Apprentissage d'un vocabulaire symbolique pour la détection d'objets dans une image

Sébastien Gadat

THÈSE DE DOCTORAT
DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Présentée par

Sébastien GADAT

Pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Domaine :

MATHÉMATIQUES

Sujet de la thèse :

Apprentissage d'un vocabulaire symbolique pour la détection d'objets dans une image

Thèse présentée et soutenue à Cachan le 17 décembre 2004 devant le jury composé de :

Romain Abraham	Professeur	Rapporteur
Michel Benaïm	Professeur	Rapporteur
Donald Geman	Chercheur associé	co-Directeur de thèse
Lionel Moisan	Professeur	Examinateur
Alain Trouvé	Professeur	Président
Laurent Younes	Directeur de recherches	Directeur de thèse

CMLA
ENS CACHAN/CNRS/UMR 8536
61 avenue du Président Wilson, 94235 CACHAN CEDEX (France)

Remerciements

Je voudrais avant tout remercier Laurent Younes pour sa disponibilité, sa patience, sa gentillesse et aussi sa rigueur et son humour décalé (mais toujours juste !) qui a été un directeur de thèse idéal pour guider mes premiers pas dans l'univers si stimulant de la recherche, ainsi que Donald Geman et ses nombreuses intuitions gagnantes pour la réalisation de mon travail.

Je remercie également Alain Trouvé pour les discussions toujours profitables pour l'avancement de mes recherches et qui me fait l'honneur de présider ce jury ainsi que Romain Abraham et Michel Benaïm qui ont spontanément accepté de rapporter mon travail, et qui par leurs remarques constructives ont contribué à améliorer la version finale de ce manuscrit. Je tiens à remercier enfin Lionel Moisan pour sa participation à ce jury et l'intérêt qu'il a porté à mon travail.

J'adresse aussi des remerciements un peu lointains à Michael Miller pour m'avoir si sympathiquement accueilli à Baltimore pour la poursuite de mes travaux, même si les températures y ont étés glaciales !

Robert Azencott m'a fait des remarques très profitables et celles-ci m'ont permis d'éclaircir certains points de mon travail, je ne voudrais donc pas l'oublier dans cette longue page de remerciements.

J'adresse une pensée particulière à Christophe qui est sans doute une des rares personnes qui aura eu le courage incensé de lire l'integralité de mon mémoire, d'améliorer et exécuter des codes - certes parfois inutiles - sur les surpuissants centres de calculs du CMLA.

Pour les divers éclaircissements sur des points qui se sont avérés essentiels dans mon travail, je veux exprimer ma gratitude à Paul Dupuis, Amarjit Budhiraja, François Fleuret et Hichem Sahbi.

De manière un peu moins sérieuse, je voudrais aussi souligner l'influence positive de toutes les personnes qui ont animé le laboratoire autour de moi et sans qui l'environnement de travail aurait été bien morose. Parmi eux, à part les moultes cafetières usées au long de ces trois ans, quelques autres noms me viennent à l'esprit :

Junior, même si parfois sa musique est un peu bizarre et son art de la glissade controlée avec souris pas toujours au point.

BérEnger pour sa persévérance rassurante dans l'exécution à la lettre d'un régime hypocalorique, la fameuse inégalité triangulaire et son intégrale légendaire sur 2n variables.

Céline et Fifi pour leurs invitations au fond de l'Essonne qui j'espére, se répèteront encore dans les mois à venir, surtout s'il y a du gâteau au chocolat.

Samy sans qui n'importe quelle ligne de mon C++ n'aurait pas dépassé le "Seg Fault", et qui a réussi à me donner (malgré lui) une vague idée de ce que représentait mach 1 aux jeux videos.

Julien et Anthony, même s'ils font partie de l'équipe d'analyse numérique (...), se sont révéles d'excellents camarades sauf peut-être dans un domaine que la décence m'empêche de citer.

Jérémie pour son goût avéré pour le sprint, l'offrande, la glissade, les stomps, claps ou autres blasts en tous genres.

Benjamin (gogogo !) pour sa gentillesse et son calme parfois énervant et Julie pour avoir réussi à canaliser l'énergie débordante de certains de mes collocataires de bureau.

Micheline et Véronique pour leur organisation toujours nickel, leurs disponibilités mais aussi leurs penchants toujours trop excessifs pour certains breuvages.

Pascal et ses fameux services windows, son indécrottable volonté d'hélas utiliser drakconf et d'aller si dangeureusement crousser tous les jours.

Je tiens aussi à remercier mes parents, mon frère et mes grands parents pour m'avoir encouragé et enseigné depuis tout petit leur goût du travail. Je pense particulièrement à mon papy qui, je suis sûr, aurait été si content de contempler le parcours accompli.

Je n'oublie pas Solveig, Olivier, Vincent, Julien, Olivia, Mathilde et Laurence qui me font la joie d'être venus m'écouter sans vraissemblablement comprendre quoi que ce soit et Élie, Hélène, Adrien et Véronique qui m'ont permis de me détourner de mon travail dans le paisible environnement de Manosque.

Enfin, je pense à Mélanie pour tout ça, en plus du reste ...

Table des matières

Chapitre 1 - Introduction - État de l'art

1.1 Problématique

La détection d'objets dans une image ainsi que l'analyse et la classification d'un signal représentent des enjeux majeurs pour le traitement du signal par ordinateurs. En particulier, l'analyse de divers types de signaux comme par exemple les images satellitaires, les mammographies ou les enregistrements audio pose le problème évident de l'extraction de petites quantités de caractéristiques (« features » ou détecteurs) car l'espace initial dans lequel vivent les données est de très grande dimension.

Il paraît par ailleurs raisonnable de penser que l'exécution de tâches algorithmiques appliquées à un ensemble de signaux peut être optimisée à condition de disposer initialement du bon espace de caractéristiques, et ce quel que soit l'algorithme utilisé. La recherche du bon « feature space » est donc d'une importance capitale pour l'efficacité de la résolution de problèmes en traitement du signal. Enfin, ce sont ces caractéristiques élémentaires qui vont permettre d'interpréter le signal étudié, c'est alors en ce sens que nous utiliserons le mot « vocabulaire » pour désigner l'ensemble des caractéristiques élémentaires que l'on pourrait extraire des signaux issus d'une base de données.

Les différentes idées qui animent cette thèse pour la recherche de ces bonnes caractéristiques se basent sur différents objectifs comme l'exhaustivité et la parcimonie du vocabulaire. Et c'est en définitive la nature des propriétés que l'on souhaite obtenir sur la composition de notre vocabulaire qui détermine la manière de construire un tel ensemble de features.

Nous pouvons dans un premier temps énumérer différentes motivations pour la pré-sélection d'un ensemble de features pour l'analyse d'un signal.

- Cette pré-sélection de variables permet en effet de comprendre, du point de vue cognitif, ce que sont les caractéristiques principales qui permettent de distinguer particulièrement une classe de signal d'une autre. On peut alors exhiber et mettre de côté les features qui sont fondamentaux (qui apportent une quantité d'information substentielle pour le traitement du signal que l'on souhaite faire) et écarter au contraire les features qui ne permettent pas d'avoir des conclusions tangibles sur la tâche de traitement souhaitée.
- Du point de vue de la complexité algorithmique, on peut souhaiter manipuler un vocabulaire concis. De ce fait, le codage d'une telle liste de détecteurs correspondant à ce vocabulaire va alors permettre de n'écrire qu'une quantité réduite de bits en mémoire virtuelle ou sur le disque dur du système informatique, ce qui entraînera dès lors une meilleure portabilité du système d'analyse du signal.
- Toujours du point de vue de la complexité algorithmique, on peut vouloir qu'il existe une certaine redondance entre les différents features du vocabulaire symbolique permettant de décrire les données. Outre la rapidité de calcul de ces features qui peut alors en être accrue,

il ne faut pas négliger non plus la vitesse de transmission d'un tel ensemble de réalisation de features sur un signal. En effet, le codage d'une telle liste de réalisation de détecteurs est d'autant plus compact qu'il existe des propriétés de redondance entre les différents features que l'on souhaite coder (codage LZW - [CT91]).

- En suivant des considérations statistiques, on constate que la variance introduite par le vocabulaire dont on dispose est globalement une fonction croissante de la quantité d'éléments qui appartiennent à ce vocabulaire. En revanche, le biais inhérent à toute modélisation dépendant d'un ensemble de détecteurs applicables à un signal est lui décroissant en fonction la quantité de features. Ainsi, ce dilemme Biais-Variance ([GBD92]) aboutit à un choix à faire entre
  - la réduction de la variance du système lors de la sélection des bons features tout en maintenant un biais raisonnable (en ne supprimant par exemple que les détecteurs apportant peu d'informations à l'interprétation du signal traité).
  - la diminution du biais en ajoutant au système de features un nouveau caractère en contrôlant alors l'augmentation de la variance du modèle.
- La réduction de la dimensionnalité des données est fondamentale pour le problème de la reconnaissance de formes dans la mesure où l'augmentation du nombre de caractéristiques des données n'augmente pas nécessairement la qualité d'apprentissage. En effet, le phénomène de Hughes (également connu sous le nom de « Curse of dimensionality ») implique que la quantité de données N nécessaire pour apprendre statistiquement un modèle à p dimensions à une précision fixée augmente exponentiellement avec p ([HTF01], paragraphe 2.5).

Dans un second temps, l'extraction de nouvelles caractéristiques à partir des features primaires permet de percevoir d'autres avantages pour la reconnaissance de formes dans un signal.

- Dans le domaine cognitif de l'intelligence artificielle, l'agrégation de détecteurs peut permettre d'identifier la composition (cachée) de champs de variables qui génère le signal manipulé. Cela constituerait alors une avancée majeure dans l'apprentissage par l'informatique de sources de données pour des tâches d'apprentissage supervisé.
- En ce qui concerne les performances des tâches de classification ou reconnaissance, une telle extraction de nouvelles variables peut aboutir à de meilleures propriétés de discrimination d'un signal dans le cas où celui-ci n'est justement pas intéressant.

Les applications d'une telle amélioration du feature space peuvent être nombreuses et variées. La fouille de connaissances dans des bases de données souligne bien les contraintes de gestion et d'efficacité nécessaire à l'exploration de ces grandes bases de données ([UCI], [MIT]). Par exemple, lorsque l'on souhaite analyser des fichiers .log de connections internet sur des serveurs WEB, on constate que ces fichiers peuvent avoir plusieurs millions d'entrées pour des milliers de variables. Sans une sélection de ces variables, la plupart des méthodes de classification des données échouent sur ce type de données car les algorithmes élaborés dans des dimensions réduites ne sont pas toujours transposables dans des cas où les dimensions sont bien supérieures. Il paraît cependant intuitif que sur la quantité de données manipulées, une faible proportion de ces données peut permettre de résoudre le problème de reconnaissance de formes dans le signal.

Nous commencerons par énumérer des méthodes algorithmiques classiques pour la reconnaissance de formes et la détection d'objets avant de citer quelques méthodes existantes pour la sélection de features vis-à-vis d'un problème d'apprentissage.

1.2 Algorithmes de classification

La reconnaissance de formes consiste en l'automatisation de tâches de perception artificielle réalisées par un système informatique alors qu'elles sont usuellement effectuées par le cerveau humain. Une forme est une représentation simplifiée de l'univers extérieur définie d'une certaine manière pour l'ordinateur, par exemple un vecteur de réels, un mot d'un langage donné, ... Nous pouvons représenter la reconnaissance de forme par ordinateur en utilisant le schéma :

Un système de reconnaissance de formes ou de classification comprend la plupart du temps une phase d'apprentissage qui consiste à « apprendre » (à reconnaitre) certaines classes d'objets sur une base d'échantillon (Training-Set). Lors de cette phase d'apprentissage, le système sélectionne alors les règles qui lui permettront de décider sur les données à classer (Test Set), quelles sont les formes qu'il pense être les bonnes.

Dans notre étude, les problèmes de reconnaissance de formes seront tous des problèmes dits supervisés : le nombre de classes est connu ainsi qu'un échantillon de données pour chaque classe. L'algorithme d'apprentissage que nous construisons utilisera alors différents classifieurs pour sélectionner les variables retenues pour la tâche de reconnaissance de formes.

1.2.1 Décision Bayésienne

La décision bayésienne est la théorie centrale des méthodes stochastiques où les problèmes de décision sont traités en termes de probabilités. Le point névralgique de cette théorie est la règle de Bayes qui permet en fait de choisir l'hypothèse ayant la probabilité la plus élevée.

Dans notre cadre, on suppose que le problème de reconnaissance de forme fait intervenir s classes que l'on énumère en $\mathcal{C}_1, \dots \mathcal{C}_s$ et on se dote de fonctions réelles $\lambda(\mathcal{C}_i, \mathcal{C}_j)$ qui quantifient le coût de la décision de classe $\mathcal{C}_i$ quand le signal appartient en réalité à $\mathcal{C}_j$. Si l'on note X le signal d'entrée, $\mathrm{P}(\mathrm{X}|\mathcal{C}_i)$ la loi de probabilité d'obtenir le signal X lorsque la classe est $\mathcal{C}_i$ et $\mathrm{P}(\mathcal{C}_i)$ la probabilité *a priori* de la classe $\mathcal{C}_i$, alors :

$$\mathrm{P}(\mathcal{C}_i|\mathrm{X}) = \frac{\mathrm{P}(\mathrm{X}|\mathcal{C}_i)\mathrm{P}(\mathcal{C}_i)}{\mathrm{P}(\mathrm{X})}$$

avec

$$\mathrm{P}(\mathrm{X}) = \sum_{k=1}^{s} \mathrm{P}(\mathrm{X}|\mathcal{C}_k)\mathrm{P}(\mathcal{C}_k)$$

La fonction de coût qui est associée à un signal X et une classe $\mathcal{C}_i$ est elle donnée par :

$$\mathrm{R}(\mathcal{C}_i|\mathrm{X}) = \sum_{k=1}^{s} \lambda(\mathcal{C}_i|\mathcal{C}_j)\mathrm{P}(\mathcal{C}_j|\mathrm{X})$$

La règle de décision Bayésienne est alors de choisir la classe $\mathcal{C}_i$ qui minimise la fonction de risque R connaissant le signal X. Lorsqu'on décide de prendre comme fonction de coût la fonction symétrique :

$$\lambda(\mathcal{C}_i, \mathcal{C}_j) = 1 - \delta_{i,j}$$

on obtient la règle de Bayes classique qui consiste, étant donné un signal X, à maximiser $\mathrm{P}(\mathcal{C}_i|\mathrm{X})$ puisque :

$$\mathrm{R}(\mathcal{C}_i, \mathrm{X}) = \sum(1 - \delta_{i,k})\mathrm{P}(\mathcal{C}_k|\mathrm{X}) = \sum_{k \neq i} \mathrm{P}(\mathcal{C}_k|\mathrm{X}) = 1 - \mathrm{P}(\mathcal{C}_i|\mathrm{X})$$

On choisit donc dans ce cas de sélectionner la classe $\mathcal{C}_i$ qui maximise la probabilité conditionnelle sachant X, probabilité alors évaluée sur les échantillons du Training Set. Le taux d'erreur commis par la décision Bayésienne est alors appelé taux de Bayes. La règle de décision Bayésienne est une méthode couramment utilisée pour classer un signal.

1.2.2 k Plus Proche Voisin

Nous ne rentrons pas dans les détails du déroulement de cet algorithme et renvoyons par exemple à [Kni99] ou [HTF01] pour la connaissance de différents aspects théoriques de cet algorithme. Nous retiendrons qu'étant donné un ensemble de données labélisées (« training-set »), on décide de classer un signal d'entrée en étudiant le voisinage formé par les k plus proches voisins de ce signal dans le training-set puis en choisissant comme réponse de l'algorithme la classe majoritaire parmi les labels du voisinage calculé.

Cette méthode :

- ne nécessite aucune analyse nécessaire du modèle ni aucun calcul de densité.
- réclame la définition d'une métrique entre les différents signaux traités.
- nécessite de conserver tous les échantillons du « training-set ».
- demande d'effectuer de nombreuses mesures de distance.

On perçoit donc ici l'intérêt de n'avoir qu'un faible nombre de variables appliquées aux signaux puisque le calcul des distances est d'autant plus long que le nombre de features est grand.

Nous avons pris le parti d'utiliser en particulier cet algorithme car il possède des propriétés assez performantes du point de vue des taux de classification. Si $e^\star$ désigne l'erreur commise par le classificateur de Bayes (classificateur optimal), et si e est l'erreur commise par l'algorithme de k plus proche voisin, on a l'inégalité :

$$e^\star < e < 2e^\star$$

En ce qui concerne le temps de calcul pour l'algorithme de k plus proche voisin, il est crucial de contourner l'énumération totale de tous les points du Learning-Set ainsi que la considération de toutes les variables puisque, si N désigne le nombre de points du Learning-Set et si p désigne la quantité de features disponibles, l'exécution du kPPV nécessite $\mathrm{O}(k\mathrm{N}p)$ calculs. Le premier point (réduction de N) peut être effectué en utilisant des techniques de Clustering ([HTF01], paragraphe 14.3) tandis que la sélection des variables (réduction de p) peut être fait *via* une des méthodes évoquées plus loin ou celle présentée dans notre travail.

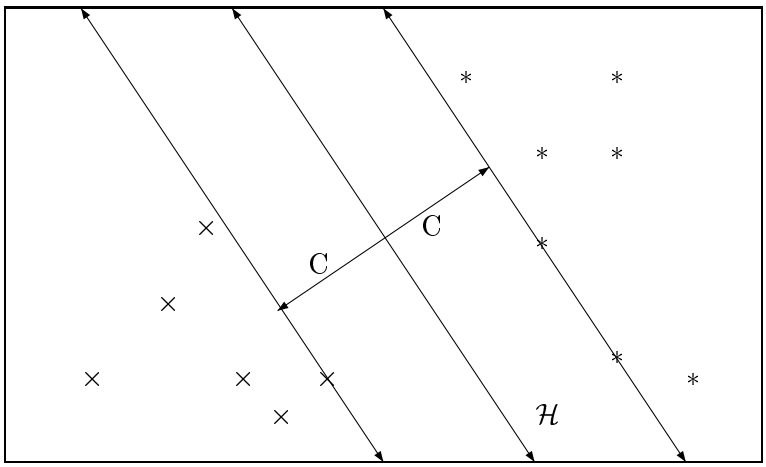
Enfin, le calcul du kPPV est très sensible à la présence de points non représentatifs (« outliers ») et une pré-sélection des variables du problème peut permettre de supprimer l'effet de ces « outliers » si les variables responsables de la présence de ces « outliers » sont alors identifiées et supprimées du vocabulaire.

1.2.3 Support Vector Machine

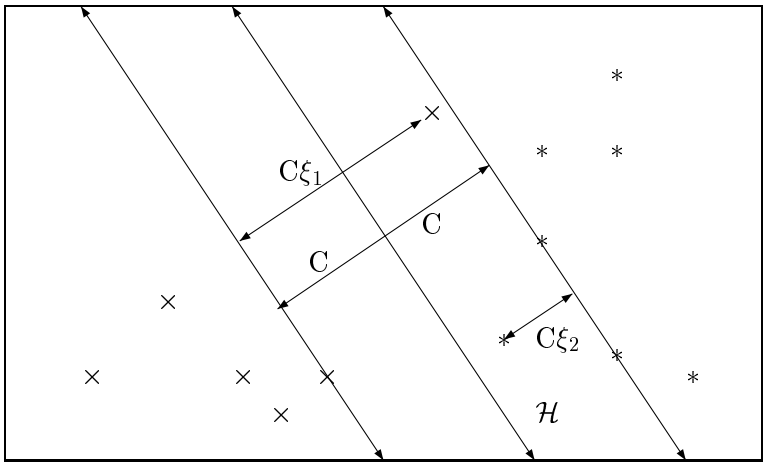
Les machines à Vecteur de Support sont issues des travaux de Vapnik ([Vap00], [CM98]) et permettent d'obtenir de bonnes performances de classification initialement dans le cas d'un

problème à deux classes, mais cet outil peut également être adapté à des problèmes multi-classes. La problématique de Vapnik ne consiste pas à minimiser un taux d'erreur sur l'ensemble d'apprentissage mais plutôt à trouver un hyperplan optimal séparant les deux classes de données. Étant donné un jeu de données $(x_1, y_1), \dots (x_n, y_n)$ de $\mathbb{R}^p \times \{-1; 1\}$, on peut représenter le problème par les schémas suivants.

- dans le cas séparable :



- dans le cas non séparable :



Les variables y_i prennent alors leurs valeurs dans $\{-1; 1\}$ et les x_i sont des points de $\mathbb{R}^p$. Dans l'exemple précédent, les points tels que y_i vaut 1 sont les $\times$ tandis que $*$ représente les points tels que $y_i = -1$.

On cherche alors l'hyperplan $\mathcal{H}$ maximisant la marge de séparation C des deux classes, cet hyperplan a une équation donnée par l'application affine f de la forme :

$$\mathcal{H} = \left[x \in \mathbb{R}^p \quad | \quad x^t\beta + \beta_0 = f(x) = 0\right]$$

où β est le vecteur normal unitaire à l'hyperplan $\mathcal{H}$.

La recherche d'un tel hyperplan peut également s'adapter au cas où les deux nuages ne sont pas séparables en paramétrant chaque point x_i par un réel positif ξ_i qui mesure la distance du point à l'un des deux hyperplans d'appui $\mathcal{H}_1$ ou $\mathcal{H}_2$. Dans ce cas, il s'agit alors de maximiser la marge C sous les contraintes :

$$y_i({x_i}^t\beta + \beta_0) > \mathrm{C}(1 - \xi_i)$$

La formalisation « duale » de cette maximisation revient à minimiser la norme $\|\beta\|$ sous les mêmes contraintes. La résolution d'un tel problème amène alors à étudier le minimum du Lagrangien grâce aux conditions de Karush-Kuhn-Tucker :

$$\mathrm{L_P} = {\|\beta\|_2}^2 + \mathrm{C}\sum\xi_i - \sum\lambda_i \left(y_i \left((x_i \mid \beta) + \beta_0\right) - 1 + \xi_i\right) - \sum\mu_i\xi_i$$

où μ_i sont les multiplicateurs de Lagrange associés à la condition de positivité de ξ_i, et sous les contraintes :

$$\begin{cases} \dfrac{\partial \mathrm{L_P}}{\partial \beta_j} = 0 = \beta_j - \sum\lambda_i y_i x_{i,j} = 0 \\ \dfrac{\partial \mathrm{L_P}}{\partial \beta_0} = 0 - \sum\lambda_i y_i \end{cases}$$

Pour plus de détails sur la résolution d'un tel système quadratique, on pourra se référer à [Bur98], [Vap00] ou [JK00].

On notera que l'algorithme de séparation de deux classes par SVM permet également d'obtenir des séparations non-linéaires des deux classes. On utilise une application Φ et un noyau K tels que Φ est une application de $\mathbb{R}^p \mapsto \mathcal{E}$ où $\mathcal{E}$ est un espace euclidien et K est défini par :

$$\mathrm{K}(x_i, x_j) = (\Phi(x_i) \mid \Phi(x_j))_{\mathcal{E}}$$

Il s'agit alors de trouver un hyperplan séparateur des deux classes dans $\mathcal{E}$ pour les deux nuages de points transformés par l'application Φ. Dans notre approche de sélection des variables, on pourra interpréter en réalité la suppression ou l'ajoût de nouvelles variables comme l'apprentissage d'un noyau K pour mieux séparer les différentes classes d'objet.

Enfin, dans notre travail, nous avons utilisé l'implémentation SVM_{light}, algorithme optimisé de l'algorithme de Support Vector Machine établi par T. Joachims ([Joa02], [JK00]).

1.3 Sélection de features

Nous allons maintenant exposer brièvement quelques méthodes de sélection de features plus ou moins classiques permettant de restreindre la dimensionalité du problème de reconnaissance de formes.

1.3.1 Analyse en composantes principales

L'analyse en composantes principales est une technique classique en statistique linéaire ([Sap90]) pour représenter de façon concise un nuage de points d'un espace affine $\mathcal{E}$, ce nuage de points

représentant la plupart du temps une population d'individus. On suppose donc donnés des points $\mathrm{X}_1, \ldots \mathrm{X}_\mathrm{N}$ correspondant à N individus. L'objectif de l'analyse en composantes principales est alors la recherche des vecteurs e_i dans l'espace vectoriel $\overrightarrow{\mathcal{E}}$ orthonormés tels que les points X_i soient représentés en :

$$\mathrm{X}_k = \overline{\mathrm{X}} + \sum \alpha_{k,i} e_i + \mathrm{R}_k$$

où $\overline{\mathrm{X}}$ est un point de l'espace affine $\mathcal{E}$ et $\alpha_{k,i}$ les coordonnées des points X_k sur les vecteurs e_i qui forment une famille libre de $\overrightarrow{\mathcal{E}}$. Le but est alors de minimiser l'erreur quadratique ε donnée par

$$\varepsilon = \sum_{k=1}^{\mathrm{N}} \|\mathrm{R}_k\|_2{}^2$$

On constate immédiatement que dès que les vecteurs orthonormés e_i sont choisis, la solution (choix des $\alpha_{k,i}$) est en réalité déterminée puisque la meilleure représentation correspond finalement à la projection du vecteur $\mathrm{X}_k - \overline{\mathrm{X}}$ sur l'espace engendré par les (e_i).

Ainsi $$\alpha_{k,i} = (\mathrm{X}_k - \overline{\mathrm{X}} \mid e_i)$$

Le choix des vecteurs e_i se réduit à la minimisation de

$$\varepsilon = \sum_{k=1}^{\mathrm{N}} \left\| \mathrm{X}_k - \overline{\mathrm{X}} - \sum_{i=1}^{p} (\mathrm{X}_k - \overline{\mathrm{X}} \mid e_i) e_i \right\|_2^2$$

Donc $$\varepsilon = \sum_{k=1}^{\mathrm{N}} \left\| \mathrm{X}_k - \overline{\mathrm{X}} \right\|_2{}^2 - \sum_{i=1}^{p} \sum_{k=1}^{\mathrm{N}} (\mathrm{X}_k - \overline{\mathrm{X}} \mid e_i)^2$$

De ce fait, il s'agit de maximiser la somme

$$\sum_{i=1}^{p} \|e_i\|_{\mathcal{N}}{}^2$$

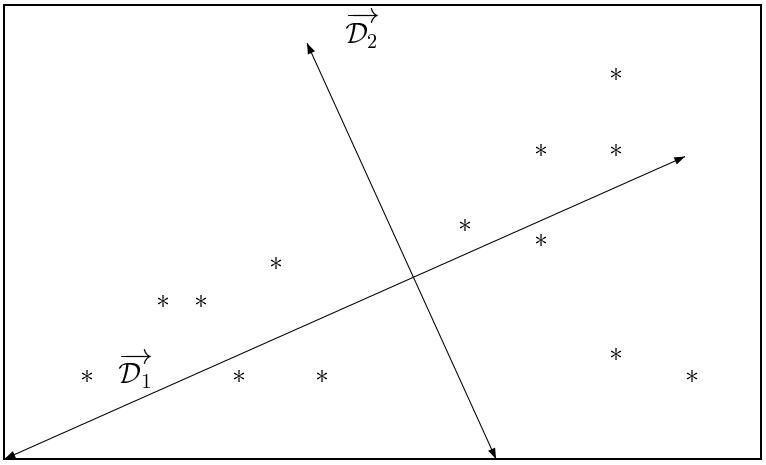
où la semi-norme $\|.\|_{\mathcal{N}}$ est issue du produit scalaire :

$$(\mathrm{X} \mid \mathrm{Y})_{\mathcal{N}} = \frac{1}{\mathrm{N}} \sum_{k=1}^{\mathrm{N}} (\mathrm{X}_k - \overline{\mathrm{X}} \mid \mathrm{X})(\mathrm{X}_k - \overline{\mathrm{X}} \mid \mathrm{Y})$$

Les vecteurs (e_i) sont alors les vecteurs propres associés à la forme quadratique $\|.\|_{\mathcal{N}}{}^2$, et sont appelés les directions principales du nuage de points.

On peut de plus interpréter statistiquement cette résolution comme étant en réalité la recherche de la base $\mathcal{B} = (e_i)$ telle que les projections de la variable aléatoire $(\mathrm{X} - \mathbb{E}\mathrm{X})$ sur les e_i représentent des variables α_i qui ne sont pas corrélées :

$$\mathbb{E}[\alpha_i \alpha_j] = 0$$



Dans le schéma précédent, l'axe $\overrightarrow{\mathcal{D}_1}$ désigne l'axe principal de l'ACP associée au nuage de points tandis que $\overrightarrow{\mathcal{D}_2}$ est le second axe de l'ACP. Par définition, il est bien entendu orthogonal à $\overrightarrow{\mathcal{D}_1}$.

La sélection des features peut alors s'effectuer en utilisant cette ACP en choisissant comme features les coordonnées sur les axes de l'analyse en composantes principales. En effet, en écrivant tous les vecteurs propres de $\|.\|_{\mathcal{N}}{}^2$ et en les ordonnant par ordre décroissant de valeurs propres, on obtient une liste d'axes et coefficients $(e_1, \lambda_1) \dots (e_{\mathrm{N}}, \lambda_{\mathrm{N}})$ tels que

$$\lambda_1 \geqslant \lambda_2 \geqslant \dots \geqslant \lambda_{\mathrm{N}}$$

quitte à poser $\lambda_p = 0$ si un nombre strictement inférieur à N suffit pour décrire le nuage de points. La sélection des features s'effectue alors en fonction de la précision souhaitée pour la description des données en prenant successivement comme caractéristiques les coordonnées des points du nuage sur les axes e_1, puis e_2 ...

Il est de plus remarquable que la précision de l'approximation des points du nuage par leurs projections sur l'espace engendré par $(e_1, \dots e_p)$ est donnée par la somme

$$\lambda_{p+1}^2 + \dots \lambda_{\mathrm{N}}^2$$

1.3.2 Analyse en composantes indépendantes

L'analyse en composantes indépendantes s'inspire de la problématique précédente. Si X est une variable aléatoire d'un espace euclidien $\mathcal{E}$ on recherche les vecteurs (e_i) tels que les coordonnées s_i de X sur les vecteurs e_i sont alors des variables indépendantes. On peut également résumer le problème en la recherche de W tel que

$$s = \mathrm{WX} \quad \text{ou} \quad \mathrm{X} = \mathrm{A}s$$

et s a ses coordonnées indépendantes. Si $p_i(.)$ désigne la densité de probabilité de s_i et p la densité de probabilité jointe des sources s, cela signifie que l'on a :

$$p(s_1, \dots s_n) = \prod_{i=1} p_i(s_i)$$

Néanmoins, l'analyse ne garantit pas toujours la détermination de sources indépendantes, mais plutôt approche la solution où l'on a des sources aussi indépendantes que possible. Pour quantifier cette optimalité, plusieurs mesures existent ([Car98], [Jut87], [JH91]). Nous allons présenter brièvement une méthode utilisant une fonction de contraste pour en déduire une ACI.

