times, and test weeks (2020-01-13 – 2020-02-17) with 520 observed test waiting times. We make predictions for only the last test week (2020-02-10 – 2020-02-17), so the number of prediction weekdays remains $\tilde{N} = 5$.

In Figure 4.15 are the box plots of the observed pseudo waiting times and the quantiles for the posterior predictions, for hourly intervals for the Test scenario #7. The observed pseudo waiting times are displayed as the grey box plots, and the 50%, 75%, 95% quantiles of the posterior predicted waiting times are the light, medium and dark purple circles. The advantage of the BHML is that we have the entire sampling distribution of the predicted waiting times, which is more comprehensive than point or interval predictions of the usual regression models. The median and upper quartile of the predicted pseudo waiting times tend to track those for the observed waiting times, especially for the 06:00-07:00, 17:00-18:00 and 18:00-19:00 intervals. From anecdotal evidence provided by Ecov, 15 minutes corresponds roughly to the maximum time that passengers are willing to wait for a driver to arrive if a pre-arranged meeting time has not been made. With the BHML predictions, we can assert that 95% of the waiting times for passenger requests do not exceed this 15 minutes threshold during most of the operating hours. Whilst this information could also be established with the empirical quantiles of the observed waiting times, the advantage of

the BHML is that it gives a more solid basis that this performance will continue into the future.

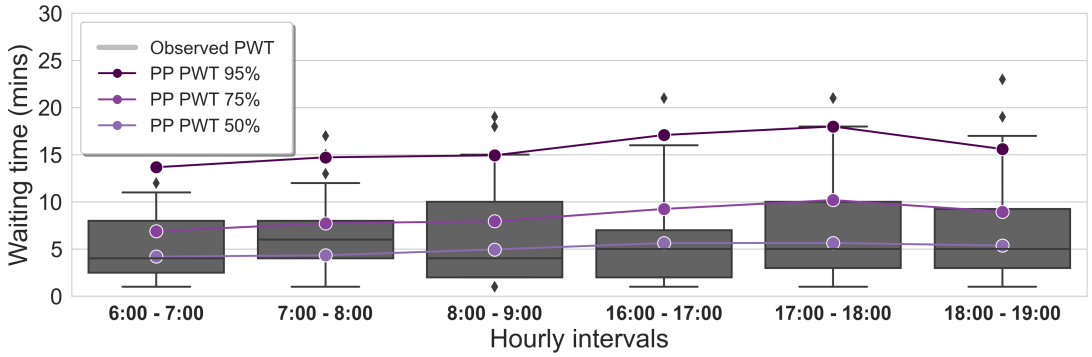


Figure 4.15 – Predictions of passenger pseudo waiting times for hourly time intervals for the Test scenario #7. The observed waiting times are the grey box plots, and the 50%, 75%, 95% quantiles of the posterior predicted waiting times are the light, medium and dark purple circles.

Lastly we consider our custom PE metric from Equation (4.9) on the BHML posterior predictions. This metric is illustrated in Figure 4.16, for both the training phase $\mathrm{PE}(\mathbb{W}, \tilde{\mathbb{W}}; \delta)$ (blue curve) and the test phase $\mathrm{PE}(\mathbb{W}_{\text{test}}, \tilde{\mathbb{W}}_{\text{test}}; \delta)$ (red curve). The blue curve dominates the red curve for most values of δ. This implies that the posterior predictions are more accurate during the test phase than in the training phase. This gives us confidence that the BHML posterior predictions are robust and are not based on over-fitting on the training data.

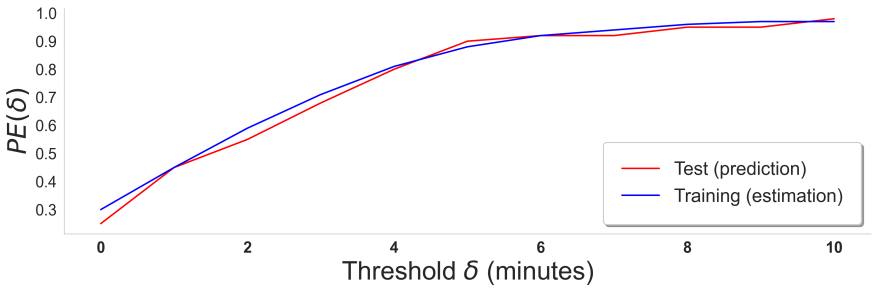


Figure 4.16 – Evolution of the PE metric of observed and BHML posterior predicted passenger pseudo waiting times, as a function of the threshold δ. The blue curve is for the training phase, and the red curve for the test phase.

For the service operator, the BHML implies that the key factor in determining the passenger waiting time (i.e. the output from the second stage) is the driver flow (i.e. the output from the first stage). It is thus imperative that a consistent level of driver participation in the carpooling service is maintained so that

consistent waiting times can be provided to passengers. Since they are non-professional drivers, then non-monetary incentives are crucial in maintaining their participation in a stochastic carpooling service (Zhu, 2017; Zhu, 2021).

In contrast to the comparison of the predicted driver flows from the BHML to those from competing models in the previous section, the comparison of the Gamma regression with other possibilities in the second stage of the BHML is not considered here. According to Papoutsis et al. (2021), the main predictor for the passenger waiting times is the driver flow, and so we conjecture that the choice of the passenger waiting times prediction model is of secondary importance with respect to the choice of the driver flow prediction model.

4.4.3 Temporal profiles for passenger perceived waiting times

We have focused on the pseudo waiting times $w_{i,j}$, though the perceived waiting times $w^*_{i,j}$ are more pertinent, since the latter are the true waiting times from the passenger point-of-view. The current set-up of the Lane carpooling service is not able to collect reliable perceived waiting times since the passenger arrivals are not reliably tracked. Also, as alluded to earlier, the unconditional prediction of the perceived waiting times is a non-identifiable problem. So a complete analysis of them is out of scope of this manuscript. Nonetheless, we can provide a framework for their future analysis based on conditioning on simulated passenger arrival processes. Since the complete posterior predictive distribution in Equation (4.8) is available, we can integrate it with respect to the passenger arrival distribution to obtain the pseudo waiting time distribution. Then we are able to reconstruct the perceived waiting times using $w^*_{i,j+1} = w_{i,j+1} + [w_{i,j} - \zeta_i | (w_{i,j} > \zeta_i)]$ with $\zeta_i = t_{i,j+1} - t_{i,j}$.

Let the passenger arrivals be a Poisson process, for $t > 0$, $N(t) = \max\{n : \sum_{k=0}^{n} A_k \leq t\}$ where $N(0) = 0$, $A_0 = 0$, and $A_k \sim \mathcal{E}(\lambda)$ and $\lambda > 0$ are independent exponential random variables. The parameter λ is the rate of the passenger arrivals, and it measures the mean number of arrivals over a unit of time. In our case, the unit of time is one hour and we focus on the opening hours of the service ($I_1 =$ 6:00-7:00, $I_2 =$ 7:00-8:00, $I_3 =$ 8:00-9:00, $I_4 =$ 16:00-17:00, $I_5 =$ 17:00-18:00, $I_6 =$ 18:00-19:00). The base passenger arrival rate is denoted by λ_1, as shown in Table 4.2. The two other scenarios involve λ_2, an increase by 50%, and λ_3, an increase of 100%.

For the base passenger arrival scenario, there are few situations where the passenger requests overlap each other, and so there is little difference between the pseudo and the perceived waiting times. The second and third scenarios with increased rates of passenger arrivals lead to increased overlapping passenger requests. This is of intense interest to the service operator because it informs them how the passenger perceived waiting times respond to the increased passenger requests whilst maintaining the current driver flow. In Figure 4.17

	Morning intervals			Evening intervals		
	06:00–07:00	07:00–08:00	08:00–09:00	16:00–17:00	17:00–18:00	18:00–19:00
λ_1	8	6	4	6	4	4
λ_2	12	9	6	9	6	6
λ_3	16	12	8	12	8	8

Table 4.2 – Poisson passenger arrival rates per hourly intervals. λ_1 is the base passenger arrival rate, λ_2 is an increase by 50% and λ_3 is an increase of 100%.

are the box plots for the pseudo and perceived waiting times for the three passenger arrival scenarios from Table 4.2 with a constant driver flow. The pseudo waiting times of the left panel are similar for the three scenarios since they do not account for overlapping passenger arrivals. On the other hand, we observe that the perceived waiting times on the right panel tend to increase as the number of passengers arriving increases. For λ_2 with a 50% increase in the passenger arrivals, the perceived waiting times remain acceptable for a stochastic carpooling service (median less than 15 minutes). However for λ_3 with a 100% increase in the passenger arrivals, the perceived waiting times exceed 15 minutes for many passengers. For the service operator to reduce the waiting times, the driver flow must be increased by increasing the driver participation rate.

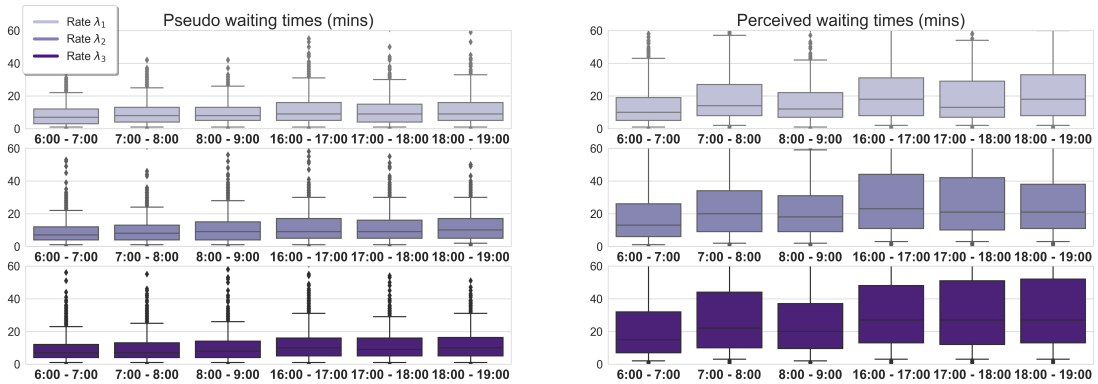


Figure 4.17 – Evolution of passenger waiting times as a function of increased passenger arrival rates. (Left) Pseudo waiting times. (Right) Perceived waiting times. λ_1 is the base passenger arrival rate, λ_2 is an increase by 50% and λ_3 is an increase of 100%.

4.5 Conclusions

The main contribution of this paper is the prediction of the daily driver flows and the hourly passenger waiting times using a nested two-stage Bayesian hierarchical model. The first stage is a multi-level moving average model of the daily driver flows, where the multi-level coefficient depends on if the current day is a work day, a school holiday or a public holiday/weekend. The second stage is a Gamma regression where the covariates are the daily driver flows from the first stage, and the response variables are the hourly passenger waiting times. The predicted driver flows and passenger waiting times are robust going into the future, since we demonstrated that they are not due to over-fitting. Furthermore, since we analyse the data from an operational carpooling service, we are able to provide operational advice. For the service operator, the baseline frequentist model is the simplest to implement, and so may be sufficient under certain cost-benefit scenarios. However only the more complex BHML can be utilised for more in-depth data analysis, such as quantiles and confidence regions of passenger waiting times, and for forward planning with the effect of increased passenger requests on waiting times.

We focused on modelling the driver arrival processes and assumed the passenger arrivals to be non-random in the first stage, and on pseudo waiting times in the second stage. One main advantage of the Bayesian hierarchical framework is that it is straightforward to generalise any of the models in the constituent stages (i) to allow the passenger arrivals to also be a random process, and (ii) to predict both the pseudo and perceived passenger waiting times. These perceived waiting times are of intense operational interest to stochastic carpooling service providers.

4.6 Acknowledgements

The authors thank Ecov for providing the data sets of the driver GPS traces and the passenger waiting times. The authors also thank Safa Fennia, Madeleine Zuber, Flavien Sindou and Constant Bridon from Ecov, and Gérard Biau from Sorbonne University for their feedback.

4.7 Appendix

4.7.1 GPS traces pre-processing

A GPS trace is an ℓ-sequence of triplets $\boldsymbol{X} = \{(X_i, Y_i, T_i)\}_{i=1}^{\ell}$ where (X_i, Y_i) are the longitude, latitude coordinates of the GPS sensor at the i^{th} timestamp T_i. The m pick-up/drop-off locations in the carpooling network are represented by their

GPS coordinates $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m$. Around each of the m pick-up/drop-off locations $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m$, a ball of 1 km radius is drawn to obtain $B(\boldsymbol{M}_1), \ldots, B(\boldsymbol{M}_m)$. The intersection of these balls and the GPS trace, $\boldsymbol{X} \cap B(\boldsymbol{M}_1), \ldots, \boldsymbol{X} \cap B(\boldsymbol{M}_m)$, is m sub-sequences of the GPS points of $\boldsymbol{X}$. For those pick-up/drop-off locations with non-empty intersections, we consider that the driver is able to collect a passenger at these points without onerous detours.

This only considers the spatial proximity of the driver to a passenger at a pick-up/drop-off location. For the carpooling to succeed, they also need to be also in temporal proximity. Among the spatial intersections $\boldsymbol{X} \cap B(\boldsymbol{M}_1), \ldots, \boldsymbol{X} \cap B(\boldsymbol{M}_m)$, we examine the corresponding timestamps and retain only those in a suitably restrained time interval. If this set of spatio-temporal intersections is non-empty then we proceed to compute the closest GPS points in $\boldsymbol{X}$ to the pick-up/drop-off locations $\boldsymbol{M}_j$, as defined by $\boldsymbol{X}_{\boldsymbol{M}_j} = \{(X_k, Y_k, T_k) : k = \operatorname{argmin}_{1 \leq i \leq \ell} \|((X_i, Y_i) - \boldsymbol{M}_j\|\}, j = 1, \ldots, m$. From this closest point $\boldsymbol{X}_{\boldsymbol{M}_j}$, we extract the corresponding timestamp T_k to be an estimate of the driver arrival time at $\boldsymbol{M}_j$.

As an example, suppose that there two pick-up/drop-off points $\boldsymbol{M}_1, \boldsymbol{M}_2$ at which the GPS trace $\boldsymbol{X}$ has well-defined estimated arrival times. Then the ℓ points of $\boldsymbol{X}$ can be reduced to the sequence of 4 points $\tilde{\boldsymbol{X}} = \{(X_1, Y_1, T_1) > \boldsymbol{X}_{\boldsymbol{M}_1} > \boldsymbol{X}_{\boldsymbol{M}_2} > (X_\ell, Y_\ell, T_\ell)\}$ where (X_1, Y_1, T_1) is the driver origin and (X_ℓ, Y_ℓ, T_ℓ) is the driver destination. With this simplified trace $\tilde{\boldsymbol{X}}$, we are still able to determine if the driver can fulfil a passenger request at $\boldsymbol{M}_1$ for a trip going to $\boldsymbol{M}_2$ at time t. The complex topology of $\boldsymbol{X}$ is simplified by retaining a small number of key derived indicators (Lee et al., 2011).

4.7.2 Simulation algorithms

Algorithm 4.1 simulates a driver flow for a single day with no day types. Equation (4.2) with no day types simplifies to $y_i = \alpha \sum_{k=1}^{K} y_{i-k} + \varepsilon_i$, which is a true autoregressive model. The inputs are the day i, the coefficient α, the autoregression order K, and the error variance σ_ε^2. The output is a single driver flow for day i. The repeat loop ensures that the simulated driver flow is strictly positive. To simulate a sequence of N driver flows, we initialise the values generated by Algorithm 4.1 for $i = 1, \ldots, K$ days, and then iterate Algorithm 4.1 sequentially for $i = K + 1, \ldots, N$.

With Algorithm 4.1 defined, it is straightforward to define one with day types (i.e. Equation (4.2)) in Algorithm 4.2. The latter has similar inputs: the day i, the day type coefficients $\boldsymbol{\theta}$, the autoregression order K, the error variance σ_ε^2, and the vector coefficients $\boldsymbol{\theta}$. The output is the daily driver flow for day i, accounting for the day types before day i.

Algorithm 4.3 simulates the passenger pseudo waiting times in Equation (4.7) for a sequence of days. The inputs are the number of days N, the day type co-

Algorithm 4.1: Daily driver flow without day types

procedure TRAFFICFLOW($i, \alpha, K, \sigma_\varepsilon^2$)
if i <= K **then**
  initialise $y \longleftarrow \mathcal{N}(30, \sigma_\varepsilon^2)$
else
  repeat
    $y \longleftarrow \mathcal{N}(\alpha \sum_{k=1}^{K} \text{TRAFFICFLOW}(i-k, \alpha, K, \sigma_\varepsilon^2), \sigma_\varepsilon^2)$
  until $y > 0$;
return: y driver flow for day i

Algorithm 4.2: Daily driver flow with day types

procedure TRAFFICFLOWDT($i, \boldsymbol{\theta}, K, \sigma_\varepsilon^2$)
if $\text{DT}(i) == \text{ORD}$ **then**
  $y \longleftarrow \text{TRAFFICFLOW}(i, \alpha_{\text{ORD}}, K, \sigma_\varepsilon^2)$
else
  if $\text{DT}(i) == \text{SCH}$ **then**
    $y \longleftarrow \text{TRAFFICFLOW}(i, \alpha_{\text{SCH}}\eta_{\text{SCH}}, K, \sigma_\varepsilon^2)$
  else
    if $\text{DT}(i) == \text{PWE}$ **then**
      $y \longleftarrow \text{TRAFFICFLOW}(i, \alpha_{\text{PWE}}\eta_{\text{PWE}}, K, \sigma_\varepsilon^2)$

efficients $\boldsymbol{\theta}$, the autoregression order K, the error variance σ_ε^2, the first shape parameter for the Gamma distribution ν, the S regression parameters $\boldsymbol{\beta}$, and the number of replicates of the waiting times J. The output are J replicates of a pseudo waiting time for each time interval $I_s, s = 1, \ldots, S$, for each day $i = 1, \ldots, N$. The TRAFFICFLOWDT procedure (Algorithm 4.2) is called outside of the replicates loop since all waiting times on a given day are simulated from the same daily driver flow.