Si P_s désigne la loi des sources reconstruites $s = \mathrm{WX}$, il s'agit alors d'estimer et de minimiser la fonction de contraste :

$$\phi^{\mathrm{IM}}(y) = \mathrm{K}\left(\mathrm{P}_y | \prod p_i(s_i)\right)$$

Cette méthode utilise la divergence de Kullback-Leibler de la loi jointe à la loi produit et mesure donc l'indépendance des variables au sens où elle donne une distance à l'indépendance.

On résout alors une recherche d'une telle matrice W en effectuant une descente de gradient. [Car98] ou [Hyv99].

La sélection des variables s'effectue une fois que la matrice W est déterminée en choisissant les coordonnées s_i données par $(\mathrm{WX})_i$ qui minimisent la fonction de contraste ϕ^{IM}.

1.3.3 Construction de Features à partir d'arbres de décisions

De nombreux travaux ont été effectués pour parvenir à la construction de features sous la forme d'arbres binaires de décisions ([AG97a], [Bre98]). L'approche consiste généralement à construire des arbres de décisions de plus en plus complexes, à partir de features élémentaires binaires. Les arbres sont construits récursivement, en prenant en compte soit des propriétés géométriques ([FG01]), soit des propriétés statistiques ([AG97b], [AGW97]). En général, les algorithmes de construction montants des arbres utilisent des notions de théorie de l'information ([CT91]) comme l'entropie d'une variable aléatoire, entité qui mesure le désordre ou l'incertitude statistique de la réalisation d'une variable aléatoire.

Définition 1.3.1 (Entropie d'une variable aléatoire)
Si X *est une variable aléatoire à valeurs dans* Ω *et* $\mathrm{P}(\mathrm{X})$ *sa loi de probabilité, on définit* $\mathrm{H}(\mathrm{X})$ *par*

$$\mathrm{H}(\mathrm{X}) = \mathbb{E}[-\log \mathrm{P}] = \sum_{\omega \in \Omega} -\mathrm{P}(\mathrm{X} = \omega) \log \mathrm{P}(\mathrm{X} = \omega)$$

Dans les travaux sur les arbres de décision, si l'on suppose construits des features complexes représentés par des arbres, on décide de former un nouvel arbre à partir de critères statistiques utilisant :

- l'entropie condionnelle : si l'on nomme $\mathrm{Q}_1, \ldots, \mathrm{Q}_{k-1}$ les $k-1$ features sélectionnés dans l'arbre binaire de décision, on forme un nouvel arbre binaire dont l'arbre précédent est un sous-arbre si Q_k minimise

  $$\mathrm{H}\left(\mathrm{Y} | \mathrm{Q}_k, \mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1\right)$$

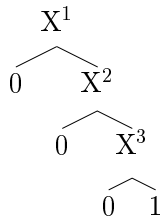
  Cela revient à chercher le feature Q_k qui va répartir un poids à peu près équivalent sur les éléments du Learning-Set qui sont réalisés pour $\mathrm{Q}_k, \mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1$ et ceux qui sont réalisés pour $\mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1$ mais pas pour $\mathrm{Q}_k, \mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1$ ([AG97b]).
- la probabilité de réussite de $\mathrm{Q}_k, \mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1$ conditionnée à la réussite des précédents $\mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1$:

  $$\mathrm{P}\left[(\mathrm{Q}_k, \mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1)(\mathrm{Y}) = 1 | (\mathrm{Q}_{k-1}, \ldots \mathrm{Q}_1)(\mathrm{Y}) = 1\right] \geqslant \tau$$

  où $\tau = 1/2$ dans [AG97a].

- la corrélation statistique ρ pour la fusion de deux arbres de décisions binaires ([FG01]).

L'idée est alors d'obtenir des features complexes discriminants en utilisant des disjonctions de features binaires, en s'assurant que sur une des classes de signaux, le nouveau feature est réalisé avec une probabilité suffisante. Cela a donné lieu à l'approche Coarse-to-Fine (du plan large au détail) et à un mode de parcours des arbres vérifiant un critère d'optimalité ([Fle00], paragraphe 5.6).



Dans le schéma précédent, X^1 désigne une variable aléatoire binaire (détecteur de bord « coarse »), et si cette variable vaut 1 (il y a eu détection *via* X^1), on appplique alors X^2 détecteur plus précis. Le parcours de l'arbre de décision binaire permet alors d'obtenir un algorithme de classification efficace : il est peu couteux en quantité de stockage de données et en temps de calcul.

Plus précisément, supposant construit un ensemble de features $\mathcal{F}_k$ (qui peuvent être représentés sous forme d'arbres binaires de décisions), la construction de $\mathcal{F}_{k+1}$ s'effectue en parcourant toutes les concaténations possibles d'arbres binaires de $\mathcal{F}_k$ et en choisissant d'ajouter de telles concaténations à $\mathcal{F}_{k+1}$ sur des considérations statistiques. Par exemple, si $\mathcal{A}_1$ et $\mathcal{A}_2$ sont deux arbres de décisions de $\mathcal{F}_k$ vérifiant

$$\rho(\mathcal{A}_1;\mathcal{A}_2) \geqslant \rho_0$$

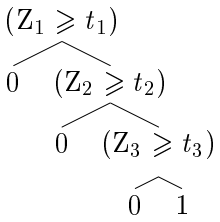
puis $\mathcal{A}_1 = \widehat{\mathcal{A}_1} \in \mathcal{F}_k$ (avec $\mathcal{A}_1.g$ et $\mathcal{A}_1.d$) et $\mathcal{A}_2 = \widehat{\mathcal{A}_2} \in \mathcal{F}_k$ (avec $\mathcal{A}_2.g$ et $\mathcal{A}_2.d$)

on décide alors de former le nouvel arbre

$$\mathcal{A}_1 :: \mathcal{A}_2 = \widehat{\mathcal{A}_1 :: \mathcal{A}_2} \in \mathcal{F}_{k+1}$$

($\mathcal{A}_1$ et $\mathcal{A}_2$)

Les tests séquentiels issus de features sont alors des arbres de décisions formés à partir de tests binaires, les tests sont de la forme



où les Z_i sont des variables aléatoires formées à partir du compte du nombre de réalisations d'arbres binaires de certains ensembles $\mathcal{F}_k$. Pour plus de détails, nous renvoyons aux travaux [FG01].

1.3.4 Sélection de features binaires par critère d'information mutuelle

F. Fleuret exploite dans ses travaux récents un modèle qui utilise à partir de M features $f_{n(1)}, \dots f_{n(\mathrm{M})}$, binaires un algorithme du type perceptron en choisissant une règle de décision de la forme

$$f(x) = \sum_{i=1}^{\mathrm{M}} w_i f_i(x) + b$$

On recherche les w_i optimaux pour obtenir un taux d'erreur minimal *via* une descente de gradient [Ros58]. Pour construire ces M features, il utilise alors un critère basé sur l'information mutuelle I qu'apporte un nouveau feature aux features déjà existants.

Définition 1.3.2 (information mutuelle)
Soient X *et* Y *deux variables aléatoires de lois* p *et* q *et de loi jointe* r*, l'information mutuelle* $\mathrm{I(X;Y)}$ *est définie par*

$$\mathrm{I(X;Y)} = \sum_{x \in \Omega} \sum_{y \in \Omega'} r(x,y) \log \frac{r(x,y)}{p(x)q(y)}$$

Si les F_i sont les features dont on dispose, on commence donc par sélectionner le feature apportant le plus d'information au modèle :

$$n(1) = \arg \operatorname*{Max}_i \mathrm{I(Y, F}_i)$$

Puis on sélectionne récursivement tous les autres features en choisissant à l'étape k celui qui possède la meilleure minoration (la plus grande) de l'information mutuelle avec l'ensemble des features construits à l'étape $k-1$:

$$n(k+1) = \arg \operatorname*{Max}_i \left\{ \operatorname*{Min}_j \mathrm{I(Y, F}_i | \mathrm{F}_{n(j)}) \right\}$$

La sélection de features ainsi effectuée, l'algorithme du perceptron de Rosenblatt exécuté sur un tel sous-ensemble de features permet alors d'obtenir des résultats comparables à l'algorithme de Boosting (*Cf* paragraphe 1.4) exécuté sur l'ensemble des features dans le problème de la détection de visages ([Fle03]).

1.3.5 Maximisation de la marge des SVMs pour la sélection de variables

Nous avons vu dans la section 1.2 que l'algorithme de Support Vector Machine permettait de séparer de façon optimale un nuage de points appartenant à deux classes dans un espace de grande dimension. Deux méthodes de sélection de features basées sur la structure de l'hyperplan de séparation en deux classes ont été étudiées. Les deux méthodes utilisent la variation de la marge de séparation en deux classes, la première supprime récursivement des variables tandis que la seconde effectue un algorithme de descente de gradient pour apprendre un noyau optimal pour le SVM.

1.3.5.1 Élimination récursive de features (ERF)

Séparation linéaire Étant donné un nuage de points appartenant à deux classes sur p variables réelles, on peut décider de calculer un hyperplan séparateur linéaire entre ces deux classes pour ces variables. Le résultat de l'algorithme de SVM donne donc une application f affine donnée par :

$$f(x) = (w \mid x) + b = \sum_{j=1}^{p} w_j x^j + b$$

où $(\mid)$ est le produit scalaire euclidien standard de $\mathbb{R}^p$ et chaque point x de $\mathbb{R}^p$ a pour coordonnées $\left(x^j\right)_{j=1..p}$.

L'idée guidant l'ERF ([WMC+00]) est de calculer le vecteur w et de classer les valeurs absolues de $|w_i|$ par ordre croissant. Comme les variables x_i telles que w_i est grand sont les variables les plus influentes pour l'hyperplan de séparation, on suppose que ces variables sont celles qui ont le plus d'importance pour le problème de classification traité par le SVM. On décide alors de supprimer les features qui correspondent à des quantités $|w_i|$ relativement faibles. On peut par exemple décider de supprimer les 10% de features ayant le moins grand $|w_i|$ puisque dans la détection par hyperplan, ce sont les features qui influencent le moins la détection. On procède récursivement en recommençant un nouveau calcul de SVM sur les $9n/10$ variables restantes, et ceci jusqu'à obtenir la quantité de features souhaitée.

Séparation non linéaire Dans le cas où l'on utilise un noyau pour le SVM, l'idée de base est identique puisqu'il s'agit également de supprimer les features affectant le moins la marge. Si l'équation de l'application f est donnée par

$$f(x) = \sum c_i \mathrm{K}(x, x_i) + b$$

la marge M est alors donnée ([Vap00]) par

$$\frac{1}{\mathrm{M}} = \sum_{i,j} c_i c_j \mathrm{K}(x_i, x_j)$$

et la mesure d'influence du feature j sur la marge vaut alors

$$\mathrm{S}(j) = \frac{\partial\,(1/\mathrm{M})}{\partial x_j}$$

On choisit là encore de supprimer les 10% de features ayant la quantité $\mathrm{S}(j)$ la plus petite et la procédure récursive est itérée à nouveau jusqu'à l'obtention de la quantité souhaitée de features.

1.3.5.2 Apprentissage d'un noyau

On peut également utiliser une autre méthode à base de Support Vector Machine pour sélectionner des features. La technique ([CVBM02]) est un peu différente de ce qui a été évoqué plus haut puisqu'on paramètre le noyau K par un vecteur $\theta \in \mathbb{R}^n$ où n est le nombre de features disponibles initialement :

$$\mathrm{K}_\theta(x, z) = \mathrm{K}(\theta^t x, \theta^t y)$$

On décide alors de minimiser l'erreur estimée *via* une descente de gradient sur le paramètre θ. On pourra obtenir tous les calculs nécessaires dans [CVBM02].

1.4 Le Boosting : complément naturel à la sélection de features

Nous avons choisi de présenter rapidement l'algorithme de Boosting, même si ce n'est pas à proprement parler un algorithme de sélection de Features, car cet algorithme peut permettre à

partir de classifieurs $(f_m)_{m=1...\mathrm{M}}$ binaires à valeurs dans $\{-1;1\}$ de chercher des quantités c_m pour que

$$\mathrm{F}(x) = \sum_{m=1}^{\mathrm{M}} c_m f_m(x)$$

renvoie une erreur de classification inférieure, à chaque étape de l'algorithme.

Le principe de l'algorithme est le suivant : on peut appliquer un certain nombre de règles de décisions d'« experts » pour un problème de classification, et chacun de ces experts fournit une règle aux performances faibles, mais néanmoins meilleures qu'une décision précise au hasard. Les questions auxquelles répond l'algorithme du Boosting sont alors les suivantes :

- Quels experts doit-on interroger lorsqu'un échantillon à classer nous est présenté ?
- Comment combiner les avis de ces experts pour atteindre la meilleure décision ?
- Est-il possible de rendre aussi bon que l'on veut un algorithme d'apprentissage « faible » ?

Schapire donne les différentes réponses à ces questions et nous allons brièvement présenter l'algorithme AdaBoost introduit dans [FS99]. L'algorithme utilise une distribution de probabilité sur le Training Set qui donne plus de poids aux points de l'ensemble d'apprentissage qui sont mal-classés pour concentrer l'attention de l'algorithme précisément sur ces points. Voici comment l'algorithme se déroule précisément :

1. $(x_1, y_1), \ldots (x_{\mathrm{N}}, y_{\mathrm{N}})$ couples de données x_i et réponses $(y_i \in \{\pm 1\})$.
2. Initialisation de w_i en $1/\mathrm{N}$
3. Utiliser f_m pour calculer son erreur ε_m sur la distribution de données $\sim w$:

   $$\varepsilon_m = \mathbb{E}_w \left[\chi_{y \neq f_m(x)}\right]$$

   et poser

   $$c_m = \log\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)$$

4. Mettre à jour la distribution apprise sur le Learning-Set en utilisant les formules :

   $$w_i e^{c_m \chi_{y_i \neq f_m(x_i)}} \longmapsto w_i$$

5. Renormaliser les coefficients w_i et retourner en 2.

En fin d'algorithme, on choisit alors de classer les données en prenant le signe de F donnée par $\mathrm{F} = \sum c_m f_m$:

$$\widetilde{y} = \mathrm{signe}\left[\mathrm{F}(x)\right]$$

L'algorithme AdaBoost présenté précédemment permet d'obtenir des améliorations notables des performances des classifieurs f_m. On peut ([FHT00]) par ailleurs interpréter un tel algorithme comme la recherche des coefficients c_i qui minimisent la fonction de coût :

$$\mathrm{J}(c) = \mathbb{E}\left[e^{-y\mathrm{F}(x)}\right]$$

Le boosting est donc une méthode permettant, à l'issue d'une sélection de classifieurs f_i, d'augmenter la performance de classification en organisant un vote optimisé de ces classifieurs f_i. Ainsi, pour obtenir une bonne sélection de variables, nous aurons donc intérêt à proposer un ensemble non restreint à un seul classifieur pour permettre l'utilisation du boosting. Cette idée simple guidera notre modélisation dans la suite du mémoire.

1.5 Organisation du mémoire

Dans tout le mémoire, nous allons donc chercher à utiliser divers algorithmes de classification et à optimiser leurs performances en sélectionnant les variables sur lesquelles les taux d'erreur de classification sont les plus faibles.

Dans le chapitre 2, nous donnons les définitions précises des objets que nous allons manipuler dans le mémoire : nous définissons les features élémentaires pour le cas particulier des images, situation où en général il n'existe pas de définition « intrinsèque » de caractéristiques élémentaires. Puis nous précisons ce que sont les dictionnaires et structures arborescentes des features plus complexes que les features élémentaires que nous manipuleront. Enfin, nous rappelons la définition des entités informatives qui nous permettent de mesurer l'efficacité d'agrégation de features comme l'information commune ou la corrélation fonctionnelle de variables aléatoires.

Le chapitre 3 présente une nouvelle manière de sélectionner certaines variables d'un signal lorsque le dictionnaire de features est figé, en effectuant une descente de gradient d'une énergie. Nous modélisons notre problème de sélection des variables par un tirage aléatoire de ces variables *via* une loi de probabilité $\mathbb{P}$ sur l'ensemble des features, ce qui constitue un modèle tout à fait applicable à différents problèmes de classification de signaux et sélections de variables. Les techniques utilisées sont classiques, comme les méthodes d'approximation d'équation différentielle du type Robbins-Monro. Des applications précises sont données sur divers types de signaux : données synthétiques, messages électroniques ou images réelles. Nous obtenons par ailleurs un résultat de convergence de notre schéma d'apprentissage qui, sous des conditions certes restrictives, converge vers le minimum absolu de l'énergie de notre système.

Dans le chapitre 4, nous définissons et utilisons l'application de Skorokhod pour construire un processus stochastique contraint à un simplexe $\mathcal{S}_{\mathcal{F}}$ qui permettra d'organiser une méthode de sélection de variables parmi l'ensemble $\mathcal{F}$ des features fixés. La contrainte d'appartenance au simplexe de notre processus stochastique est alors naturellement satisfaite, ce qui représente un avantage majeur par rapport aux conditions obtenues en fin de chapitre 3 sur notre descente de gradient exacte, ou approchée.

Nous donnons une méthode précise pour faire évoluer notre espace de features dans le chapitre 5 en donnant des règles de transitions entre différents ensembles de features. Ces transitions sont basées sur une dynamique de type MCMC pour des chaînes faiblement réversibles et n'est pas sans rappeler l'évolution de certains algorithmes d'évolution des populations tels les algorithmes génétiques ou les réseaux de neurones. Dans ce chapitre, nous construisons également un processus stochastique représentant à la fois l'évolution de notre population de tests et les règles de tirage de ces tests. Ces règles sont toujours dédiées au problème de la minimisation d'une énergie $\mathcal{E}$ basée sur un taux d'erreur de classification d'un algorithme fixé.

Le chapitre 6 est une étude succinte du comportement asymptotique et infinitésimal du processus couplé défini dans le chapitre 5. On précise notamment une propriété importante de récurrence du processus avant d'exprimer le générateur du processus. Nous donnons enfin la mesure stationnaire associée à ce processus qui est précisément le champ de Gibbs associée à l'énergie $\mathcal{E}$.

Enfin, le chapitre 7 donne un algorithme d'approximation stochastique du processus défini au chapitre 5. Plusieurs définitions et propriétés sur l'approximation au sens faible y sont données, avant de montrer que le processus approché construit converge bien faiblement vers le processus défini au chapitre 5. Nous appliquons enfin notre étude d'approximation au cas des exemples synthétiques du chapitre 3 ainsi qu'à la détection de visages issus de [MIT], les performances sont alors nettement améliorées par rapport aux résultats obtenus au chapitre 3 puisque l'ensemble des features construits possède alors de grandes propriétés discriminantes pour les images de « fond », propriétés très utiles pour le problème de la détection de visages.

Chapitre 2 - Obtention de features et mesure de l'information

2.1 Features élémentaires

2.1.1 Introduction

Le problème de la détection et de la classification d'objets dans un signal implique tout d'abord que l'on puisse accéder à des données quantifiées dans ce signal. Cette quantification des données nécessite alors la définition d'attributs sur les signaux manipulés. De plus, le choix des caractéristiques retenues dans le signal est d'une importance capitale pour l'obtention de bonnes performances lors de tâches de détection ou de classification. C'est alors précisément la recherche de features élémentaires, puis composées qui motivera toute la suite de ce travail. La recherche du bon « feature space » sera conditionnée par les propriétés discriminantes et informatives de ces attributs.

Afin de traiter divers problèmes de classification, l'utilisateur dispose de plus ou moins de libertés pour le choix de ces features primitifs, selon la nature des données qu'il doit traiter.
Par exemple :

- Dans le cas où le signal correspond à un flux binaire, lorsque l'on souhaite par exemple analyser les diverses couches du protocole de communication TCP/IP par paquets, les features élémentaires peuvent alors correspondre exactement aux éléments binaires reçus par la carte réseau.
- Dans le cas du problème de la détection de SPAM dans les courriers électroniques, les features peuvent correspondre par exemple au pourcentage d'occurences de mots dans le texte, mais aussi à la nature du document (texte, page html, pièces jointes, ...) .
- Dans le cas particulier des images numériques, la notion de feature devient plus complexe. Il n'y a, en effet, pas de définition intrinsèque pour des features sur des images numériques si ce n'est la donnée des valeurs exactes en niveaux de gris en chaque pixel d'une image. Mais on perçoit vite la limite d'une telle représentation : il y a vraisemblablement des zones d'une image beaucoup plus informatives que d'autres et l'utilisation d'autres critères géométriques comme la fermeture, la convexité, l'alignement, la présence de bords orientés ou enfin les caractéristiques comme les couleurs, le nombre de composantes connexes peuvent se substituer avantageusement à la manipulation de la totalité de ces niveaux de gris.

Le calcul de la valeur d'un feature sur un exemple issu d'une base de données impose également la connaissance d'une règle quasi-instantanée pour son calcul. De tels features seront alors vus au sens statistique, c'est-à-dire comme étant la réalisation d'une variable aléatoire sur l'espace $\mathcal{I}$ des données qui peuvent être l'ensemble des images, des gènes, des messages électroniques, ...

Il est à noter qu'un tel mode de calcul de features sur un élément $\mathcal{I}$ peut parfois imposer, selon la nature des propriétés d'invariance sur $\mathcal{I}$, des invariances structurelles sur les détecteurs

et le calcul des variables aléatoires.

Par exemple, si l'on considère le cas particulier des images et de la reconnaissance de formes et la classification d'objets, on désire construire des détecteurs qui renvoient la même réponse, indépendamment de la position de l'objet dans l'image. Cela signifie donc que le détecteur construit doit renvoyer la même réponse de classification de façon invariante quelle que soit la translation que l'on pourrait appliquer à l'image. Ces propriétés d'invariance des détecteurs seront discutées dans la section suivante concernant le cas particulier de la détection d'objets dans une image.

2.1.2 Cas particulier des images

Une image est « la reproduction exacte ou représentation analogique d'un être ou d'une chose ». Mathématiquement, une telle reproduction est bien entendu impossible, on peut tout de même représenter de façon abstraite une image comme une application de $\mathbb{R}^2$ (ou $\mathbb{R}^3$ si l'on manipule des images en 3 dimensions) dans $[\,0\,;256\,]$ si l'image est en niveaux de gris continus ou dans $[\,0\,;256\,]^3$ si l'image est en couleur.

Du point de vue du traitement de l'image par ordinateur (image numérique), l'espace est alors discrétisé par une grille dont les noeuds sont appelés pixels et l'image en niveau de gris est en fait donnée comme une application de l'ensemble des pixels dans $[\![\,0\,;\,255\,]\!]$ où $[\![\,a\,;\,b\,]\!]$ désigne l'ensemble de tous les entiers compris entre a et b au sens large. La conversion d'une image analogique en image numérique nécessite donc deux opérations :

- la discrétisation des coordonnées spatiales (dépendant de la résolution fixée par l'utilisateur).
- la discrétisation de l'amplitude, c'est-à-dire la quantification en niveaux de gris (8 bits pour une amplitude variant dans $[\![\,0\,;\,255\,]\!]$) ou en couleurs (trois canaux variant sur 8, 24 ou 32 bits).

Dans tout notre mémoire, l'ensemble des images manipulées seront des images à deux dimensions codées en niveaux de gris de taille variable selon les bases de données étudiées. La taille des images numériques (taille de la grille définissant les pixels) sera notée génériquement $\mathrm{N}_x \times \mathrm{N}_y$ où N_x et N_y désignent le nombre de coordonnées horizontales et verticales.

Le point de vue que nous allons adopter est le point de vue probabiliste classique ([GG84]) : une image I en niveau de gris est vue comme la réalisation d'une variable aléatoire dans l'espace $\mathcal{I}$ des applications de $[\![\,0\,;\,\mathrm{N}_x\,]\!] \times [\![\,0\,;\,\mathrm{N}_y\,]\!]$ dans $[\![\,0\,;\,255\,]\!]$.

2.1.2.1 Détecteurs de bords « positifs » dans les images

Nous avons pris le parti, dans ce mémoire, de ne prendre comme features élémentaires sur les images numériques que des détecteurs de bords, ceci en raison de la facilité d'interprétation de la sélection de plusieurs détecteurs pour les tâches de classification. Ces détecteurs de bords sont par ailleurs locaux, et permettront plus tard de construire des features plus complexes, possédant des propriétés d'invariance notamment par petite translation sur les images.

La définition des features élémentaires que l'on utilisera tout au long du mémoire pour les images reprend ce qui a été fait dans [AG97a]. Ces détecteurs de bords sont très simples, et extrêmement rapides à calculer. Ce sont des fonctions booléennes qui possèdent de grandes propriétés d'invariance par rapport à la modification de l'intensité lumineuse, ainsi que par transformation croissante du niveau de gris.

Ces détecteurs de bords se décomposent en deux familles :

- Les détecteurs « positifs » du type $\varepsilon_1, \dots \varepsilon_8$ qui renvoient 1 lorsqu'un bord est détecté, et 0 si aucun bord n'est détecté.
- Les détecteurs « négatifs » du type $\varepsilon_9, \dots \varepsilon_{16}$ qui renvoient 1 lorsque précisément un bord n'est pas détecté.

Cet ensemble de détecteurs de bords primitifs aboutira alors à un premier ensemble de features $\mathcal{D}_0{}^+$ qui correspondra en réalité à la construction du « dictionnaire » initial des tests positifs, « dictionnaire » qui sera converti en « forêt » dans le chapitre V, tous ces termes restant très largement à définir.

Le but de notre algorithme final de sélection et composition de features sera alors de poursuivre la construction dynamique de ces « dictionnaires », permettant d'éclaircir la tâche ardue de classification d'une image I dans une des classes $\mathcal{C}_i$. On remarquera également que, plus généralement, cet algorithme pourra se généraliser à d'autres problèmes de classifications d'objets dans un signal.

Nous renvoyons à l'annexe A pour la définition précise de ces détecteurs de bords, issus des travaux de Geman et Amit ([AG97a]).

2.1.2.2 Sélection de détecteurs de bords pour une base de données

Dans les deux cas qui nous intéresseront, les images que nous aurons à traiter seront issuses de deux bases de données possédant un ensemble d'apprentissage clairement défini. Ces deux bases de données mettent en scène pour la première base des visages et des images de « fond » [MIT], tandis que la seconde base correspond à une liste de chiffres manuscrits issus de [USP] correspondant aux chiffres des codes postaux scannés par l'US Postal.

Que ce soit pour la détection de visages ou la reconnaissance de chiffres manuscrits, il est opéré une première pré-sélection des détecteurs de bords positifs possibles que l'on peut appliquer à une image. Cette première pré-sélection aboutit alors à la construction d'un dictionnaire de tests $\mathcal{D}_0^+$. Cette sélection de tests est détaillée dans l'annexe A. La grande quantité de détecteurs de bords obtenus à l'issue de cette sélection (plus de 2000 tests pour des images de taille à peine 20×20 pixels) permet donc de disposer d'une grande quantité d'information, ce qui est un avantage réel pour des problèmes de séparation de classes.

2.1.2.3 Détecteurs primitifs de bords négatifs

Afin de pouvoir s'autoriser la discrimination de certaines classes, on voit qu'il est nécessaire de considérer des tests négatifs, c'est-à-dire des détecteurs d'absence de bords en certaines zones de l'image. Ces détecteurs d'absence de bords sont également définis à partir des ε_i précédents. On ajoutera donc comme features primitifs les détecteurs $\varepsilon_9, \dots \varepsilon_{16}$ qui seront définis par

$$\forall i \in \{9, \dots 16\} \quad \forall \mathrm{I} \in \mathcal{I} \qquad \varepsilon_i(\mathrm{I}) = 1 - \varepsilon_{i-8}(\mathrm{I})$$

Ces détecteurs correspondent donc aux « non » logiques des variables booléennes ε_i du paragraphe précédent.

On peut légitimement se demander à quoi peuvent servir ces détecteurs de bords négatifs, puisqu'ils sont obtenus directement à partir de la formule : $\varepsilon_i = 1 - \varepsilon_{i-8}$. Nous verrons pourquoi il peut être bienvenu de manipuler également ces tests, notamment lors de la phase de composition de features.

Pour définir $\mathcal{D}_0^-$, ensemble des détecteurs initiaux négatifs, il est nécessaire de se baser sur les détecteurs de $\mathcal{D}_0^+$. En effet, dans le cas particulier de nos images [USP], il est facile de comprendre

qu'il y a de très nombreux tests négatifs qui sont réalisés, notamment dans les régions des images qui ne sont pas informatives. Afin de limiter la quantité de tests négatifs retenus, on définit donc

Définition 2.1.1 (Dictionnaire initial $\mathcal{D}_0{}^-$)
L'ensemble $\mathcal{D}_0{}^-$ des détecteurs élémentaires négatifs initiaux est donné par des variables aléatoires δ dépendant d'une orientation codée par ε, d'un flou f et d'une localisation c_x, c_y dans la grille de pixels tels que

$$\delta_{\varepsilon,f,c_x,c_y} \in \mathcal{D}_0{}^- \Longleftrightarrow \begin{cases} \exists\, i \in \{1 \dots \mathrm{C}\} \qquad \mathrm{P}_{\mathcal{C}_i}(\delta_{\varepsilon,f,c_x,c_y} = 1) \geqslant 1/2 \\ \varepsilon \in \{\varepsilon_9, \dots \varepsilon_{16}\} \\ f = \max\left\{\widetilde{f} \,|\, \delta_{\varepsilon,\widetilde{f},c_x,c_y} \in \mathcal{D}_0{}^-\right\} \\ \delta_{\varepsilon-8,\widetilde{f},c_x,c_y} \in \mathcal{D}_0{}^+ \end{cases}$$

Ce dictionnaire est donc l'ensemble des tests négatifs $\overline{\delta}$ réalisés avec une probabilité supérieure à $1/2$ sur une classe au moins des données manipulées et tels que le test « opposé » (qui est un test positif) δ soit réalisé avec une probabilité également $1/2$ sur une autre classe de données.

Au final, nous obtenons un premier dictionnaire de features élémentaires par réunion des deux ensembles de détecteurs précédents.

On peut de plus représenter l'ensemble des détecteurs élémentaires sélectionnés par le critère précédent. Plus l'image est foncée en un pixel donné, plus le détecteur représenté est précis. Cela se traduit pour les tests positifs par une valeur du flou petite tandis que pour les tests négatifs, cela signifie que le flou est grand :

ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7	ε_8	ε_9	ε_{10}	ε_{11}	ε_{12}	ε_{13}	ε_{14}	ε_{15}	ε_{16}
															

2.1.3 Composition de détecteurs élémentaires, agrégation de détecteurs

L'objectif de ce travail a été de trouver de nombreuses combinaisons de détecteurs élémentaires, la plupart du temps binaires, pour obtenir de meilleurs résultats de classification et une meilleure concision du dictionnaire représenté par l'agrégation de ces détecteurs élémentaires.

Ces arrangements de détecteurs élémentaires peuvent se construire, que l'on traite le cas particulier des images ou d'autres formes de signaux.

Il est possible de formaliser cette agrégation de détecteurs élémentaires. Nous appelerons « alphabet » et « mot » issu de l'alphabet les entités suivantes

Définition 2.1.2 (Alphabet $\mathcal{A}$)
$\mathcal{A}$, alphabet associé au problème de classification, est l'ensemble des détecteurs élémentaires dont on dispose. Si l désigne alors une lettre de $\mathcal{A}$, on notera $l(\mathrm{I})$ l'évaluation de l sur I élément de $\mathcal{I}$.

On peut illustrer cette définition dans quelques cas particuliers :

- En ce qui concerne le problème de classification d'objets dans une image, l'alphabet $\mathcal{A}$ est formé de l'ensemble des tests $\delta_{\varepsilon,f,c_x,c_y}$, où $\varepsilon \in \{\varepsilon_1, \dots, \varepsilon_{16}\}$, $f \in [\![\, 0\,;\, \mathrm{F}\,]\!]$ et $(c_x, c_y) \in [\![\, 0\,;\, \mathrm{N}_x - 1\,]\!] \times [\![\, 0\,;\, \mathrm{N}_y - 1\,]\!]$. La valeur de $l(\mathrm{I})$ est alors binaire : 1 s'il y a une détection, 0 sinon.

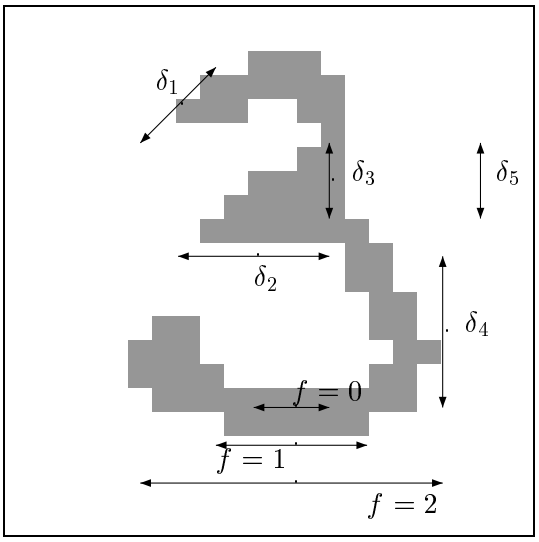
- En ce qui concerne le problème de la détection du SPAM dans des emails, l'alphabet est formé de l'ensemble des tests calculant les pourcentages d'occurences de certains mots spécifiques. $l(\mathrm{I})$ est alors un réel de $[\,0\,;1\,]$.

Définition 2.1.3 (Mots $\mathcal{A}^\star$)
Un mot m issu de $\mathcal{A}$ est formé d'une succession sans ordre précis et sans répétition de lettres de l'alphabel $\mathcal{A}$. L'évaluation d'un mot $m(l) = l_1 \dots l_p$ sur un élément de la base de données $\mathcal{I}$ sera alors

$$\forall \mathrm{I} \in \mathcal{I} \qquad m(\mathrm{I}) = \psi(l_1(\mathrm{I}), \dots, l_p(\mathrm{I}))$$

L'application ψ est une application directement dépendante de la nature des données extraites par les applications $l_1, \dots l_p$. L'ensemble des mots possibles sera noté $\mathcal{A}^\star$.

- Dans le cas de variables aléatoires binaires ou ternaires l_i, l'application $\psi(l_1, \dots l_p)$ peut alors correspondre simplement au produit de telles variables l_i puisque cette opération possède alors un sens logique parfaitement défini. La multiplication de deux variables binaires correspond alors au « et » logique tandis que la multiplication de deux variables ternaires correspond au « et ou ni-ni ».
- Dans le cas de variables réelles, la multiplication de telles variables ne possède plus de sens logique précis. En réalité, cette application ψ correspond alors plutôt à l'exploration de noyaux polynômiaux comme ce qui est pratiqué dans les algorithmes de Support Vector Machine (nous utiliserons désormais l'abbréviation classique SVM) pour des données autres que des données binaires. En effet, si x_1 et x_2 désignent deux variables, la possibilité de concaténer x_1 et x_2 revient à manipuler la variable $x_1 x_2$. L'espace dans lequel sont alors quantifiées les données est l'ensemble des polynômes à $|\mathcal{D}_0|$ variables de degré 2 si on autorise la concaténation d'au plus deux variables ou de degré supérieur sinon.



En utilisant la figure précédente, on voit donc dans le cas de variables binaires que le mot $m = \delta_1\delta_2\delta_3\delta_4$ est réalisé sur l'image tandis que le mot $\widetilde{m} = \delta_1\delta_2\delta_5$ ne l'est pas. En revanche, le mot $m' = \delta_1\delta_2\overline{\delta_5}$ est lui bien vrai sur l'image.

2.2 Représentation des mots sous forme d'arbres binaires

2.2.1 Définitions

La définition précédente des mots de $\mathcal{A}^\star$ nous suggère une représentation pratique des mots sous forme d'arbres binaires. Une agrégation de features sera désormais représentée par un arbre binaire grâce à l'algorithme de construction récursif suivant :

- Feature élémentaire : si le feature m est en fait une lettre de $\mathcal{A}$ (m est de longueur 1), on représente m par l'arbre $a(m)$:

$$a(m) = \begin{array}{c} \mathrm{m} \\ \varnothing \quad \varnothing \end{array}$$

- Feature composé : si le feature m est issu de l'agrégation de deux features « fils » m_g et m_d représentés par les arbres $a(m_g)$ et $a(m_d)$, alors $a(m)$ vaut :

$$a(m) = \begin{array}{c} \mathrm{m} \\ a(m_g) \quad a(m_d) \end{array} \qquad (\mathbf{A})$$

L'évaluation du feature représenté par l'arbre de (**A**) est alors fondé sur le noeud principal m : on pose donc

Définition 2.2.1 (Évaluation d'un arbre binaire sur une donnée)

$$\forall \mathrm{I} \in \mathcal{I} \qquad a(m)(\mathrm{I}) = m(\mathrm{I})$$

Dans la suite de la construction et sélection des features pour la classification, on manipulera de préférence l'arborescence complète d'un feature plutôt que le noeud principal. Cela peut paraître paradoxal du point de vue de la complexité algorithmique puisque l'évaluation de l'arbre $a(m)$ sur $\mathcal{I}$ ne dépend en fait que de m, mais la suite de l'algorithme nécessitera une « mémoire » sur la façon dont ont été construits les features.

Enfin, notons que les répétitions de lettres dans les noeuds principaux des deux fils ne sont pas répétées dans le noeud principal de l'arbre père, ainsi l'agrégation formelle de

$$\mathcal{A}_1 = \begin{array}{c} \mathrm{ab} \\ \mathrm{a} \quad \mathrm{b} \end{array} \qquad \text{et} \qquad \mathcal{A}_2 = \begin{array}{c} \mathrm{ac} \\ \mathrm{a} \quad \mathrm{c} \end{array}$$

donne

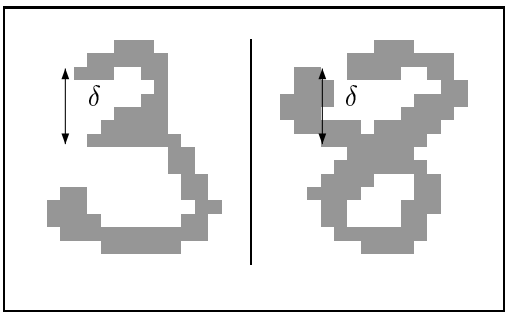
$$\mathcal{A}_1 :: \mathcal{A}_2 = \begin{array}{c} \mathrm{abc} \\ \mathrm{ab} \qquad \mathrm{ac} \\ \mathrm{a} \quad \mathrm{b} \quad \mathrm{a} \quad \mathrm{c} \end{array}$$

Définition 2.2.2 (Ensemble d'arbres binaires)
Nous appelerons $\mathcal{A}^\star$ l'ensemble des arbres binaires que l'on peut construire à partir de l'ensemble des features élémentaires présents dans l'alphabet $\mathcal{A}$.

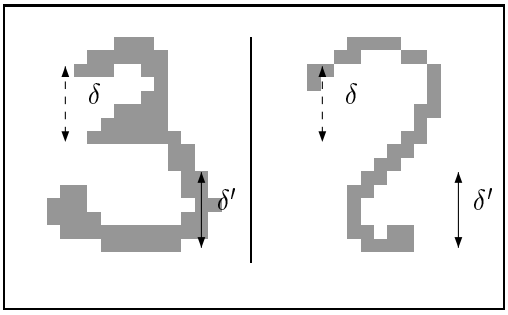
2.2.2 Motivation pour l'utilisation de tests élémentaires négatifs $\varepsilon_9, \ldots \varepsilon_{16}$

L'utilisation de tests élémentaires négatifs nous permet, *via* la composition de features évoquée précédemment, d'utiliser le caractère d'absence de bords dans des features composés. Cette utilisation de tests négatifs peut être intéressante du fait que la composition de tests pour obtenir des features composés permet d'engendrer une réutilisation de ces features pour d'autres classes que celles qui ont permis de les former.

Par exemple, la classe $\mathcal{C}_8$ peut être discriminée par rapport à la classe $\mathcal{C}_3$ par un détecteur de bord δ mais également par $\overline{\delta}$:



Mais le test $\overline{\delta}$ peut alors être réutilisé pour discriminer la classe $\mathcal{C}_3$ par rapport à la classe $\mathcal{C}_2$ dans la composition $\overline{\delta}\delta'$:



tandis que le test δ ne peut être réutilisé pour discriminer les deux classes précédentes. Ainsi, c'est plutôt en vue d'une réutilisation des features composés formés au temps t dans un temps ultérieur à t que l'on manipule les détecteurs de bords « négatifs ».

2.2.3 Détecteurs de bords invariants par translation

2.2.3.1 Invariance par translation

Dans le cas où l'on étudie des problèmes d'images numériques et où les données ne sont pas déjà centrées sur la grille des pixels, il peut être nécessaire de gérer l'invariance par translation de ces détecteurs. Plus précisément, si m désigne un mot du vocabulaire de features disponibles et I une image de la base de données, on peut souhaiter imposer que quelle que soit l'opération de translation $\overrightarrow{\tau}$ effectuée sur I, le résultat renvoyé par $m(\overrightarrow{\tau}(\mathrm{I}))$ soit indépendant de $\overrightarrow{\tau}$.

Cette nécessaire invariance par translation des détecteurs est issue du fait que lors de l'extraction du signal analogique et la conversion en image numérique, la pose n'influence pas la nature du signal et ne doit donc pas influencer l'interprétation par tout algorithme de traitement de l'image.

Dans notre situation (images issues de [MIT] ou [USP]), les images sont préalablement centrées et nous n'avons donc pas à implémenter cette invariance par translation. Cependant, nous pouvons donner les pistes qui permettent d'implémenter une telle invariance par translation des features construits.

Il s'agit tout d'abord de construire des features élémentaires invariants par translation, ces détecteurs élémentaires doivent donc renvoyer le même résultat quelle que soit la pose de l'image. Si l'on définit la relation d'équivalence $\sim_\tau$, sur l'ensemble des images par :

$$\forall \mathrm{I}_1, \mathrm{I}_2 \in \mathcal{I} \qquad \mathrm{I}_1 \sim_\tau \mathrm{I}_2 \Longleftrightarrow \exists\, \overrightarrow{\tau} \quad | \quad \overrightarrow{\tau}(\mathrm{I}_1) = \mathrm{I}_2$$

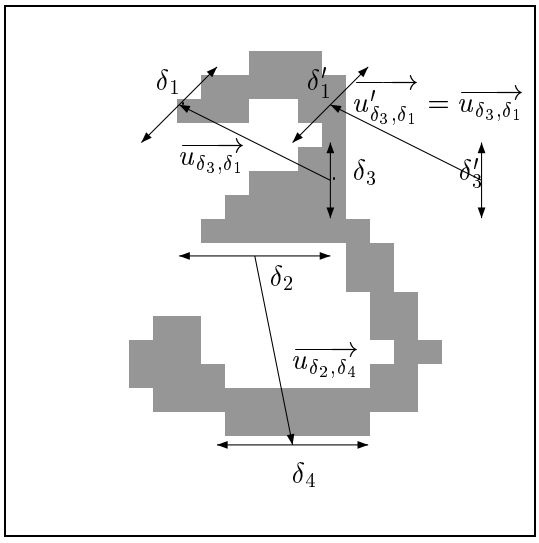
les détecteurs peuvent être définis sur le quotient $(\mathcal{I}/\sim_\tau)$ ensemble des images quotientées par la relation d'équivalence $\sim_\tau$.

Enfin, nous constatons immédiatement que pour qu'un détecteur élémentaire ait un sens, il faut nécessairement que ce détecteur soit composé d'au moins deux tests de bords. Ainsi, les détecteurs élémentaires que nous considèrerons sont de la forme :

$$((\varepsilon_1, f_1), (\varepsilon_2, f_2), \overrightarrow{u_{12}})$$

où le vecteur $\overrightarrow{u_{12}}$ quantifie la position du test (ε_1, f_1) par rapport à (ε_2, f_2), ε_1 et ε_2 sont les deux orientations des tests de bords tandis que f_1 et f_2 sont les deux valeurs des flous associées à ces tests de bords. Nous parlerons également de l'orbite d'un tel détecteur élémentaire puisqu'en somme, si la paire de tests (δ_1, δ_2) est obtenue par translation quelconque à partir d'une autre paire de tests (δ_3, δ_4) alors ces deux paires de tests renverront la même valeur sur l'ensemble des images $\mathcal{I}$.

Nous pouvons alors représenter la réalisation de deux tests élémentaires invariants par translation dans l'image de la section 2.6.3 par :



Il faut voir en réalité ce passage au quotient comme la possibilité de « déplacer » la paire de tests δ_1, δ_3 dans toute l'image en imposant que le vecteur de translation $\overrightarrow{u_{\delta_3,\delta_1}}$ reste constant. La

paire de tests est alors réalisée lorsqu'il existe une position dans l'image qui réalise à la fois δ_1 et δ_3.

Par ailleurs, (δ_1, δ_3) et (δ'_1, δ'_3) appartiennent à la même orbite de tests invariants par translation.

2.2.3.2 Dictionnaire invariant par translation

En énumérant comme dans le paragraphe 2.6.2.2 toutes les paires de tests positifs possibles, on obtient un premier dictionnaire de tests invariants par translation consituté uniquement de conjonction de bords. Ce dictionnaire est encore noté $\mathcal{D}_0^+$. Puis nous construisons le dictionnaire complet en enrichissant $\mathcal{D}_0^+$ de conjonctions de tests positifs et négatifs issus de $\mathcal{D}_0^+$:

$$\mathcal{D}_0^- = \left\{ \left((\varepsilon, f), (\varepsilon', f'), \overrightarrow{u} \right) \quad | \quad \left((\varepsilon - 8, f), (\varepsilon', f'), \overrightarrow{u} \right) \in \mathcal{D}_0^+ \right\}$$

Le dictionnaire initial est alors formé là aussi de la réunion des deux dictionnaires précédents :

$$\mathcal{D}_0 = \mathcal{D}_0^+ \cup \mathcal{D}_0^-$$

2.2.3.3 Agrégation de détecteurs invariants par translation

Dans le cas où l'on décide d'effectuer une agrégation de features invariants par translation, l'agrégation arborescente du paragraphe 2.7.1 n'est pas bien posée puisque l'objet naturel obtenu n'est pas la réalisation d'une agrégation de tests de bords modulo une translation. Pour maintenir une telle représentation, il est nécessaire de fixer une pose entre les deux orbites de tests θ_1 et θ_2 que l'on décide de « fusionner ». Le choix d'un tel vecteur $\overrightarrow{u}$ positionnant l'orbite θ_1 par rapport à θ_2 pour former $(\theta_1, \theta_2, \overrightarrow{u})$ peut se faire selon différents critères. Nous proposons le critère suivant, adapté à l'étude de la détection d'objets. Si $\mathcal{D}$ désigne le dictionnaire d'orbites de tests, on choisit d'enrichir $\mathcal{D}$ de l'orbite $(\theta_1, \theta_2, \overrightarrow{u})$ sous la condition :

$$(\theta_1, \theta_2, \overrightarrow{u}) \notin \mathcal{D}$$

et $\overrightarrow{u}$ est le vecteur positionnant θ_1 par rapport à θ_2 qui maximise la probabilité de réussite de $(\theta_1, \theta_2, \overrightarrow{u})$ sur une classe des données.

2.3 Mesure de l'information commune

L'objectif central de notre travail sera de construire un ensemble composé de mots de $\mathcal{A}^\star$ à partir des lettres de $\mathcal{A}$. Nous avons vu que les lettres de $\mathcal{A}$ étaient considérées comme des variables aléatoires sur $\mathcal{I}$. Former un mot à partir de plusieurs lettres revient donc à considérer la loi jointe ou la loi du produit de plusieurs variables aléatoires. Afin d'optimiser la recherche de mots dans $\mathcal{A}^\star$ améliorant la tâche de classification d'objets, nous allons nous doter de critères de formations de mots à partir de lettres d'autres mots. En effet, l'absence de stratégie de parcours de $\mathcal{A}^\star$ paraît peu efficace dans la mesure où

$$| \mathcal{A}^\star | \geqslant | \mathcal{A} | !$$

et l'énumération de tous les éléments de $\mathcal{A}^\star$ est donc impossible.

Nous allons envisager plusieurs façons de mesurer la valeur informative de deux variables aléatoires X et Y afin de privilégier la construction de mots à partir de variables aléatoires possédant de fortes propriétés de corrélation, soit en terme de quantité d'information, soit en terme de corrélation statistique.

2.3.1 Cas de variables aléatoires binaires

Dans le cas où les variables aléatoires sont binaires, le produit de deux détecteurs revient finalement au « et » logique. La concaténation de deux détecteurs pour δ_1 et δ_2 a cela d'agréable qu'elle permet d'interpréter facilement la réalisation de la nouvelle variable aléatoire. Mais il ne s'agit pas d'opérer des concaténations au hasard de certaines variables. En effet, la concaténation naïve de deux features complètement indépendants ne permettra peut être pas d'obtenir un gain de performance. Pour parvenir à une richesse de vocabulaire, il faut au moins que la concaténation de deux mots permette de s'adapter au maximum à une ou plusieurs classes de la base de données.

2.3.1.1 Définitions et propriétés des corrélations fonctionnelles

On procède aux calculs des corrélations fonctionnelles des tests contre chacune des classes $\mathcal{C}_i$, idée qui est développée dans [Fle00] . Si l'on note P loi de probabilité sur les données, on peut définir les corrélations fonctionnelles par :

Définition 2.3.1 (Corrélations fonctionnelles)
Si X *et* Y *sont deux variables aléatoires, on note le coefficient de corrélation :*

$$\rho_{\mathrm{P}}(\mathrm{Y};\mathrm{X}) = \frac{\mathrm{C}ov_{\mathrm{P}}(\mathrm{X},\mathrm{Y})}{\sigma_{\mathrm{P}}(\mathrm{X})\sigma_{\mathrm{P}}(\mathrm{Y})}$$

On pourra se référer à [Sap90] pour de nombreux détails sur ce coefficient. Dans la définition précédente, les quantités $\mathrm{C}ov_{\mathrm{P}}$ et σ_{P} sont les covariance et écart type relativement à la distribution empirique P. Ce que nous retiendrons de ce coefficient est résumé dans ce qui suit.

Propriété 2.3.1
Étant données deux variables aléatoires X *et* Y*, on a :*

$$0 \leqslant \rho_{\mathrm{P}}(\mathrm{Y};\mathrm{X})^2 \leqslant 1$$

Propriété 2.3.2 (Cas de variables aléatoires binaires)
Étant données deux variables aléatoires X *et* Y *binaires, si* V *désigne la variance, on a :*

$$\rho_{\mathrm{P}}(\mathrm{X};\mathrm{Y})^2 = \frac{\mathrm{V}(\mathbb{E}_{\mathrm{P}}[\mathrm{Y}|\mathrm{X}])}{\mathrm{V}_{\mathrm{P}}(\mathrm{Y})}$$

et si $\rho_{\mathrm{P}}(\mathrm{Y};\mathrm{X})^2 = 1$, Y *est fonctionnellement lié à* X P*-presque sûrement, c'est-à-dire il existe une fonction* g *non constante telle que* $\mathrm{Y} = g(\mathrm{X})$ P*-presque sûrement. De plus, on a dans ce cas*

$$\mathrm{Y} = \mathrm{X} \qquad ou \qquad \mathrm{Y} = \overline{\mathrm{X}}$$

Propriété 2.3.3 (Cas de variables aléatoires binaires)
Étant données deux variables aléatoires X *et* Y*, si* $\rho_{\mathrm{P}}(\mathrm{Y};\mathrm{X}) = 0$, $\mathbb{E}_{\mathrm{P}}(\mathrm{Y}|\mathrm{X})$ *est presque sûrement une constante.*

Propriété 2.3.4 (Cas de variables aléatoires binaires)
Étant données deux variables aléatoires X *et* Y*,* $\rho_{\mathrm{P}}(\mathrm{Y};\mathrm{X})$ *est le cosinus de l'angle formé par* $\mathrm{Y} - \mathbb{E}_{\mathrm{P}}[\mathrm{Y}]$ *avec le sous-espace de dimension 2 de* $\mathrm{L}^2_{\mathrm{X}}$ *engendré par la variable* X *et la droite des constantes.*

Propriété 2.3.5 (Cas de variables aléatoires binaires)
Si X *et* Y *sont deux variables aléatoires à valeurs dans* $\{0;1\}$*, alors en notant* $p_{\mathrm{X}} = \mathrm{P}(\mathrm{X}=1)$, $p_{\mathrm{Y}} = \mathrm{P}(\mathrm{Y}=1)$ *et* $p_{\mathrm{XY}} = \mathrm{P}(\mathrm{X}=1 \text{ et } \mathrm{Y}=1)$*, on a :*

$$\rho_{\mathrm{P}}(\mathrm{Y};\mathrm{X}) = \frac{p_{\mathrm{XY}} - p_{\mathrm{X}} p_{\mathrm{Y}}}{\sqrt{p_{\mathrm{X}}(1-p_{\mathrm{X}}) p_{\mathrm{Y}}(1-p_{\mathrm{Y}})}} = \rho_{\mathrm{P}}(\mathrm{X};\mathrm{Y})$$

- Par conséquent, on dira que deux tests (variables aléatoires) sont fortement corrélés si le rapport $\rho_{\mathrm{P}}(\mathrm{X};\mathrm{Y})$ est proche de 1.
  Inversemement, deux tests sont anti-corrélés si $= \rho_{\mathrm{P}}(\mathrm{X};\mathrm{Y})$ est proche de -1. Le rapport de corrélation est lui maximal si Y est lié fonctionnellement à X, c'est-à-dire lorsque $\mathrm{Y} = g(\mathrm{X}) \quad \mathrm{P} - p.s.$
- Par ailleurs, Y est non corrélé à X s'il y a absence de dépendance en moyenne. C'est en particulier le cas lorsque X et Y sont indépendantes mais la réciproque est inexacte. En effet $\rho_{\mathrm{P}}(\mathrm{X};\mathrm{Y}) = 0$ signifie seulement que $\mathrm{Y} - \mathrm{E}_{\mathrm{P}}(\mathrm{Y})$ est orthogonal à $\mathrm{L}^2_{\mathrm{X}}$.
- Enfin, on peut remarquer une dernière propriété triviale :

$$\rho_{\mathrm{P}}(\mathrm{X};\mathrm{Y}) = -\rho_{\mathrm{P}}(\overline{\mathrm{X}};\mathrm{Y})$$

2.3.1.2 ρ-décomposabilité pour des arrangements de features élémentaires

Nous pouvons donc utiliser ces notions pour aménager la notion de ρ-décomposabilité utilisée dans [Fle00]. Cet aménagement est dû précisément au fait que le problème de classifiction est multi-classe dans notre cas alors qu'il est à deux classes pour les données étudiées dans [Fle00]. Il est tout d'abord nécessaire de définir la profondeur des features (représentés sous forme d'arbres) de notre ensemble $\mathcal{F}$.

Définition 2.3.2 (Profondeur des features)
On calcule la profondeur $|\delta|$ d'un feature δ donné sous la forme :

$$\delta = \quad \begin{array}{c} \delta \\ \diagup \quad \diagdown \\ \delta.g \quad \delta.d \end{array}$$

par la définition récursive

$$|\delta| = 1 + \max\{|\delta.g|; |\delta.d|\}$$

avec en plus

$$|\varnothing| = 0$$

Nous notons $\mathrm{P}_{\mathcal{C}_i}$ la loi des détecteurs sous l'hypothèse que l'image étudiée appartient à la classe $\mathcal{C}_i$. Pour un problème de détection à plusieurs classes, la notion de ρ-décomposabilité sera définie conditionnellement à une loi $\mathrm{P}_{\mathcal{C}_i}$, ce qui revient pour un problème multi-classe à définir des corrélations relativement à une des classe $\mathcal{C}_i$.

Définition 2.3.3 (ρ_0-décomposabilité pour les classes $(\mathcal{C}_i)_{i \in [\![1\,;\, |\mathrm{C}|]\!]}$)

1. *Tout feature élémentaire δ(de profondeur égale à 1 exactement) est ρ_0-décomposable si*

$$\exists i \in [\![1\,;\, |\mathrm{C}|]\!] \qquad \mathrm{P}_{\mathcal{C}_i}(\delta = 1) \geqslant 1/2$$

2. *a(m), feature composé de profondeur strictement supérieur à 1, issu d'un arrangement écrit de façon générique :*

$$a(m) = \begin{array}{ccc} & m & \\ & \diagup\quad\diagdown & \\ a(m_g) & & a(m_d) \end{array}$$

est un arrangement ρ_0 décomposable si, et seulement si,

- *Les fils gauche et droit $a(m_g)$ et $a(m_d)$ vérifient*

$$\exists i \in [\![1\,;\, |\mathrm{C}|]\!] \qquad \begin{cases} \rho_{\mathcal{C}_i}(a(m_g); a(m_d)) \geqslant \rho_0 & (\mathbf{I}) \\ \mathrm{P}_{\mathcal{C}_i}(a(m_g) = 1) \geqslant \dfrac{1}{2}\rho_0^{|m|} & (\mathbf{a_g}) \\ \mathrm{P}_{\mathcal{C}_i}(a(m_d) = 1) \geqslant \dfrac{1}{2}\rho_0^{|m|} & (\mathbf{a_d}) \end{cases}$$

- *$a(m_g)$ et $a(m_d)$ sont ρ_0-décomposables.*

Ainsi, un arrangement de tests est ρ_0-décomposable s'il existe une classe de données pour laquelle chacun des tests fils est réalisé avec une probabilité suffisante et tel que la corrélation de ces deux fils soient supérieures à ρ_0.

Propriété 2.3.6 (Minoration de $\mathrm{P}_{\mathcal{C}_i}(a(m))$)
Si $a(m)$ est un arrangement ρ-décomposable, alors

$$\exists i \in [\![1\,;\, |\mathrm{C}|]\!] \qquad \mathrm{P}_{\mathcal{C}_i}(a(m) = 1) \geqslant \frac{1}{2}\rho_0^{|m|}$$

Preuve : La démonstration établie dans [Fle00], section 3.3 peut être reprise point par point pour établir le résultat par récurrence sur $|m|$. □

Ainsi, le fait de manipuler un arrangement ρ-décomposable assure qu'il va exister une classe d'objets qui assure la réalisation de l'arrangement avec une probabilité suffisante.

On peut également utiliser d'autres définitions pour la ρ-décomposabilité, dans la mesure où les minorations $(\mathbf{a_g})$ et $(\mathbf{a_d})$ de la définition 2.8.2 peuvent être substituées à des minorations un peu différentes qui se répercutent dans la proposition 2.8.6. On établit alors des minorations de $\mathrm{P}_{\mathcal{C}_j}(a(m) = 1)$ comparables à celles obtenues dans [Fle00]. Il est nécessaire de définir $\mathcal{D}_{0,j}$ l'ensemble des lettres élémentaires de $\mathcal{D}_0$ étant satisfaites avec une probabilité d'au moins 1/2 sur les données issus de $\mathcal{C}_j$.

Définition 2.3.4 (Dictionnaire $\mathcal{D}_{0,j}$)
$\mathcal{D}_{0,j}$ est l'ensemble des δ de $\mathcal{D}_0$ tels que

$$\mathrm{P}_{\mathcal{C}_j}(\delta = 1) \geqslant \frac{1}{2}$$

Muni de ces nouveaux ensembles de variables, on peut alors donner deux autres définitions de ce qu'on pourrait appeler ρ_0-décomposabilité des arrangements, ces définitions ne sont pas équivalentes à la définition précédente mais illustrent la même notion de lien de corrélation entre un père et ses deux fils sur une classe de la base de données.

Définition 2.3.5 (Deuxième définition de ρ_0-décomposabilité pour $(\mathcal{C}_i)_{i\in[\![1\,;\,|\mathrm{C}|]\!]}$)
La définition est là-encore récursive.