An iteration of the nested loop in Algorithm 4.3 in the Appendix results in a single $N \times S$ matrix of pseudo waiting times drawn from the appropriate Gamma distributions

$$\boldsymbol{W}^{(j)} \sim \begin{bmatrix} \Gamma(\nu, \beta_1 y_1) & \ldots & \Gamma(\nu, \beta_S y_1) \\ \vdots & & \vdots \\ \Gamma(\nu, \beta_1 y_N) & \ldots & \Gamma(\nu, \beta_S y_N) \end{bmatrix}.$$

Algorithm 4.3: Passenger pseudo waiting times

procedure WAITINGTIME($N, \boldsymbol{\theta}, K, \sigma_\varepsilon^2, \nu, \boldsymbol{\beta}, J$)
  $S \longleftarrow$ LEN($\boldsymbol{\beta}$)
  for i **in** *1:N* **do**
    $Y[i] \longleftarrow$ TRAFFICFLOWDT($i, \boldsymbol{\theta}, K, \sigma_\varepsilon^2$)
  for j **in** *1:J* **do**
    for i **in** *1:N* **do**
      for s **in** *1:S* **do**
        $\boldsymbol{W}^{(j)}[i, s] \longleftarrow \Gamma(\nu, \beta_s Y[i])$
  return: $\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(J)}$ sequence of waiting time matrices

for $j = 1, \ldots, J$. These are collated into the sequence

$$\mathbb{W} = \{\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(J)}\} = \left\{ \begin{bmatrix} w_{1,1}^{(1)} & \ldots & w_{1,S}^{(1)} \\ \vdots & & \vdots \\ w_{N,1}^{(1)} & \ldots & w_{N,S}^{(1)} \end{bmatrix}, \ldots, \begin{bmatrix} w_{1,1}^{(J)} & \ldots & w_{1,S}^{(J)} \\ \vdots & & \vdots \\ w_{N,1}^{(J)} & \ldots & w_{N,S}^{(J)} \end{bmatrix} \right\}.$$

As Equation (4.8) generates only a single posterior prediction $\tilde{w}_s$ for a time interval I_s, we collate these $\tilde{w}_s$ for $s = 1, \ldots S$ into an S-vector, and in turn collate N of these S-vectors of posterior prediction distributions row-wise into a $N \times S$ matrix.

4.7.3 Competing models for driver flows

In addition to the multi-level moving average model for driver flows, we consider a baseline frequentist model and a Bayesian Prophet model. The baseline frequentist model has multi-levels like our model, but without the Bayesian moving average structure. To account for the school/public holidays, as proposed by Gould et al., 2008, if day i is not a school/public holiday then the average is calculated over all previous days with the same day of week as day i; and if day i is a school/public holiday, then the average is over all previous school/public holidays. That is,

$$y_i = \frac{1}{|T_d(i)|} \sum_{k \in T_d(i)} y_{i-k} \mathbf{1}\{\mathrm{DT}'(i) \neq \mathrm{HOL}\} + \frac{1}{|T_{\mathrm{HOL}}(i)|} \sum_{k \in T_{\mathrm{HOL}}(i)} y_{i-k} \mathbf{1}\{\mathrm{DT}'(i) = \mathrm{HOL}\} + \varepsilon_i \tag{4.10}$$

where the day type function is

$$\mathrm{DT}'(i) = \begin{cases} \mathrm{ORW} & \text{if day } i \text{ is an ordinary workday or a weekend day} \\ \mathrm{HOL} & \text{if day } i \text{ is a school or a public holiday;} \end{cases}$$

$T_d(i) = \{k : k < i, \mathrm{DN}(i-k) = d\}$ is the set of days with the same day of week before day i; $T_{\mathrm{HOL}}(i) = \{k : k < i, \mathrm{DT}'(i-k) = \mathrm{HOL}\}$ is the set of school/public holidays before day i; and DN is the day of week number function, $\mathrm{DN}(i) = 1$ if day i is a Monday, $\mathrm{DN}(i) = 2$ if day i is a Tuesday etc.

The Bayesian Prophet model, devised by Taylor et al., 2018; Facebook Core Data Science Group, 2019, is an additive model with three components:

$$y_i = g(i) + s(i) + h(i) + \varepsilon_i \tag{4.11}$$

where $g(i)$ is the trend, $s(i)$ is the seasonality, and $h(i)$ is the holiday effect. The linear trend is $g(i) = (k + \boldsymbol{a}(i)^\top \boldsymbol{\delta})i + (m + \boldsymbol{a}(i)^\top \boldsymbol{\gamma})$ where k is the growth rate, m is the offset, $\boldsymbol{a}$ is the change point indicator, $\boldsymbol{\delta}$ is the growth rate adjustment, and $\boldsymbol{\gamma}$ is the piece-wise continuity adjustment to ensure that g is continuous. The seasonality component is a Fourier decomposition $s(i) = \sum_{\ell=1}^{L}[\alpha_\ell \cos(2\pi\ell i/P) + \beta_\ell \sin(2\pi\ell i/P)]$ where $(\alpha_\ell, \beta_\ell)$ are the Fourier coefficients, L is the number of Fourier coefficients and P is the period (in days). The holiday effect is $h(i) = \boldsymbol{h}(i)^\top \boldsymbol{\kappa}$ where, say, $\boldsymbol{h}(i) = (\mathbf{1}\{\mathrm{DT}(i) = \mathrm{SCH}\}, \mathbf{1}\{\mathrm{DT}(i) = \mathrm{PWE}\})$ is the vector of indicator variables of the type of holiday of day i, and $\boldsymbol{\kappa}$ is the weight vector, usually equal to the all-ones vector. Taylor et al., 2018 provide the details for the construction of the change point function $\boldsymbol{a}(t)$ and the continuity adjustment parameter $\boldsymbol{\gamma}$. These authors set the number of Fourier coefficients to be $L = 10$ for yearly cycles and $L = 3$ for weekly cycles. What remains is to estimate the trend growth rate k, the offset m, the growth rate adjustments $\boldsymbol{\delta}$ and the Fourier coefficients $\boldsymbol{\alpha}$.

CHAPTER 5

CONSTRUCTION D'UN PRIOR INFORMATIF PAR TRANSFERT BAYÉSIEN : APPLICATION AU COVOITURAGE

5.1 Introduction

Dans de nombreuses situations en apprentissage statistique on dispose d'un modèle qui a été entraîné sur un premier jeu de données et on souhaite utiliser cet apprentissage pour un second jeu de données. Une situation typique est lorsque le second jeu de données est de petite taille et suit une distribution proche de celle du premier jeu de données. L'approche retenue s'appuie sur la construction d'un prior informatif pour l'ajustement d'un prédicteur provenant d'un modèle hiérarchique bayésien. Et plus spécifiquement celui du Chapitre 4. L'application visée ici, comme pour le reste de ce manuscrit, concerne le covoiturage. Nous rappelons tout d'abord quelques notions de base du paradigme bayésien dans le cas des modèles paramétriques.

Notons $\theta \in \mathbb{R}^d$ le paramètre inconnu à estimer à partir des observations $x_1, \ldots, x_n$. On note aussi $f(x|\theta)$ la densité de probabilité selon laquelle les observations sont échantillonnées. Contrairement aux statistiques fréquentistes, le paramètre θ est modélisé par une loi de probabilité π appelée loi *a priori*. Notons que la fonction $f(x|\theta)$ représente la densité de la loi des observations conditionnellement au paramètre θ. La règle de Bayes permet d'obtenir la loi du paramètre θ conditionnellement aux observations x. Il s'agit de la loi a posteriori définie par

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta}.$$

En statistique bayésienne, le *prior* permet d'ajouter à une modélisation une connaissance *a prior*, telle que la connaissance du métier ou la connaissance du terrain apportée par exemple par un expert. Le choix de cette loi a priori est important car il influence directement l'inférence du modèle. Lorsque nous n'avons pas de connaissance a priori nous avons recours à un prior dit non-informatif. Plusieurs priors non-informatif peuvent être construits, et nous

utilisons ici celui de Jeffreys, 1946. Ce prior a l'avantage d'être invariant lors de la reparamétrisation des coordonnées du paramètre. Il est défini par

$$\pi(\theta) \propto \det^{\frac{1}{2}} \mathcal{I}(\theta) \tag{5.1}$$

avec I, l'information de Fisher du modèle. Dans la communauté statistique, la définition des priors informatifs fait parfois l'objet de débats. En effet, les intuitions et les hypothèses portant sur un prior sont subjectifs. Il s'agit d'une question cruciale car un prior informatif bien construit permettra souvent d'éviter des modélisations aberrantes entraînant des conclusions erronées. Cette situation est d'autant plus sensible dans un cas de jeu de données de petite taille (voir Vanpaemel, 2011).

La construction des priors informatifs est un sujet d'intérêt pour les applications industrielles. Dans le domaine de l'énergie, Launay et al., 2015 ont construit un prior informatif pour améliorer les prédictions de consommation d'énergie des particuliers. Le prior développé dans cet article est construit sur un jeu de données long afin d'être ultérieurement appliqué sur un jeu de données court. De même, Cucchi et al., 2019 propose un prior informatif dans le secteur de l'hydrogéologie. Cette fois le prior informatif est construit à partir de données provenant de divers sites hydrogéologiques. Egidi, 2018 propose une autre application qui concerne l'industrie du sport et plus particulièrement le football et utilise un prior informatif afin de prédire les performances individuelles des joueurs. Ce dernier prior est basé sur les données historiques de chacun des membres de l'équipe sportive.

Dans ce chapitre nous nous appuyons sur des approches similaires pour proposer à notre tour un prior informatif dans le domaine du covoiturage. Plus précisément, nous développons un prior informatif afin d'améliorer les performances d'un modèle bayésien lors du lancement d'une nouvelle ligne de covoiturage. Disposer d'un modèle prédictif performant dès l'ouverture d'une nouvelle ligne de covoiturage est un enjeu stratégique pour assurer son bon fonctionnement dès son lancement. Dans ce contexte, la difficulté principale rencontrée est évidemment le manque de données historiques. L'objet de ce chapitre est de développer des solutions par transfert bayésien afin de pallier à ce problème.

5.2 Méthodologie: construction du prior informatif

La construction du prior informatif dans le cas du covoiturage Ecov est motivé d'une part par la ressemblance des données d'un territoire à un autre, et d'autre part par le manque de données historiques à l'ouverture. La Figure 5.1 illustre la ressemblance entre les flux des conducteurs sur trois lignes distinctes. Nous remarquons que les flux quotidiens des trois lignes suivent des évolutions similaires au cours de la même période. Plus précisément, nous pouvons

aussi observer la manière dont les flux sont distribués dans une même journée. La Figure 5.2 illustre le comportement des flux moyens par intervalles de 15 minutes pour les différents jours de la semaine.

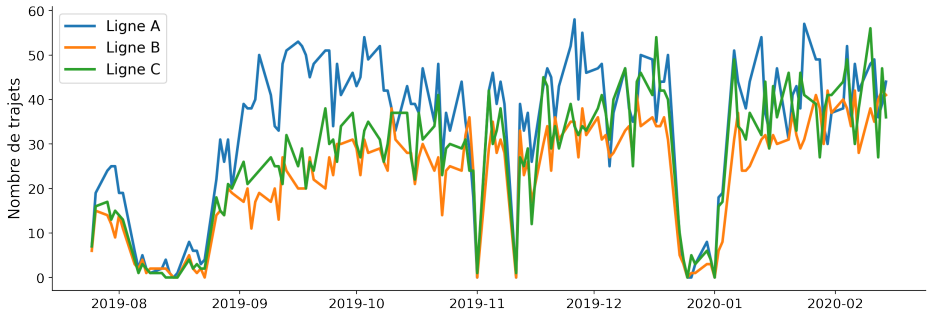


Figure 5.1 – Exemple des flux quotidiens sur trois lignes de covoiturage différentes.

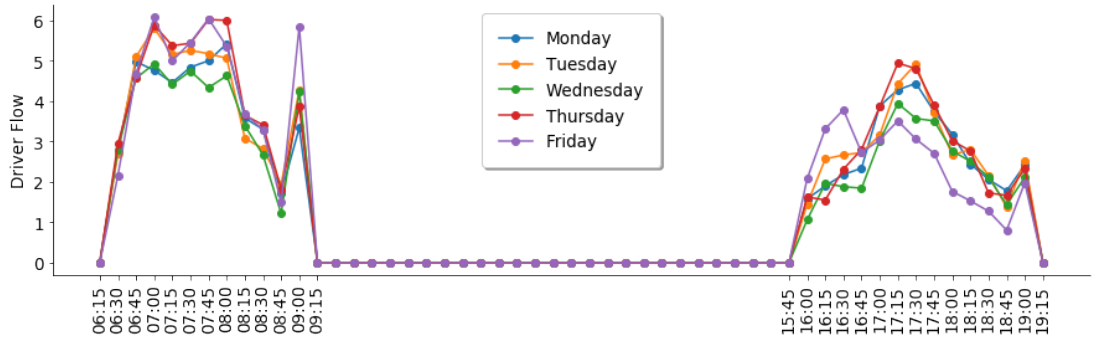


Figure 5.2 – Flux moyens des conducteurs par intervalles de 15 minutes pour chaque jour de semaine sur la période du 2018-05-15 au 2019-05-31. Le lundi est en bleu, le mardi en orange, le mercredi en vert, le jeudi en rose, le vendredi en violet.

La méthodologie que nous allons exposer s'appuie principalement sur les travaux de Launay et al., 2015. On appelle $\mathcal{L}$ le jeu de données long et $\mathcal{C}$ le jeu de données court. Nous proposons de construire le prior informatif $\pi^{\mathcal{C}}$ pour le jeu de données $\mathcal{C}$, à partir de la loi a posteriori $\pi^{\mathcal{L}}(\bullet|X^{\mathcal{L}})$. Cette dernière est estimée sur le jeu de donnée $\mathcal{L}$. Sachant que les observations $X^{\mathcal{L}}$ sont nombreuses, on suppose que la loi a posteriori $\pi^{\mathcal{L}}(\bullet|X^{\mathcal{L}})$ a eu le temps de converger vers sa loi asymptotique et que les performances du modèle en prior non-informatif sont satisfaisantes. On choisit ensuite de modéliser la loi a priori $\pi^{\mathcal{C}}$ par une loi Gaussienne. Afin de rendre cette loi a priori informative nous imposons que sa moyenne et sa variance s'appuient sur la moyenne $\mu^{\mathcal{L}}$ et la variance $\Sigma^{\mathcal{L}}$ de la loi a posteriori $\pi^{\mathcal{L}}(\theta|X^{\mathcal{L}})$ sur le jeu de données long. On introduit la matrice

$K = k\,\mathrm{Id}$ avec k une variable aléatoire centré autour de la valeur 1. La loi a priori informative proposée par Launay et al., 2015 est de la forme suivante

$$\theta|k, r \sim \mathcal{N}(k\mu^{\mathcal{L}}, r\Sigma^{\mathcal{L}}), \tag{5.2}$$
$$k|m, l \sim \mathcal{N}(m(1, \ldots, 1)', l\,\mathrm{Id}). \tag{5.3}$$

Les lois a priori des hyperparamètres r, m et l sont définies par

$$\begin{aligned} r &\sim \mathcal{N}(a_r, b_r), \\ m &\sim \mathcal{N}(1, \sigma_m^2), \\ l &\sim \mathcal{N}(1, \sigma_l^2), \end{aligned}$$

avec a_r et b_r des réels positifs choisis afin que la loi a priori de r soit non-informative. Au final, le prior informatif est défini par

$$\pi(\theta, k, r, m, l) \sim \pi(\theta|k, r)\pi(k|m, l)\pi(m)\pi(r)\pi(l)\pi(\sigma_m^2)\pi(\sigma_l^2), \tag{5.4}$$

avec

$$\begin{aligned} &\pi(\sigma_l^2) \propto \sigma_l^{-2}, \\ &\pi(\sigma_m)^2 \propto \sigma_m^{-2}, \\ &\pi(l) \propto |\sigma_l^{-2}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\sigma_l^{-2}(l-1)^2\right), \\ &\pi(m) \propto |\sigma_m^{-2}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\sigma_m^{-2}(m-1)^2\right), \\ &\pi(r) \propto |b_r|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}b_r^{-1}(r-a_r)^2\right), \\ &\pi(k|m, l) \propto |s|^{-\frac{d}{2}} \exp\left(-\frac{1}{2}l^{-1}\sum_{i=1}^{d}(k_i - m)^2\right), \\ &\pi(\theta|k, r) \propto |r|^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(\theta - K\mu^{\mathcal{L}})' r^{-1}(\Sigma^{\mathcal{L}})(\theta - K\mu^{\mathcal{L}})\right). \end{aligned}$$

La Figure 5.3 représente de manière schématique la méthode proposée pour la construction du prior informatif.