1. *Tout feature élémentaire δ est ρ_0-décomposable.*
2. *$a(m)$, feature composé (de profondeur supérieur strictement supérieur à 1), issu d'un arrangement écrit de façon générique :*

$$a(m) = \begin{array}{ccc} & m & \\ a(m_g) & & a(m_d) \end{array}$$

est un arrangement ρ_0 décomposable si, et seulement si,

- *Les fils gauche et droit $a(m_g)$ et $a(m_d)$ vérifient*

$$\exists i \in [\![1\,;\, |\mathrm{C}|]\!] \qquad \begin{cases} \rho_{\mathcal{C}_i}(a(m_g); a(m_d)) \geqslant \rho_0 & (\mathbf{I}) \\ \mathrm{P}_{\mathcal{C}_i}(a(m_g) = 1) \geqslant \left[\underset{\delta \in \mathcal{D}_{0,j}}{\mathrm{Min}} \mathrm{P}_{\mathcal{C}_j}(\delta = 1) \right]^{|m_g|} & (\mathbf{a'_g}) \\ \mathrm{P}_{\mathcal{C}_i}(a(m_d) = 1) \geqslant \left[\underset{\delta \in \mathcal{D}_{0,j}}{\mathrm{Min}} \mathrm{P}_{\mathcal{C}_j}(\delta = 1) \right]^{|m_d|} & (\mathbf{a'_d}) \end{cases}$$

- *$a(m_g)$ et $a(m_d)$ sont ρ_0-décomposables.*

Pour cette deuxième définition de la ρ_0-décomposabilité, on peut alors établir le pendant de la propriété 2.8.6 pour cette nouvelle définition :

Propriété 2.3.7 (Minoration de réalisation d'un arrangement)
Si $a(m)$ est un arrangement ρ-décomposable, alors

$$\exists\, i \in [\![1\,;\, |\mathcal{C}|]\!] \qquad \mathrm{P}_{\mathcal{C}_i}(a(m) = 1) \geqslant \left[\underset{\delta \in \mathcal{D}_{0,i}}{\mathrm{Min}} \right]^{|a(m)|}$$

La troisième définition différente (et non équivalente que nous donnons) est également récursive, et améliore encore un peu plus la borne minorante des arrangements

Définition 2.3.6 (Troisième définition de ρ_0-décomposabilité pour $(\mathcal{C}_i)_{i \in [\![1\,;|\mathrm{C}|]\!]}$)

1. *Tout feature élémentaire δ est ρ_0-décomposable.*
2. *$a(m)$, feature composé (de profondeur supérieur strictement supérieur à 1), issu d'un arrangement écrit de façon générique :*

$$a(m) = \begin{array}{ccc} & m & \\ a(m_g) & & a(m_d) \end{array}$$

est un arrangement ρ_0 décomposable si, et seulement si,

- *Les fils gauche et droit $a(m_g)$ et $a(m_d)$ vérifient*

$$\exists i \in [\![1\,;\, |\mathrm{C}|]\!] \qquad \begin{cases} \rho_{\mathcal{C}_i}(a(m_g); a(m_d)) \geqslant \rho_0 & (\mathbf{I}) \\ \mathrm{P}_{\mathcal{C}_i}(a(m_g) = 1) \geqslant \underset{\delta \in \mathcal{D}_{0,j}}{\mathrm{Min}} \mathrm{P}_{\mathcal{C}_j}(\delta = 1) \rho_0^{\log_2(|a(m_g)|)} & (\mathbf{a''_g}) \\ \mathrm{P}_{\mathcal{C}_i}(a(m_d) = 1) \geqslant \underset{\delta \in \mathcal{D}_{0,j}}{\mathrm{Min}} \mathrm{P}_{\mathcal{C}_j}(\delta = 1) \rho_0^{\log_2(|a(m_d)|)} & (\mathbf{a''_d}) \end{cases}$$

- *$a(m_g)$ et $a(m_d)$ sont ρ_0-décomposables.*

Cette définition assure alors une nouvelle fois que tout arrangement ρ_0-décomposable se réalise avec une probabilité suffisante sur au moins une des classes des données, c'est ce qu'exprime la propriété suivante :

Propriété 2.3.8 (Minoration de réalisation d'un arrangement)
i $a(m)$ est un arrangement ρ-décomposable, alors

$$\mathrm{P}_{\mathcal{C}_j}(a(m)=1) \geqslant \min_{i\in\mathcal{D}_{0,j}} \mathrm{P}_{\mathcal{C}_j}(m_i=1)\,\rho_0^{\log_2(|a(m)|)}$$

Les minorations obtenues des probabilités de réalisation des features sous $\mathrm{P}_{\mathcal{C}_i}$ expliquent l'apport d'efficacité de ces arrangements. On montre expérimentalement ([Fle00], section 3.4) que sous P_0 (loi de probabilité des tests élémentaires et composés sous l'hypothèse que la donnée appartient à une image de « fond »), la probabilité de réussite d'un arrangements $a(m)$ décroît exponentiellement avec $|m|$.

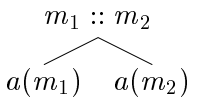
Il faut à présent examiner pourquoi les conditions $(\mathbf{a_g})$ et $(\mathbf{a_d})$ sont absolument fondamentales dans la définition 2.8.2. Imaginons que l'on dispose de deux arbres $a(m_1)$ et $a(m_2)$ tels que pour trois classes distinctes $\mathcal{C}_i, \mathcal{C}_j$ et $\mathcal{C}_k$:

$$(\mathbf{H})\begin{cases} \mathrm{P}_{\mathcal{C}_i}(a(m_1)=1) \geqslant \xi\rho^{|m_1|} \quad \text{et} \quad \mathrm{P}_{\mathcal{C}_i}(a(m_2)=1) \leqslant \varepsilon \\ \mathrm{P}_{\mathcal{C}_j}(a(m_2)=1) \geqslant \xi\rho^{|m_2|} \quad \text{et} \quad \mathrm{P}_{\mathcal{C}_i}(a(m_1)=1) \leqslant \varepsilon \\ \mathrm{P}_{\mathcal{C}_k}(a(m_1)=1) = \varepsilon \quad \text{et} \quad \mathrm{P}_{\mathcal{C}_k}(a(m_2)=1) = \varepsilon \\ \mathrm{P}_{\mathcal{C}_k}(a(m_2)=1 \,|\, a(m_1)=1) = 9/10 \end{cases}$$

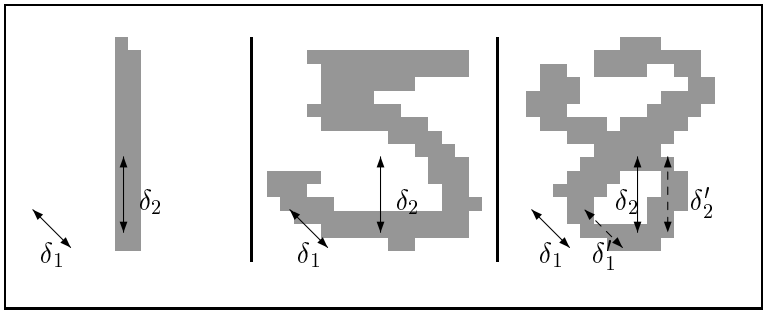
alors $$\rho_{\mathcal{C}_k}(a(m_1); a(m_2)) \sim 9/10\frac{1}{1-\varepsilon}$$

tandis que $$\mathrm{P}_{\mathcal{C}_k}(a(m_1)=1 \quad \text{et} \quad a(m_2)=1) = 9/10\varepsilon$$

Si l'on décide de respecter la règle de construction conditionnée par $(\mathbf{I})$ sans pour autant vérifier $(\mathbf{a_g})$ et $(\mathbf{a_d})$, on peut très bien décider de former l'arbre



par le biais de la classe $\mathcal{C}_k$ alors qu'en fait, l'arbre $a(m_1 :: m_2)$ n'est réalisé qu'avec une très faible probabilité (de l'ordre de ε) sur cette même classe. La réalisation des hypothèses $(\mathbf{H})$ peut se produire, par exemple dans le cas suivant :



Dans le cas des classes précédentes, et en considérant les deux lettres δ_1 et δ_2, on constate en effet que le test δ_1 possède une probabilité très forte d'être vrai sur la classe $\mathcal{C}_1$, et il en est de même pour δ_2 sur $\mathcal{C}_5$. Par ailleurs, l'arbre composé

est réalisé avec une très faible probabilité sur $\mathcal{C}_1$ et sur $\mathcal{C}_5$. Cependant, même si cet arbre est également peu probable sur $\mathcal{C}_8$, on constate que les deux features élémentaires δ_1 et δ_2 sont fortement corrélés sur $\mathcal{C}_8$: il suffit pour cela de considérer la petite translation transformant δ_1 en δ_1' et δ_2 en δ_2'.

Il faut donc imposer comme hypothèse supplémentaire que l'arrangement des features soit réalisé (dans la classe où les features sont corrélés) avec une probabilité suffisante. La définition de la ρ_0-décomposabilité pour un problème multi-classe implique donc la présence de conditions du type $(\mathbf{a_g})$ et $(\mathbf{a_d})$ dans la définition 2.8.2.

2.3.2 Cas de variables aléatoires réelles

Cette étude concerne donc le cas où les données ne sont plus binaires, mais réelles. En réalité, cette étude concerne en particulier le cas où les données sont ternaires, ce qui est le cas lorsque l'on considère des détecteurs de bords orientés comme des variables ternaires par exemple.

Dans le cas des variables réelles, l'approche par corrélation peut toujours être menée, mais il est alors impossible à partir d'un seuillage des corrélations de déduire une propriété sur la loi de probabilité d'un feature composé comme ce qui a été fait à la proposition 2.8.6. On peut cependant décider de poursuivre l'approche en imposant une définition construite sur la dernière remarque du paragraphe précédent.

Il est également possible de décider de la composition du feature $a(m)$ à partir de l'information mutuelle des variables aléatoires fils. Cette quantité est décrite en détails dans [CT91]. L'information mutuelle entre deux variables aléatoires est définie *via* la distance de Kullback Leibler :

Définition 2.3.7 (Distance de Kullback Leibler)
Si p et q désignent deux lois de probabilités sur Ω, alors la distance D

$$\mathrm{D}(p;q) = \sum_{x\in\Omega} p(x)\log\left(\frac{p(x)}{q(x)}\right) = \mathbb{E}_p\left[\log\left(\frac{p}{q}\right)\right]$$

est appelée distance de Kullback Leibler. Nous utilisons pour que cette définition ait un sens les conventions suivantes :

$$0\log\frac{0}{q} = 0 \quad et \quad p\log\frac{p}{0} = \infty \quad et \quad 0\log\frac{0}{0} = 0$$

Cette distance permet alors de définir l'information mutuelle entre deux variables aléatoires :

Définition 2.3.8 (Information mutuelle)
Soient X *et* Y *deux variables aléatoires de lois p et q et de loi jointe r, l'information mutuelle* $\mathrm{I}(\mathrm{X};\mathrm{Y})$ *est définie par*

$$\mathrm{I}(\mathrm{X};\mathrm{Y}) = \mathrm{D}(r(x,y);p(x)q(y)) = \sum_{x\in\Omega}\sum_{y\in\Omega'}\log\frac{r(x,y)}{p(x)q(y)}$$

On peut alors démontrer les résultats fondamentaux de la théorie de l'information.

Théorème 2.3.1
Pour toutes distributions de probabilités p et q, on a

$$\mathrm{D}(p;q) \geqslant 0$$

avec égalité si et seulement si $p = q$ pp.

Enfin, on peut établir les relations :

Théorème 2.3.2

- $$\mathrm{I}(\mathrm{X};\mathrm{Y}) \geqslant 0$$

  avec égalité si et seulement si X *et* Y *sont indépendantes*

- $$\mathrm{I}(\mathrm{X};\mathrm{Y}) = \mathrm{H}(\mathrm{X}) - \mathrm{H}(\mathrm{X}|\mathrm{Y}) = \mathrm{H}(\mathrm{Y}) - \mathrm{H}(\mathrm{Y}|\mathrm{X})$$

- $$\mathrm{I}(\mathrm{X};\mathrm{Y}) = \mathrm{H}(\mathrm{X}) + \mathrm{H}(\mathrm{Y}) - \mathrm{H}(\mathrm{X},\mathrm{Y})$$

- $$\mathrm{I}(\mathrm{X};\mathrm{X}) = \mathrm{H}(\mathrm{X})$$

- $$\mathrm{I}(\mathrm{X};\mathrm{Y}) \leqslant \min\{\mathrm{H}(\mathrm{X});\mathrm{H}(\mathrm{Y})\}$$

La quantité d'information mutuelle $\mathrm{I}(\mathrm{X};\mathrm{Y})$ permet donc de quantifier le degré d'indépendance de X vis-à-vis de Y. On s'efforcera, dans nos constructions de features composés, de faire que l'information mutuelle de deux fils construisant un père, soit la plus grande possible.

> Le calcul de l'information mutuelle lorsque les variables sont réelles n'est pas toujours aisé puisqu'il faut estimer la densité de la loi jointe r, ainsi que celle de chacune des variables réelles p et q. En fait, nous utiliserons à chaque fois un critère basé sur l'information mutuelle dans le cas des variables ternaires : l'évaluation de p, q et r est alors possibles puisqu'il s'agit d'énumérer 9 cas pour le calcul de r et 3 cas pour le calcul de p et q.

Nous avons donc vu une manière de structurer l'ensemble des features sous la forme d'arbres. Cette structure est légèrement différente de celle de [AG97a] puisque le parcours des arbres en profondeur n'est pas nécessaire pour évaluer chaque feature.

Nous expliquerons dans le chapitre 5 la nécessité d'une représentation arborescente des features manipulés.

Enfin, nous avons donné les éléments qui permettront de mettre un « poids » sur chaque arbre en définissant des quantités comme la ρ-corrélation ou l'information mutuelle. Nous utiliserons de manière pratique les arbres et forêts dans le chapitre 5.

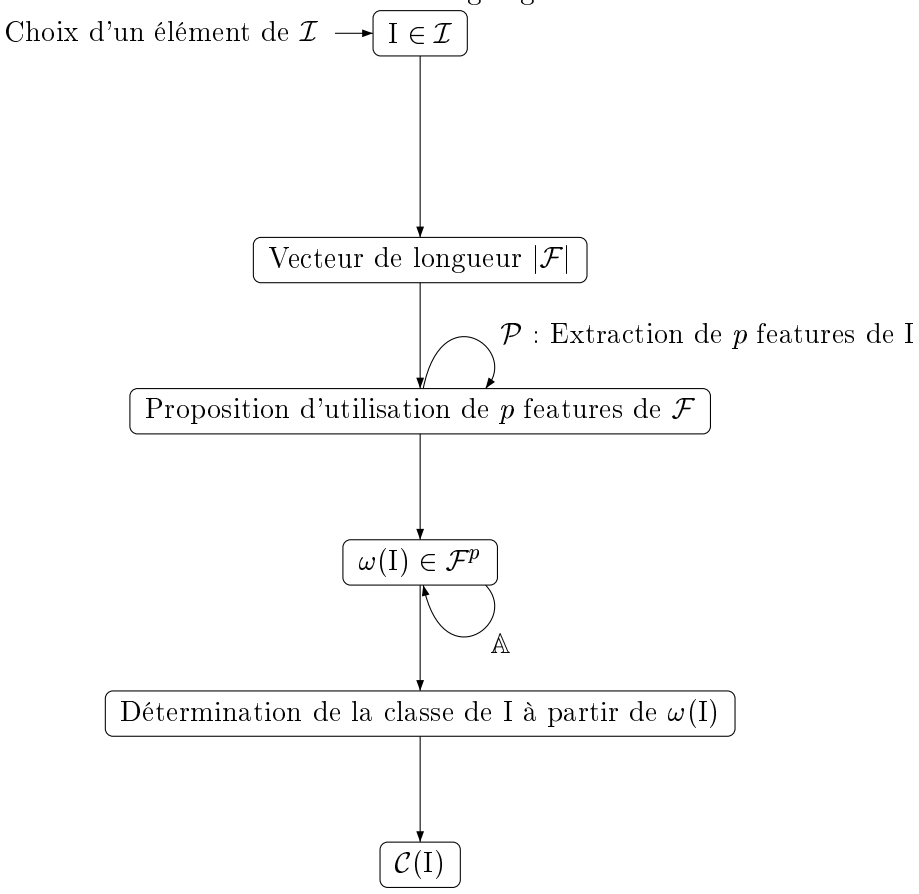
Chapitre 3 - Sélection de features par minimisation d'une énergie

3.1 Problématique

Nous allons tout d'abord rappeler à quel niveau se situe notre intervention sur l'extraction de features puis définir notre modèle par l'énergie $\mathcal{E}$. La méthode de recherche de la solution, ou du moins d'une solution locale pour la sélection de features est ensuite décrite avant d'appliquer les algorithmes établis sur différents exemples ([GY04]).

Notons $\mathcal{F}$ notre ensemble de features sélectionnés, on imagine que l'on dispose d'un algorithme $\mathbb{A}$ de classification qui soit utilisable pour tous les problèmes de classification d'une classe $\mathcal{C}_i$ quelconque contre la réunion des autres classes $\bigcup_{j \neq i} \mathcal{C}_j$. On suppose de plus que cet algorithme peut fonctionner si on lui passe en entrée un champ restreint d'informations plutôt que la totalité des informations disponibles sur les données. Typiquement, on suppose que $\mathbb{A}$ permet de déterminer une classe approchée d'un élément de la base de données à partir de n'importe quel sous-ensemble de features issu de $\mathcal{F}$.

On peut schématiser l'action de $\mathbb{A}$ via l'organigramme suivant :



On suppose de plus que l'on peut mesurer la performance de classification de $\mathbb{A}$ sur la base d'apprentissage de I, que l'on appellera classiquement « training set ». Cette mesure de performance de $\mathbb{A}$ sera par exemple :

- un taux d'erreur relatif à un algorithme de k-plus proche voisins ($k-\mathrm{NN}$) utilisant donc un sous-ensemble ω extrait de $\mathcal{F}$.
- un taux d'erreur relatif à un algorithme de Support Vector Machine (SVM) utilisant également un sous-ensemble ω extrait de $\mathcal{F}$.

Plus précisément, dans le cas où l'algorithme $\mathbb{A}$ est basé sur un SVM pour un problème de classification muli-classe, on utilisera l'algorithme de SVM pour chacun des problèmes $\left(\mathcal{C}_i \,||\, \bigcup_{j\neq i} \mathcal{C}_j\right)$ et si l'on désigne par $g_i(\omega)$ le taux d'erreur commis par $\mathbb{A}$ pour ce problème, on posera

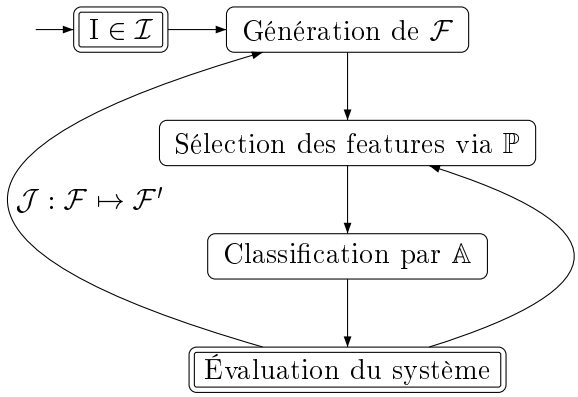
$$g(\omega) = \sum_{i=1}^{|\mathrm{C}|} g_i(\omega) \tag{3.1}$$

Dans le cas où l'algorithme est basé sur un $k-\mathrm{NN}$, il n'y a pas d'aménagement à apporter à cet algorithme puisqu'il est directement adapté aux problèmes multi-classes.

Notre approche consiste à établir les « meilleures » propositions de features ω de $\mathcal{F}^p$, c'est-à-dire à optimiser la phase $\mathcal{P}$ en vue de l'obtention de meilleures performances pour $\mathbb{A}$. Pour caractériser l'utilité d'un feature de $\mathcal{F}$, nous utiliserons donc un critère de performance basé sur les quantités $g(\omega)$. Par ailleurs, nous voulons imposer une collaboration des features f de $\mathcal{F}$, ceci afin d'exhiber des features réutilisables dans diverses combinaisons de sous-ensembles $\omega \subset \mathcal{F}^p$. Ainsi, nous munissons $\mathcal{F}$ d'une probabilité $\mathbb{P}$: si f est un feature de $\mathcal{F}$, $\mathbb{P}(f)$ correspond alors à la probabilité de proposer f dans l'étape $\mathcal{P}$ du diagramme précédent. On cherche alors à minimiser une énergie comprenant le terme suivant utilisant (3.1)

$$\mathcal{E}_1(\mathbb{P}) = \mathbb{E}_{\mathbb{P}}[g(\omega)] = \sum_{\omega\in\mathcal{F}^p} \mathbb{P}(\omega)g(\omega) = \sum_{(\omega_1\ldots\omega_p)\in\mathcal{F}^p} \mathbb{P}(\omega_1)\ldots\mathbb{P}(\omega_p)g(\omega)$$

Le schéma de recherche du $\mathbb{P}$ optimal est alors classique en apprentissage :



Mise à jour des paramètres de sélection

Ultérieurement, le schéma de recherche inclura également le parcours de l'ensemble des caractéristiques $\mathcal{F}$ « optimales » dans un sens qui reste encore à définir ; nous aurons donc une mise à jour de $\mathcal{F}$ par une procédure $\mathcal{J}$.

L'algorithme d'apprentissage de la bonne méthode de sélection de features sera donc itératif, voire doublement itératif puisque à un ensemble de features $\mathcal{F}$ fixés, on cherchera la meilleure probabilité sur $\mathcal{F}$ en approchant ${\mathbb{P}_{\mathcal{F}}}^{\star}$ par une suite de probabilités $\left(\mathbb{P}_{n,\mathcal{F}}\right)_{n\in\mathbb{N}}$; puis l'on procèdera dans les chapitres suivants à la mise à jour de $\mathcal{F}$ par une dynamique markovienne.

Dans la suite de ce chapitre, comme l'ensemble des features $\mathcal{F}$ n'est pas modifié, on désignera plus simplement par $(\mathbb{P}_n)_{n\in\mathbb{N}}$ la suite qui approche ${\mathbb{P}_{\mathcal{F}}}^{\star}$.

3.2 Algorithme de recherche

3.2.1 Énergie

Commençons par préciser la notation, que nous utiliserons tout au long de ce mémoire, de la loi uniforme $\mathcal{U}_{\mathcal{F}}$.

Définition 3.2.1 (Loi uniforme $\mathcal{U}_{\mathcal{F}}$)
Nous noterons $\mathcal{U}_{\mathcal{F}}$ la loi de probabilités uniforme sur l'ensemble des features $\mathcal{F}$. Pour des raisons de simplicité des formules, nous confondrons systématiquement la loi $\mathcal{U}_{\mathcal{F}}$ avec la valeur de cette probabilité en n'importe quel feature de $\mathcal{F}$.

L'énergie que l'on souhaite minimiser s'écrit :

$$\mathcal{E}(\mathbb{P}) = \alpha \underbrace{\sum_{(\omega_1\ldots\omega_p)\in\mathcal{F}^p} \mathbb{P}(\omega_1)\ldots\mathbb{P}(\omega_p)g(\omega)}_{=\mathcal{E}_1(\mathbb{P})} + \beta \underbrace{\sum_{\delta\in\mathcal{F}} \left[\frac{1}{|\mathcal{F}|} - \mathbb{P}(\delta)\right]^2}_{=\|\mathcal{U}_{\mathcal{F}}-\mathbb{P}\|_2{}^2=\mathcal{E}_2(\mathbb{P})} \qquad (\mathbf{E})$$

Le terme $\mathcal{E}_1$ est un terme destiné à optimiser la performance de détection tandis que le terme $\mathcal{E}_2$ est un terme de régularisation par une attache vers la loi uniforme sur $\mathcal{F}$. On remarque enfin que le tirage des p-uplets ω se fait avec remise (expression de $\mathcal{E}_1(\mathbb{P})$).

On peut d'emblée justifier la modélisation de notre énergie à minimiser.

- Si le tirage s'effectuait sans remise, il est certain que le minimum $\mathbb{P}^{\star}$ de $\mathcal{E}_1$ conduirait à une solution ayant un support dégénéré. Il suffit en effet de considérer qu'il est possible d'énumérer tous les p-uplets possibles de features issus de $\mathcal{F}$ et de ne garder que ceux qui réalisent la meilleure performance pour la quantité définie en (3.1) $g(\omega)$.

  Par ailleurs, l'expression de l'énergie $\mathcal{E}_1$ est grandement simplifiée pour un tirage avec remise alors qu'un tirage sans remise imposerait un terme de la forme

$$\sum_{\omega\in\mathcal{F}^p} g(\omega) \sum_{\sigma\in\Sigma_p} \frac{\mathbb{P}(\omega_{i_{\sigma(1)}})\ldots\mathbb{P}(\omega_{i_{\sigma(p)}})}{\left[1-\mathbb{P}(\omega_{i_{\sigma(1)}})\right]\ldots\left[1-\mathbb{P}(\omega_{i_{\sigma(1)}})-\ldots\mathbb{P}(\omega_{i_{\sigma(p-1)}})\right]} \qquad (3.2)$$

  On constate en effet la grande simplicité du terme composant $\mathcal{E}_1$ issu de (**E**) par rapport à celui qui intervient dans (3.2).

- Le terme d'attache à la loi uniforme sur $\mathcal{F}$ permet de sélectionner un plus grand nombre de features en fin d'algorithme dans la mesure où ce terme empêche d'obtenir un support dégénéré sur un petit nombre de features de $\mathcal{F}$. En effet, notre objectif est de donner de « bonnes » règles pour proposer des features (phase $\mathcal{P}$) dans le premier schéma d'apprentissage précédent. Il est nécessaire que $\mathcal{P}$ propose des p-uplets de features qui ne soient par toujours identiques ou quasi-identiques afin de pouvoir enclencher par la suite une étape de

collaboration de ces classifieurs via un boosting ou un vote de classifieurs (cependant, tous les p-uplets proposés par $\mathcal{P}$ devront avoir une efficacité tangible). Le terme de rappel vers $\mathcal{U}_\mathcal{F}$ permet de s'absoudre de cette contrainte puisque les features de faibles performances sont tout de même « rehaussés » par cette force de rappel. Par ailleurs, cette force de rappel apporte un autre avantage, théorique cette fois, puisqu'il assure (certes sous une condition un peu contraignante sur α et β) que la fonctionnelle est convexe, ce qui confèrera une plus grande stabilité de l'algorithme de minimisation et l'unicité du minimum.

Nous allons utiliser une méthode basée sur une descente de gradient pour minimiser l'énergie $\mathcal{E}$. Il faudra prendre garde au fait que $\mathcal{E}$ n'est définie que sur l'ensemble $\mathcal{S}_\mathcal{F}$:

Définition 3.2.2 (Simplexe $\mathcal{S}_\mathcal{F}$)
Les éléments P *de* $\mathcal{S}_\mathcal{F}$ *sont les éléments de* $\mathbb{R}^{|\mathcal{F}|}$ *vérifiant*

$$\mathrm{P} \in \mathcal{S}_\mathcal{F} \iff \begin{cases} \forall i \in \mathcal{F} \qquad \mathrm{P}(i) \geqslant 0 \\ \sum\limits_{i \in \mathcal{F}} \mathrm{P}(i) = 1 \end{cases}$$

On notera par ailleurs $\mathcal{S}^\star_\mathcal{F}$ *l'ensemble des* P *satisfaisant en plus des inégalités strictes :* $\mathrm{P}(i) > 0$.

Autrement dit, $\mathcal{S}_\mathcal{F}$ désigne simplement l'ensemble des probabilités sur les features de support coïncidant avec $\mathcal{F}$.

Définition 3.2.3 (Hyperplan $\mathcal{H}_\mathcal{F}$)
On définit par ailleurs l'hyperplan $\mathcal{H}_\mathcal{F}$ *comme étant l'hyperplan portant le simplexe* $\mathcal{S}_\mathcal{F}$. *Sa définition se résume donc à*

$$\sum_{i \in \mathcal{F}} \mathrm{P}(i) = 1$$

De plus, le vecteur $\overrightarrow{n_\mathcal{F}(\mathbb{P})}$ *désigne le vecteur unitaire normal donné par*

$$\overrightarrow{n_\mathcal{F}(\mathbb{P})} = \frac{1}{\sqrt{|\mathcal{F}|}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Ce vecteur ne dépendant pas en réalité du point $\mathbb{P}$, *nous le noterons plus simplement* $\overrightarrow{n_\mathcal{F}}$.

Par ailleurs, il est nécessaire de définir la projection $\Pi_{\mathcal{H}_\mathcal{F}}$ sur le plan tangent à l'hyperplan précédent. Cela correspond ici à la définition 3.2.3 suivante.

Définition 3.2.4 (Projection $\Pi_{\mathcal{H}_\mathcal{F}}$)
Pour $\mathbb{P}$ *un point de* $\mathcal{S}_\mathcal{F}$, $\Pi_{\mathcal{H}_\mathcal{F}}(\mathbb{P})$ *est la projection vectorielle donnée par*

$$\Pi_{\mathcal{H}_\mathcal{F}}(\mathbb{P}) \left(\overrightarrow{u} \right) = \overrightarrow{u} - (\overrightarrow{u} \mid \overrightarrow{n_\mathcal{F}(\mathbb{P})}) \overrightarrow{n_\mathcal{F}(\mathbb{P})}$$

On peut par ailleurs expliciter précisément $\Pi_{\mathcal{H}_\mathcal{F}}(\mathbb{P})$:

$$\begin{aligned} \forall \delta \in \mathcal{F} \qquad \Pi_{\mathcal{H}_\mathcal{F}}(\overrightarrow{u})(\delta)) &= \overrightarrow{u}(\delta) - (\overrightarrow{u} \mid \overrightarrow{n_\mathcal{F}(\mathbb{P})}) \\ &= \overrightarrow{u}(\delta) - \sum_{\delta' \in \mathcal{F}} \frac{\overrightarrow{u}(\delta')}{|\mathcal{F}|} \end{aligned}$$

Là aussi, la projection ne dépend en réalité pas du point $\mathbb{P}$ *de l'hyperplan* $\mathcal{H}_\mathcal{F}$ *et nous la noterons plus simplement* $\Pi_{\mathcal{H}_\mathcal{F}}$.

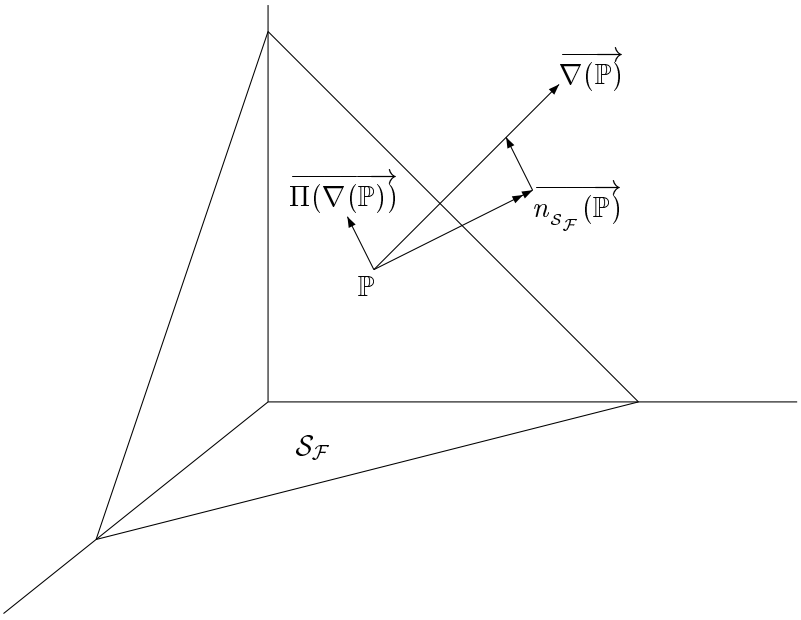
Afin de minimiser $\mathcal{E}$, on utilisera une descente de gradient de la forme

$$\frac{\mathrm{d}\mathbb{P}_t}{\mathrm{d}t} = -\Pi_{\mathcal{H}_\mathcal{F}}\left(\nabla\mathcal{E}(\mathbb{P}_t)\right) \tag{3.3}$$

La discrétisation en temps de l'équation d'évolution (3.3) revient alors à utiliser une formule de récurrence de la forme :

$$\forall \delta \in \mathcal{F} \qquad \mathbb{P}_{n+1}(\delta) = \mathbb{P}_n(\delta) - \gamma_n\left[\Pi_{\mathcal{H}_\mathcal{F}}(\nabla\mathcal{E}(\mathbb{P}_n)(\delta))\right] \tag{3.4}$$

Il faut noter dans la précédente formule (3.4) que c'est précisément le terme $\Pi_{\mathcal{H}_\mathcal{F}}$ qui permet de contraindre $\mathbb{P}$ à $\mathcal{H}_\mathcal{F}$. Ce qui correspond au schéma suivant :



Enfin, pour assurer que la suite $(\mathbb{P}_n)_{n\in\mathbb{N}}$ évolue toujours dans $\mathcal{S}_\mathcal{F}$, il faut s'assurer que tous les $\mathbb{P}_n(\delta)$ sont positifs au cours du temps (discrétisé par n). Plusieurs conditions assurant une telle stabilité dans $\mathcal{S}_\mathcal{F}$ seront données dans les paragraphes 3.3 et 3.4.