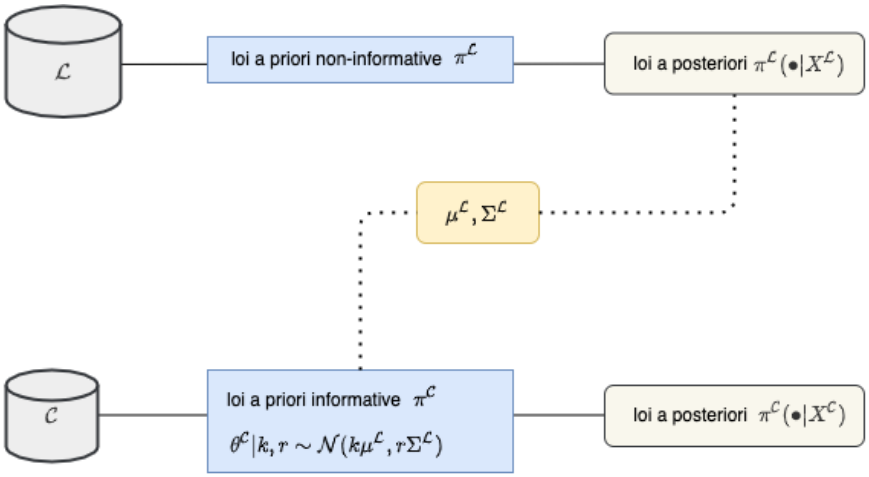


Figure 5.3 – Schéma de la méthode proposée pour la construction du prior informatif en situation d'historique court.

5.3 Description du modèle de prédiction des flux et temps d'attente

Nous allons construire le prior informatif sur un modèle de flux et de temps d'attente dérivé de celui présenté dans le Chapitre 4. Comme au Chapitre 4 le modèle est constitué de deux étages. Une illustration de la structure du modèle est donnée par la Figure 5.4.

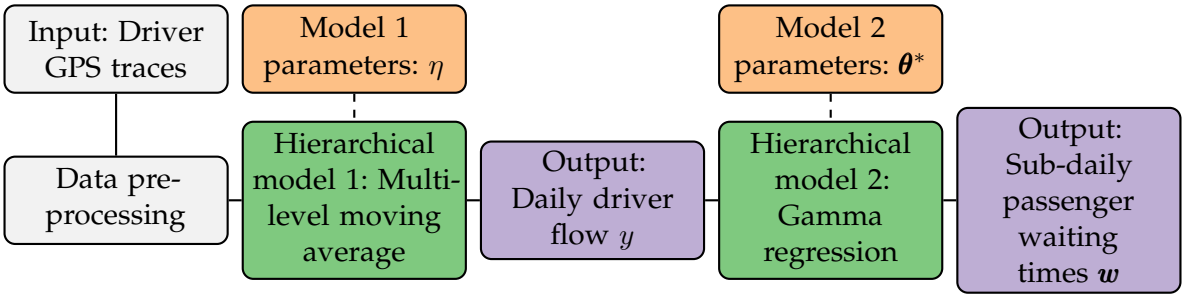


Figure 5.4 – Schéma du modèle hiérarchique bayésien pour la prédiction du flux de conducteurs et du temps d'attente des passagers. Les données d'entrée (traces GPS des conducteurs) sont en gris, les modèles hiérarchiques en vert, les paramètres du modèle en orange et les sorties du modèle en violet.

Le premier étage modélise les flux quotidiens sur une ligne de covoiturage. Il est plus simple que celui proposé au Chapitre 4, aussi il permet une meilleure

interprétation des résultats. Ce premier étage est défini par le modèle

$$y_i = \sum_{k=1}^{K} \eta_{\mathrm{DT}(i-k),\mathrm{DT}(i)} y_{i-k} + \varepsilon_i \tag{5.5}$$

avec ε l'erreur et DT une fonction qui caractérise les types de jours,

$$\mathrm{DT}(i) = \begin{cases} \mathrm{ORD} & \text{si le jour } i \text{ est un jour ordinaire,} \\ \mathrm{SCH} & \text{si le jour } i \text{ est un jour férié scolaire.} \end{cases} \tag{5.6}$$

Le paramètre η représente le coefficient de transition du type de jour qui précède le jour i sur le flux y_i.

Le deuxième étage modélise les temps d'attente par créneaux dans la journée. Il est strictement identique au modèle présenté au Chapitre 4. On rappelle qu'il s'agit d'un modèle de régression Gamma qui permet de modéliser les flux journaliers en fonction des temps d'attente par créneaux. Pour le jour i, nous avons n_i temps d'attente qu'on note $w_{i,1}, \ldots, w_{i,n_i}$ pour des demandes de covoiturages effectuées aux instants $t_{1,i} < \cdots < t_{1,n_i}$. L'intervalle s de la journée est noté I_s. On suppose que les temps d'attente vérifient

$$w_{i,j} | (y_i, \boldsymbol{\beta}, t_{i,j} \in I_s) \sim \Gamma(\nu, \beta_s y_i) \quad \text{pour } i = 1 \ldots N \text{ et } j = 1, \ldots, n_i. \tag{5.7}$$

Le paramètre $\boldsymbol{\beta}$ est un vecteur de taille S. Il répartit les flux quotidiens sur les S différents intervalles de la journée. Le paramètre ν est le premier paramètre de la loi Gamma modélisant les temps d'attente.

On note $\boldsymbol{\theta}$ le vecteur de paramètre des deux étages du modèle. Il est défini

$$\boldsymbol{\theta} = (\eta_{\mathrm{ORD,\,ORD}}, \eta_{\mathrm{SCH,SCH}}, \eta_{\mathrm{ORD,SCH}}, \eta_{\mathrm{SCH,ORD}}, \nu, \beta_1, \ldots, \beta_S, \sigma_\varepsilon^2).$$

Dans la suite nous construisons un prior informatif uniquement pour les paramètres du deuxième étage qui concerne les temps d'attente. On note θ^* le vecteur de taille $S+1$ qui représente les paramètres de la loi Gamma, on obtient

$$\boldsymbol{\theta}^* = (\nu, \beta_1, \ldots, \beta_S). \tag{5.8}$$

5.4 Résultats

5.4.1 Détails de la procédure d'évaluation

La construction du prior informatif est faite suivant l'équation 5.4. De plus, pour les lois a priori des hyperparamètres σ_l^2 et σ_m^2 nous avons utilisé des priors non-informatifs de Jeffreys comme énoncé dans l'introduction. Le paramètre $\boldsymbol{\theta}^*$ est estimé grâce à une méthode de Monte-Carlo par chaînes de Markov

(MCMC). Concernant l'échantillonneur MCMC, nous utilisons l'échantillonneur NUT (Hoffman et al., 2014), qui est implémenté dans le paquet Python pyStan (https://pystan.readthedocs.io). Nous avons utilisé des chaînes MCMC longues pour s'assurer que les lois ont pu converger correctement.

Nous avons évalué les résultats du modèle avec prior informatif et sans prior informatif dans un premier temps sur des données simulées puis sur les données réelles d'Ecov. Dans le cadre de l'évaluation de l'apport du prior informatif on discutera deux aspects i) la qualité de l'estimation des paramètres et ii) les performances prédictives. Plus précisément, la qualité de l'estimation des paramètres va être étudiée dans le cadre de données simulées pour lesquelles les vrais paramètres sont connus. En ce qui concerne les performances prédictives, nous allons évaluer l'apport du prior informatif sur les données réelles issues des lignes de covoiturage.

5.4.2 Jeu de données simulées

Présentation des jeux de données simulées

Le protocole de simulation est celui présenté dans le Chapitre 4 et décrit en Appendice 4.7.2. Les valeurs de ν et $\boldsymbol{\beta}$ sont choisies de manière à obtenir des valeurs cohérentes avec les flux et les temps d'attente observés dans le covoiturage. Le jeu de données $\mathcal{L}$ correspond aux données simulées sur une période de 390 jours avec en moyenne une dizaine d'observations de temps d'attente par intervalles (six intervalles par jour). Le jeu de données court $\mathcal{C}$ correspond aux données simulées sur une période de 90 jours avec en moyenne trois observations de temps d'attente par intervalles. On note $\nu_{\mathcal{L}}$,$\beta_{\mathcal{L}}$ les paramètres de simulation pour le jeu de données $\mathcal{L}$ et $\nu_{\mathcal{C}}$,$\beta_{\mathcal{C}}$ pour le jeu de données $\mathcal{C}$. On choisit $\beta_{\mathcal{C}} = \lambda_1 \beta_{\mathcal{L}}$ et $\nu_{\mathcal{C}} = \lambda_2 \nu_{\mathcal{L}}$ avec $\lambda_1 = 2$ et $\lambda_2 = 1.5$ pour perturber les paramètres initiaux. Le récapitulatif des paramètres de simulation pour chaque jeu de données est détaillé par la Table 5.1.

Jeu de données	ν	β_1	β_2	β_3	β_4	β_5	β_6	Nombre de jours simulés
$\mathcal{L}$	10	0.12	0.1	0.11	0.13	0.18	0.16	390
$\mathcal{C}$	15	0.24	0.2	0.22	0.26	0.36	0.32	90

Table 5.1 – Tableau récapitulatif des paramètres de simulation pour le jeu de données long $\mathcal{L}$ et court $\mathcal{C}$.

Les temps d'attente issus de cette simulation sont représentés sur la Figure 5.5. Le premier jeu de données $\mathcal{L}$ correspond à une ligne de covoiturage ayant un historique de données long. Le deuxième jeu de données $\mathcal{C}$ correspond à la situation de lancement d'une nouvelle ligne de covoiturage avec peu d'observations.

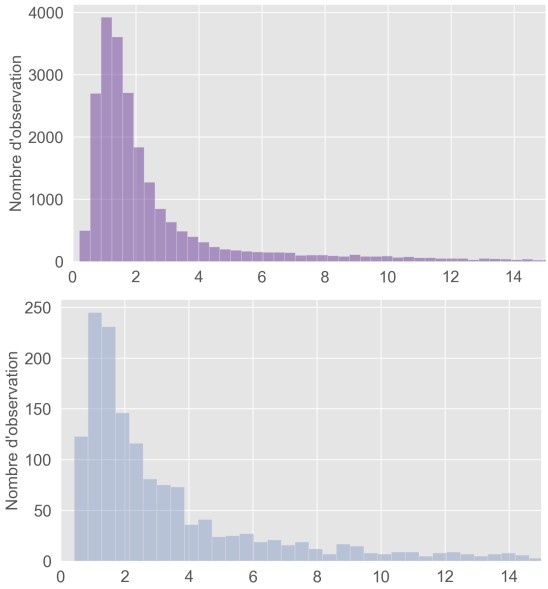


Figure 5.5 – À gauche sont représentés les temps d'attente simulés du jeu de données long $\mathcal{L}$ et à droite les temps d'attente simulés du jeu de données court $\mathcal{C}$.

Résultats : Estimation des paramètres du modèle

La qualité de l'estimation des paramètres du modèle en utilisant le prior informatif, par rapport à l'utilisation du prior non-informatif est visible sur les lois a posteriori des paramètres $\beta_{\mathcal{C}}$ et $\nu_{\mathcal{C}}$. La Figure 5.6 représente les lois a posteriori des paramètres du modèle informatif et non-informatif. Nous constatons que les distributions a posteriori dans le cas du modèle informatif sont moins étendues et plus piquées sur les vraies valeurs des paramètres recherchés. La Table 5.2 compare les écarts-types des paramètres du modèle pour les deux cas (informatif et non-informatif). Les distributions a posteriori des paramètres issues du modèle informatif sont plus précises; le prior informatif facilite l'estimation des paramètres du modèle. Nous rappelons que dans les deux cas, informatif et non-informatif, les lois a posteriori sont issues des chaînes MCMC de même taille, à savoir longue, permettant leurs convergences. De ce fait, nous pouvons nous assurer que dans le cas du modèle non-informatif les lois a posteriori ont pu converger vers leurs lois limites.

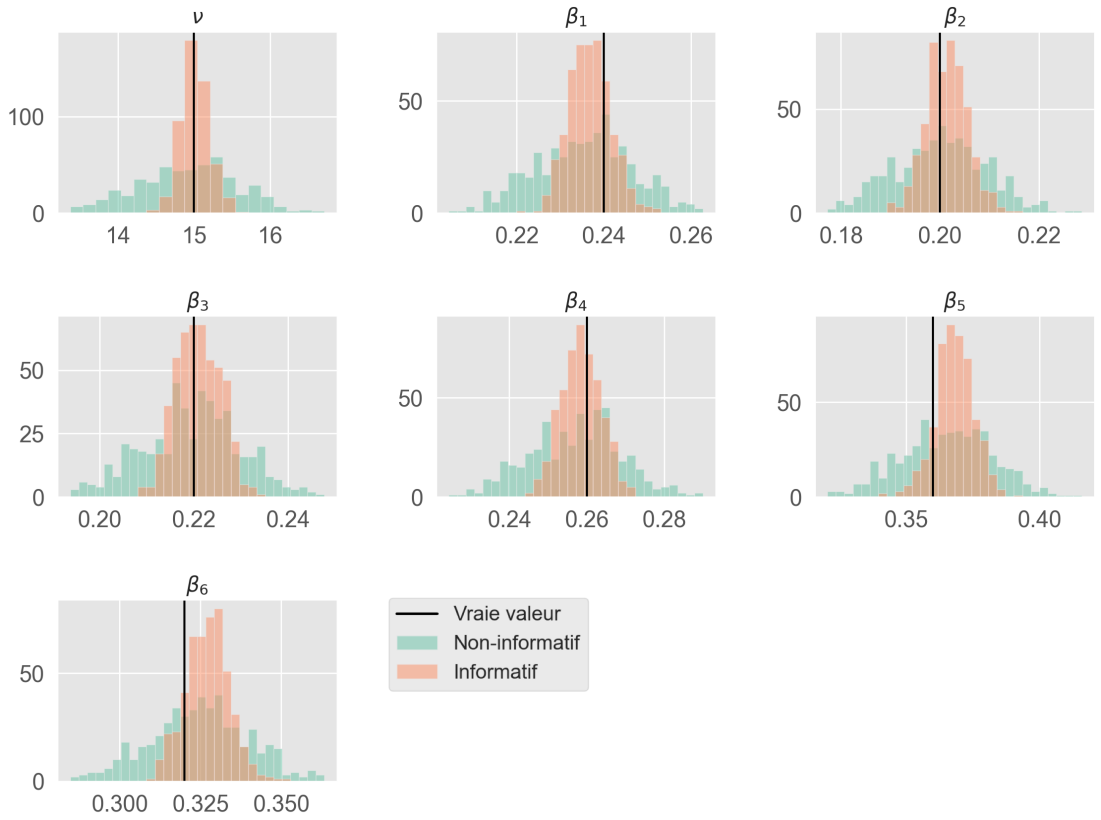


Figure 5.6 – Distributions des lois a posteriori des paramètres du modèle informatif en orange et du modèle non-informatif en vert.

	ν	β_1	β_2	β_3	β_4	β_5	β_6
Non-informatif	0.64	0.011	0.009	0.010	0.011	0.017	0.015
Informatif	0.18	0.004	0.004	0.004	0.004	0.007	0.006

Table 5.2 – Comparaison des écarts-types des lois a posteriori des paramètres du modèle informatif et non-informatif.

Par la suite nous allons évaluer l'apport du prior informatif sur des données réelles d'Ecov. Nous allons évaluer la qualité de l'estimation des paramètres mais aussi les performances prédictives.

5.4.3 Jeu de données d'Ecov

Présentation des jeux de données d'Ecov

Nous nous intéressons maintenant au transfert de prior bayésien pour les données de covoiturage Ecov. Nous allons appliquer la méthode à trois jeux de données qui correspondent à trois scénarios différents. Le premier jeu de

données est le jeu de données long, il s'étale sur une plus grande période d'usage. Ce jeu de données est appelé $\mathcal{A}$. Le deuxième jeu de données $\mathcal{A}'$ correspond à un sous-échantillonnage du jeu de données $\mathcal{A}$. Il s'étale sur une plus petite période et contient à peu près la moitié des observations du jeu de données $\mathcal{A}$. Le troisième et dernier jeu de données $\mathcal{B}$ correspond lui aussi à un jeu de données court mais cette fois les observations proviennent d'une toute nouvelle ligne de covoiturage avec encore peu d'observations. Il s'agit de la situation d'historique court sur un nouveau territoire. Les détails des jeux de données sont résumés dans la Table 5.3 et dans la Figure 5.7.

	Période	Nombre d'observations	Nom
Jeu de données long	2019-07-26 – 2020-02-14	935	$\mathcal{A}$
Jeu de données long sous-échantillonné	2019-11-08 – 2020-01-28	405	$\mathcal{A}'$
Jeu de données court	2019-10-07 – 2020-02-14	405	$\mathcal{B}$

Table 5.3 – Récapitulatif des jeux de données pour les trois différents scénarios.

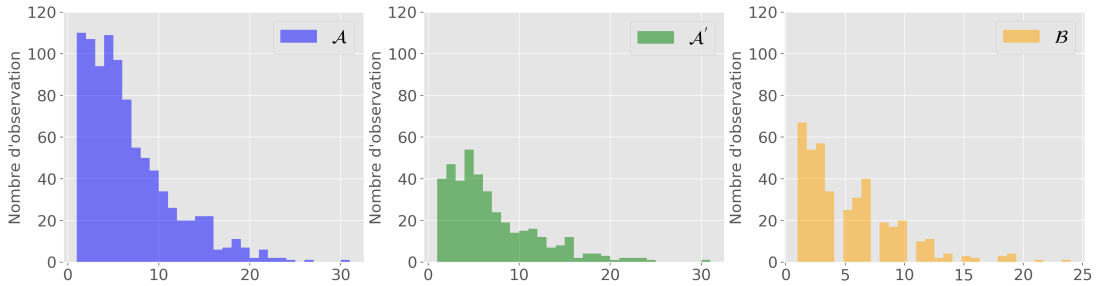


Figure 5.7 – Distributions des temps d'attente observés pour les trois différents jeux de données. En bleu, le jeu de données $\mathcal{A}$, en vert, le jeu de données $\mathcal{A}'$ et en jaune le jeu de données $\mathcal{B}$.