3.2.2 Gradient de $\mathcal{E}$ en métrique euclidienne

Dans le cadre classique de la métrique euclidienne, on peut calculer le gradient de l'énergie $\mathcal{E}$. Au vu de (**E**) celui-ci est donné par :

$$\nabla\mathcal{E}(\mathbb{P}) = \alpha\nabla\mathcal{E}_1(\mathbb{P}) + \beta\nabla\mathcal{E}_2(\mathbb{P})$$

avec $\forall \delta \in \mathcal{F}$

$$\begin{cases} \nabla\mathcal{E}_1(\mathbb{P})(\delta) = \displaystyle\sum_{\omega\in\mathcal{F}^p} \frac{\partial\,\mathbb{P}(\omega)}{\partial\,\mathbb{P}(\delta)} g(\omega) \\ \nabla\mathcal{E}_2(\mathbb{P})(\delta) = 2\left(\mathbb{P}(\delta) - \dfrac{1}{|\mathcal{F}|}\right) \end{cases}$$

Pour expliciter facilement le gradient de $\mathcal{E}_1$ précédent, introduisons les fonctions $\mathrm{C}(\omega, \delta)$.

Définition 3.2.5 (Fonctions $\mathrm{C}(\omega, \delta)$)

$$\forall \omega \in \mathcal{F}^p \quad \forall \delta \in \mathcal{F} \qquad \mathrm{C}(\omega, \delta) = \Big| \{ i \in [\![1 ; p]\!] \quad \omega_i = \delta \} \Big|$$

Ainsi, $\mathrm{C}(\omega, \delta)$ *représente le nombre d'occurences du feature* δ *dans le* p*-uplet* ω*.*

Cet artifice de notation va alors nous permettre d'exprimer simplement le gradient de $\mathcal{E}_1$. Par contre, cette notation est nécessaire puisque nous avons choisi d'effectuer un tirage des variables ω avec remises dans l'ensemble $\mathcal{F}$. L'introduction du terme $\mathrm{C}(\omega, \delta)$ est donc le « prix à payer » de l'approche avec remises : un tirage sans remise nous permettrait de s'absoudre de l'utilisation d'un tel terme mais alors le calcul du gradient de $\mathcal{E}_1$ serait nettement plus complexe.

On peut alors expliciter le calcul de $\nabla \mathcal{E}_1(\mathbb{P})$:

$$\forall \delta \in \mathcal{F} \qquad \mathcal{E}_1(\mathbb{P})(\delta) = \sum_{\omega \in \mathcal{F}^p} \frac{\mathrm{C}(\omega, \delta) \mathbb{P}(\omega)}{\mathbb{P}(\delta)} g(\omega) \tag{3.5}$$

Dans l'expression précédente, il n'y a pas de problème de continuité pour $\mathbb{P}(\delta) = 0$ puisque dès que $\mathrm{C}(\omega, \delta) \geqslant 1$, ω s'écrit

$$\omega = \delta, \omega_2, \ldots, \omega_p$$

et $\mathbb{P}(\omega)/\mathbb{P}(\delta)$ se simplifie en $\mathbb{P}(\omega_2) \times \cdots \times \mathbb{P}(\omega_p)$.

Cette expression amène donc à la formule de mise à jour :

$$\mathbb{P}_{n+1}(.) = \mathbb{P}_n(.) - \gamma_n \Pi_{\mathcal{H}_\mathcal{F}} \left(\alpha \sum_{\omega \in \mathcal{F}^p} \mathbb{P}(\omega) g(\omega) \frac{\mathrm{C}(\omega, .)}{\mathbb{P}(.)} - 2\beta \left(\frac{1}{|\mathcal{F}|} - \mathbb{P}(.) \right) \right) \tag{3.6}$$

La linéarité de la projection sur l'hyperplan $\mathcal{H}_\mathcal{F}$ implique alors que :

$$\boxed{\begin{array}{l} \forall n \in \mathbb{N} \qquad \mathbb{P}_{n+1}(.) = \mathbb{P}_n(.) - \gamma_n \alpha \displaystyle\sum_{\omega \in \mathcal{F}^p} \mathbb{P}(\omega) g(\omega) \left(\frac{\mathrm{C}(\omega, .)}{\mathbb{P}(.)} - \sum_{\delta \in \omega} \frac{\mathrm{C}(\omega, \delta)}{|\mathcal{F}| \mathbb{P}(\delta)} \right) \\ \qquad\qquad\qquad\qquad + 2\gamma_n \beta \left(\frac{1}{|\mathcal{F}|} - \mathbb{P}(.) \right) \end{array}} \quad (\mathbf{E-1})$$

Il est important de noter que pour le but que l'on poursuit (à savoir, à chaque étape de l'algorithme, mettre à jour de $(\mathbb{P}_n)_{n \in \mathbb{N}}$) il est nécessaire de calculer le gradient de $\mathcal{E}$. Ceci n'est pas tout le temps réaliste puisque le calcul de $\nabla \mathcal{E}$ impose le parcours de tous les ω de $\mathcal{F}^p$ ainsi que le calcul de la performance $g(\omega)$. Lorsque c'est impossible (du point de vue du temps de calcul), il faut alors procéder à une approximation de ce gradient. Cette possibilité dépend en réalité de la nature des données, ainsi que du nombre de features tirés dans ω. Nous verrons différents cas où un tel parcours est impossible.

3.2.3 Gradient de $\mathcal{E}$ en variables exponentielles

Il n'apparaît pas trivial au vu de $(\mathbf{E-1})$ de restreindre la suite $(\mathbb{P}_n)_{n \in \mathbb{N}}$ à l'espace $\mathcal{S}_\mathcal{F}$ (il paraît difficile de vérifier que la contrainte $\mathbb{P}(\delta) \geqslant 0$ est valable, quel que soit n). On pourrait penser alors à paramétrer le simplexe en variables exponentielles comme suit.

3.2.3.1 Changement de variable :

On pose plus précisément :

$$\forall \delta \in \mathcal{F} \qquad \mathbb{P}(\delta) = e^{y(\delta)}$$

Le but est alors de chercher le vecteur y qui minimise l'énergie $\mathcal{E}(e^y) = \widetilde{\mathcal{E}}(y)$ sur l'espace des contraintes qui est cette fois définit par :

$$\mathcal{C} = \left\{ y \quad / \quad \sum e^{y(\delta)} = 1 \right\}$$

Dans ce nouveau jeu de coordonnées, l'énergie $\widetilde{\mathcal{E}}$ se reécrit en

$$\widetilde{\mathcal{E}}(y) = \alpha \sum_{\omega \in \mathcal{F}^p} e^{y(\omega_1)+\cdots+y(\omega_p)} g(\omega) + \beta \sum_{\delta \in \mathcal{F}} \left(e^{y(\delta)} - \frac{1}{|\mathcal{F}|} \right)^2 \tag{3.7}$$

et le gradient de $\widetilde{\mathcal{E}}$ vaut cette fois :

$$\forall \delta \in \mathcal{F} \qquad \nabla\widetilde{\mathcal{E}}(y)(\delta) = \alpha \sum_{\omega \in \mathcal{F}^p} e^{y(\omega_1)+\cdots+y(\omega_p)} g(\omega) \mathrm{C}(\omega, \delta) + 2\beta e^{y(\delta)} \left(e^{y(\delta)} - \frac{1}{|\mathcal{F}|} \right) \tag{3.8}$$

La projection sur l'espace des contraintes en variables « exponentielles » est un peu différente du cas des variables « en $\mathbb{P}$ » et se fait parallélement au vecteur normal (ici non normalisé) au plan tangent à $\mathcal{C}$ en y défini par :

$$\overrightarrow{n_{\mathcal{C}}(y)} = \begin{pmatrix} e^{y_1} \\ \vdots \\ e^{y_n} \end{pmatrix}$$

La descente de gradient sur les variables y s'écrit alors :

$$\boxed{y_{n+1} = y_n - \gamma_n \nabla\widetilde{\mathcal{E}}(y_n) + \gamma_n \underbrace{\frac{(\nabla\widetilde{\mathcal{E}}(y_n) \mid \overrightarrow{n_{\mathcal{C}}(y_n)})}{\|\overrightarrow{n_{\mathcal{C}}(y_n)}\|^2}}_{=\mathrm{K}}}$$

En revenant aux variables $\mathbb{P}$, on en déduit une autre formule de mise à jour de $(\mathbb{P}_n)_{n\in\mathbb{N}}$:

$$\boxed{\forall n \in \mathbb{N} \qquad \mathbb{P}_{n+1}(\delta) = \mathbb{P}_n(\delta) \frac{e^{-\gamma_n \nabla\widetilde{\mathcal{E}}(y_n)(\delta)}}{\mathrm{K}_n}} \qquad (\mathbf{E-2})$$

où K_n est donné par l'expression

$$\mathrm{K}_n = \sum_{\delta \in \mathcal{F}} \mathbb{P}_n(\delta) e^{-\gamma_n \nabla\widetilde{\mathcal{E}}(y_n)(\delta)} \tag{3.9}$$

et
$$\nabla\widetilde{\mathcal{E}}(y)(\delta) = \alpha \sum_{\omega \in \mathcal{F}^p} \mathbb{P}(\omega) g(\omega) \mathrm{C}(\omega, \delta) + 2\beta \mathbb{P}(\delta) \left(\mathbb{P}(\delta) - \frac{1}{|\mathcal{F}|} \right)$$

avec
$$\mathbb{P}(\delta) = e^{y(\delta)}$$

On constate en examinant $(\mathbf{E-2})$ qu'il n'est pas nécessaire de calculer explicitement K_n : en effet, renormaliser $\mathbb{P}_{n+1}$ afin que la somme des probabilités soit égale à 1 suffit puisque de fait, les entités manipulées sont toutes positives. L'avantage de cette descente de gradient « en variables exponentielles » est que finalement, il y a peu de soins à apporter au pas de l'algorithme (γ_n) pour contraindre la suite $(\mathbb{P}_n)_{n\in\mathbb{N}}$ dans $\mathcal{S}_{\mathcal{F}}$.

3.2.3.2 Vitesse de décroissance de $\mathbb{P}_n(\delta)$ vers 0 :

Cette approche paraît séduisante puisqu'elle permet de s'absoudre des conditions $\mathbb{P}(i) \geqslant 0$ qui sont naturellement vérifiées. En revanche, l'évolution est plus lente, notamment lorsque $\mathbb{P}_n$ tend vers 0 et ne peut atteindre 0 qu'avec une vitesse nulle. En effet, en écrivant un développement limité, lorsque γ_n est petit, de $(\mathbf{E}-\mathbf{2})$ (ce qui sera précisément le cas lorsque nous choisirons numériquement le pas de l'algorithme), on obtient

$$\mathbb{P}_{n+1} = \mathbb{P}_n - \gamma_n \frac{\mathbb{P}_n}{\mathrm{K}_n} \nabla\widetilde{\mathcal{E}}(y_n) + \mathrm{o}\left(\gamma_n\right)$$

On constate bien que dans la formule précédente, le point 0 ne peut être atteint par $\mathbb{P}_n(\delta)$ qu'à vitesse nulle puisque le terme de vitesse d'atteinte de 0 varie au plus en

$$\mathrm{M}\gamma_n \mathbb{P}_n(\delta)$$

puisque grossièrement $\quad \forall \mathrm{P} \in \mathcal{S}_{\mathcal{F}} \qquad \|\nabla\widetilde{\mathcal{E}}(y(\mathrm{P}))\| \in [\, m_1 \,;\, m_2 \,]$

où $$\begin{cases} m_1 = \alpha \ \mathrm{Inf}\ g - 2\beta/|\mathcal{F}| \\ m_2 = \alpha \ \mathrm{Sup}\,(g) + 2\beta \end{cases}$$

et $$\mathrm{K}_n \in [\, e^{-\gamma_n m_2} \,;\, e^{-\gamma_n m_1} \,]$$

Ceci assure que $$\frac{\|\nabla\widetilde{\mathcal{E}}(y_n)\|}{\mathrm{K}_n} \in [\, m_1 e^{-\gamma_n m_2} \,;\, m_2 e^{-\gamma_n m_1} \,]$$

Enfin, $$\frac{\mathbb{P}_{n+1}(\delta) - \mathbb{P}_n(\delta)}{\mathbb{P}_n(\delta)} = -\gamma_n \frac{\nabla\widetilde{\mathcal{E}}(\mathbb{P}_n)(\delta)}{\mathrm{K}_n} + \mathrm{o}\left(\gamma_n\right)$$

La vitesse d'atteinte de 0 varie donc comme annoncé.

3.2.3.3 Interprétation en métrique Riemannienne

Il est possible d'interpréter le changement de variable précédent en utilisant une métrique non plus Euclidienne, mais une métrique proche d'un cadre Riemannien (au sens strict, les définitions ne sont pas vérifiées pour satisfaire au cadre Riemannien). Cela nous permettra d'interpréter intuitivement le comportement de l'algorithme paramétré en variables exponentielles en terme de distance à la frontière du simplexe $\mathcal{S}^\star_{\mathcal{F}}$. Nous munissons donc $\mathcal{S}^\star_{\mathcal{F}}$ de l'ensemble des produits scalaires $\langle \mathrm{Q}, \mathrm{R} \rangle_{\mathbb{P}}$.

Définition 3.2.6 (Produit scalaire $\langle .,. \rangle_{\mathbb{P}}$)
Soit $\mathbb{P} \in \mathcal{S}^\star_{\mathcal{F}}$, pour Q *et* R *deux éléments de l'espace tangent à $\mathcal{S}^\star_{\mathcal{F}}$ en $\mathbb{P}$ noté $\mathcal{T}\mathcal{S}^\star_{\mathcal{F}}(\mathbb{P})$,*

$$\langle \mathrm{Q}, \mathrm{R} \rangle_{\mathbb{P}} = \sum_{\delta \in \mathcal{F}} \frac{\mathrm{Q}(\delta)\mathrm{R}(\delta)}{\mathbb{P}(\delta)} \tag{3.10}$$

Cet ensemble de produits scalaires ne définit pas exactement une métrique Riemannienne sur $\mathcal{S}_{\mathcal{F}}$ mais en définit une sur la variété différentiable $\mathcal{S}^\star_{\mathcal{F}}$ On vérifie en plus les deux propriétés qui assurent la structure Riemanienne de l'espace $\mathcal{S}^\star_{\mathcal{F}}$.

Propriété 3.2.1 (Vérifications des propriétés des tenseurs métriques)
L'ensemble des produits scalaires $\langle .,. \rangle_{\mathbb{P}}$ vérifient bien les deux propriétés :

1. *Pour tout* $\mathbb{P}$ *de* $\mathcal{S}^\star_\mathcal{F}$, *on a*

$$(\mathrm{Q},\mathrm{R}) \longmapsto \langle \mathrm{Q},\mathrm{R}\rangle_\mathbb{P} \qquad \mathrm{Q},\mathrm{R} \in \mathcal{T}\mathcal{S}^\star_\mathcal{F}$$

est une forme bilinéaire, symétrique, définie positive.

2. *Pour tout ouvert* U *de* $\mathcal{S}^\star_\mathcal{F}$, *et pour tout* Q,R *de* $\mathcal{S}^\star_\mathcal{F}$, *la fonction*

$$\mathbb{P} \in \mathrm{U} \longmapsto \langle \mathrm{Q},\mathrm{R}\rangle_\mathbb{P}$$

est différentiable sur U.

Ces propriétés nous permettent donc de considérer la variété Riemannienne $(\mathcal{S}^\star_\mathcal{F};\langle .,.\rangle_.)$ ainsi que l'espace tangent $\mathcal{T}_\mathbb{P}$ à cette variété en $\mathbb{P}$; et de calculer ainsi dans cet espace le gradient de la fonctionnelle $\mathcal{E}$ qui y est définie. Enfin, notons que l'espace tangent à $\mathcal{S}^\star_\mathcal{F}$ est confondu en tout point avec $\mathcal{S}_\mathcal{F}$ puisque la variété est en réalité une variété linéaire. Cela justifie alors la définition de la quantité $\langle \mathrm{Q},\mathrm{R}\rangle_\mathbb{P}$ dans le second point de la propriété 3.2.1 précédente.

Propriété 3.2.2 (Calcul de $\nabla^\star\mathcal{E}$ dans $(\mathcal{S}^\star_\mathcal{F};\langle .,.\rangle_.)$)
Dans l'espace Riemannien précédent, le gradient de $\mathcal{E}$ *vaut*

$$\forall \mathbb{P} \in \mathcal{S}^\star_\mathcal{F} \quad \forall \delta \in \mathcal{F} \qquad \nabla^\star\mathcal{E}(\mathbb{P})(\delta) = 2\beta\mathbb{P}(\delta)\left[\mathbb{P}(\delta) - \mathcal{U}(\delta)\right] + \alpha \sum_{\omega\in\mathcal{F}^p} g(\omega)\mathrm{C}(\omega,\delta) \tag{3.11}$$

Preuve : On effectue le calcul de ce gradient en remarquant que l'on doit avoir pour $\mathbb{P}$ une probabilité de $\mathcal{S}^\star_\mathcal{F}$:

$$\forall \mathrm{Q} \in \mathcal{T}_\mathbb{P} \qquad \lim_{\varepsilon\to 0} \frac{\mathcal{E}(\mathbb{P}+\varepsilon\mathrm{Q}) - \mathcal{E}(\mathbb{P})}{\varepsilon} = \langle \nabla^\star\mathcal{E}(\mathbb{P});\mathrm{Q}\rangle_\mathbb{P}$$

On obtient ainsi la formule (3.11). □

Il est désormais important de remarquer que si l'on pose $\mathbb{P} = e^y$ via le changement de variables précédent, on a alors

$$\nabla\widetilde{\mathcal{E}}(y) = \nabla^\star\mathcal{E}(\mathbb{P}) \tag{3.12}$$

Ainsi, l'égalité (3.11) exprime que le passage en variables exponentielles pour paramétrer $\mathcal{S}^\star_\mathcal{F}$ est une autre façon de voir la descente de gradient en métrique Riemannienne de $\mathcal{S}^\star_\mathcal{F}$. La métrique en question a pour propriété d'« envoyer » tous les points de sa frontière ($\mathbb{P}$ tels que $\mathbb{P}(\delta) = 0$ pour un feature δ) à une distance infinie (division par un terme nul dans la définition 3.2.5 de $\langle ;\rangle_.$). Ce résultat est donc à mettre en parallèle avec le fait que l'on ne peut atteindre la frontière de $\mathcal{S}^\star_\mathcal{F}$ par la descente de gradient (**E** − **2**) qu'à une vitesse nulle.

3.2.3.4 Identification de l'espace $\mathcal{S}_{\mathcal{F}^\star}, \langle ;\rangle_\mathbb{P}$

Nous allons montrer que l'espace métrique précédent peut s'interpréter facilement comme l'espace métrique $\mathcal{S}_{\mathcal{F}^\star}$ muni de la distance de Kullback. Commençons par considérer l'espace $\mathcal{S}_{\mathcal{F}^\star}$ muni de la distance de Kullback D donnée par la définition 2.8.3. Nous allons étudier le produit scalaire qui est issu de cette distance. Appelons $(\,|\,)_.$ le produit scalaire dérivé de cette distance, nous allons identifier ce produit scalaire au produit scalaire Riemanien du paragraphe précédent. Pour cela, développons l'expression de D localement en $\mathbb{P}$ dans la direction $\mathrm{d}\mathbb{Q}$. On a alors la propriété suivante.

Propriété 3.2.3 (Développement de $\mathrm{D}(.,.)$)
Si $\mathrm{d}\mathbb{Q}$ désigne un point de l'espace tangent $(\mathcal{TS}_{\mathcal{F}^\star})_{\mathbb{P}}$, alors

$$\mathrm{D}(\mathbb{P}, \mathbb{P} + \mathrm{d}\mathbb{Q}) = \sum_x \frac{\mathrm{d}\mathbb{Q}(x)^2}{\mathbb{P}(x)} + o\left(\|\mathrm{d}\mathbb{Q}\|^2\right)$$

Preuve : Il suffit de développer l'expression précédente :

$$\mathrm{D}(\mathbb{P}, \mathbb{P} + \mathrm{d}\mathbb{Q}) = \sum_x \mathbb{P}(x) \log\left(\frac{\mathbb{P}(x)}{\mathbb{P}(x) + \mathrm{d}\mathbb{Q}(x)}\right)$$

Soit

$$\begin{aligned}
\mathrm{D}(\mathbb{P}, \mathbb{P} + \mathrm{d}\mathbb{Q}) &= -\sum_x \mathbb{P}(x) \log\left(1 + \frac{\mathrm{d}\mathbb{Q}(x)}{\mathbb{P}(x)}\right) \\
&= -\sum_x \mathbb{P}(x) \left(\frac{\mathrm{d}\mathbb{Q}(x)}{\mathbb{P}(x)} - \frac{1}{2}\frac{\mathrm{d}\mathbb{Q}(x)^2}{\mathbb{P}(x)^2}\right) + \mathrm{o}\left(\|\mathrm{d}\mathbb{Q}\|^2\right) \\
\mathrm{D}(\mathbb{P}, \mathbb{P} + \mathrm{d}\mathbb{Q}) &= \underbrace{-\sum_x \mathrm{d}\mathbb{Q}(x)}_{=0} + \frac{1}{2}\sum_x \frac{\mathrm{d}\mathbb{Q}(x)^2}{\mathbb{P}(x)} \mathrm{o}\left(\|\mathrm{d}\mathbb{Q}\|^2\right)
\end{aligned}$$

Finalement, on obtient bien le développement limité annoncé. □

Remarquons enfin que cette métrique est reliée à la métrique du χ^2 puisqu'on peut établir également la propriété 3.2.4.

Propriété 3.2.4 (Développement de $\chi^2(.,.)$)
Si $\mathbb{P}$ désigne un point de $\mathcal{S}_{\mathcal{F}^\star}$ et $\mathrm{d}\mathbb{Q}$ est point de l'espace tangent à $\mathcal{S}_{\mathcal{F}^\star}$ en $\mathbb{P}$, alors à l'ordre 2 on a

$$\frac{1}{2}\chi^2(\mathbb{P}, \mathbb{P} + \mathrm{d}\mathbb{Q}) = \frac{1}{2}\sum_x \frac{\mathrm{d}\mathbb{Q}(x)^2}{\mathbb{P}(x)} = \mathrm{D}(\mathbb{P}, \mathbb{P} + \mathrm{d}\mathbb{Q}) + o\left(\mathrm{d}\mathbb{Q}(x)^2\right)$$

Par conséquent, le développement limité de la distance de Kullback en $\mathbb{P}$ ainsi que celui du χ^2 correspondent et définissent la même métrique Riemanienne dont la norme est définie par

$$(\mathrm{d}\mathbb{Q} \mid \mathrm{d}\mathbb{Q})_{\mathbb{P}} = \sum_x \frac{\mathrm{d}\mathbb{Q}^2(x)}{\mathbb{P}(x)}$$

L'identité de polarisation permet alors de conclure que pour cette distance (celle du χ^2 ou deux fois celle de Kullback), on a

$$\forall \mathbb{Q}, \mathbb{R} \in \mathcal{S}_{\mathcal{F}^\star} \qquad (\mathbb{Q} \mid \mathbb{R})_{\mathbb{P}} = \sum_x \frac{\mathbb{Q}(x)\mathbb{R}(x)}{\mathbb{P}(x)} = \langle \mathrm{d}\mathbb{Q}, \mathrm{d}\mathbb{R} \rangle_{\mathbb{P}}$$

On peut donc remarquer que la métrique $(\,|\,)_{.}$ dérivée de la distance du χ^2 (ou deux fois celle de Kullback) est exactement la métrique $\langle .,. \rangle_{.}$ du paragraphe précédent. La descente de gradient en variables exponentielles étant identifiées à la descente de gradient en variable en $\mathbb{P}$ pour la métrique $\langle .,. \rangle_{.}$, on en déduit donc que cette descente de gradient en variables exponentielles est équivalente à une descente de gradient « en $\mathbb{P}$ » pour la métrique dérivée de la métrique du χ^2.

3.3 Équations différentielles associées aux descentes de gradient

Pour obtenir la résolution de la recherche du minimum de $\mathcal{E}$, on pourrait tout d'abord songer à simuler l'équation différentielle associée aux descentes de gradients évoquées ci-dessus *via* (3.3) :

$$\frac{\mathrm{d}\mathbb{P}_t}{\mathrm{d}t} = -\alpha \mathbb{E}_{\omega\sim\mathbb{P}_t{}^{\times p}}\left[\frac{\mathrm{C}(\omega,.)g(\omega)}{\mathbb{P}(.)} - \sum_{\mu\in\omega}\frac{g(\omega)\mathrm{C}(\omega,\mu)}{|\mathcal{F}|\mathbb{P}(\mu)}\right] - 2\beta\left(\mathbb{P}(.) - \mathcal{U}_{\mathcal{F}}\right) \qquad (\mathbf{E-3})$$

ou bien

$$\frac{\mathrm{d}\mathbb{P}_t}{\mathrm{d}t} = -\frac{\mathbb{P}_t}{\mathrm{K}_t}\left(\alpha\mathbb{E}_{\omega\sim\mathbb{P}_t{}^{\times p}}\left[g(\omega)\mathrm{C}(\omega,.)\right] + 2\beta\mathbb{P}_t(\mathbb{P}_t - \mathcal{U}_{\mathcal{F}})\right) \qquad (\mathbf{E-4})$$

où K_t est une constante qui assure que la somme des probabilités $\mathbb{P}_t(\delta)$ est constante égale à 1 au cours du temps et $\mathbb{E}_{\omega\sim\mathbb{P}_t{}^{\times p}}$ désigne l'espérance sur ω tirée selon la loi $\mathbb{P}_t$ avec p remises.

3.3.1 Étude de $(\mathbf{E-3})$

Il faut commencer par étudier l'existence de solutions à ces équations, et si tel est le cas, déterminer si elles définissent bien des éléments de $\mathcal{S}_{\mathcal{F}}$. Ce n'est pas évident dans le cas de la première équation différentielle puisque dans ce cas, on peut réécrire l'équation différentielle ordinaire en

$$\frac{\mathrm{d}\mathbb{P}_t}{\mathrm{d}t} = \mathrm{F}(\mathbb{P}_t)$$

La fonction F est alors définie par

$$\begin{aligned}\mathrm{F}(\mathbb{P})(\delta) = &-\alpha\sum_{\omega\,|\,\delta\in\omega} g(\omega)\mathrm{C}(\omega,\delta)\mathbb{P}(\omega\smallsetminus\delta) + \frac{\alpha}{|\mathcal{F}|}\sum_{\omega}\sum_{\mu\in\omega} g(\omega)\mathrm{C}(\omega,\mu)\mathbb{P}(\omega\smallsetminus\mu)\\ &-2\beta\left(\mathbb{P}(\delta) - \frac{1}{|\mathcal{F}|}\right)\end{aligned}$$

La fonction est de classe $\mathscr{C}^\infty(\mathcal{S}_{\mathcal{F}}, \mathbb{R}^{|\mathcal{F}|})$. Le théorème de Cauchy-Lipschitz assure l'existence d'une solution maximale définie sur un intervalle $[\,0\,;t_{max}\,[$ de $\mathbb{R}$.

Raisonnons par l'absurde et supposons que $t_{max} < +\infty$, le théorème de prolongement des solutions maximales assure alors que :

$$\lim_{t\mapsto t_{max}} \|\mathbb{P}_t\|_\infty = +\infty$$

On sait que

$$\sum_{\mu\in\mathcal{F}}\mathbb{P}_t(\mu) = 1$$

Démontrons alors que sous certaines conditions, on ne peut avoir $\mathbb{P}_{\widetilde{t}}(\delta) < 0$, ce qui assurera dès lors que $t_{max} = +\infty$. Supposons donc l'existence d'une telle éventualité, la fonction $\mathbb{P}_t$ étant $\mathscr{C}^1$, cela signifie donc que

$$(\mathbf{D}) \qquad \begin{cases} \exists t_0 \leqslant \widetilde{t} & \mathbb{P}_{t_0}(\delta) = 0 \\ \forall t \in \left[\,t_0\,;\widetilde{t}\,\right] & \mathbb{P}_t(\delta) < 0 \end{cases}$$

Mais alors

$$\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}t}\right)_{t=t_0} = -\alpha \sum_{\omega \mid \delta \in \omega} g(\omega)\mathrm{C}(\omega,\delta)\mathbb{P}(\omega \setminus \delta) + \frac{\alpha}{|\mathcal{F}|}\sum_{\omega,\mu\in\omega} g(\omega)\mathrm{C}(\omega,\mu)\mathbb{P}(\omega \setminus \mu) + \frac{2\beta}{|\mathcal{F}|}$$
$$\geqslant -\alpha|\omega|\ \mathrm{Sup}\,(g) + \frac{2\beta}{|\mathcal{F}|}$$

Si l'on suppose que

$$\alpha < \frac{2\beta}{|\mathcal{F}||\omega|\ \mathrm{Sup}\,(g)} \tag{3.13}$$

est vérifiée, on peut alors en déduire que

$$\left(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}t}\right)_{t=t_0} > 0$$

On peut alors trouver $\eta > 0$ tel que $\mathbb{P}_t(\delta) > 0$ pour $t \in [\,t_0\,;t_0+\eta\,]$, ce qui contredit la définition (**D**) de t_0. Par conséquent, si α et β vérifient l'inégalité ci-dessus, $\mathbb{P}_t(\delta)$ ne peut devenir négatif, pour tout t et pour tout δ. Comme en plus ces réels ont pour somme 1, on peut en déduire que ces nombres sont également bornés par 1, soit

$$\forall t \in \mathbb{R} \qquad \|\mathbb{P}_t\|_\infty \leqslant 1$$

Ceci est en contradiction avec l'hypothèse

$$\lim_{t\mapsto t_{max}} \|\mathbb{P}_t\|_\infty = +\infty$$

Ainsi, la solution de $(\mathbf{E}-\mathbf{3})$ est définie sur $\mathbb{R}$ et vérifie

$$\forall t \in \mathbb{R} \quad \forall \delta \in \mathcal{F} \qquad \mathbb{P}_t(\delta) > 0$$

> A noter que la formulation de l'équation différentielle ordinaire $(\mathbf{E}-\mathbf{3})$ utilisant la notation d'espérance sous $\mathbb{P}$ est donc bien définie puisque $\mathbb{P}$ est en tout temps une probabilité sur $\mathcal{F}$.

On a donc établi le premier résultat

Théorème 3.3.1 (Existence des solutions de $(\mathbf{E}-\mathbf{3})$)
Si l'on suppose que α et β vérifient

$$\alpha < \frac{2\beta}{|\mathcal{F}||\omega|\ \mathrm{Sup}\,(g)} \tag{3.14}$$

il existe alors une unique solution maximale de $(\mathbf{E}-\mathbf{3})$ définie sur $\mathbb{R}^+$. De plus, la solution de $(\mathbf{E}-\mathbf{3})$ appartient à $\mathcal{S}_\mathcal{F}$ en tout temps.