Transfert bayésien

La méthodologie appliquée est la suivante :

1. On commence par ajuster le modèle non-informatif sur le jeu de données $\mathcal{A}$.
2. On construit ensuite le prior informatif comme détaillé en Section 5.2 pour le jeu de données $\mathcal{A}'$.

3. Puis, on construit le prior informatif comme détaillé en Section 5.2 pour le jeu de données $\mathcal{B}$.

L'étape 2 a pour but de vérifier si l'apport du prior informatif est correct car le jeu de données $\mathcal{A}'$ est un sous-échantillon du jeu de données $\mathcal{A}$ tandis que la dernière étape correspond à l'enjeu essentiel de la construction du prior informatif. Effectivement, l'amélioration des performances du modèle sur un nouveau jeu de données, en l'occurrence le jeu de données $\mathcal{B}$ à partir d'un autre jeu de données long, est stratégique d'un point de vue industriel. Nous rappelons que les paramètres du modèle concernés par la construction du prior informatif sont $\boldsymbol{\theta}^* = (\nu, \beta_1, \ldots, \beta_6)$. Ce sont les paramètres du deuxième étage du modèle hiérarchique bayésien détaillé en Section 5.3 modélisant les temps d'attente. Nous avons aussi ajusté le modèle non-informatif sur les deux jeux de données $\mathcal{A}$ et $\mathcal{B}$ de façon à pouvoir comparer l'intérêt du prior informatif. Les lois a posteriori obtenues avec prior non-informatif sont illustrées par la Figure 5.8.

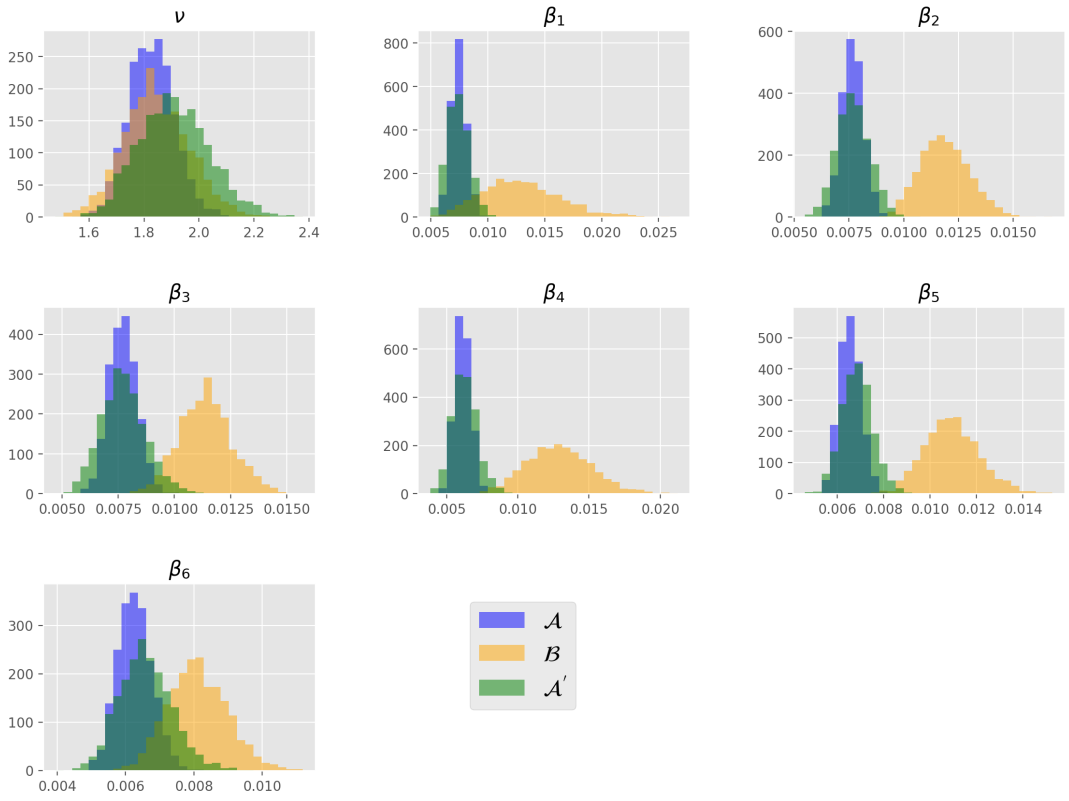


Figure 5.8 – Distributions a posteriori des paramètres du modèle non-informatif concernant les temps d'attente pour chaque jeu de données. En bleu, le jeu de données $\mathcal{A}$, en jaune, le jeu de données $\mathcal{B}$ et en vert, le jeu de données $\mathcal{A}'$.

Résultats : Estimation des paramètres du modèle

Nous remarquons tout d'abord que les lois a posteriori du modèle non-informatif des jeux de données $\mathcal{A}$ et $\mathcal{A}^{'}$ se ressemblent fortement. Néanmoins, celles du jeu de données $\mathcal{A}^{'}$ sont plus étendues que celles du jeu de données $\mathcal{A}$ ce qui est naturel puisque $\mathcal{A}^{'}$ est une sous population de $\mathcal{A}$. Les lois a posteriori des paramètres du modèle non-informatif pour le jeu de données $\mathcal{B}$ sont aussi moins précises et par ailleurs centrées sur des valeurs différentes.

Un des objectifs de la construction du prior informatif est de diminuer l'incertitude autour des distributions a posteriori des paramètres, ce qui va entraîner une amélioration de la précision du modèle. La Figure 5.9 compare les distributions a posteriori des paramètres issues du modèle informatif et non-informatif pour le jeu de données $\mathcal{A}^{'}$. De même, la Figure 5.10 représente la même comparaison mais pour le jeu de données $\mathcal{B}$. Ces résultats confirment que le prior informatif est efficace. En effet, la totalité des lois a posteriori des paramètres du modèle informatif dans les deux cas ($\mathcal{A}^{'}$ et $\mathcal{B}$) sont plus concentrées que celles du modèle non-informatif. On rappelle que, pour les jeux de données $\mathcal{A}^{'}$ et $\mathcal{B}$, nous n'avons pas accès aux vraies valeurs a posteriori des paramètres. Sachant que les résultats sur les données simulées indiquent que le modèle informatif est plus efficace, on peut espérer que les estimations des répartitions depuis le prior informatif ont un biais comparable à celles provenant du modèle non-informatif. Nous allons voir dans la section suivante comment cette amélioration se traduit sur la performance des lois prédictives a posteriori.

Résultats : Performances prédictives

Le deuxième objectif de la construction du prior informatif est d'améliorer les performances prédictives du modèle. Rappelons qu'en statistique bayésienne nous pouvons estimer la loi prédictive a posteriori. Ceci nous donne la possibilité d'évaluer le gain du modèle informatif par rapport au non-informatif pour différents niveaux de quantiles de la loi prédictive. Il est en effet intéressant ici d'évaluer cet apport en terme de quantiles pour deux raisons. La première raison est que dans le domaine du covoiturage nous nous intéressons à prédire les quantiles de loi des temps d'attente (cf. Chapitre 2). La deuxième raison est que l'amélioration des performances prédictives est largement plus visible au niveau des valeurs des quantiles élevés. Les moyennes des lois a posteriori prédictives du modèle informatif et du modèle non-informatif sont similaires. En revanche, les quantiles des deux lois se distinguent car l'information apportée par le prior informatif rend la première plus précise. Afin de quantifier le gain du modèle informatif nous avons recours au score PE. Ce score, introduit dans cette thèse, est adapté pour le covoiturage. Il mesure le pourcentage de bonnes

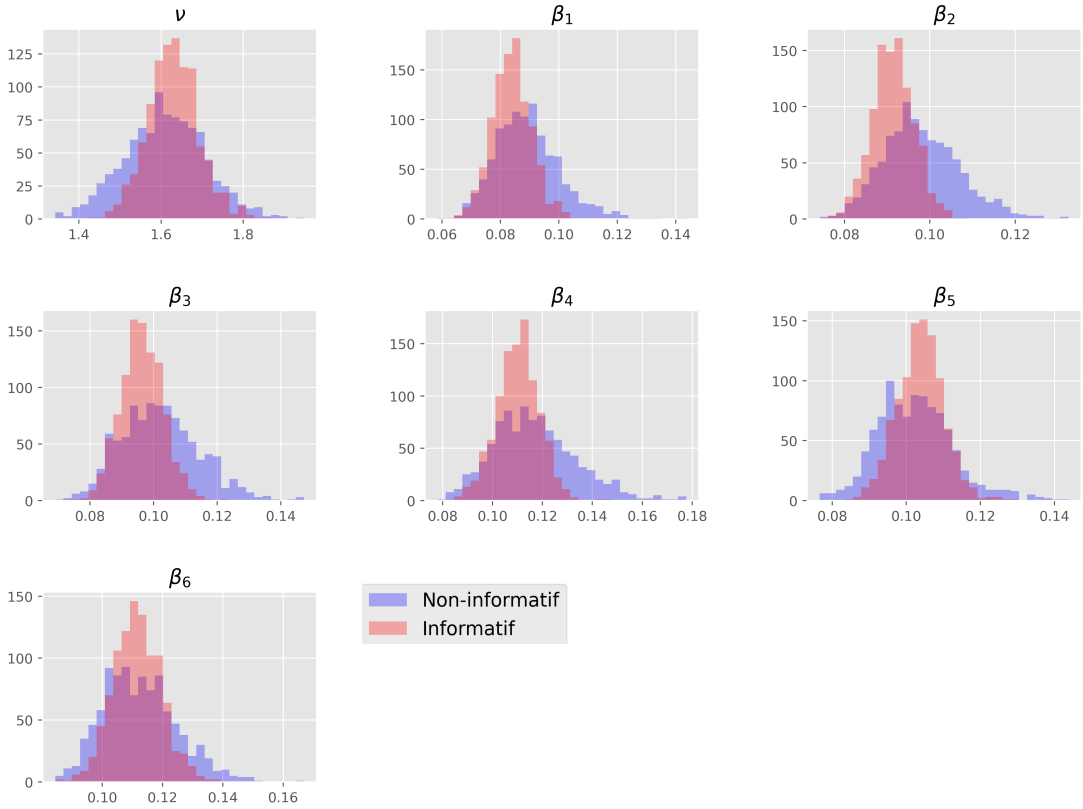


Figure 5.9 – Histogrammes des lois a posteriori des paramètres du modèle de temps d'attente pour le jeu de données $\mathcal{A}'$. En bleu, les lois a posteriori avec un prior non-informatif et en rouge, les lois a posteriori avec un prior informatif.

prédictions en dessous d'un seuil δ (en minutes) donné. Son expression est

$$\mathrm{PE}(\delta) = \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} \mathbf{1}\{|q_\alpha(\tilde{w}_i) - w_i| < \delta\} \tag{5.9}$$

avec $q_\alpha(\tilde{w}_i)$ est le quantile au niveau α de la loi prédictive a posteriori de la $i^{ième}$ observation et w_i le $i^{ième}$ temps d'attente observé pour un échantillon de taille $\tilde{N}$. Le niveau du quantile choisi pour notre comparaison est $\alpha = 95\%$. Ce niveau avait été retenu lors de l'industrialisation du modèle parce qu'il permettait d'assurer un bon niveau de service à nos utilisateurs. La Figure 5.11 compare les scores PE du quantile niveau 95% issues du modèle informatif et non-informatif pour le jeu de données $\mathcal{A}'$. La Figure 5.12 représente la même comparaison mais pour le jeu de données $\mathcal{B}$. Nous constatons que dans le cas du transfert de $\mathcal{A}$ vers $\mathcal{A}'$ comme dans le cas du transfert $\mathcal{A}$ vers $\mathcal{B}$ le score PE du modèle informatif est meilleure. Ce résultat est d'autant plus visible sur le jeu de données $\mathcal{B}$ ce qui montre que la loi a priori informative apporte une réelle

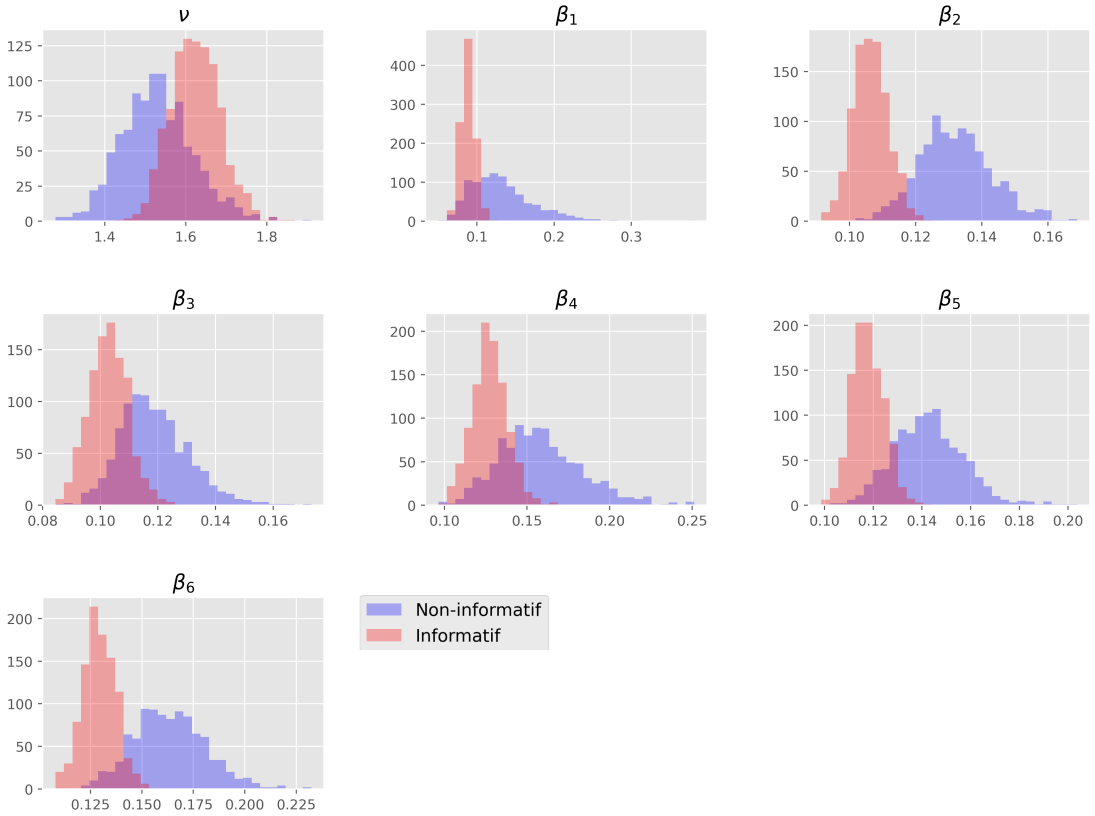


Figure 5.10 – Histogrammes des lois a posteriori des paramètres du modèle de temps d'attente pour le jeu de données $\mathcal{B}$. En bleu, les lois a posteriori avec un prior non-informatif et en rouge, les lois a posteriori avec un prior informatif.

information supplémentaire. Effectivement, dans le cas du jeu de données $\mathcal{B}$, nous observons que la totalité des prédictions du quantile posterior issues du modèle informatif, au niveau 95%, ne dépassent pas une erreur de plus de 9 minutes. Par contre, pour le même seuil de 9 minutes, le quantile posterior du modèle non-informatif atteint seulement 40% de ses prédictions. Cette nette amélioration illustre l'intérêt et l'efficacité du prior informatif sur un jeu de données court.

Une autre façon de résumer la loi prédictive a posteriori des temps d'attente est de calculer son intervalle de plus haute densité, HDI (Highest Density Interval). Un point se trouvant dans cet intervalle a une crédibilité plus élevée que tout point se trouvant à l'extérieur de ce même intervalle. Nous fixons donc un niveau de crédibilité à 94% et comparons les intervalles de plus haute densité des lois a posteriori prédictives du modèle informatif et non-informatif. De cette manière nous étudions les performances prédictives des deux modèles. Cette comparaison doit au préalable s'accompagner d'une étude sur les scores PE

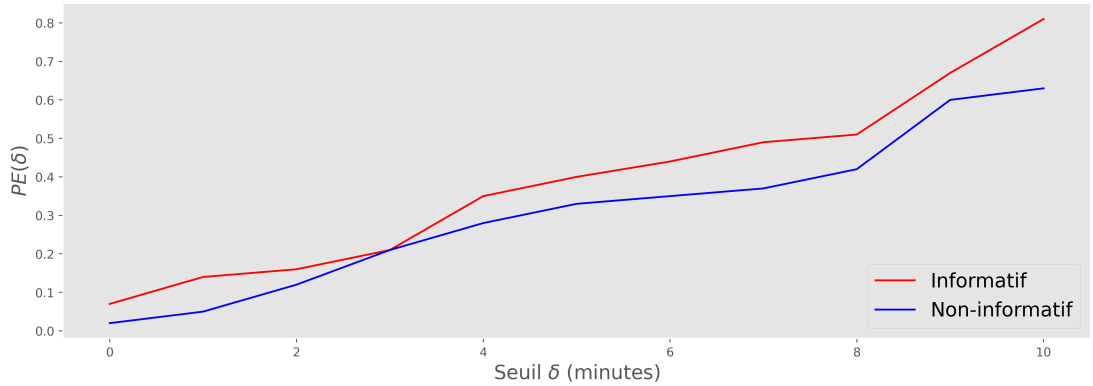


Figure 5.11 – Comparaison du score PE du quantile posterior 95% pour le jeu de données $\mathcal{A}^{'}$. En bleu le score PE pour le modèle avec prior non-informatif et en rouge en avec prior informatif.

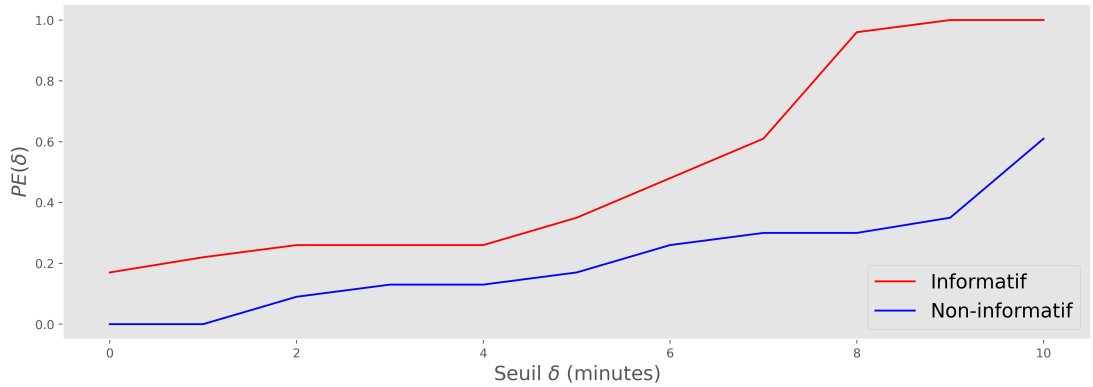


Figure 5.12 – Comparaison du score PE du quantile posterior 95% pour le jeu de données $\mathcal{B}$. En bleu le score PE pour le modèle avec prior non-informatif et en rouge en avec prior informatif.

comme établie ci-dessus. En effet, une loi prédictive a posteriori plus précise, c'est à dire plus courte n'est pas nécessairement plus performante qu'une loi prédictive a posteriori plus longue. Si la loi plus courte est mal centrée on obtiendra un score PE moins bon que dans le cas de la loi plus longue cependant bien centrée. Comme nous l'avons expliqué pour les Figures 5.11 et 5.12 nous avons éliminé cette dérive car le modèle informatif a des meilleurs résultats d'un point vue PE que le modèle non-informatif. Au final on peut donc affirmer que les intervalles de plus haute densité des lois posteriori prédictives des temps d'attente en modèle informatif sont plus petits que ceux du modèle non-informatif. Ceci implique sans ambiguïté que le modèle informatif est plus performant sur ses tâches prédictives. En particulier, les Figures 5.13 et 5.14 comparent les intervalles de plus haute densité des lois prédictives en

informatif et non-informatif pour les jeux de données $\mathcal{A}'$ et $\mathcal{B}$ respectivement. Cette comparaison est faite pour les six intervalles de la première journée de prédiction. Comme dans le cas du score PE l'amélioration des performances est nettement plus grande pour le jeu de données $\mathcal{B}$. Notons que pour le premier jour de prédiction du jeu de données $\mathcal{B}$ nous obtenons un gain de 4 minutes en moyenne sur les lois a posteriori prédictives des temps d'attente. Nous résumons à la Table 5.4 ce gain en pourcentage de diminution des intervalles de plus haute densité des lois a posteriori prédictives provenant du modèle informatif.

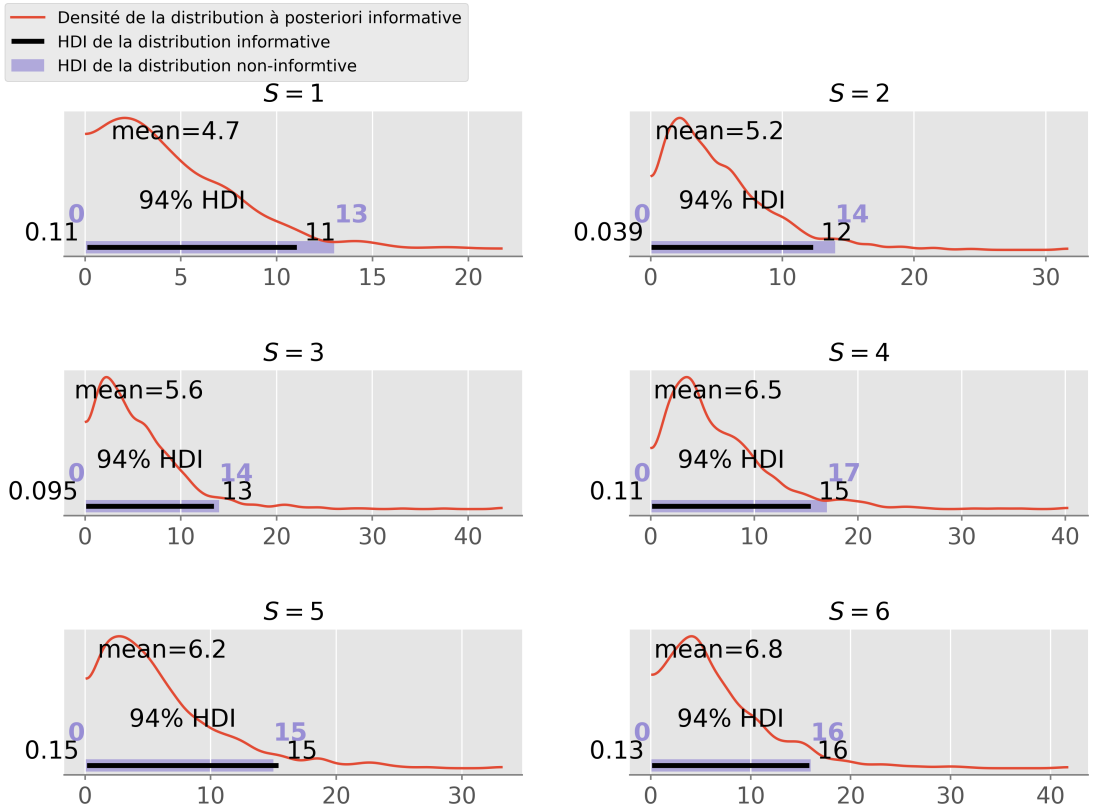


Figure 5.13 – Comparaison des lois a posteriori prédictives des temps d'attente pour le premier jour de prédiction pour les six créneaux du dataset $\mathcal{A}'$. La courbe en rouge est la loi prédictive lorsque le prior est informatif et en noir sa région HDI (Highest Density Interval). En violet, la région HDI de la loi prédictive des temps d'attente en non-informatif.

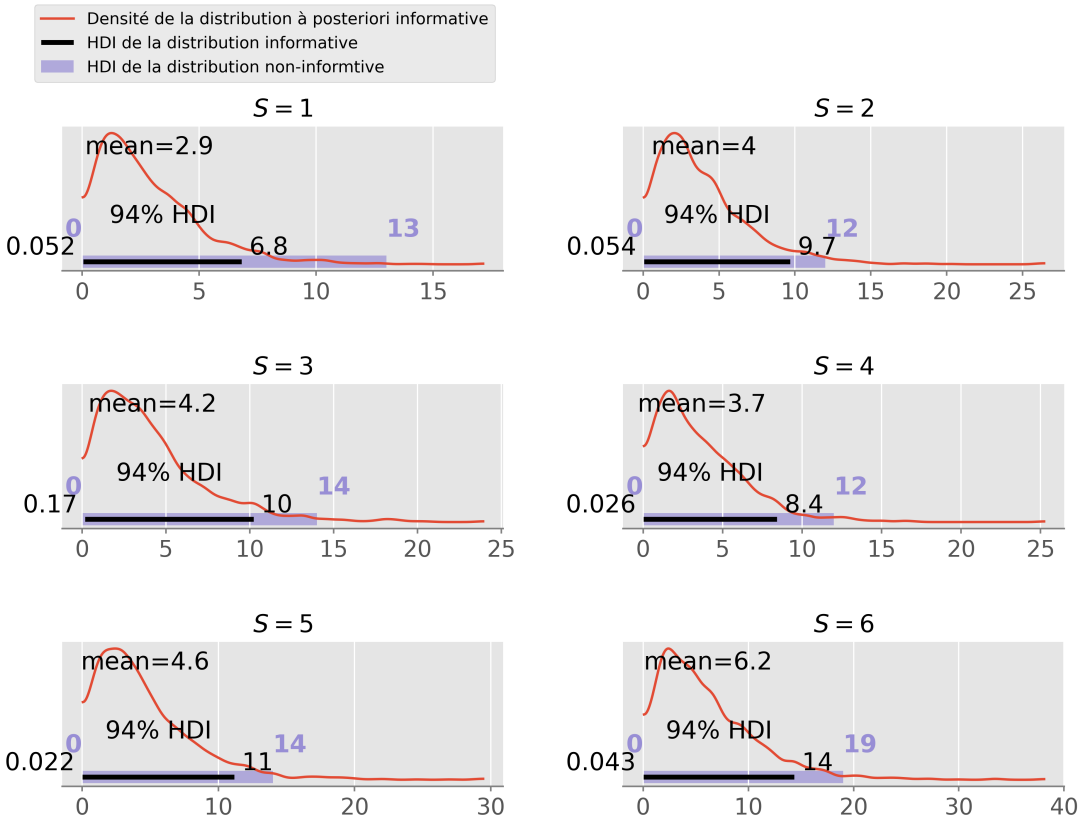


Figure 5.14 – Comparaison des lois a posteriori prédictives des temps d'attente pour le premier jour de prédiction pour les six créneaux du dataset $\mathcal{B}$. La courbe en rouge est la loi prédictive lorsque le prior est informatif et en noir sa région HDI (Highest Density Interval). En violet, la région HDI de la loi prédictive des temps d'attente en non-informatif.

S	1	2	3	4	5	6
$\mathcal{A}'$	16%	14%	8%	12%	1%	1%
$\mathcal{B}$	48%	20%	30%	30%	22%	26%

Table 5.4 – Pourcentages de diminution de l'intervalle de plus haute densité des lois prédictives a posteriori issues du modèle informatif sur les six créneaux du premier jour de prédiction.

5.5 Conclusions

Dans ce chapitre nous avons développé une méthode pour la construction d'une loi a priori informative par transfert bayésien dans le cas d'un jeu données court.

Cette construction est faite à partir d'un jeu de données long $\mathcal{L}$. Puis, le modèle informatif est utilisé sur un jeu de données court $\mathcal{C}$, qui est supposé ressembler au jeu de données long. Nous avons montré, à l'aide de données simulées et réelles, que l'apport de cette loi a priori informative améliore la qualité de l'estimation des paramètres du modèle ainsi que les performances prédictives.

Cette méthode proposée repose sur l'incorporation des informations apprises par le modèle sur le jeu de données long $\mathcal{L}$ à la loi a priori du modèle sur le jeu de données court $\mathcal{C}$. Une perspective possible des travaux de ce chapitre est l'utilisation d'autres données disponibles, externes à celles utilisées actuellement par le modèle, afin de mieux construire le prior informatif. En effet, Cucchi et al., 2019 ont proposé de mutualiser des données de plusieurs sites différents pour renforcer la qualité du prior. Nous pensons que cette perspective, en parallèle avec notre méthode, pourra améliorer davantage encore les performances du prior informatif.

CHAPTER 6

VALORISATION INDUSTRIELLE : MISE EN PRODUCTION DU MODÈLE HIÉRARCHIQUE BAYÉSIEN

6.1 Introduction

Ecov propose aux collectivités plusieurs solutions de mobilité comme par exemple la solution *Lane* https://www.lanemove.com/. Il s'agit d'un service de covoiturage structuré en ligne se situant au Sud-Est de Lyon. Afin d'améliorer l'expérience des passagers, la connaissance des flux de trafic et des temps d'attente est essentielle. Nous rappelons que tout au long de cette thèse nous avons développé des outils et des modèles dans ce but. L'objet de ce dernier chapitre est de montrer les différentes étapes et enjeux pour mettre en production le modèle hiérarchique bayésien développé dans le Chapitre 4 permettant la prédiction des flux de trafic et de temps d'attente.

L'étape de la mise en production du modèle va donner la possibilité à la société Ecov de proposer un service de covoiturage plus complet. En effet, les informations concernant le flux de trafic et les temps d'attente vont alimenter les différents outils d'information et de synthèse disponibles de la société. Ensuite, chaque outil expose ces informations de manière à répondre aux différents besoins que ce soit des passagers ou des différentes équipes au sein d'Ecov. Dans les faits, un passager peut s'informer de son temps d'attente et du potentiel de covoiturage à plusieurs moments. Par exemple, la veille d'un covoiturage le passager peut consulter ces informations à l'aide du site internet et de l'application mobile d'Ecov. Également, au moment même du covoiturage le temps d'attente est affiché sur la borne d'informations des voyageurs (BIV) qui se trouve à l'arrêt de covoiturage. De plus, l'équipe "gestion relation utilisateur" d'Ecov qui accompagne les passagers lors d'un covoiturage a lui aussi besoin de connaître les temps d'attente et les flux de trafic potentiel. Dans ce cas, l'équipe "gestion relation utilisateur" consulte un logiciel de tableau de bord. C'est dans le but de répondre à ces besoins que nous avons développé un paquet en Python appelé *bayesian-hierarchical-model*. Ce paquet permet la collecte des données, le pré-traitement des données, l'apprentissage du modèle et l'exposition des résultats du modèle vers les différents canaux énumérés précédemment (site

internet, application mobile, la BIV, logiciel de tableau de bord).

Le paquet *bayesian-hierarchical-model* a été conçu à partir des outils développés dans ce manuscrit et constitue un aboutissement de ces travaux de thèse. Nous allons décrire les différentes étapes et la structure du paquet ainsi que l'automatisation de son utilisation.

6.2 Description des données et du modèle à industrialiser

Rappelons tout d'abord que les données utilisées par le modèle proviennent de deux sources distinctes i) les traces GPS des conducteurs et ii) les temps d'attente observés des passagers. Nous rappelons aussi que le modèle développé pour la prédiction des flux de trafic et des temps d'attente est composé de deux étages imbriqués. Comme pour le Chapitre 5, nous travaillons sur un modèle dérivé de celui présenté au Chapitre 4. Le premier étage modélise les flux quotidiens sur une ligne de covoiturage. Ce premier étage est défini par le modèle

$$y_i = \sum_{k=1}^{K} \eta_{\mathrm{DT}(i-k),\mathrm{DT}(i)} y_{i-k} + \varepsilon_i \tag{6.1}$$

avec ε l'erreur et DT une fonction qui caractérise les types de jours,

$$\mathrm{DT}(i) = \begin{cases} \mathrm{ORD} & \text{si le jour } i \text{ est un jour ordinaire,} \\ \mathrm{SCH} & \text{si le jour } i \text{ est un jour férié scolaire.} \end{cases} \tag{6.2}$$

Le paramètre η représente le coefficient de transition du type de jour qui précède le jour i sur le flux y_i.

Le deuxième étage modélise les temps d'attente par créneaux dans la journée. Il est strictement identique au modèle présenté au Chapitre 4. On rappelle qu'il s'agit d'un modèle de régression Gamma modélisant les flux journaliers en fonction des temps d'attente par créneaux. Pour le jour i, nous avons n_i temps d'attente qu'on note $w_{i,1}, \ldots, w_{i,n_i}$ pour des demandes de covoiturages effectuées aux instants $t_{1,i} < \cdots < t_{1,n_i}$. L'intervalle s de la journée est noté I_s. On suppose que les temps d'attente vérifient

$$w_{i,j} | (y_i, \boldsymbol{\beta}, t_{i,j} \in I_s) \sim \Gamma(\nu, \beta_s y_i) \quad \text{pour } i = 1 \ldots N \text{ et } j = 1, \ldots, n_i. \tag{6.3}$$

Le paramètre $\boldsymbol{\beta}$ est un vecteur de taille S et il répartit les flux quotidiens sur les S différents intervalles de la journée. Le paramètre ν est le premier paramètre de la loi Gamma modélisant les temps d'attente.

6.3 Industrialisation du modèle hiérarchique bayésien

6.3.1 Les logiciels et outils informatiques

Le passage d'un modèle prédictif expérimental vers une version stable et industrielle de celle-ci nécessite plusieurs pratiques et outils informatiques. Durant la phase de conceptualisation du modèle, nous construisons un modèle prototype, dans ce cas, il fonctionne localement sur un Jupyter notebook (Kluyver et al., 2016). Ensuite, pour la phase d'industrialisation du modèle nous avons développé un paquet en Python autoportant de manière à ce qu'il soit exécuté automatiquement à des moments clés (tous les matins, à chaque demande de covoiturage). Nous avons cartographié à la Figure 6.1 les outils informatiques principaux nécessaires pour les deux phases.