L'inégalité (3.14) assure ainsi la stabilité de $\mathcal{S}_\mathcal{F}$, mais cette condition n'est pas du tout réaliste du point de vue de l'implémentation, et satisfaisante théoriquement puisque la condition porte sur une majoration de α, c'est-à-dire une majoration du terme contrôlant l'influence du terme d'erreur $\mathcal{E}_1$. En réalité, la contrainte sur le coefficient α devient encore plus forte lorsque l'on s'intéresse à un algorithme d'approximation de cette équation différentielle (voir section 4.4 et Appendice B), ce qui nous incitera à contourner la difficulté des contraintes « $\mathbb{P}(i) \geqslant 0$ ».

3.3.2 Étude de $(\mathbf{E}-\mathbf{4})$

Au vu de l'expression de K_t :

$$\mathrm{K}_t = \frac{(\nabla\widetilde{\mathcal{E}}(y_t) \mid \overrightarrow{n_{\mathcal{C}}(y_t)})}{\|\overrightarrow{n_{\mathcal{C}}(y_t)}\|^2} = \frac{\sum_{i=1}^{|\mathcal{F}|} e^{y_i} \nabla\widetilde{\mathcal{E}}(y_t)_i}{\sum_{i=1}^{|\mathcal{F}|} e^{2y_t}}$$

et de celle de $\nabla\widetilde{\mathcal{E}}$:

$$\nabla\widetilde{\mathcal{E}}(y)(\delta) = \alpha \sum_{\omega\in\mathcal{F}^p} e^{y(\omega_1)+\cdots+y(\omega_p)} g(\omega)\mathrm{C}(\omega,\delta) + 2\beta e^{y(\delta)}\left(e^{y(\delta)} - \frac{1}{|\mathcal{F}|}\right)$$

on constate que l'équation différentielle sur y est de la forme

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = \mathrm{F}(y_t) = -\Pi\left(\nabla\widetilde{\mathcal{E}}(y_t)\right)$$

où F est de classe $\mathscr{C}^\infty(\mathbb{R}^{|\mathcal{F}|}, \mathbb{R}^{|\mathcal{F}|})$. Cette application F est bornée en tout temps puisque

$$\sum_{\delta\in\mathcal{F}} e^{y_t(\delta)} = 1$$

La solution de $(\mathbf{E}-\mathbf{4})$ est donc globale. Il n'y a par ailleurs aucun problème de contraintes aux données par construction de $\mathbb{P}$ et y. Le choix des constantes α et β peut donc être effectué sans se préoccuper du poids de α par rapport à β.

Lorsque les conditions $(\mathbf{C})$ seront difficilement applicables, par exemple dans le cas où l'ensemble des features est de cardinal important, nous privilégierons donc l'approche en variables exponentielles et l'équation d'évolution $(\mathbf{E}-\mathbf{4})$.

En revanche, si $|\mathcal{F}|$ est petit (moins de 100 éléments), nous utiliserons les équations $(\mathbf{E}-\mathbf{3})$ qui sont des équations d'évolution plus rapides.

3.4 Approximation stochastique de la descente de gradient $(\mathbf{E}-\mathbf{1})$

3.4.1 Nécessité d'une approximation stochastique

Que ce soit dans l'approche en variables exponentielles, ou en variables « en P », on constate qu'il peut être numériquement très long de calculer le gradient de la fonctionnelle à minimiser. On doit donc songer à effectuer une approximation de ce gradient, de sorte que le comportement limite de l'algorithme d'évolution soit identique à la descente de gradient théorique. On aura donc à étudier un algorithme s'écrivant sous la forme :

$$\mathbb{P}_{n+1} = \mathbb{P}_n - \gamma_{n+1}\left(d_n - \mathrm{C}_n\right) \tag{3.15}$$

en variables « en $\mathbb{P}$ » avec C_n constante de recentrage aux données qui assure que la suite de probabilités appartient à $\mathcal{S}_{\mathcal{F}}$ et $d_n$variable aléatoire approchant la valeur du gradient.

En s'inspirant des travaux sur les algorithmes de Robbins et Monro ([BMP], [Ben96]), on calcule les termes d_n en imposant que le comportement limite de l'algorithme doit être identique à celui de l'équation différentielle :

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}t} = -\Pi_{\mathcal{H}_{\mathcal{F}}}\left(\nabla\mathcal{E}(\mathbb{P})\right)$$

Mais le comportement asymptotique de la mise à jour (3.15) s'écrit

$$\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}t} = -\mathbb{E}[d(\delta) - \mathrm{C}(\delta)]$$

En examinant l'expression (3.5) du gradient de $\mathcal{E}$, on constate qu'il suffit de poser dans un premier temps :

$$d_n = \alpha \frac{\mathrm{C}(\omega_n, .)}{\mathbb{P}_n(.)} + 2\beta \left(\mathbb{P}_n(.) - \frac{1}{|\mathcal{F}|} \right)$$

pour ω_n variable aléatoire tirée dans $\mathcal{F}^p$ selon la loi $\mathbb{P}_n$ avec remise. Là encore, il n'y a pas de problème de division par 0 puisque si $\mathbb{P}_n(\delta)$ est nul, on ne peut donc tirer un feature ω_n contenant δ et la quantité d'occurences $\mathrm{C}(\omega_n, \delta)$ de δ dans ω est également nul. Ainsi, on prendra la convention que

$$\mathbb{P}_n(\delta) = 0 \Longrightarrow d_n(\delta) = 0$$

L'espérance « à la limite » du terme précédent peut se calculer en :

$$\mathbb{E}[d_n(\delta)] = \mathbb{E}_{\mathbb{P}_n}[d_n(\delta)] = \alpha \sum_{\omega \in \mathcal{F}^p} \mathbb{P}_n(\omega) g(\omega) \frac{\mathrm{C}(\omega, \delta)}{\mathbb{P}_n(\delta)} + 2\beta \left(\mathbb{P}_n(\delta) - \frac{1}{|\mathcal{F}|} \right)$$

Finalement, en ajoutant le terme de recentrage à $\mathcal{S}_\mathcal{F}$, on obtient la première descente de gradient stochastique :

$$\boxed{\begin{array}{l} \forall n \in \mathbb{N} \qquad \mathbb{P}_{n+1}(.) = \mathbb{P}_n(.) - \gamma_{n+1} \alpha g(\omega_n) \left(\dfrac{\mathrm{C}(\omega_n, .)}{\mathbb{P}_n(.)} - \displaystyle\sum_{\delta \in \omega_n} \frac{\mathrm{C}(\omega_n, \delta)}{|\mathcal{F}| \mathbb{P}_n(\delta)} \right) \\ \qquad\qquad\qquad\qquad + 2\gamma_{n+1} \beta \left(\dfrac{1}{|\mathcal{F}|} - \mathbb{P}_n(.) \right) \end{array}} \quad (\mathbf{E-5})$$

La suite de cette section va consister à montrer en quel sens une telle équation d'évolution $(\mathbf{E-5})$ approche le comportement de la descente de gradient exacte $(\mathbf{E-1})$ et donc de faire baisser l'énergie générale $\mathcal{E}$.

3.4.2 Stabilité de $\mathcal{S}_\mathcal{F}$

Dans le cas de la première équation d'évolution $(\mathbf{E-5})$; on cherche un intervalle stable de la forme $[\, m\,;\mathrm{M}\,]$. Le théorème suivant assure que sous certaines conditions relativement restricitives, l'évolution de $(\mathbf{E-5})$ se fait bien dans $\mathcal{S}_\mathcal{F}$.

Théorème 3.4.1 (Stabilité de $\mathcal{S}_\mathcal{F}$ sous l'équation d'évolution $(\mathbf{E-5})$)
Si l'on suppose que α et β vérifient

$$\frac{\alpha}{\beta} \leqslant \frac{\mathcal{U}_\mathcal{F}^2}{2|\omega| \ \mathrm{Sup}\,(g)} \tag{3.16}$$

et que le nombre ω de features tirés dans $\mathcal{F}$ vérifie

$$\omega \leqslant \frac{|\mathcal{F}|}{2}$$

alors on peut trouver un pas γ_n variant en $1/n$ suffisamment petit tel que l'intervalle $\left[\dfrac{\mathcal{U}_\mathcal{F}}{2} ; \dfrac{2\omega}{|\mathcal{F}|} \right]$ *est laissé stable par l'équation d'évolution $(\mathbf{E-5})$. Plus précisément, si $\gamma = \alpha/\beta$:*

$$\exists \gamma_0 \in \mathbb{R}^+ \quad 0 \leqslant \gamma \leqslant \gamma_0 \Longrightarrow \forall n \in \mathbb{N} \quad \forall \delta \in \mathcal{F} \qquad \mathbb{P}_n(\delta) \in \left[\frac{\mathcal{U}_\mathcal{F}}{2} ; \frac{2\omega}{|\mathcal{F}|} \right]$$

Par ailleurs, la condition qui donne le plus de poids à $\mathcal{E}_1$ par rapport à $\mathcal{E}_2$ (α/β maximal) est

$$\frac{\alpha}{\beta} = \frac{2\beta}{|\omega| \ \mathrm{Sup}\,(g)} \frac{\mathcal{U}_\mathcal{F}^2}{4} \tag{C}$$

Preuve : Voir Annexe B □

Le théorème précédent donne donc une condition déterministe pour assurer que l'équation de récurrence $(\mathbf{E}-\mathbf{5})$ définit une suite d'éléments de $\mathcal{S}_\mathcal{F}$. Mais l'importance qu'accorde cette condition $(\mathbf{C})$ à l'énergie $\mathcal{E}_1$ est petite par rapport à la contribution de $\mathcal{E}_2$, le rapport entre α et β étant en effet trop faible. En réalité, ce problème provient de la discrétisation de l'équation différentielle limite (*cf* paragraphe précédent où il est démontré la stabilité en temps continu de $\mathcal{S}_\mathcal{F}$ sous une contrainte moins « dure » : l'inégalité (3.16) est en effet plus exigeante que (3.13)).

3.4.3 Pistes pour contourner (C)

Nous avons dès lors envisagé de modifier l'énergie $\mathcal{E}$ pour s'affranchir de conditions comme $(\mathbf{C})$ ou (3.16), et pouvoir donner autant d'importance que souhaité au terme d'erreur $\mathcal{E}_1$ dans $\mathcal{E}$.

- Une première solution imaginée a été de remplacer la définition de $\mathcal{E}$ donnée par $(\mathbf{E})$ en

$$\forall \mathbb{P} \in \mathcal{S}_\mathcal{F} \qquad \widetilde{\mathcal{E}}(\mathbb{P}) = \mathcal{E}_1(\mathbb{P}) + \underbrace{\sum_{\delta \in \mathcal{F}} \mathbb{P}(\delta) \log \mathbb{P}(\delta)}_{=\mathcal{E}_3(\mathbb{P})}$$

Cette énergie à minimiser est encore une fois constituée de deux termes : celui d'erreur $\mathcal{E}_1$ est identique à celui de $(\mathbf{E})$, le terme $\mathcal{E}_3$ est un terme entropique. Comme il s'agit de minimiser $\mathcal{E}$, le terme $\mathcal{E}_3$ constitue là encore un rappel vers la loi uniforme $\mathcal{U}_\mathcal{F}$. La nouvelle équation différentielle (remplaçant $(\mathbf{E}-\mathbf{3})$) sur $(\mathbb{P}_t)_{t\in\mathbb{R}}$ devient alors

$$\begin{aligned} \frac{\mathrm{d}\mathbb{P}_t(\delta)}{\mathrm{d}t} &= -\alpha \Pi_{\mathcal{H}_\mathcal{F}}(\mathbb{P}_t)(\nabla \mathcal{E}_1(\mathbb{P})(\delta)) + \beta \log \left(\frac{\mathbb{P}_t(\delta_1) \dots \mathbb{P}_t\left(\delta_{|\mathcal{F}|}\right)}{\mathbb{P}_t(\delta) \dots \mathbb{P}_t(\delta)} \right)^{1/|\mathcal{F}|} \\ &= -\alpha \Pi_{\mathcal{H}_\mathcal{F}}(\mathbb{P}_t)(\nabla \mathcal{E}_1(\mathbb{P})(\delta)) - \beta \Pi_{\mathcal{H}_\mathcal{F}}(\mathbb{P}_t)\left(\nabla \mathcal{E}_3(\mathbb{P}_t)\right) \end{aligned}$$

On constate immédiatement que pour cette équation différentielle, on ne peut avoir $\lim\limits_{t\mapsto+\infty} \mathbb{P}_t(\delta) = 0$ puisque sans condition sur α et dès que $\beta > 0$, on a

$$\lim_{\mathbb{P}_t(\delta)\mapsto 0} \frac{\mathrm{d}\mathbb{P}_t(\delta)}{\mathrm{d}t} = +\infty$$

Le terme en $\alpha \Pi_{\mathcal{H}_\mathcal{F}}(\mathbb{P}_t)(\nabla \mathcal{E}_1(\mathbb{P})(\delta))$ étant borné, pour tout δ par $\alpha|\omega| \ \mathrm{Sup}\,(g)$, on peut donc trouver ε tel que $\mathcal{S}_\mathcal{F} \setminus_{\mathcal{V}_\varepsilon(\partial \mathcal{S}_\mathcal{F})}$ est stable par l'équation différentielle précédente, inconditionnellement en α et β, ou encore :

$$\exists\, \varepsilon > 0 \quad \forall \mathbb{P} \in \mathcal{S}_\mathcal{F} \qquad \varepsilon < \left|\mathbb{P}(\delta)\right| < 2\varepsilon \Rightarrow \frac{\mathrm{d}\mathbb{P}_t(\delta)}{\mathrm{d}t} > 0$$

Cependant, cette approche n'a pas été retenue puisque pour simuler cette équation différentielle, il s'agit là encore d'effectuer une approximation stochastique. Malheureusement, l'approximation stochastique s'avère performante mais inutile car le terme de rappel $\mathcal{E}_3$

« tire » trop fort vers $\mathcal{U}$, de sorte que l'apprentissage ne permet pas d'obtenir des performances de diminution de $\mathcal{E}_1$ convaincantes. Nous n'avons pas réussi, dans ce cas, à trouver une combinaison de coefficients α et β satisfaisants pour un tel apprentissage.

- L'autre solution étudiée a été de choisir de définir structurellement la stabilité de $\mathcal{S}_\mathcal{F}$. On propose, à chaque fois qu'une frontière de $\mathcal{S}_\mathcal{F}$ est heurtée par une trajectoire (ou trajectoire approchée) de l'équation différentielle, une direction de réflexion qui permet de maintenir l'algorithme dans le simplexe étudié. L'approche théorique est alors plus ardue et nous consacrerons les chapitres suivants à un tel algorithme.

3.5 Approximation stochastique de la descente de gradient $(\mathbf{E}-\mathbf{2})$

Le cas où l'on cherche à approcher le comportement limite de $(\mathbf{E}')$ se traite de manière identique à celui concernant l'étude de $(\mathbf{E})$. On cherche donc d_n et k_n pour que l'équation de récurrence en variables « exponentielles »

$$y_{n+1} = y_n - \gamma_{n+1}(d_n - k_n) \tag{3.17}$$

ou bien

$$\mathbb{P}_{n+1} = \mathbb{P}_n e^{-\gamma_{n+1}(d_n - k_n)} \tag{3.18}$$

soit proche à la limite de $(\mathbf{E}-\mathbf{2})$. d_n est ici encore une approximation du gradient tandis que k_n est là aussi une constante de recentrage issue de la projection sur l'espace des contraintes. Or, l'équation d'évolution $(\mathbf{E}-\mathbf{2})$ s'écrit

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -\mathbb{E}[d_t(\delta) - k_t(\delta)] \qquad \text{pour (3.17)}$$

où
$$k_t(\delta) = \frac{\sum\limits_{\mu} \mathbb{P}_t(\mu) \nabla \widetilde{\mathcal{E}}(y_t)(\mu)}{\sum\limits_{\mu} \mathbb{P}_t(\mu)^2} \mathbb{P}_t(\delta)$$

De la même façon que dans le paragraphe précédent, l'approximation en y se traduit par la descente de gradient stochastique :

$$\boxed{y_{n+1} = y_n - \gamma_{n+1}\left[\alpha g(\omega_n)\mathrm{C}(\omega_n, .) - 2\beta\gamma_n \mathbb{P}_n(.)\left(\frac{1}{|\mathcal{F}|} - \mathbb{P}_n(.)\right) + k_n(\delta)\right]} \qquad (\mathbf{E}-\mathbf{6})$$

soit finalement comme équation « en $\mathbb{P}$ » :

$$\boxed{\forall n \in \mathbb{N} \qquad \mathbb{P}_{n+1}(.) = \mathbb{P}_n(.)\frac{e^{-\gamma_{n+1}\left[\alpha g(\omega_n)\mathrm{C}(\omega_n,.) - \beta\gamma_n \mathbb{P}_n(.)(\frac{1}{|\mathcal{F}|} - \mathbb{P}_n(.)) + k_n(\delta)\right]}}{\mathrm{K}_n}} \qquad (\mathbf{E}-\mathbf{6}')$$

où K_n est une constante de normalisation assurant que

$$\sum_{\delta} \mathbb{P}_{n+1}(\delta) = 1$$

En développant la somme $\sum \mathbb{P}_n(\delta) e^{-\gamma_{n+1}(d_n - k_n)}$ à l'ordre 2 en γ_n, on constate alors que

$$\mathrm{K}_n = 1 + \mathrm{O}(\gamma_n^2)$$

et les équations $(\mathbf{E}-\mathbf{6})$ et $(\mathbf{E}-\mathbf{6}')$ sont donc équivalentes à l'ordre 1.

Là encore, ω_n est une variable aléatoire de loi $\mathbb{P}_n^{\times p}$ avec remise sur $\mathcal{F}^p$. Il n'y a pas comme dans la section précédente d'étude à effectuer sur l'appartenance à $\mathcal{S}_{\mathcal{F}}$ pour chaque itération de la suite de probabilités puisque par construction, on a bien les conditions réunies :

$$\forall n \in \mathbb{N} \qquad \begin{cases} \mathbb{P}_n(\delta) \geqslant 0 \\ \sum\limits_{\delta \in \mathcal{F}} \mathbb{P}_n(\delta) = 1 \end{cases}$$

Le seul inconvénient de cette approche est, nous l'avons vu au paragraphe 4.2.3.2, sa lenteur et notamment lorsque $\mathbb{P}$ approche 0.

3.6 Convergence de l'apprentissage $(\mathbf{E}-\mathbf{6})$ de $\mathbb{P}$ vers $(\mathbf{E}-\mathbf{2})$

Nous allons établir dans un premier temps quelques définitions et propriétés sur les trajectoires associées à un flot d'équation différentielle. Nous utiliserons notamment une topologie sur ces trajectoires, topologie qui est précisément décrite dans [Ben96] et [Ben00]. Par ailleurs, les algorithmes d'apprentissages qui seront utilisés appartiennent à la classe générale d'algorithmes stochastiques de Robbins-Monro. Ces algorithmes font l'objet de nombreuses études : on pourra se reporter à [BMP], [Duf96] pour une étude exhaustive de ces algorithmes et à [You91], [You89] pour des utilisations de tels algorithmes pour l'estimation de paramètres issus de champs de Gibbs.

3.6.1 Généralités sur les équations différentielles

Commençons par définir les trajectoires $\Phi_t(x)$, trajectoires en temps initialisées en x, point quelconque d'un espace métrique.

Définition 3.6.1 (semi-flot)
On appelle semi-flot Φ sur (M, d) espace métrique une application continue

$$\Phi : (t,x) \in \mathbb{R}^+ \times \mathrm{M} \longmapsto \Phi_t(x) = \Phi(t,x) \in \mathrm{M}$$

telle que

$$\begin{cases} \Phi_0 = \mathrm{I}d \\ \Phi_{t+s} = \Phi_t \circ \Phi_s \end{cases}$$

La définition du semi-flot nous permet de définir une convergence d'un ensemble de trajectoires $\left(\Theta^h(\mathrm{X})\right)_{h \geqslant 0}$ vers la solution d'une équation différentielle définie par son flot. Ceci est rendu possible grâce à la notion développée dans la définition 3.6.2.

Définition 3.6.2 (Pseudo-trajectoire asymptotique)
Une fonction continue $\mathrm{X} : \mathbb{R}^+ \longrightarrow \mathrm{M}$ est une pseudo-trajectoire de Φ si

$$\forall \mathrm{T} > 0 \qquad \lim_{t \mapsto \infty} \operatorname*{Sup}_{0 \leqslant h \leqslant \mathrm{T}} d(\mathrm{X}(t+h), \Phi_h(\mathrm{X}(t))) = 0$$

Cette notion de pseudo-trajectoire a été introduite par Benaïm et Hirsh et se révèle particulièrement adaptée pour l'étude des processus stochastiques. Nous verrons dans le paragraphe suivant comment cette définition sera utilisée dans le cadre de trajectoires de processus aléatoires.

Définition 3.6.3 (Opérateur de translation Θ)
Θ *est l'opérateur de* $\mathcal{C}^0(\mathbb{R},\mathrm{M})\times\mathbb{R}\longrightarrow\mathcal{C}^0(\mathbb{R},\mathrm{M})$ *tel que*

$$\forall \mathrm{X}\in\mathcal{C}^0(\mathbb{R},\mathrm{M})\quad\forall(s,t)\in\mathbb{R}^2\qquad\Theta^t(\mathrm{X})(s)=\mathrm{X}(s+t)$$

Θ correspond donc à l'opérateur de décalage temporel d'une trajectoire X. Il sera alors intéressant de regarder l'ensemble des propriétés topologiques de l'ensemble des trajectoires $\left(\Theta^h(\mathrm{X})\right)_{h\geqslant 0}$.

Définition 3.6.4 (Trajectoires S_Φ associées au flot Φ)
S_Φ *est l'ensemble des trajectoires associées au flot* Φ*, et une trajectoire est précisément la donnée de* $\Phi^p : t\longmapsto\Phi_t(p)$.

Ainsi $$\mathrm{S}_\Phi=\{t\mapsto\Phi_t(p)\quad|\quad p\in\mathrm{M}\}$$

On note H l'homéomorphisme qui à un point p de M associe sa trajectoire dans S_Φ. On obtient immédiatement la propriété de conjugaison :

$$\forall t>0\qquad\Theta^t\circ\mathrm{H}=\mathrm{H}\circ\Phi_t$$

Enfin, pour définir la notion de convergence vers des trajectoires, il est nécessaire de définir une métrique sur l'ensemble des trajectoires continues de $\mathbb{R}$ dans $\mathcal{S}_\mathcal{F}$. La topologie alors obtenue est celle de la convergence uniforme sur tous les intervalles compacts de $\mathbb{R}$, basée sur la métrique donnée dans la définition 3.6.5 qui suit.

Définition 3.6.5 (Métrique sur $\mathcal{C}^0\left(\mathbb{R},\mathcal{S}_\mathcal{F}\right)$)
Si f *et* g *désignent deux trajectoires continues, alors l'application* d *définie une distance sur* $\mathcal{C}^0\left(\mathbb{R},\mathcal{S}_\mathcal{F}\right)$ *par :*

$$d(f;g)=\sum_{k\in\mathbb{N}}\frac{1}{2^k}\left\{\mathrm{Min}\left\{1;\left\|\left.(f-g)\right|_{[-k\,;k]}\quad\right\|_\infty\right\}\right\}$$

On peut dès lors énoncer un théorème de caractérisation des pseudo-trajectoires utilisant la métrique précédente. Ce théorème nous permettra par la suite de caractériser l'approximation d'une trajectoire par l'algorithme stochastique défini précédemment.

Théorème 3.6.1 (Caractérisation des pseudo-trajectoires, Benaïm)
Soit $\mathrm{X}:\mathbb{R}^+\longrightarrow\mathrm{M}$ *une application continue précompacte, les propositions suivantes sont équivalentes*

- *(i)* X *est une pseudo-trajectoire asymptotique de* Φ.
- *(ii)* X *est uniformément continu et tout point de* $\overline{\{\Theta^t(\mathrm{X})\}_{t\geqslant 0}}$ *appartient à* S_Φ.

3.6.2 Convergence vers une pseudo-trajectoire asymptotique

Afin de se référer très précisément aux travaux [Ben00], nous allons suivre les notations utilisées dans ce travail. On suppose donc (ce qui est le cas au vu de $(\mathbf{E}-\mathbf{6})$) que l'équation d'évolution satisfait l'équation stochastique :

$$y_{n+1}-y_n=\gamma_{n+1}\left(\mathrm{F}(y_n)+\delta\mathrm{U}_{n+1}\right)\tag{3.19}$$

Pour coller parfaitement au cadre théorique des algorithmes de Robbins-Monro, on impose que

1. $(\gamma_n)_{n\in\mathbb{N}}$ est une suite de réels strictement positifs, décroissante vers 0.

2. $\delta \mathrm{U}_n$ est une suite de perturbations aléatoires de $\mathbb{R}^{|\mathcal{F}|}$.
3. Si $(\mathcal{F}_n)_{n\in\mathbb{N}}$ désigne la filtration croissante adaptée au processus $(y_n)_{n\in\mathbb{N}}$ donnée par

$$\mathcal{F}_n = \sigma(y_0, \ldots, y_n)$$

alors le processus $(\delta \mathrm{U}_n)_{n\in\mathbb{N}}$ est adapté à la filtration et vérifie

$$\mathbb{E}[\delta \mathrm{U}_{n+1} | \mathcal{F}_n] = 0 \tag{3.20}$$

> La formule (3.19) est à rapprocher du schéma d'Euler explicite
>
> $$\mathrm{Y}_{k+1} - \mathrm{Y}_k = \gamma_{k+1} \mathrm{F}(\mathrm{Y}_k)$$
>
> mais ici, dans le cas de la formule (3.19), l'action de $\delta \mathrm{U}$ est en moyenne nulle.

On essaie donc de rapprocher le processus $\mathbb{P}_n$ ainsi que son processus continu interpolé avec les trajectoires de l'équation différentielle $(\mathbf{E}-\mathbf{3})$. Il sera alors commode d'utiliser les notations suivantes qui permettent de transformer une itération n de $\mathbb{N}$ en un temps $\tau(n)$ de $\mathbb{R}$.

Définition 3.6.6 (Transformation $\tau - m$)
Si n est un entier naturel, on pose

$$\tau_n = \sum_{k=0}^{n} \gamma_{k+1}$$

On note alors l'application « réciproque » de τ :

$$\forall u \in \mathbb{R} \qquad m(u) = \mathrm{Inf}\ \{n \in \mathbb{N} \quad | \quad \tau_n \geqslant u\}$$

Ces deux applications permettent d'associer à une itération donnée le temps qui correspond à l'évolution du schéma d'Euler explicite évoqué précédemment.
On constate d'emblée que le fait de discrétiser le temps variant dans $\mathbb{R}^+$ par la transformation $\tau - m$ implique que τ varie lui aussi dans $\mathbb{R}^+$, et atteigne également ses bornes. Finalement, on peut d'ores et déjà établir que la condtion

$$\lim_{n \mapsto +\infty} \tau_n = +\infty$$

doit être vérifiée. C'est-à-dire

$$\sum_{n\in\mathbb{N}} \gamma_n = +\infty$$

On définit également les processus (en temps continus) interpolés de $(\mathbb{P}_n)_{n\in\mathbb{N}}$ aux temps τ_n par la définition suivante.

Définition 3.6.7 (Interpolation du processus discret $\mathbb{P}$)
Si n désigne un entier naturel quelconque, les processus en temps continus $(y_t)_{t\geqslant 0}$ et $(\overline{y}_t)_{t\geqslant 0}$ sont définis sur tous les intervalles $]\,\tau_n\,;\tau_{n+1}\,[$ par

$$\forall n \in \mathbb{N} \quad \forall s \in\,]\,0\,;\gamma_{n+1}\,] \qquad \begin{cases} y_{s+\tau_n} = y_n + s\dfrac{y_{n+1} - y_n}{\tau_{n+1} - \tau_n} \\ \\ \overline{y}_{s+\tau_n} = y_n \end{cases}$$

Les processus y et $\overline{y}$ sont donc affines par morceaux et constants par morceaux. Par ailleurs, ils sont égaux en tous temps τ_n à $\mathbb{P}_n$, d'où l'interpolation. De manière exactement identique, on note enfin le processus continu $\overline{\delta \mathrm{U}}$:

$$\forall n \in \mathbb{N} \quad \forall s \in]\,0\,;\gamma_{n+1}\,] \qquad \overline{\delta \mathrm{U}}_{\tau_n+s} = \delta \mathrm{U}_{n+1}$$

Toutes ces notations nous permettent alors d'écrire l'équation intégrale approchée du processus discret :

$$y_t - y_0 = \int_0^t \left[\mathrm{F}(\overline{y}_s) + \overline{\delta \mathrm{U}}_s\right] ds \tag{3.21}$$

De plus, il est possible d'expliciter exactement la fonction F et la perturbation δU dans la propostion qui suit.

Propriété 3.6.1 (Expression de F et de δU :)
La fonction F *et le processus* δU *sont donnés par*

$$\mathrm{F}(y) = -\Pi(\nabla \widetilde{\mathcal{E}}(y))$$

et $\qquad \forall n \in \mathbb{N} \qquad \delta \mathrm{U}_{n+1} = -\Pi\left(\alpha g(\omega_n)\mathrm{C}(\omega_n,\delta) + 2\beta e^{y_n}(\mathcal{U}_{\mathcal{F}} - e^{y_n})\right) + \Pi(\nabla \widetilde{\mathcal{E}}(y_n))$

Preuve : Il suffit d'utiliser l'égalité (3.20). On a en effet :

$$\mathrm{E}[\delta \mathrm{U}_{n+1}|\mathcal{F}_n] = 0$$

Mais $\delta \mathrm{U}_{n+1}$ est également donné par l'expression :

$$\mathrm{F}(y_n) + \delta \mathrm{U}_{n+1} = \frac{y_{n+1} - y_n}{\gamma_{n+1}}$$

En prenant l'espérance conditionnée à $\mathcal{F}_n$ dans l'expression précédente, on obtient :

$$\underbrace{\mathrm{E}[\mathrm{F}(y_n)|\mathcal{F}_n]}_{=\mathrm{F}(y_n)} + \underbrace{\mathrm{E}[\delta \mathrm{U}_{n+1}|\mathcal{F}_n]}_{=0} = \frac{\mathrm{E}[y_{n+1}|\mathcal{F}_n] - \overbrace{\mathrm{E}[y_n|\mathcal{F}_n]}^{=y_n}}{\gamma_{n+1}}$$

Soit

$$\begin{aligned} \mathrm{F}(y_n) &= \frac{\mathrm{E}[y_{n+1}|\mathcal{F}_n] - \mathbb{P}_n}{\gamma_{n+1}} \\ &= -\nabla \widetilde{\mathcal{E}}(y) \end{aligned}$$

L'expression du processus ponctuel δU s'en déduit alors par soustraction :

$$\delta \mathrm{U}_{n+1} = \frac{y_{n+1} - y_n}{\gamma_{n+1}} - \mathrm{F}(y_{n+1}) \qquad \square$$

3.6.2.1 Comportement des perturbations δU

On peut de plus étudier la convergence du processus $(\mathrm{U}_n)_{n\in\mathbb{N}}$ défini par

$$\forall n \in \mathbb{N} \qquad \mathrm{U}_n = \sum_{i=1}^{n} \gamma_i \delta \mathrm{U}_i$$

ainsi que son interpolé $\left(\overline{\mathrm{U}_t}\right)_{t\in\mathbb{R}}$ constant par morceaux défini comme $\overline{\delta \mathrm{U}}$. Le point crucial de la démonstration repose sur le fait que (U_n) est une $\mathcal{F}_n$-martingale ([MPB98], [Wil91]). Nous allons, pour T une constante temporelle fixée, étudier la quantité

$$\lim_{n\to\infty} \mathrm{P}\left(\mathop{\mathrm{Sup}}_{j\geqslant n} \max_{0\leqslant t\leqslant \mathrm{T}} \left| \sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+t)-1} \gamma_i \delta \mathrm{U}_i \right| \geqslant \varepsilon \right)$$

où m est la fonction qui à un temps donné de $\mathbb{R}$ associe son itération correspondante dans $\mathbb{N}$. On peut tout d'abord énoncer le lemme évident (par construction de $\delta \mathrm{U}$)

Lemme 3.6.1
Le processus $(\mathrm{U}_n)_{n\in\mathbb{N}}$ *est une* $\mathcal{F}_n$*-martingale.*

Le lemme suivant donne un critère de comportement asymptotique de la queue du processus U.