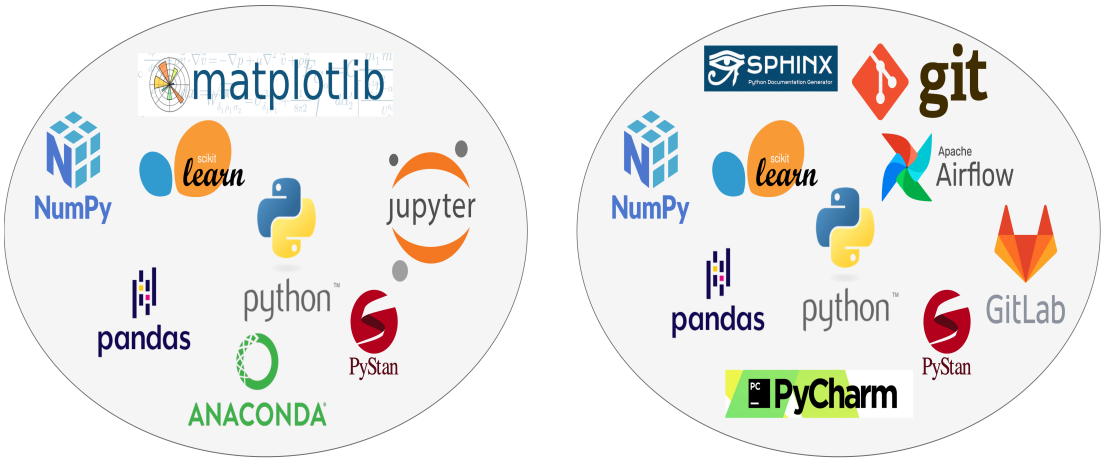


Figure 6.1 – Outils principaux utilisés pour le prototypage du modèle (à gauche) et pour son industrialisation (à droite).

Plusieurs de ces outils sont communs pour les deux phases comme le langage Python (Van Rossum et al., 1995) et la bibliothèque PyStan (Stan Development Team, 2012). Cette dernière est une interface en Python pour le langage Stan qui est un langage de programmation probabiliste pour l'inférence statistique bayésienne. De plus, la bibliothèque Pandas (The pandas development team, 2020) permet la manipulation et l'analyse des données ainsi que NumPy (Harris et al., 2020) qui a pour but de manipuler des matrices et des tableaux multidimensionnels ainsi que des fonctions mathématiques. Nous avons aussi emprunté des fonctions développées dans la bibliothèque Scikit-Learn (Pedregosa et al., 2011) qui est dédiée à l'apprentissage automatique.

Dans le cadre du prototypage du modèle nous avons eu besoin d'utiliser

des outils de gestion des différents paquets Python. Anaconda (Anaconda Software Distribution, 2020) permet la gestion de différents paquets proposant directement un environnement où la plupart des paquets populaires en science des données sont pré-installés. Un autre outil est la bibliothèque Matplotlib (Hunter, 2007) pour visualiser les données sous formes de graphiques. Cette liste n'est pas exhaustive mais donne une vue globale sur les différents outils utiles lors du prototypage d'un modèle d'apprentissage statistique.

Concernant l'industrialisation du modèle, des outils de gestion des versions du code tel que le Git (Chacon et al., 2014) et GitLab[1] sont utilisés. Ils donnent la possibilité de garder un historique de l'évolution du code et de mieux partager le code entre collaborateurs. De plus, plusieurs bonnes pratiques existent afin de bien utiliser ce genre d'outils, il s'agit d'un sujet important pour la bonne organisation des équipes de développement. Ensuite, un autre outil incontournable est PyCharm[2] qui est un environnement de développement intégré pour le langage Python. Il permet de faciliter les bonnes pratiques de développement en offrant un environnement hautement configurable et contient une extension pratique de débogage graphique. Au moment du déploiement du modèle nous avons fait le choix d'utiliser une plateforme de planification de flux de travail qui s'appelle Apache Airflow[3]. La dernière étape à ne pas négliger est la documentation qui doit être claire et à jour, pour cela nous avons fait le choix d'utiliser Sphinx[4] qui est un générateur de documentation.

Il est important de noter que le prototypage du modèle est un travail de recherche où les aspects statistiques du modèle sont privilégiés ainsi que les performances prédictives. Tandis que pour l'industrialisation du modèle, on s'intéresse à la qualité du code, la rapidité d'exécution du modèle et l'automatisation de son entraînement et de son déploiement.

6.3.2 La structure du paquet

Le paquet développé s'appelle *bayesian-hierarchical-model* et est entièrement écrit en Python suivant les standards des paquets utilisés en apprentissage automatique. Nous avons emprunté la structure des modèles implémentés dans la bibliothèque Scikit-Learn (Pedregosa et al., 2011). Cela signifie que nous avons développé une méthode nommée ".fit()" afin d'entraîner le modèle sur les données d'entraînement et une méthode ".predict()" permettant la prédiction du modèle sur les nouvelles données. Nous avons aussi implémenté plusieurs autres méthodes et fonctions concernant la récupération et le pré-traitement des données et la simulation des données jouets. La structure du paquet, illustrée

1. https://about.gitlab.com/
2. https://www.jetbrains.com/fr-fr/pycharm/
3. https://airflow.apache.org/
4. https://www.sphinx-doc.org/en/master/

par son arborescence en dossier à la Figure 6.2, est divisée en cinq parties.

1. Les fichiers "README.rst" et "README_files" contiennent des informations importantes sur le projet pour les futurs utilisateurs. Par convention, il est placé au tout premier niveau de l'arborescence pour que les utilisateurs trouvent immédiatement les informations.

2. Le dossier *bayeshierarchical* contient les fichiers Python et Stan qui correspondent aux différents modules du modèle. Le fichier "__init__.py" permet l'interprétation des répertoires en tant que modules, il est toujours chargé en premier dans un module.

3. Le dossier *doc* contient tous les fichiers textes qui constituent la documentation du paquet. Ce répertoire est généré automatiquement par l'outil Sphinx.

4. Le fichier "setup.py" est indispensable pour la bonne construction et distribution d'un paquet Python. En effet, il est responsable de la transmission des instructions indispensables à la construction du paquet tels que le nom, la version et les dépendances.

5. Le dossier *tests* contient les fichiers Python avec les tests unitaires afin de vérifier le bon fonctionnement des différentes méthodes et fonctions. L'exécution du modèle est aussi testée sur des données artificielles dans le Jupyter Notebook "Tests.ipynb".

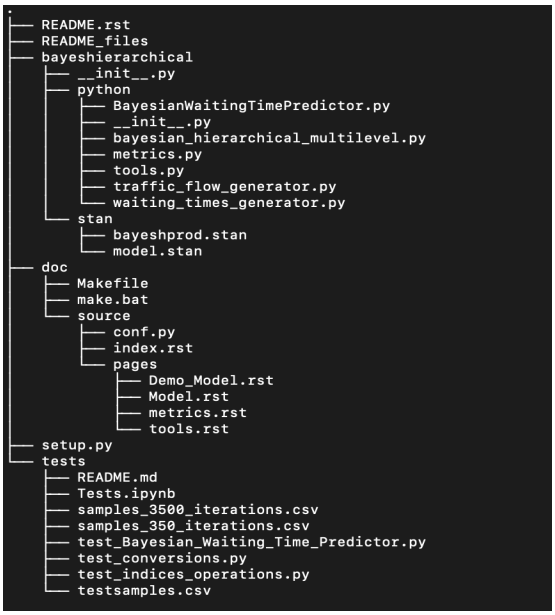


Figure 6.2 – La structure du paquet *bayesian-hierarchical-model*, vue sous forme d'arborescence affichant les dossiers et fichiers présent du paquet.

Après avoir détaillé la structure du paquet nous illustrons son utilisation par des exemples concrets.

6.3.3 Utilisation du paquet *bayesian-hierarchical-model*

Simulation de flux de trafic. L'utilisateur du paquet a la possibilité de générer des données simulées selon le protocole de simulation détaillé dans le Chapitre 4. La fonction développée pour cela s'appelle *TrafficFlowGenerator*. Les arguments nécessaires sont le nom de la colonne contenant les dates ("date_col"), la période concernée pour la simulation en donnant la date de début et de fin ("start" et "end") au format "aaaa-mm-jj", le pays concerné ("country") et la zone souhaitée ("zone") quand cela s'applique pour le pays choisi. Nous devons préciser le pays ainsi que la zone voulue pour permettre au simulateur de prendre en compte l'impact des types de jour sur le trafic. L'exemple suivant illustre l'utilisation de cette fonction pour la simulation de trafic de flux pour la période "2019-01-01" à "2020-01-01" en France dans la zone "A" (Poitiers, Bordeaux, Limoges, Clermont-Ferrand, Lyon, Grenoble, Dijon, Besançon). Les données obtenues sont illustrées par la Figure 6.3.

```
from bayeshierarchical.python.traffic_flow_generator
    import TrafficFlowGenerator

t = TrafficFlowGenerator(date_col = "Date",
                         start = "2019-01-01",
                         end = "2020-01-01",
                         country = "FRA",
                         zone= "A")

simulated_traffic  = t.generate_flow()
```

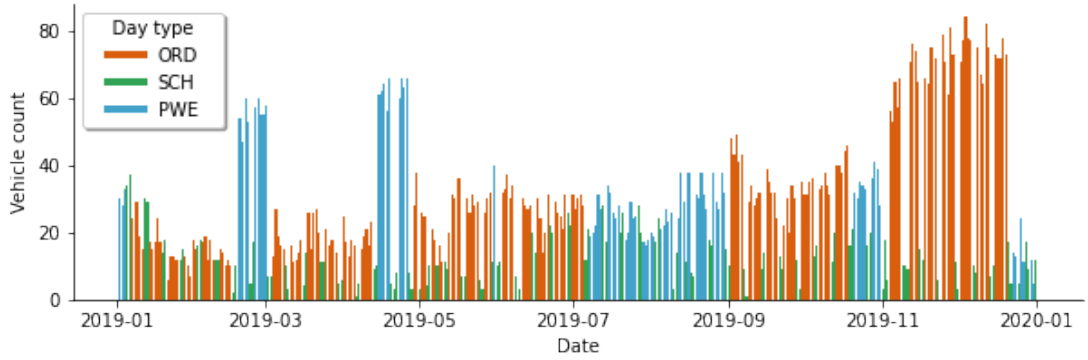


Figure 6.3 – Exemple de trafic de flux généré par la fonction *TrafficFlowGenerator* du paquet *bayesian-hierarchical-model*.

Simulation des temps d'attente. La simulation des temps d'attente, elle aussi suit le protocole de simulation décrit dans le Chapitre 4. La fonction dédiée à cela est *WaitingTimesGenerator*. Les arguments nécessaires sont les mêmes que ceux de la fonction *TrafficFlowGenerator* avec un argument supplémentaire qui est le nombre de temps d'attente à simuler par créneaux ("wt_num"). Par défaut, une journée est divisée en huit créneaux, les quatre premiers représentent des créneaux d'une demi-heure durant la période d'ouverture du service du matin et les quatre derniers pour la période d'ouverture du service du soir. L'exemple suivant illustre l'utilisation de cette fonction pour la simulation de vingt temps d'attente par créneaux pour la période "2018-01-01" à "2018-11-30" en France dans la zone "A" (Poitiers, Bordeaux, Limoges, Clermont-Ferrand, Lyon, Grenoble, Dijon, Besançon). Les données obtenues sont tabulaires, nous représentons un échantillon de celles-ci à la Figure 6.4.

```
from bayeshierarchical.python.waiting_times_generator
    import WaitingTimesGenerator

w = WaitingTimesGenerator(date_col="Date",
                          start="2018-01-01",
                          end="2018-11-30",
                          country="FRA",
                          zone="A",
                          wt_num = 20)

simulated_wt = w.generate_waiting_times()
```

	inter_1	inter_2	inter_3	inter_4	inter_5	inter_6	inter_7	inter_8	Date	day_type	flow
0	3.4	2.0	3.0	3.3	3.8	2.5	4.0	3.7	2018-01-02	3	28
1	6.0	2.1	2.1	3.2	3.0	2.6	3.4	3.6	2018-01-02	3	28
2	2.6	2.0	2.2	1.7	5.1	3.5	4.0	4.1	2018-01-02	3	28
3	3.7	2.9	2.9	4.6	4.2	3.0	3.6	4.1	2018-01-02	3	28
4	2.3	1.5	4.7	5.0	4.6	3.3	7.4	3.6	2018-01-02	3	28

Figure 6.4 – Exemple des temps d'attente générés par la fonction *WaitingTimesGenerator* du paquet *bayesian-hierarchical-model*.

Construction du modèle sur les données simulées L'objet *BayesianHierarchicalMultilevel()* permet la construction du modèle, il est également lié aux fonctions *TrafficFlowGenerator()* et *WaitingTimesGenerator()* dans le cas de l'utilisation du modèle sur des données simulées. L'exemple ci-dessous illustre comment construire le modèle sur des données simulées. Les arguments "start","week_to_predict","num_tra

sont utilisés pour la génération des données simulées comme expliqué précédemment. L'argument "chains" correspond au nombre de chaîne MCMC (Markov chain Monte Carlo) à utiliser et "iterations" correspond à la longueur de ces chaînes. L'échantillonneur implémenté par défaut dans le paquet pyStan est le NUTS (No-U-Turn Sampler) Hoffman et al., 2014. Pour préparer les données simulées qui vont être utilisées pour la construction du modèle, la méthode *data_preparation()* est développés. Les sorties de la méthode *data_preparation()* sont:

- Le DataFrame [5] **fitting_df** contenant les informations de la période d'entraînement concernant les flux de trafic et les types de jour.
- Le DataFrame **prediction_df** contenant les informations de la période de prédiction concernant les flux de trafic et les types de jour.
- La matrice **W** contenant les temps d'attente observés pour la période d'entraînement.
- La matrice **W_test** contenant les temps d'attente observés pour la période de prédiction.

Ensuite, la phase de prédiction est lancée avec la méthode ".build_model()" qui déclenche l'entraînement et l'inférence du modèle. Nous obtenons par conséquent les distributions des lois a posteriori des paramètres du modèle et des lois prédictives a posteriori des temps d'attente. Dans l'exemple ci-dessous toutes ces lois sont enregistrées dans la variable "samples".

```
from bayeshierarchical.python.
    bayesian_hierarchical_multilevel import
    BayesianHierarchicalMultilevel
iterations = 350
model = BayesianHierarchicalMultilevel(
              start="2018-01-01",
              week_to_predict="2018-11-08",
              num_train_wt=20,
              iterations=iterations,
              chains=1)

fitting_df,prediction_df,W,W_test = model.
    data_preparation()

samples = model.build_model()
```

5. Un DataFrame est une structure de données bidimensionnelle, les données sont alignées de façon tabulaire en lignes et en colonnes.

Construction du modèle sur les données réelles Dans le cas de l'utilisation du modèle sur des données réelles, voir des données externes pour d'autres applications, il suffit de créer l'objet *BayesianHierarchicalMultilevel()* avec des arguments spécifiques. L'exemple suivant illustre le cas d'utilisation du modèle sur des données réelles d'Ecov. Les arguments concernant la simulation des données ("start","week_to_predict", "wt_num) sont remplacés par les DataFrame "fitting_df" et "prediction_df" et les matrices "W" et "W_test". L'utilisateur doit préparer ces données en respectant les formats qui sont décrits dans la documentation. Évidement, les arguments "prediction_df" et "W_test" sont facultatifs car en production nous n'avons pas accès aux données futures. L'exemple suivant illustre l'utilisation du modèle hiérarchique bayésien sur des données réelles d'Ecov qu'on a nommées "fitting_df_real", "prediction_df_real", "W_real" et "W_test_real".

```
from bayeshierarchical.python.
    bayesian_hierarchical_multilevel import
    BayesianHierarchicalMultilevel
iterations = 350
model = BayesianHierarchicalMultilevel(
              fitting_df=fitting_df_real,
              prediction_df=prediction_df_real,
              W=W_real,
              W_test=W_test_real,
              iterations=iterations,
              chains=1)

samples = model.build_model()
```

Comme pour l'exemple précèdent, les lois a posteriori des paramètres et les lois a prédictive a posteriori des temps d'attente sont enregistrées dans la variable "samples".

Contrôles prédictifs a posteriori (Posterior predictive checks) Un outil important en statistique bayésienne est le contrôle prédictif a posteriori. Il consiste à vérifier graphiquement les distributions a posteriori des paramètres du modèle ainsi que les lois prédictives. La comparaison des lois a posteriori du modèle avec les lois qu'on observe est un excellent moyen pour valider un modèle. Nous avons développé deux fonctions permettant cette comparaison. La première s'appelle *graphical_ppc_hist()*, elle compare graphiquement la distribution des temps d'attente simulés (ou observés) avec la distribution prédictive a posteriori. Nous illustrons par la Figure 6.5 un exemple de comparaison des lois a posteriori prédictive du modèle avec les distributions des données simulées de temps d'attente.