Lemme 3.6.2
Si la série de terme général positif $q_j(\varepsilon)$ *défini par*

$$q_j(\varepsilon) = \mathrm{P}\left(\max_{0\leqslant t\leqslant \mathrm{T}} \left| \sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+t)-1} \gamma_i \delta \mathrm{U}_i \right| \geqslant \varepsilon \right)$$

converge, alors

$$\lim_{n\to\infty} \mathrm{P}\left(\mathop{\mathrm{Sup}}_{j\geqslant n} \max_{0\leqslant t\leqslant \mathrm{T}} \left| \sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+t)-1} \gamma_i \delta \mathrm{U}_i \right| \geqslant \varepsilon \right) = 0$$

Preuve : Il suffit d'appliquer le lemme de Borel-Cantelli à la série $\sum q_j(\varepsilon)$. On en déduit immédiatement que

$$\lim_{n\to\infty} \mathrm{P}\left(\mathop{\mathrm{Sup}}_{j\geqslant n} \max_{0\leqslant t\leqslant \mathrm{T}} \left| \sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+t)-1} \gamma_i \delta \mathrm{U}_i \right| \geqslant \varepsilon \right) = 0$$

C'est précisément ce qu'il fallait démontrer. □
La convergence vers 0 du terme précédent peut alors se reécrire en exprimant le lemme 3.6.3.

Lemme 3.6.3
Si la série de terme général positif $q_j(\varepsilon)$ *converge, alors*

$$\lim_{n\to+\infty} \mathop{\mathrm{Sup}}_{j\geqslant n} \mathop{\mathrm{Max}}_{0\leqslant t\leqslant \mathrm{T}} \left| \overline{\mathrm{U}_{j\mathrm{T}+t}} - \overline{\mathrm{U}_{j\mathrm{T}}} \right| = 0$$

Enfin, la proposition suivante assure que la série précédente de terme général $q_j(\varepsilon)$ converge bien, sous certaines conditions de décroissance des coefficients γ_n, ce que nous allons voir dans la prochaine propriété.

Propriété 3.6.2 (Convergence de $\sum q_j(\varepsilon)$)

Si les coefficients γ_n *varient en*

$$\gamma_n = \frac{\gamma}{n}$$

où γ *est une constante strictement positive, alors la série* $\sum q_j(\varepsilon)$ *est convergente.*

Preuve : Appliquons l'inégalité

$$\mathrm{P}_{\mathcal{F}_{t_0}}(\mathop{\mathrm{Sup}}_{t_0\leqslant t\leqslant t_1} |\mathrm{M}_t| \geqslant \lambda) \leqslant \frac{\mathrm{E}[q(\mathrm{M}_{t_1})|\mathcal{F}_{t_0}]}{q(\lambda)}$$

si M est une martingale et si q est une fonction convexe ([Bre68], chapitre 5) à la fonction $q(x) = \|x\|_2^{\,2}$ qui est convexe et à la $\mathcal{F}$-martingale M_t donnée par

$$\mathrm{M}_t = \sum_{m(j\mathrm{T})}^{m(j\mathrm{T}+t)} \gamma_i \delta \mathrm{U}_i$$

On obtient donc

$$q_j(\varepsilon) \leqslant \frac{\mathbb{E}\left[\left[\sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+\mathrm{T})} \gamma_i \delta \mathrm{U}_i\right]^2 \Big| \mathcal{F}_{m(j\mathrm{T})}\right]}{\varepsilon^2}$$

En développant la somme au carré, et en supprimant les « doubles produits » qui ont une espérance nulle sous $\mathcal{F}_{m(j\mathrm{T})}$ puisque les accroissements $\delta \mathrm{U}_i$ sont centrés, alors

$$q_j(\varepsilon) \leqslant \frac{\mathbb{E}\left[\sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+\mathrm{T})} \gamma_i^2 \delta \mathrm{U}_i^2 + \sum_{m(j\mathrm{T})\leqslant k<n\leqslant m(j\mathrm{T}+\mathrm{T})} \gamma_k \gamma_n \delta \mathrm{U}_k \delta \mathrm{U}_n \Big| \mathcal{F}_{m(j\mathrm{T})}\right]}{\varepsilon^2}$$

$$\leqslant \frac{\mathbb{E}\left[\sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+\mathrm{T})} \gamma_i^2 \delta \mathrm{U}_i^2 \Big| \mathcal{F}_{m(j\mathrm{T})}\right]}{\varepsilon^2} + \frac{\sum_{m(j\mathrm{T})\leqslant k<n\leqslant m(j\mathrm{T}+\mathrm{T})} \gamma_k \gamma_n \mathbb{E}\left[\delta \mathrm{U}_k \delta \mathrm{U}_n \Big| \mathcal{F}_{n-1} \Big| \mathcal{F}_{m(j\mathrm{T})}\right]}{\varepsilon^2}$$

$$q_j(\varepsilon) \leqslant \frac{\mathbb{E}\left[\sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+\mathrm{T})} \gamma_i^2 \delta \mathrm{U}_i^2 \Big| \mathcal{F}_{m(j\mathrm{T})}\right]}{\varepsilon^2} + \frac{\sum_{m(j\mathrm{T})\leqslant k<n\leqslant m(j\mathrm{T}+\mathrm{T})} \gamma_k \gamma_n \mathbb{E}\left[\delta \mathrm{U}_k \overbrace{\mathbb{E}\left[\delta \mathrm{U}_n \Big| \mathcal{F}_{n-1}\right]}^{=0} \Big| \mathcal{F}_{m(j\mathrm{T})}\right]}{\varepsilon^2}$$

soit

$$q_j(\varepsilon) \leqslant \frac{\mathbb{E}\left[\sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+\mathrm{T})} \gamma_i^2 \delta \mathrm{U}_i^2 \Big| \mathcal{F}_{m(j\mathrm{T})}\right]}{\varepsilon^2}$$

Par ailleurs, le terme $\delta \mathrm{U}_n^2$ est borné pour tout entier n quelconque puisque

$$\forall n \in \mathbb{N} \qquad \delta \mathrm{U}_{n+1} = -\Pi\left(\alpha g(\omega_n)\mathrm{C}(\omega_n, \delta) + 2\beta e^{y_n}(\mathcal{U}_{\mathcal{F}} - e^{y_n})\right) + \Pi(\nabla \widetilde{\mathcal{E}}(y_n))$$

On a ici une différence de deux termes de projection sur le compact $\mathcal{S}_{\mathcal{F}}$ et donc

$$\|\delta \mathrm{U}_n\|_2^{\,2} \leqslant 2\big|\mathcal{F}\big|$$

Ainsi

$$q_j(\varepsilon) \leqslant \frac{\mathrm{K}^2 \sum_{i=m(j\mathrm{T})}^{m(j\mathrm{T}+\mathrm{T})} \gamma_i^2}{\varepsilon^2}$$

Au vu du choix des coefficients γ_n, on en déduit que la série est convergente. □

Avec ce qui vient d'être établi, nous pouvons donc énoncer le théorème de convergence 3.6.2.

Théorème 3.6.2 (Convergence de U)
Les choix de décroissance des coefficients γ_n assurent que

$$\lim_{n\to+\infty} \operatorname*{Sup}_{j\geqslant n} \operatorname*{Max}_{0\leqslant t\leqslant \mathrm{T}} |\,\overline{\mathrm{U}_{j\mathrm{T}+t}} - \overline{\mathrm{U}_{j\mathrm{T}}}\,| = 0$$

En examinant les démonstrations précédentes, on constate qu'il suffit de faire des hypothèses sur le comportement de la suite $(\gamma_n)_{n\in\mathbb{N}}$ pour assurer la convergence vers 0 du processus U. Les deux hypothèses fondamentales à satisfaire sont

1. Les coefficients γ_n doivent permettre de discrétiser tout temps t réel, c'est-à-dire m est définie sur tout $\mathbb{R}$, soit encore :
$$\sum_{n\in\mathbb{N}} \gamma_n = +\infty$$
2. Les coefficients γ_n doivent décroître suffisament rapidement pour que
$$\sum_{n\in\mathbb{N}} \gamma_n^{1+\alpha} < +\infty$$
où α est un réel strictement positif. La condition suivante provient alors du fait qu'il suffit d'avoir un moment borné à un ordre quelconque de la martingale $(\mathrm{M}_t)_{t\geqslant 0}$ pour assurer la convergence de $q_j(\varepsilon)$. L'application de l'inégalité utilisée au début de la démonstration de la propriété 3.6.2, ou de l'inégalité de Burkholder [Wil91], [BMP] joue alors un rôle fondamental dans la démonstration d'une telle convergence.

Typiquement, les coefficients γ_n varieront donc en $\gamma_n = \gamma/n$

3.6.2.2 Convergence vers une pseudo-trajectoire de $(y_t)_{t\in\mathbb{R}}$

Nous avons démontré dans le paragraphe précédent que la perturbation du schéma d'Euler qui est en somme discrétisé par notre algorithme tendait vers 0. Il suffit donc pour conclure à la convergence vers une pseudo-trajectoire d'utiliser des arguments élémentaires de continuité et le théorème 4.6.1 caractérisant les pseudo-trajectoires. Plus précisément, si Φ désigne le flot associé à l'équation différentielle :

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = \mathrm{F}(y_t)$$

où F est l'application continue $-\Pi(\nabla\widetilde{\mathcal{E}}(y))$, on a alors le théorème caractérisant le comportement asymptotique du processus interpolé P qui est démontré ci-dessous.

Théorème 3.6.3 (Convergence vers une pseudo-trajectoire)
Le processus interpolé P *est une pseudo-trajectoire asymptotique de* Φ.

Preuve :

Nous allons suivre exactement la démarche de [Ben96] en appliquant le théorème de caractérisation des pseudo-trajectoires vu précédemment. Démontrons tout d'abord que le processus P est uniformément continu. Le processus $(y_t)_{t\geqslant 0}$ appartient à tout instant à $\mathcal{S}_{\mathcal{F}}$. Ainsi, par continuité de F et compacité de $\mathcal{S}_{\mathcal{F}}$, il existe un réel K tel que

$$\forall \mathrm{P} \in \mathcal{S}_{\mathcal{F}} \quad | \quad \mathrm{P} = e^y \qquad |\,\mathrm{F}(y)\,| \leqslant \mathrm{K}$$

En utilisant alors :

$$y_t - y_0 = \int_0^t \mathrm{F}(y_s) + \overline{\mathrm{U}}_s \,\mathrm{d}s$$

et le comportement asymptotique de U, on obtient alors :

$$\forall \mathrm{T} > 0 \qquad \limsup_{t \mapsto \infty} \operatorname*{Sup}_{0 \leqslant h \leqslant \mathrm{T}} \mid y_{t+h} - y_t \mid \leqslant \mathrm{KT}$$

Ainsi, le processus P est uniformément continu.

Par ailleurs, en utilisant le flot de l'équation différentielle Θ, on peut écrire que :

$$\Theta^t(y)_h = \Phi(h, \Theta^t(y)) + \mathrm{A}_t(h) + \mathrm{B}_t(h)$$

où l'on a

$$\Phi(h, y) = y_0 + \int_0^h \mathrm{F}(y_u)\mathrm{d}u$$

et

$$\begin{cases} \mathrm{A}_t(s) = \displaystyle\int_t^{t+s} \mathrm{F}(\overline{y}_u) - \mathrm{F}(y_u)\mathrm{d}u \\ \mathrm{B}_t(s) = \displaystyle\int_t^{t+s} \overline{\mathrm{U}}_u \mathrm{d}u \end{cases}$$

En vertu du comportement asymptotique de $\overline{\mathrm{U}}$, on a toujours :

$$\lim_{t \mapsto \infty} \mathrm{B}_t = 0$$

Enfin, on a :

$$y_t - \overline{y}_t = \int_{\tau(m(t))}^t \mathrm{F}(\overline{y}_s) + \overline{\mathrm{U}}_s \, \mathrm{d}s$$

Ainsi, on obtient la majoration par uniforme continuité de F :

$$\mid y_t - \overline{y}_t \mid \leqslant \mathrm{K}\gamma_{m(t)} + \mid \int_{\tau(m(t))}^t \overline{\mathrm{U}}_s \, \mathrm{d}s \mid$$

Et on obtient donc également :

$$\lim_{t \mapsto \infty} \mathrm{A}_t = 0$$

On conclut donc que si $y^\star$ est un point de $\overline{\{\Theta^t(y)\}_{t \geqslant 0}}$, alors $y^\star$ vérifie :

$$y^\star = \Phi(y^\star)$$

On peut donc appliquer le théorème de caractérisation des pseudo-trajectoires pour achever la démonstration. □

3.6.3 Convergence vers un minimum de $\mathcal{E}$

3.6.3.1 Convexité de $\mathcal{E}$

Rappelons l'expression de $\mathcal{E}$:

$$\mathcal{E}(\mathbb{P}) = \alpha \, \mathbb{E}_{\mathbb{P}}[g(\omega)] + \beta \underbrace{\left\| \mathcal{U} - \mathbb{P} \right\|_2^2}_{\text{Convexe}}$$

On ne peut donc pas conclure directement en utilisant un résultat élémentaire de convexité puisqu'ici nous sommes en présence d'une somme de deux fonctionnelles : l'une convexe et l'autre non. On peut énoncer un résultat qui assure la convexité de $\mathcal{E}$ sous des conditions restrictives comparables à (3.16), et qui ne seront donc pas respectées en pratique. Ce résultat donne une borne exacte qui assure que la perturbation d'une fonctionnelle convexe reste convexe si l'amplitude de la perturbation est suffisament faible (majoration de α).

Théorème 3.6.4 (Convexité de $\mathcal{E}$)
Si les coefficients α et β vérifient l'inégalité

$$\beta \geqslant \frac{\alpha|\omega|^2 \ \mathrm{Sup}\,(g)}{2} \tag{3.22}$$

alors la fonctionnelle $\mathcal{E}$ est convexe et admet donc un unique minimum.

Preuve : Voir Annexe B. □

Cette condition est à rapprocher de (3.16) qui assurait la convergence de l'algorithme. On constate qu'on se retrouve ici également une borne pour α moins exigeante que celle en « $1/|\mathcal{F}|^2$ » obtenue en (3.16) puisque

$$\forall(\delta,\widetilde{\delta}) \quad \forall\omega \in \mathcal{F}^2 \qquad \mathrm{C}(\omega,\delta)\mathrm{C}(\omega,\mu) \leqslant |\omega|^2$$

et que donc la condition (3.77, *cf* Annexe B) est déjà plus souple (plus facilement satisfiable) que l'inégalité :

$$\beta > \frac{\alpha|\omega|^2}{2}$$

et bien entendu, on a (3.16) $\Longrightarrow$ (3.22). Ainsi, l'inégalité (3.16) implique également la convexité de la fonctionnelle, et donc la convergence de la descente de gradient vers le minimum absolu de $\mathcal{E}$ sur $\mathcal{S}_{\mathcal{F}}$.

3.6.3.2 Convergence du processus y

On suppose ici que le processus est bien confiné à $\mathcal{S}_{\mathcal{F}}$, c'est-à-dire lorsque (3.16) est vérifiée, on sait alors que E est convexe d'après le paragraphe précédent. Pour obtenir la convergence de $\mathbb{P}_n$ vers la probabilité réalisant le minimum de $\mathcal{E}$, il suffit de remarquer que

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = -\nabla^\star\mathcal{E}(y_t) = \mathrm{F}(y_t)$$

L'équation différentielle précédente admet donc un unique équilibre, par convexité de $\mathcal{E}$ qui est différentiable, il s'agit du minimum de $\mathcal{E}$ et en repassant aux variables « en $\mathbb{P}$ » :

$$\lim_{t\to\infty} \mathrm{P}_t = \arg\operatorname*{Min}_{\mathrm{P}\in\mathcal{S}_{\mathcal{F}}} \mathcal{E}(\mathrm{P})$$

Comme la suite $(\mathbb{P}_n)_{n\in\mathbb{N}}$ est une pseudo-trajectoire asymptotique pour l'équation différentielle précédente, on peut en conclure le théorème :

Théorème 3.6.5 (Convergence vers le minimum de $\mathcal{E}$)
Dans le cas où l'inégalité (3.16) est vraie, et si la suite $(\mathbb{P}_n)_{n\in\mathbb{N}}$ suit $(\mathbf{E}-\mathbf{2})$, on a

$$\lim_{n\to\infty} \mathbb{P}_n = arg\operatorname*{Min}_{\mathrm{P}\in\mathcal{S}_{\mathcal{F}}} \mathcal{E}(\mathrm{P})$$

On peut par ailleurs étendre tous les résultats des deux dernières sections au cas de la descente de gradient $(\mathbf{E}-\mathbf{1})$ discrétisant à la limite l'équation différentielle $(\mathbf{E}-\mathbf{3})$. Il est alors nécessaire de manipuler les variables « $\mathbb{P}$ » au lieu des variables en y. L'étude des perturbations $\gamma_n\widetilde{\delta\mathrm{U}_n}$ est même identique puisque celles-ci sont bornées également.

La convexité de $\mathcal{E}$, elle, n'est pas influencée par la manipulation des variables « en $\mathbb{P}$ » ou « exponentielles ». Nous obtenons donc le résultat similaire exprimé dans le théorème qui suit.

Théorème 3.6.6 (Convergence vers le minimum de $\mathcal{E}$)
Si la suite $(\mathbb{P}_n)_{n\in\mathbb{N}}$ suit $(\mathbf{E}-\mathbf{1})$ et si (3.16) est de plus vérifiée, alors on a la convergence de $(\mathbb{P}_n)_{n\in\mathbb{N}}$ vers le minimum global de $\mathcal{E}$.

3.7 Expériences sur des données synthétiques

Nous étudions tout d'abord l'effet, sur des données synthétiques, d'un tel algorithme d'apprentissage. Il s'agit ici de valider sur ce cas simple notre modélisation, plutôt que de proposer une méthode de résolution car le problème pourra se résoudre beaucoup plus simplemenent par des considérations statistiques.

3.7.1 Description des données

L'objet de ce paragraphe est d'illustrer l'augmentation de performance moyenne que permet la descente de gradient précédente. Les données qui sont étudiées sont issues de modèles synthétiques aléatoires simples. Chacun des éléments de la base de données appartient à une des classes $\mathcal{C}_i$ et est quantifié par des valeurs ternaires $\{-1;0;1\}$ sur l'ensemble $\mathcal{F}$ des features. L'ensemble $\mathcal{F}$ est, quand à lui, supposé donné.

Nous pouvons donner explicitement les règles de construction qui nous ont permis de générer notre problème de classification synthétique.

Définition 3.7.1 (Données synthétiques)
Pour chacune des classes $\mathcal{C}_i$, on définit plusieurs sous-ensembles ${\mathcal{F}_i}^k$ de $\mathcal{F}$ tels que

$$\forall i \in [\![\, 1\,;\, |\mathrm{C}|\,]\!] \quad \exists k \quad \forall \delta \in {\mathcal{F}_i}^k \qquad \mathrm{P}\left(\delta(\mathrm{I}) = 1 \,|\, \mathrm{I} \in \mathcal{C}_i\right) = \frac{9}{10}$$

et $\quad \forall i \in [\![\, 1\,;\, |\mathrm{C}|\,]\!] \quad \exists k \quad \forall \delta \in {\mathcal{F}_i}^k \qquad \mathrm{P}\left(\delta(\mathrm{I}) = 0 \,|\, \mathrm{I} \in \mathcal{C}_i\right) = \mathrm{P}\left(\delta(\mathrm{I}) = -1 \,|\, \mathrm{I} \in \mathcal{C}_i\right) = \frac{1}{20}$

tandis que

$$\forall i \quad \forall k \quad \forall \delta \notin {\mathcal{F}_i}^k \qquad \mathrm{P}\left[\delta(\mathrm{I}) = 1 \,|\, \mathrm{I} \in \mathcal{C}_i\right] = \mathrm{P}\left[\delta(\mathrm{I}) = -1 \,|\, \mathrm{I} \in \mathcal{C}_i\right] = \mathrm{P}\left[\delta(\mathrm{I}) = 0 \,|\, \mathrm{I} \in \mathcal{C}_i\right] = \frac{1}{3}$$

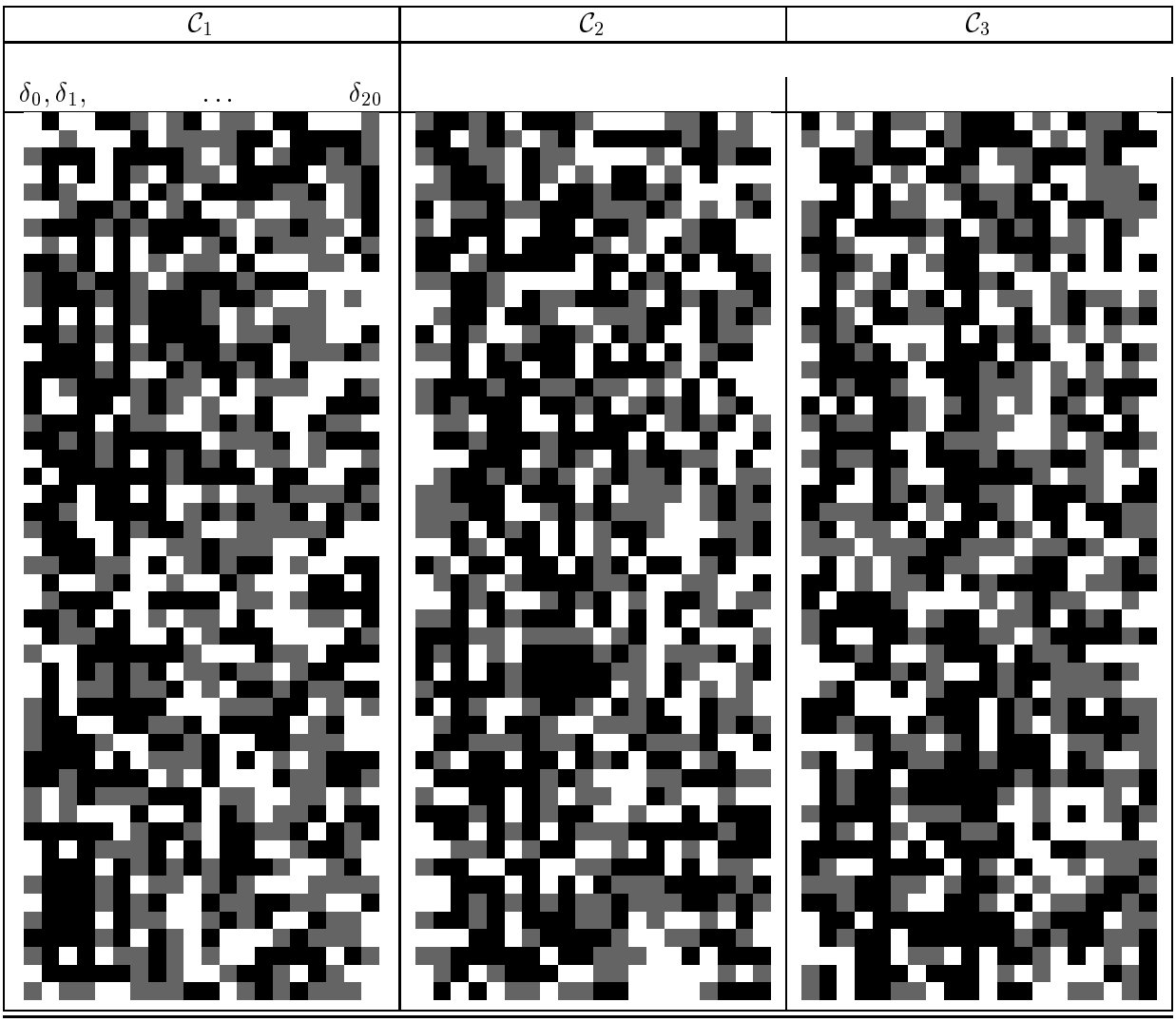
Les données ainsi générées correspondent donc à des données ternaires fortement corrélées sur certains features et complètement décorrélées sur d'autres. L'objectif est alors double :

- optimiser les performances de classification basée sur certains features
- rechercher les features caractéristiques des classes $\mathcal{C}_i$.

Les données présentées ici ont été étudiées dans le cas d'un problème de classification à trois classes avec un ensemble de features $\mathcal{F}$ de 20 tests ternaires. D'autres expériences ont été faites et ont permis d'aboutir à des conclusions similaires dans le cas d'un nombre plus grand de classes, dans le cas de tests uniquement binaires, ou bien dans le cas où l'ensemble $\mathcal{F}$ possède un cardinal plus important (plusieurs centaines).

On peut visualiser un extrait de cette base de données en représentant l'ensemble des features $\mathcal{F}$ par des cases pouvant être noires (+1), grises (-1) ou blanches (0).

Les échantillons sont représentés « par ligne » tandis que les variables δ sont rangées en colonne.



On constate ici qu'il paraît difficile, d'emblée, d'exhiber les règles de construction des différentes classes $\mathcal{C}_i$, ainsi que les features qui permettront d'obtenir le meilleur taux d'erreur lors de la phase de classification.

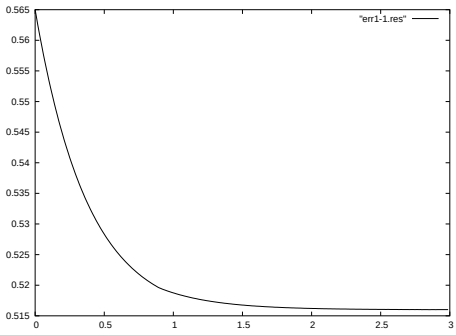
Dans notre cas précis, le calcul de g s'effectue en appliquant un algorithme $\mathbb{A}$ de k-plus proche voisin, en prenant k égal à 4. La distance choisie d est définie par

$$\forall (\mathrm{I}_1; \mathrm{I}_2) \in \mathcal{I}^2 \qquad d(\mathrm{I}_1; \mathrm{I}_2) = \sum_{\delta \in \mathcal{F}} |\delta(\mathrm{I}_1) - \delta(\mathrm{I}_2)|$$

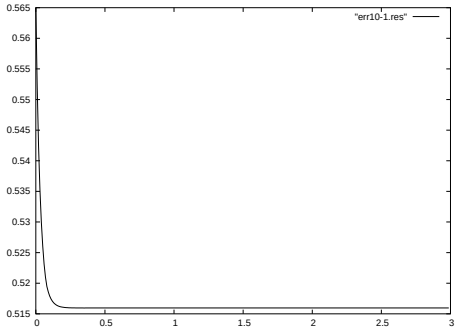
3.7.2 Descente de gradient exacte

Le cas particulier des données synthétiques est relativement intéressant vis-à-vis des expérimentations puisqu'il permet d'implémenter la descente de gradient exacte et donc de comparer les descentes de gradient exactes et stochastiques. Par ailleurs, dans le cas de données synthétiques où l'on sait exactement quelles sont les sources qui ont permis de générer ces données, on souhaite vérifier si le modèle permet de mettre en valeur ces sources ou non. Le modèle d'évolution que nous utilisons dans ce cas précis est celui des variables « en $\mathbb{P}$ ». Voici donc l'évolution

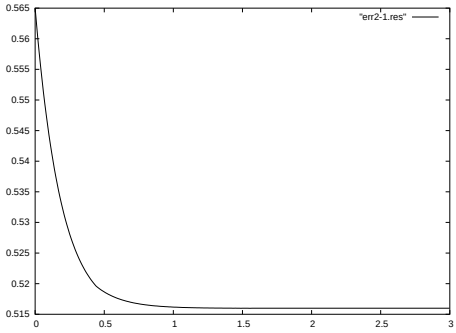
de l'erreur du 4-plus proche voisin dans le cas où l'on effectue cette descente de gradient exacte (non approchée) avec différents jeux de coefficients. Seul le rapport α/β influence donc la vitesse de convergence vers le régime stationnaire de l'équation différentielle. Tous les résultats d'erreur ont ici été calculés sur le Test-Set de la base de données.