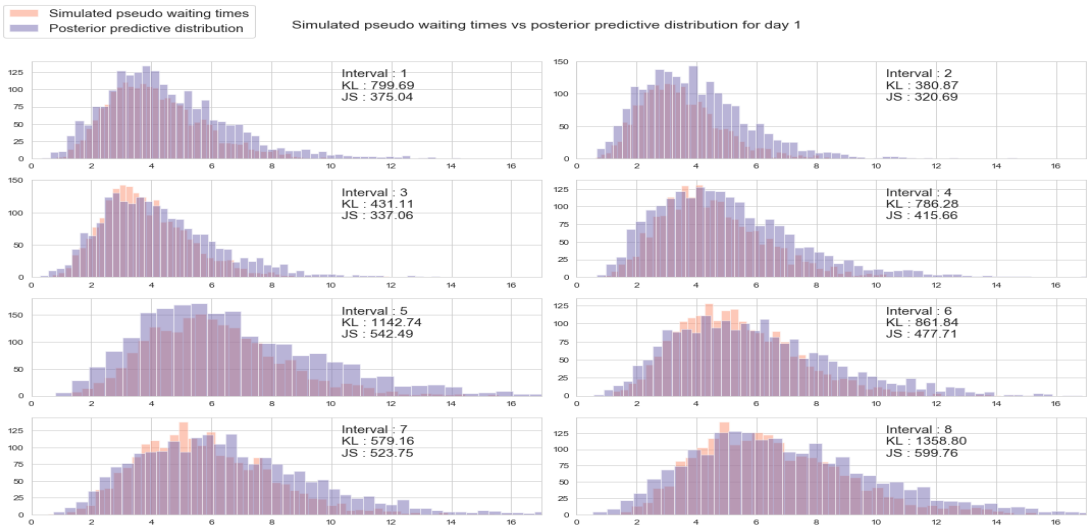


Figure 6.5 – Exemple du contrôle prédictif a posteriori des temps d'attente simulés avec la distribution prédictive a posteriori des temps d'attente via la fonction *graphical_ppc_hist()* du paquet *bayesian-hierarchical-model*.

La deuxième fonction concernant le contrôle prédictif a posteriori est *graphical_ppc_boxplots()* et permet de comparer les temps d'attente observés et les quantiles de la distribution prédictive a posteriori. La Figure 6.6 représente un exemple de comparaison des temps d'attente observés avec les quantiles de la distribution prédictive a posteriori. Les temps d'attente observés sont représentés par les boîtes à moustache et les différents quantiles prédictifs a posteriori sont illustrés par les points reliés. Ce genre de superpositions, entre loi observée et la loi prédictive, est possible grâce à l'approche bayésienne.

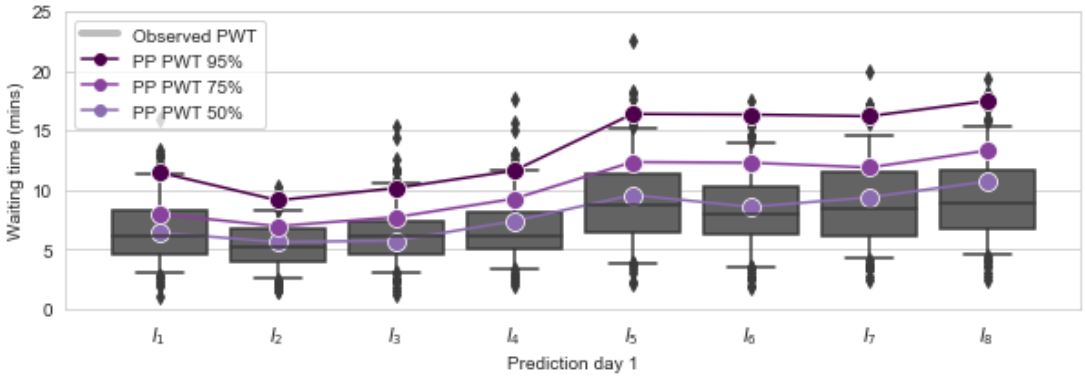


Figure 6.6 – Exemple du contrôle prédictif a posteriori des temps d'attente simulés avec les quantiles de la distribution prédictive a posteriori des temps d'attente via la fonction *graphical_ppc_boxplots()* du paquet *bayesian-hierarchical-model*.

Une dernière fonction développée pour valider le modèle est la fonction *plot_pe()*. Elle permet d'obtenir le score PE qui mesure le pourcentage de bonnes prédictions en dessous d'un seuil δ (en minutes) donné. Nous rappelons que ce score a été développée dans le Chapitre 4 spécifiquement pour le domaine des temps d'attente dans le covoiturage. La Figure 6.7 illustre un exemple de score PE obtenue sur des données simulées.

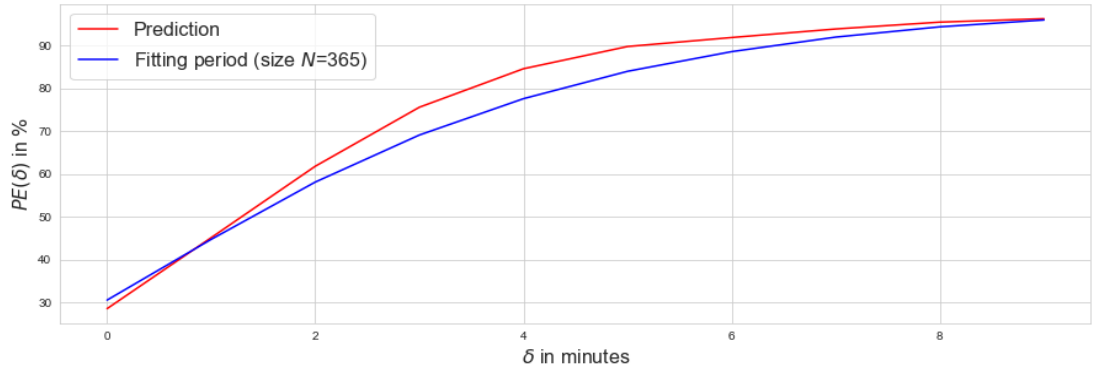


Figure 6.7 – Exemple du graphe du score PE produit par la fonction *plot_pe()* du paquet *bayesian-hierarchical-model* sur des données simulées.

Une fois la structure du paquet ainsi que les fonctions essentielles détaillées nous procédons à la présentation de son automatisation au sein d'Ecov.

6.4 Automatisation du processus

L'outil sélectionné pour l'automatisation du processus est Airflow [6]. Il s'agit d'un orchestrateur de tâches qui propose aussi une interface graphique. Cette dernière permet de vérifier la bonne exécution d'ensembles de tâches, regroupées en DAG (Directed Acyclic Graph). Le schéma choisi pour l'automatisation suit le concept d'ETL (Extract Transform Load). L'objectif d'un ETL est d'assurer la bonne gestion des différentes tâches permettant le chargement, la transformation et l'exploitation des données. En particulier pour notre application nous avons implémenté trois ETL. Chaque ETL a un rôle précis dans le but de prédire les flux de trafic et les temps d'attente. Nous détaillons le rôle de chacun ETL:

ETL1:

- **E**: La récupération de deux tables SQL (Structured Query Language) contenant les données brutes des conducteurs (les traces GPS) et les données brutes concernant les passagers (temps d'attente d'attente observés).

6. https://airflow.apache.org/

- **T** : Le nettoyage et la correction des données brutes, par exemple les temps d'attente négatifs et la conversion au bon fuseau horaire des horodatages. Une fois les données corrigées les deux tables sont fusionnées en une seule.

- **L** : L'enregistrement de cette table sur le data_warehouse d'Ecov.

ETL2:

- **E** : La récupération de la table produite par l'**ETL1**.

- **T** : La transformation des données afin de coïncider au format de données d'entrée du modèle comme illustré par la Figure 6.4 et l'entraînement du modèle avec la méthode ".build_model()".

- **L** : L'enregistrement des lois a posteriori des paramètres du modèle dans le data_warehouse d'Ecov.

ETL3:

- **E** : La récupération des lois a posteriori des paramètres du modèle par l'**ETL2**.

- **T** : La génération des lois a posteriori prédictives des flux de trafic et des temps d'attente basés sur les lois a posteriori des paramètres du modèle. Ensuite le calcul des quantiles prédictifs a posteriori qui nous intéresse.

- **L** : L'exposition des résultats (flux moyen par jour, temps d'attente par créneaux et par Origine-Destination) aux différents services d'Ecov (site web, application mobile, panneaux lumineux et BIV) et l'enregistrement des résultats dans le data_warehouse d'Ecov.

La Figure 6.8 représente de manière schématique la structure des ETL dans le cadre de la mise en production du modèle de prédiction de flux de trafic et des temps d'attente.

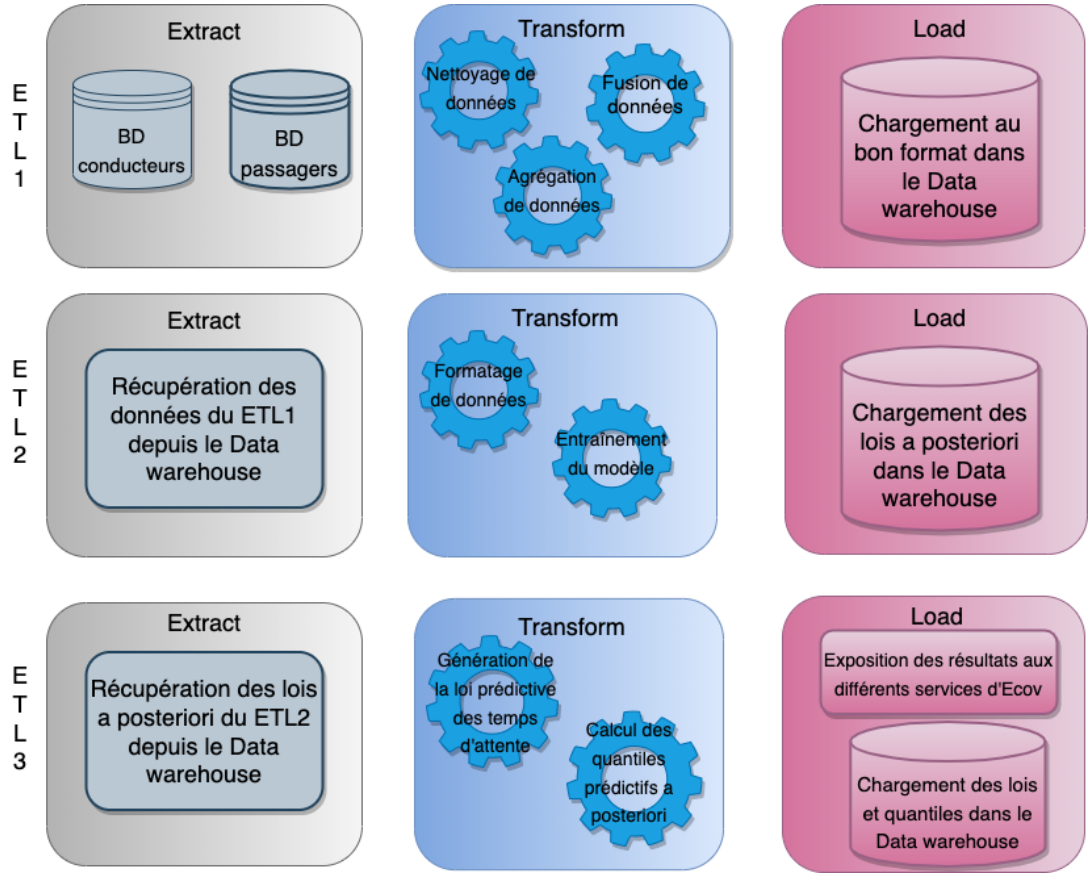


Figure 6.8 – Schéma de la structure des ETL mis en place afin d'exposer en production les prédictions des flux de trafic et des temps d'attente aux différents services d'Ecov.

6.5 Conclusions

Dans ce chapitre nous avons présenté la structure du paquet Python *bayesian-hierarchical-model* ainsi que ses fonctions. Puis, nous avons exposé comment son automatisation est faite dans le but d'avoir des prédictions en temps réel.

Nous rappelons que le paquet est développé en Python et en ce qui concerne l'inférence bayésienne nous avons utilisé PyStan. Une perspective possible est le changement de cette dernière technologie en la remplaçant par PyMC3[7] qui est un paquet pour la modélisation statistique bayésienne et l'apprentissage automatique probabiliste en python. Une deuxième perspective qui est apparue en fin de thèse est la définition d'un nouveau type de jour qui correspond aux périodes de confinement liée à la crise sanitaire. Les études préliminaires ont

7. https://docs.pymc.io/

montrées que le modèle actuel s'adapte rapidement à ce nouveau type de jour. Il serait donc intéressant d'intégrer ce type de jour dans le modèle en production. Une dernière perspective est de faire évoluer le paquet *bayesian-hierarchical-model* en ajoutant les travaux du transfert bayésien par prior informatif présenté au Chapitre 5. Cette dernière perspective permettra à Ecov d'obtenir des résultats du modèle hiérarchique bayésien sur des nouveaux territoires.

CONCLUSION

Synthèse

L'objectif de cette thèse est de développer des outils méthodologiques destinés au domaine du covoiturage, et plus précisément, à la prévision des temps d'attente et au potentiel du covoiturage sur un territoire. Le premier thème, à savoir les temps d'attente, est abordé dans les Chapitres 2 et 4. Le thème du potentiel de covoiturage est couvert par le Chapitre 3. Le Chapitre 5 présente une méthodologie pour la construction d'un prior informatif afin d'améliorer les estimations des temps d'attente dans une situation de jeu de données court. Enfin, le Chapitre 6 détaille la mise en production et l'exploitation industrielle du modèle hiérarchique bayésien.

Le Chapitre 2 présente le principe de régression quantile ainsi que quatre approches possibles de celle-ci. Nous rappelons que notre problématique est formulée de la manière suivante *"Quelle est la probabilité qu'un passager attende moins de y minutes avec $\alpha\%$ de chance?"*. Les données utilisées dans ce chapitre sont celles provenant de la structure initiale du service de covoiturage, à savoir le réseau de covoiturage. Nous proposons une comparaison des performances des quatre algorithmes de régression quantile retenus.

Le Chapitre 3 détaille, quant à lui, un processus de travail combinant les outils des Systèmes d'Information Géographiques (SIG) et de la science des données afin d'exploiter les données issues du covoiturage. Nous rappelons qu'à partir de ce chapitre les données exploitées sont celles provenant de la nouvelle structure du service, à savoir les lignes de covoiturage. Les nouveautés par rapport au chapitre précédent sont la collecte des traces GPS des conducteurs et la structure améliorée du service. Suite à l'analyse de ces données nous obtenons deux résultats : i) que la correspondance porte-à-porte de trajectoires complètes de l'origine à la destination est un obstacle structurel pour la transformation du covoiturage en un service de transport en commun et ii) que les arrêts de covoiturage fixes favorisent la rencontre entre les conducteurs et les passagers et diminuent les temps d'attente.

Le Chapitre 4 est consacré à la construction d'un modèle hiérarchique bayésien permettant la prévision des flux de trafic et des temps d'attente. Nous disposons des données industrielles provenant de deux sources distinctes : la première source concerne les traces GPS des conducteurs et la seconde les temps d'attente observés. Le modèle est constitué de deux étages imbriqués : le premier étage s'intéresse à la modélisation des flux de trafic quotidien tandis que le second étage modélise les temps d'attente pour différents créneaux de la journée.

Le Chapitre 5 expose une méthode de construction de loi a priori informative pour le modèle hiérarchique bayésien en situation d'historique court. Nous construisons cette loi informative à partir d'un jeu de données long $\mathcal{L}$ et d'une loi a priori non-informative. Par la suite, le modèle avec prior informatif est appliqué sur un jeu de données court $\mathcal{C}$ supposé semblable au jeu de données $\mathcal{L}$. Nous montrons à l'aide d'applications, sur des données simulées et réelles, que l'apport de la loi a priori construite améliore la qualité d'estimation des paramètres et les performances prédictives du modèle.

Enfin, le Chapitre 6 décrit les étapes nécessaires pour la mise en production du modèle hiérarchique bayésien. Le résultat concret du chapitre est la production d'un paquet Python *bayesian-hierarchical-model* pour la société Ecov.

Perspectives

Dans le Chapitre 3 le processus de travail constitue un prototype solide pour des flux de travail futurs de la société. Une perspective possible est l'industrialisation et la mise en production de ce processus de travail pour l'intégrer directement dans les outils de la société.

Comme nous l'avons expliqué dans le Chapitre 4, les données utilisées pour la construction du modèle hiérarchique sont issues des conducteurs et des passagers du service. Une première piste possible est l'inclusion de données externes. Des études ont montré que les données météorologiques peuvent améliorer la compréhension des flux de trafic. De plus, le modèle hiérarchique bayésien a la possibilité (avec des modifications mineures) de prendre en compte des données de comptage de trafic provenant de sources diverses. Ces comptages permettront d'améliorer à leur tour la connaissance des flux de trafic. Une dernière source de données à envisager est l'utilisation des informations des services "concurrents" pour un même territoire (la fréquence des transports en commun, les Véhicules de Tourisme avec Chauffeur disponibles, les autres applications de covoiturage). Cette dernière source de données a pour but d'améliorer la connaissance des temps d'attente. La seconde piste d'amélioration concerne directement la construction du modèle. Nous rappelons que le coefficient $\eta_{\mathrm{DT}(i-k)}$ permet de modéliser l'impact des $1 \leq k \leq K$ jours qui précèdent le $i^{\text{ième}}$ jour. Une perspective envisageable est de modéliser de la même manière l'impact entre les différents créneaux de la journée. Le coefficient $\eta_{\mathrm{DT}(i-k)}$ deviendrait une matrice représentant les interactions entre les différents créneaux de la journée.

La méthode proposée dans le Chapitre 5 repose sur l'incorporation des informations apprises par le modèle sur le jeu de données long $\mathcal{L}$ à la loi a priori du modèle sur le jeu de données court $\mathcal{C}$. Une perspective possible des travaux de ce chapitre est l'utilisation d'autres données disponibles, externes à celles utilisées actuellement par le modèle, afin de mieux construire le prior

informatif.

Un prolongement possible des travaux présentés dans le Chapitre 6 consiste à ajouter au paquet *bayesian-hierarchical-model* la méthode de construction du prior informatif par transfert bayésien introduit dans le Chapitre 5. Un autre prolongement envisageable est l'amélioration de l'exposition du modèle pour les clients interne et externe. Cela consiste à implémenter des fonctions permettant une meilleure visualisation des résultats obtenus par le modèle. Un dernier prolongement éventuel est de remplacer la librairie Python PyStan par une librairie concurrente nommée PyMC3.

BIBLIOGRAPHY

Anaconda Software Distribution (2020), version Vers. 2-2.4.0.

Armerin, F. (2014), "The Conditional Quantile as a Minimizer", Centre for Banking and Finance (Cefin).

Athey, S., J. Tibshirani & S. Wager (2019), "Generalized Random Forests", *in*: *The Annals of Statistics* **47**, pp. 1148–1178.

Baio, G. & M. Blangiardo (2010), "Bayesian hierarchical model for the prediction of football results", *in*: *Journal of Applied Statistics* **37**, pp. 253–264.

Bao, Y., F. Xiao, Z. Gao & Z. Gao (2017), "Investigation of the traffic congestion during public holiday and the impact of the toll-exemption policy", *in*: *Transportation Research Part B* **104**, pp. 58–81.

Bassett Jr, G. & R. Koenker (1982), "An empirical quantile function for linear models with iid errors", *in*: *Journal of the American Statistical Association* **77**.*378*, pp. 407–415.

Biau, G. & B. Cadre (2020), "Optimization by gradient boosting", *in*: *Advances in Contemporary Statistics and Econometrics – Festschrift in Honour of Christine Thomas-Agnan*, In press.

Biau, G., B. Cadre & L. Rouvière (2019), "Accelerated gradient boosting", *in*: *Machine Learning* **108**, pp. 971–992.

Biau, G. & B. Patra (2011), "Sequential quantile prediction of time series", *in*: *IEEE Transactions on Information Theory* **57**.*3*, pp. 1664–1674.

Biau, G. & E. Scornet (2016), "A random forest guided tour", *in*: *Test* **25**, pp. 197–227.

Bosch, R. J., Y. Ye & G. G. Woodworth (1995), "A convergent algorithm for quantile regression with smoothing splines", *in*: *Computational statistics & data analysis* **19**.*6*, pp. 613–630.

Breiman, L. (1997), *Arcing the edge*, tech. rep., Technical Report 486, Statistics Department, University of California at ...

Breiman, L. (2001), "Random forests", *in*: *Machine learning* **45**.*1*, pp. 5–32.

Bühlmann, P., T. Hothorn, et al. (2007), "Boosting algorithms: Regularization, prediction and model fitting", *in*: *Statistical Science* **22**, pp. 477–505.

Chacon, S. & B. Straub (2014), *Pro git*, Apress.

Chen, T. & C. Guestrin (2016), "XGBoost: A Scalable Tree Boosting System", *in*: DOI: 10.1145/2939672.2939785.

Congdon, P. (2014), *Applied Bayesian Modelling*, John Wiley & Sons.

Cooper, C. (2007), "Successfully changing individual travel behavior: Applying community-based social marketing to travel choice", *in*: *Transportation Research Record* **2021**, pp. 89–99.

Crambes, C., A. Gannoun & Y. Henchiri (2013), "Support vector machine quantile regression approach for functional data: Simulation and application studies", *in*: *Journal of Multivariate Analysis* **121**, pp. 50–68.
Cucchi, K., F. Heße, N. Kawa, C. Wang & Y. Rubin (2019), "Ex-situ priors: A Bayesian hierarchical framework for defining informative prior distributions in hydrogeology", *in*: *Advances in Water Resources* **126**, pp. 65–78.
De Haan, L. & A. Ferreira (2007), *Extreme Value Theory: An Introduction*, Springer Science & Business Media.
Deublein, M., M. Schubert, B. T. Adey, J. Köhler & M. H. Faber (2013), "Prediction of road accidents: a Bayesian hierarchical approach", *in*: *Accident Analysis & Prevention* **51**, pp. 274–291.
Ding, A., X. Zhao & L. Jiao (2002), "Traffic flow time series prediction based on statistics learning theory", *in*: *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 727–730.
Douglas, D. H. & T. K. Peucker (2011), "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature", *in*: *Classics in Cartography: Reflections on Influential Articles from Cartographica* **10**, pp. 15–28, DOI: 10.1002/9780470669488.ch2.
Egidi, L. (2018), "Developments in Bayesian Hierarchical Models and Prior Specification with Application to Analysis of Soccer Data", *in*:
Facebook Core Data Science Group (2019), *Forecasting at Scale*, Python package version 0.5. https://facebook.github.io/prophet. Stan model file https://github.com/facebook/prophet/blob/master/R/inst/stan/prophet.stan, Facebook.
Freund, Y. (1995), "Boosting a weak learning algorithm by majority", *in*: *Information and Computation* **121**, pp. 256–285.
Freund, Y. & R. E. Schapire (1997), "A decision-theoretic generalization of on-line learning and an application to boosting", *in*: *Journal of computer and system sciences* **55**.*1*, pp. 119–139.
Friedman, J., T. Hastie, R. Tibshirani, et al. (2000), "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)", *in*: *The Annals of Statistics* **28**, pp. 337–407.
Friedman, J. H. (2001), "Greedy function approximation: a gradient boosting machine", *in*: *Annals of statistics*, pp. 1189–1232.
Friedman, J. H. (2002), "Stochastic gradient boosting", *in*: *Computational Statistics & Data Analysis* **38**, pp. 367–378.
Furuhata, M., M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang & S. Koenig (2013), "Ridesharing: The state-of-the-art and future directions", *in*: *Transportation Research Part B: Methodological* **57**, pp. 28–46.
Gelman, A. (2006), "Multilevel (hierarchical) modeling: what it can and cannot do", *in*: *Technometrics* **48**.*3*, pp. 432–435.

Gelman, A. & J. Hill (2006), *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge University Press.
Ghoseiri, K. (2012), "Dynamic rideshare optimized matching problem", PhD thesis, University of Maryland.
Ghosh, B., B. Basu & M. O'Mahony (2007), "Bayesian time-series model for short-term traffic flow forecasting", *in*: *Journal of Transportation Engineering* **133**, pp. 180–189.
Gould, P. G., A. B. Koehler, J. K. Ord, R. D. Snyder, R. J. Hyndman & F. Vahid-Araghi (2008), "Forecasting time series with multiple seasonal patterns", *in*: *European Journal of Operational Research* **191**, pp. 207–222.
Guidotti, R., M. Nanni, S. Rinzivillo, D. Pedreschi & F. Giannotti (2017), "Never drive alone: boosting carpooling with network analysis", *in*: *Information Systems* **64**, pp. 237–257.
Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, et al. (Sept. 2020), "Array programming with NumPy", *in*: *Nature* **585**.*7825*, pp. 357–362, DOI: 10.1038/s41586-020-2649-2.
Hoffman, M. D. & A. Gelman (2014), "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo", *in*: *Journal of Machine Learning Research* **15**, pp. 1593–1623.
Hunter, J. D. (2007), "Matplotlib: A 2D graphics environment", *in*: *Computing in Science & Engineering* **9**.*3*, pp. 90–95, DOI: 10.1109/MCSE.2007.55.
INSEE (2018), *Mobilités professionnelles en 2015: déplacements domicile–lieu de travail*, In French. https://www.insee.fr/fr/statistiques/3566477, National Institute of Statistics and Economic Studies [INSEE], France.
Jeffreys, H. (1946), "An invariant form for the prior probability in estimation problems", *in*: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **186**.*1007*, pp. 453–461.
Kluyver, T., B. Ragan-Kelley, F. Pérez, B. E. Granger, M. Bussonnier, J. Frederic, et al. (2016), "Jupyter Notebooks-a publishing format for reproducible computational workflows.", *in*: *ELPUB*, pp. 87–90.
Kocherginsky, M., X. He & Y. Mu (2005), "Practical confidence intervals for regression quantiles", *in*: *Journal of Computational and Graphical Statistics* **14**.*1*, pp. 41–55.
Koenker, R., A. Chesher & M. Jackson (2005), *Quantile Regression*, Econometric Society Monographs, Cambridge University Press.
Koenker, R. & G. Bassett Jr (1978), "Regression quantiles", *in*: *Econometrica* **46**, pp. 33–50.
Koenker, R. & K. F. Hallock (2001), "Quantile regression", *in*: *Journal of economic perspectives* **15**.*4*, pp. 143–156.
Kung, K. S., K. Greco, S. Sobolevsky & C. Ratti (2014), "Exploring universal patterns in human home-work commuting from mobile phone data", *in*: *PLoS One* **9**, e96180.

Launay, T., A. Philippe & S. Lamarche (2015), "Construction of an informative hierarchical prior for a small sample with the help of historical data and application to electricity load forecasting", *in*: *Test* **24**, pp. 361–385.
Lee, D. W. & S. H. L. Liang (2011), "Crowd-sourced carpool recommendation based on simple and efficient trajectory grouping", *in*: *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pp. 12–17, DOI: 10.1145/2068984.2068987.
Li, F.-F. & P. Perona (2005), "A Bayesian hierarchical model for learning natural scene categories", *in*: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 524–531.
Li, Q., R. Xi, N. Lin, et al. (2010), "Bayesian regularized quantile regression", *in*: *Bayesian Analysis* **5**.*3*, pp. 533–556.
Li, X., S. Hu, W. Fan & K. Deng (2018), "Modeling an enhanced ridesharing system with meet points and time windows", *in*: *PLOS ONE* **13**, pp. 1–19.
Makridakis, S., R. J. Hyndman & F. Petropoulos (2020), "Forecasting in social settings: the state of the art", *in*: *International Journal of Forecasting* **36**, pp. 15–28.
McShane, W. R. & R. P. Roess (1990), *Traffic Engineering*, Prentice-Hall.
Meinshausen, N. (2006a), "Quantile Regression Forests", *in*: *Journal of Machine Learning Research* **7**, pp. 983–999, DOI: 10.1111/j.1541-0420.2010.01521.x.
Meinshausen, N. (2006b), "Quantile regression forests", *in*: *Journal of Machine Learning Research* **7**.*Jun*, pp. 983–999.
Nesterov, Y. (1983), "A method of solving a convex programming problem with convergence rate $O(1/k^2)$", *in*: *Proceedings of the Russian Academy of Sciences* **269**, pp. 543–547.
OpenStreetMap contributors (2019), *OpenStreetMap*, https://www.openstreetmap.osm.
Papoutsis, P., S. Fennia, C. Bridon & T. Duong (2021), "Relaxing door-to-door matching reduces passenger waiting times: A workflow for the analysis of driver GPS traces in a stochastic carpooling service", *in*: *Transportation Engineering* **4**, p. 100061, DOI: https://doi.org/10.1016/j.treng.2021.100061.
Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al. (2011), "Scikit-learn: Machine Learning in Python", *in*: *Journal of Machine Learning Research* **12**, pp. 2825–2830.
Portnoy, S., R. Koenker, et al. (1997), "The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators", *in*: *Statistical Science* **12**.*4*, pp. 279–300.
Ray, J.-B. (2014), "Planning a real-time ridesharing network: critical mass and role of transfers", *in*: *Transport Research Arena (TRA) 5th Conference: Transport Solutions from Research to DeploymentEuropean CommissionConference of European Directors of Roads (CEDR) European Road Transport Research Advi-*

sory Council (ERTRAC) WATERBORNEEuropean Rail Research Advisory Council (ERRAC) Institut Francais des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux (IFSTTAR) Ministère de l'Écologie, du Développement Durable et de l'Énergie.

Schapire, R. E. (1990), "The strength of weak learnability", *in*: *Machine Learning* **5**, pp. 197–227.

Schapire, R. E. (1996), "Experiments with a New Boosting Algorithm", *in*: *Update*.

Shaheen, S. A., N. D. Chan & T. Gaynor (2016), "Casual carpooling in the San Francisco Bay Area: Understanding user characteristics, behaviors, and motivations", *in*: *Transport Policy* **51**, pp. 165–173.

Shaheen, S. A., A. Stocker & M. Mundler (2017), "Online and App-Based Carpooling in France: Analyzing Users and Practices – A Study of BlaBlaCar", *in*: *Disrupting Mobility: Impacts of Sharing Economy and Innovative Transportation on Cities*, ed. by G. Meyer & S. Shaheen, Springer International Publishing, pp. 181–196.

Sherwood, B., A. Maidman, M. B. Sherwood & T. ByteCompile (2017), "Package 'rqPen'", *in*:

Smith, B. L. & M. J. Demetsky (1997), "Traffic flow forecasting: comparison of modeling approaches", *in*: *Journal of Transportation Engineering* **123**.4, pp. 261–266.

Stan Development Team (2012), *Stan Modeling Language User's Guide and Reference Manual, Version 1.0*.

Stephanedes, Y., P. G. Michalopoulos & R. A. Plum (1980), "Improved estimation of traffic flow for real-time control", *in*: *Transportation Research Record* **795**, pp. 28–39.

Stiglic, M., N. Agatz, M. Savelsbergh & M. Gradisar (2015), "The benefits of meeting points in ride-sharing systems", *in*: *Transportation Research Part B: Methodological* **82**, pp. 36–53.

Tang, J., Y. Song, H. J. Miller & X. Zhou (2016), "Estimating the most likely space–time paths, dwell times and path uncertainties from vehicle trajectory data: A time geographic method", *in*: *Transportation Research Part C: Emerging Technologies* **66**, pp. 176–194.

Taylor, S. J. & B. Letham (2018), "Forecasting at scale", *in*: *The American Statistician* **72**, pp. 37–45.

The pandas development team (Feb. 2020), *pandas-dev/pandas: Pandas*, version latest, DOI: 10.5281/zenodo.3509134.

TomTom (2019), *Routing API and Extended Routing API*, https://developer.tomtom.com/routing-api.

Van der Vaart, A. W. (2000), *Asymptotic statistics*, vol. 3, Cambridge university press.

Van Rossum, G. & F. L. Drake Jr (1995), *Python tutorial*, Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Vanpaemel, W. (2011), "Constructing informative model priors using hierarchical methods", *in*: *Journal of Mathematical Psychology* **55**.*1*, pp. 106–117.

Wang, H. & H. Yang (2019), "Ridesourcing systems: A framework and review", *in*: *Transportation Research Part B: Methodological* **129**, pp. 122–155.

Wang, S., X. Sun & U. Lall (2017), "A hierarchical Bayesian regression model for predicting summer residential electricity demand across the USA", *in*: *Energy* **140**, pp. 601–611.

Yu, K. & R. A. Moyeed (2001), "Bayesian quantile regression", *in*: *Statistics & Probability Letters* **54**.*4*, pp. 437–447.

Zammit, L. C., M. Attard & K. Scerri (2013), "Bayesian hierarchical modelling of traffic flow – with application to Malta's road network", *in*: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 1376–1381.

Zheng, S. (2012), "QBoost: Predicting quantiles with boosting for regression and binary classification", *in*: *Expert Systems with Applications* **39**.*2*, pp. 1687–1697.

Zhu, D. (2017), "More generous for small favour? Exploring the Role of Monetary and Pro-Social Incentives of Daily Ride Sharing Using a Field Experiment in Rural Île-de-France", *in*: *DigiWorld Economic Journal* **108**, pp. 77–97.

Zhu, D. (2018), *The limit of money in daily ridesharing: Evidence from a field experiment*, tech. rep., PSL, University of Paris-Dauphine.

Zhu, D. (2020), "Understanding Motivations and Impacts of Ridesharing: Three Essays on Two French Ridesharing Platforms", PhD thesis, PSL, University of Paris-Dauphine.

Zhu, D. (2021), "The limit of money in daily ridesharing: evidence from a field experiment", *in*: *Revue d'Économie Industrielle*, To appear.

CENTRALE NANTES

Titre: Potentiel et prévision des temps d'attente pour le covoiturage sur un territoire

Mot clés : covoiturage, temps d'attente, régression quantile, modèle hiérarchique bayésien, prior informatif

Resumé : Cette thèse s'intéresse au potentiel et à la prévision des temps d'attente concernant le covoiturage sur un territoire en utilisant des méthodes d'apprentissage statistique. Cinq thèmes principaux sont abordés dans le présent manuscrit. Le premier présente des techniques de régression quantile afin de prédire des temps d'attente. Le deuxième détaille la construction d'un processus de travail empruntant des outils des Systèmes d'Information Géographique (SIG) afin d'exploiter pleinement les données issues du covoiturage. Dans un troisième temps nous construisons un modèle hiérarchique bayésien en vue de prédire des flux de trafic et des temps d'attente. En quatrième partie nous proposons une méthode de construction d'une loi a priori informative par transfert bayésien dans le but d'améliorer les prédictions des temps d'attente pour une situation de jeu de données court. Enfin, le dernier thème se concentre sur la mise en production et l'exploitation industrielle du modèle hiérarchique bayésien.

Title: Potential and prediction of waiting times for carpooling in a territory

Keywords : carpooling, waiting times, quantile regression, bayesian hierarchical model, informative prior

Abstract : This thesis focuses on the potential and prediction of carpooling waiting times in a territory using statistical learning methods. Five main themes are covered in this manuscript. The first presents quantile regression techniques to predict waiting times. The second details the construction of a workflow based on Geographic Information Systems (GIS) tools in order to fully leverage the carpooling data. In a third part we develop a hierarchical bayesian model in order to predict traffic flows and waiting times. In the fourth part, we propose a methodology for constructing an informative prior by bayesian transfer to improve the prediction of waiting times for a short dataset situation. Lastly, the final theme focuses on the production and industrial exploitation of the bayesian hierarchical model.