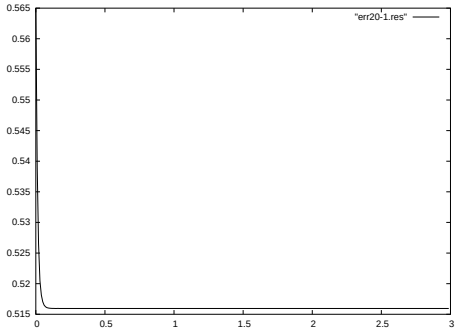
Descente Exacte avec $\alpha/\beta = 1$



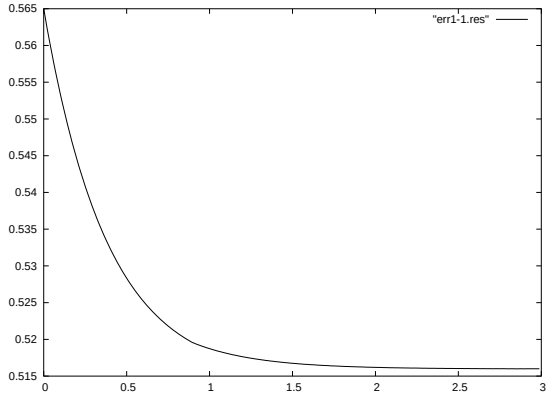
Descente Exacte avec $\alpha/\beta = 10$



Descente Exacte avec $\alpha/\beta = 2$



Descente Exacte avec $\alpha/\beta = 20$

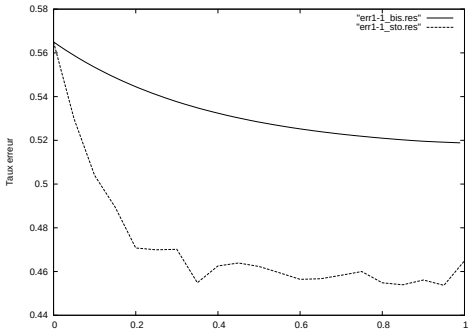


Descente Exacte avec $\alpha = 1$ et $\beta = 0$

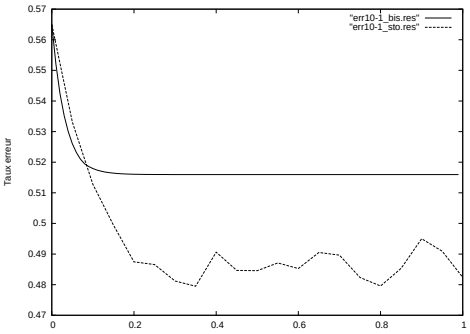
3.7.3 Descente de gradient approchée

Il n'est pas possible comme dans la descente de gradient exacte de représenter à chaque pas de temps le taux d'erreur exact correspondant, en revanche, on peut calculer en certains instants choisis à l'avance le taux d'erreur obtenu par descente de gradient stochastique. La courbe en traits pleins rappelle les résultats de la descente de gradient exacte effectuée sur

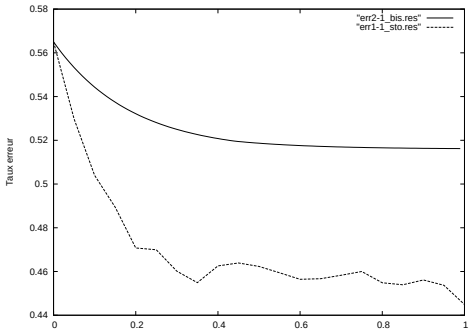
le même ordinateur, avec la même échelle de temps. Les courbes discontinues représentent de même l'évolution du taux d'erreur avec la méthode d'approximation stochastique, cette courbe est obtenue en effectuant une interpolation affine par morceaux aux instants où sont faits les relevés exacts de ces taux d'erreur. Mais l'échantillonage est donc moins précis que dans les descentes de gradient précédentes, ce qui explique le manque de régularité des courbes obtenues.



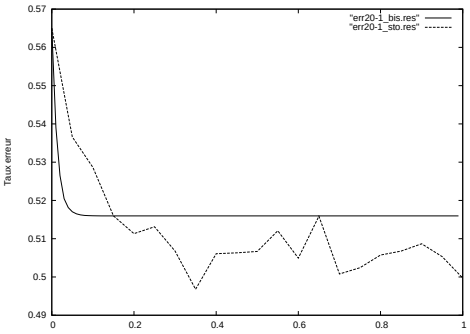
Descente Exacte/Approchée $\alpha/\beta = 1$



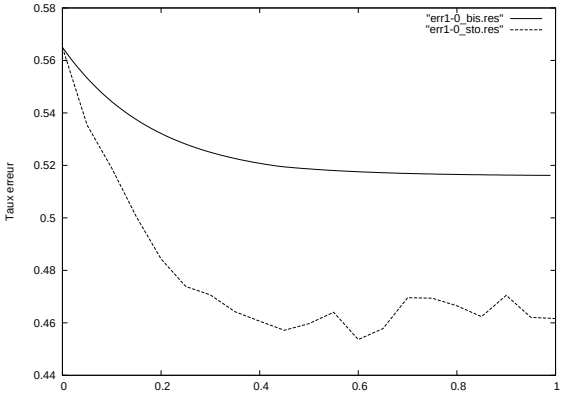
Descente Exacte/Approchée $\frac{\alpha}{\beta} = 10$



Descente Exacte/Approchée $\alpha/\beta = 2$



Descente Exacte/Approchée $\alpha/\beta = 20$



Descente Exacte/Approchée $\alpha/\beta = \infty$

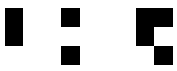
On peut d'ores et déjà faire plusieurs commentaires sur les résultats obtenus :

- La décroissance est plus rapide (en temps d'horloge processeur) lors de l'approximation stochastique que lors de la descente de gradient exacte. Ceci nous conforte dans l'idée que l'algorithme stochastique est ici un outil efficace pour caractériser l'utilité des features et les propriétés de « mélange » d'efficacité de ces features.

- L'algorithme obtient des scores meilleurs « à la limite » que la descente de gradient réelle puisque les taux d'erreurs limites obtenus sont largement inférieurs (0.51 contre 0.46).
- La variance de l'algorithme augmente (visuellement) avec le rapport α/β. Ceci était prévisible puisque le terme qui caractérise la stabilité de l'algorithme est pondéré par le réel β.

3.7.4 Features « sélectionnés »

Dans l'exemple du problème de détection précédent, on peut également mettre en valeur les features qui ont la plus forte probabilité $\mathbb{P}_\infty$ d'être tirés. Ces features sont supposés apporter le plus de valeurs informatives à la classification. Dans le problème précédent de détection, les ensembles ${\mathcal{F}_i}^k$ qui ont permis de former les classes $\mathcal{C}_1$,$\mathcal{C}_2$ et $\mathcal{C}_3$ sont précisément représentés dans le tableau suivant pour chacune des classes. Ce tableau se lit par ligne, chacune des lignes correspond à une des sources ${\mathcal{F}_i}^k$ dont est issue une partie des données.

$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$
		

On a ici :

$$ {\mathcal{F}_1}^1 = \{\delta_1;\delta_3;\delta_5;\delta_7\} \quad \text{et} \quad {\mathcal{F}_1}^1 = \{\delta_1;\delta_5\} \quad \text{et} \quad \{\mathcal{F}_1\}^3 = \{\delta_3;\delta_7\} $$

puis

$$ {\mathcal{F}_2}^1 = \{\delta_2;\delta_4;\delta_6;\delta_8\} \quad \text{et} \quad {\mathcal{F}_2}^2 = \{\delta_2;\delta_4\} \quad \text{et} \quad \{\mathcal{F}_2\}^3 = \{\delta_6;\delta_8\} $$

et enfin

$$ {\mathcal{F}_3}^1 = \{\delta_1;\delta_4;\delta_8;\delta_9\} \quad \text{et} \quad {\mathcal{F}_3}^2 = \{\delta_1;\delta_8\} \quad \text{et} \quad \{\mathcal{F}_3\}^3 = \{\delta_4;\delta_9\} $$

Il est important de calculer les entropies conditionnelles de la loi des classes sachant la connaissance des réalisations de variables. Le calcul de telles entropies permet en effet de juger la complexité statistique des données en fonction de la connaissance de certaines variables. Les features générant le bruit, c'est-à-dire les variables δ n'appartenant à aucune des sources $\mathcal{F}_i^k$ ont une entropie conditionnelle maximale, puisque :

$$ \forall j \geqslant 10 \qquad \mathrm{P}(\mathcal{C}_i|\delta_j = 1) = \frac{\mathrm{P}(\delta_j = 1|\mathcal{C}_i)\mathrm{P}(\mathcal{C}_i)}{\mathrm{P}(\delta_j = 1)} = \frac{1}{3} $$

Soit

$$ \mathrm{H}(\mathrm{P}(.|\delta_j = 1)) = \log(3) \simeq 1.099 $$

En ce qui concerne les variables appartenant à certaines sources, on constate que les entropies conditionnelles sont un peu moins importantes dans notre cas synthétique, mais sont tout de même élevées. Par ailleurs, il y a peu de différence entre les entropies conditionnelles pour les variables appartenant à une seule source, ou appartenant à l'intersection de plusieurs sources. En effet, pour les variables δ_1 et δ_3 sur les données synthétiques précédentes, on a :

$$ \mathrm{H}(\mathrm{P}(.|\delta_1 = 1)) \simeq 0.852 $$

alors que

$$ \mathrm{H}(\mathrm{P}(.|\delta_3 = 1)) \simeq 0.856 $$

Même si les variables correspondant au « bruit » peuvent facilement être déterminées en étudiant l'entropie conditionnelle des classes sur ces variables, il n'y a en revanche pas de différence

importante pour les tests δ_1 et δ_3 en ce qui concerne les entropies conditionnelles sur ces variables puisque celles-ci sont également importantes.

Par conséquent, l'examen des entropies conditionnelles permet clairement de mettre en avant les variables qui bruitent les données (entropies conditionnelles très grandes), alors que les variables qui génèrent ces données ne peuvent pas être classées par ordre d'influence de façon évidente.

Notre algorithme, en revanche, permet d'opérer cette distinction. Voici en effet un exemple d'histogramme de probabilité limite $\mathbb{P}_\infty$ pour la tâche de classification précédente avec le jeu de coefficient $\alpha/\beta = 20$ et des valeurs de $|\omega|$ valant 2 pour le premier histogramme, puis 3 et 4 pour les suivants.

$\mathcal{F}$	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8	δ_9	δ_{10}	δ_{11}	δ_{12}	δ_{13}	δ_{14}	δ_{15}	δ_{16}	δ_{17}	δ_{18}	δ_{19}	δ_{20}
L = 2																				
L = 3																				
L = 4																				

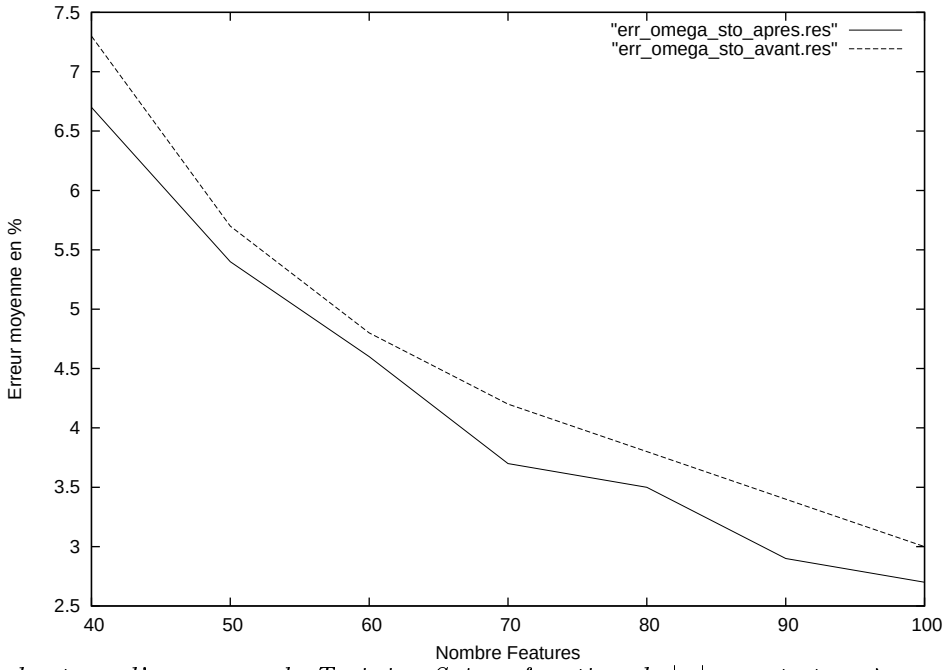
On constate ici que ce sont effectivement les tests δ qui appartiennent aux sources $\{\mathcal{F}_i\}^k$ qui sont privilégiés par la probabilité $\mathbb{P}_\infty$. De plus, les tests les plus chargés par la probabilité $\mathbb{P}_\infty$ sont de loin les tests qui permettent une collaboration d'information. Plus exactement, δ_1, δ_4 et δ_8 sont les tests qui ont la plus forte probabilité, et on peut remarquer que ce sont précisément ces tests qui appartiennent aux plus grand nombre de sources ${\mathcal{F}_i}^k$. C'est précisément le genre de résultat que nous souhaitons obtenir : parvenir à sélectionner les variables informatives qui s'adaptent le mieux aux classes en jeu dans le problème de la détection. Notre approche permet donc de mettre en valeur certaines variables qui n'auraient pas été mises en avant naturellement par l'examen des entropies conditionnelles.

Cependant, en l'état actuel des possibilités de sélection de variables de l'algorithme, il est dans l'immédiat impossible de composer exactement ces sources intégralement. Ce sera possible par la suite lorsque nous aurons formalisé un nouvel algorithme incluant des processus de sauts.

3.8 Détection de chiffres manuscrits

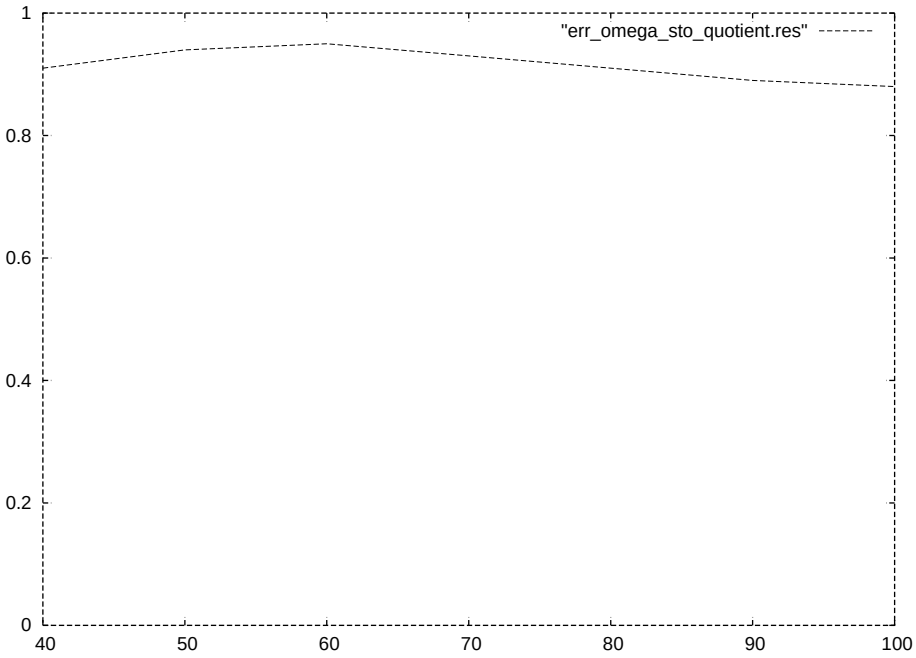
3.8.1 Taux d'erreur g

Nous présentons ici l'évolution du taux d'erreur g basé sur un algorithme de Support Vector Machine. Les features tirés sont en quantité $|\omega|$ et sont tirés selon la loi $\mathbb{P}_0 = \mathcal{U}_\mathcal{F}$ puis selon la loi $\mathbb{P}_t$ pour t grand. L'évolution du taux d'erreur lors des expériences effectuées pour cette base de données n'est malheureusement pas représentable de manière continue à cause de la grande quantité de calculs à effectuer. Mais on peut néanmoins tracer la courbe d'erreur g en fonction du nombre de coordonnées $|\omega|$ tirées, ainsi que la courbe d'erreur moyenne obtenue après apprentissage. Cette courbe a été obtenue en calculant la moyenne de g sur le Learning Set de la base de données [USP] de chiffres manuscrits.



Évolution des taux d'erreur sur le Training-Set en fonction de $|\omega|$ avant et après apprentissage

Pour mesurer l'efficacité de notre apprentissage de $\mathbb{P}_\infty$, nous pouvons également tracer la courbe représentant l'évolution du rapport $\mathcal{E}_{err}(\mathbb{P}_\infty)/\mathcal{E}_{err}(\mathcal{U}_\mathcal{F})$ en fonction du nombre de variables $|\omega|$ que nous extrayons à chaque pas de notre algorithme.

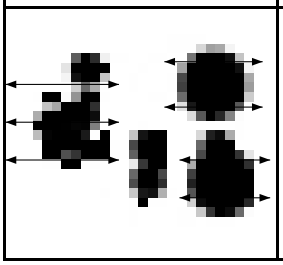
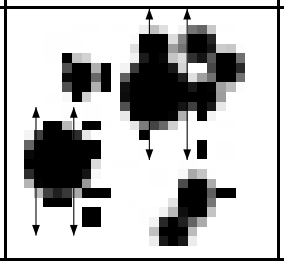
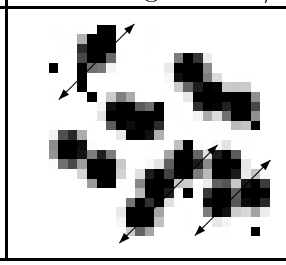
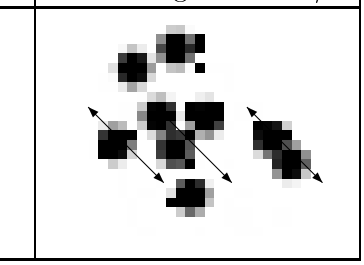


Évolution du rapport des taux d'erreur sur le Training-Set en fonction de $|\omega|$

Le gain de notre algorithme est donc à peu près stable, indépendamment du nombre de variables sélectionnées à chaque pas. Le gain réalisé est d'environ 10% pour chaque taux d'erreur sur l'ensemble d'apprentissage.

3.8.2 Organisation spatiale des tests

Il peut également être intéressant de visualiser les zones de l'image qui ont été mises en valeur par l'algorithme d'apprentissage. Ce sont les zones où la probabilité de sélectionner un feature est particulièrement importante ($\mathbb{P}_n$ grand). Ces zones dépendent en réalité un peu de l'orientation du détecteur de bord. Dans le tableau ci-dessous figurent donc en foncé ces zones « informatives » pour la détection de chiffres manuscrits, pour l'algorithme de recherche utilisé.

Bords horizontaux	Bords verticaux	Bords diagonaux $\pi/4$	Bords diagonaux $-\pi/4$
			

Il est difficile d'interpréter pour quelles classes d'objets ces différentes zones sont pertinentes pour discriminer ou favoriser la détection d'une telle classe. Cependant, l'organisation des zones « informatives » est bien cohérente avec l'intuition qu'on pourrait en avoir. Ces zones sont (relativement) concentrées dans l'image puisqu'elles n'occupent qu'un pourcentage restreint de pixels de l'image, et sont regroupées en « amas ».

3.8.3 Performance de classification

3.8.3.1 Protocole de classification

L'algorithme de Support Vector Machine est adapté aux problèmes de détections d'objets à deux classes. Afin d'obtenir un classifieur multi-classes à partir d'un algorithme de SVM , il est nécessaire d'organiser un « vote » de plusieurs détecteurs issus de problèmes à deux classes.

Si $(\mathcal{C}_i)_{i\in\{0\ldots9\}}$ désigne l'ensemble des classes, on désigne par ${D_i}^{\omega}$ un détecteur issu de l'algorithme de SVM pour le problème à deux classes :

$$\mathcal{C}_i \quad || \quad \cup_{j\neq i}\mathcal{C}_j$$

en utilisant alors uniquement un algorithme de SVM linéaire (noyau polynomial de degré 1). Le détecteur ${D_i}^{j}$ renvoie 1 lorsqu'il décide que l'image appartient à la classe $\mathcal{C}_i$ et 0 sinon.

La probabilité $\mathbb{P}_\infty$ est utilisée lors du tirage des features ω qui fournit par conséquent un grand nombre de sous-ensembles de features ${\omega_i}^1, \ldots {\omega_i}^p$ et par conséquent plusieurs détecteurs de la classe $\mathcal{C}_i$ que l'on note ${D_i}^1; \ldots {D_i}^p$.

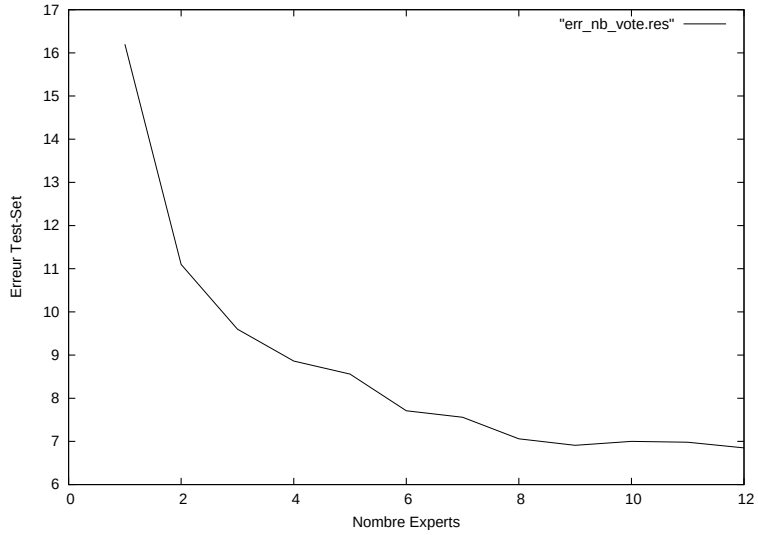
Le vote à la majorité relative est alors organisé comme suit, si I désigne une image issue du « test-set » $\mathcal{I}_{test}$, la classe choisie est :

$$\text{Arg} \max_{i\in\{0\ldots9\}} \#\left\{j \quad | \quad {D_i}^j(\text{I}) = 1\right\}$$

Ainsi, la classification est organisée en un choix de p experts (les D_i^j) pour chaque classe et un vote de ces $p \times 10$ experts pour décider de la classe des signaux étudiés.

3.8.3.2 Efficacité du vote d'experts

Commençons par illustrer l'efficacité du vote d'experts évoqués dans le paragraphe précédent. Nous avons choisi de représenter le taux d'erreur de classification *via* le protocole précédent, en sélectionnant les experts par la loi uniforme sur l'ensemble des variables $\mathcal{F}$. Voici l'évolution du taux d'erreur obtenu en fonction du nombre d'experts choisis pour chacune des classes $\mathcal{C}_i$, dans le cas où l'on tire 100 features de $\mathcal{F}$ selon la loi uniforme sur ces features.

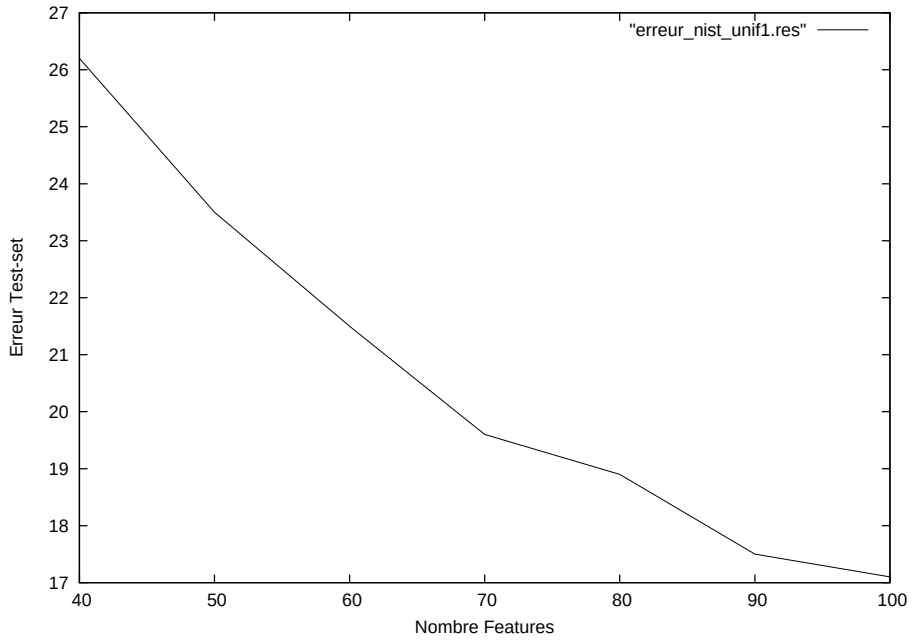


Évolution du taux d'erreur en fonction du nombre d'experts pour chaque classe sur le Test-Set

On constate immédiatement l'efficacité d'un tel vote pour améliorer les performances du taux de classification puisque le taux d'erreur est divisé par un facteur supérieur à 2 entre les cas où $p = 1$ et $p = 10$. Cependant, nous pouvons également remarquer que l'amélioration du taux d'erreur ne se fait pas « indéfiniment » en augmentant le nombre d'experts pour chaque classe. En effet, les performances de classification sont sensiblement équivalentes en utilisant 8 experts par classe ou 12. Ainsi, nous choisirons la plupart du temps d'utiliser 10 experts pour chaque classe puisque l'utilisation d'un nombre plus grand de votants pour chaque classe ne permettrait pas d'apporter de meilleures performances.

3.8.3.3 Taux de classification pour $\mathbb{P} = \mathcal{U}_{\mathcal{F}}$

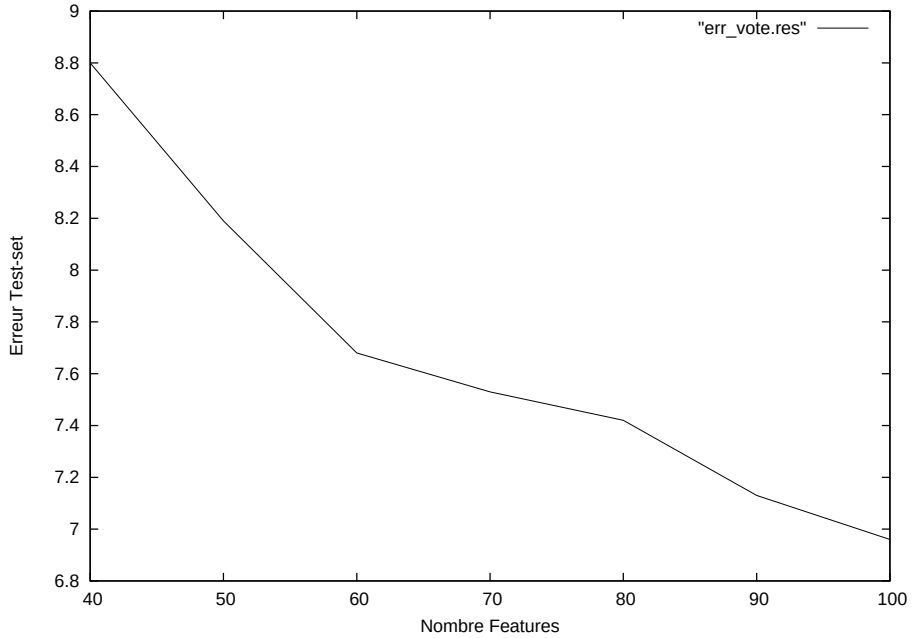
Plusieurs quantités de détecteurs $\mathrm{D}_i{}^j$ ont été retenues (p varie de 1 à 20), ces détecteurs étant issus de l'exécution de l'algorithme de SVM pour des variables tirées selon la loi uniforme sur l'ensemble des variables $\mathcal{F}$. Par ailleurs, le nombre de features utilisés pour le calcul du meilleur hyperplan séparateur du SVM a également été modifié de 40 à 100 durant nos expériences. Nous pouvons représenter la courbe de l'erreur commise dans le cas où p vaut 1 et que l'on tire au sort les tests selon la loi uniforme.



Taux d'erreur moyen pour 1 expert par classe et $\mathbb{P} = \mathcal{U}_{\mathcal{F}}$ en fonction de $|\omega|$ sur le Test-Set

Les performances obtenues dans le cas où on utilise seulement 10 hyperplans basés sur 100 features sont relativement mauvaises (17% d'erreur). Ceci s'explique par la faible quantité d'information dont on dispose dans ce cas puisqu'il n'y a pas de vote à la majorité, et également par le fait que les features sont tirés au hasard, sans *a priori* sur l'utilité ou non des features tirés.

Enfin, lorsque nous utilisons un vote de 10 experts pour chacune des 10 classes, voici les taux d'erreurs de classification que nous obtenons la courbe suivante.

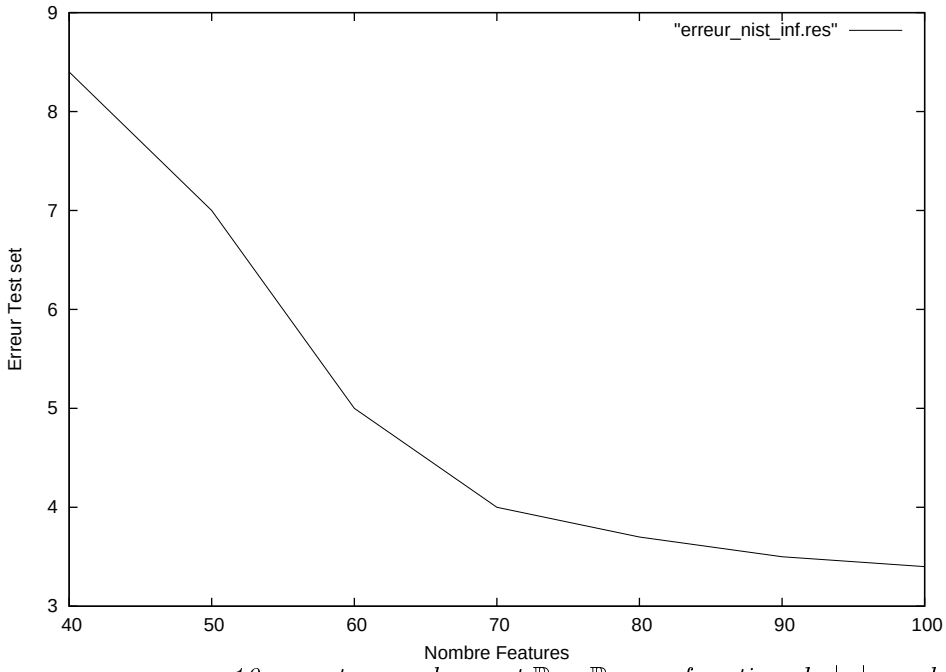


Taux d'erreur moyen pour 10 experts par classe et $\mathbb{P} = \mathcal{U}_{\mathcal{F}}$ en fonction de $|\omega|$ sur le Test-Set

Les taux d'erreurs obtenus sont, nous l'avons vu dans le paragraphe consacré à l'efficacité du vote d'experts, nettement améliorés par rapport au cas où $p = 1$. Ces résultats sont bien loin d'égaler les résultats de [SC98], mais nous verrons dans le paragraphe suivant que ces taux peuvent être nettement améliorés lorsque la sélection des variables est effectuée *via* la loi de probabilité $\mathbb{P}_\infty$ obtenue par notre algorithme d'apprentissage.

3.8.3.4 Taux de classification pour $\mathbb{P}_\infty$

Voici en revanche la courbe obtenue lorsque l'on choisit d'effectuer un vote basé sur 10 hyperplans par classe, après l'apprentissage de $\mathbb{P}_\infty$ et tirage des features selon cette loi de probabilité.



Taux d'erreur moyen pour 10 experts par classe et $\mathbb{P} = \mathbb{P}_\infty$ en fonction de $|\omega|$ sur le Test-Set

Les résultats sont nettement plus probants. Ils ne permettent cependant pas de rivaliser avec les taux d'erreurs obtenus par Yann LeCun ([SC98], [LBBH98]) sur cette base de données (2.7% de taux d'erreur sur la base [USP] de chiffres manuscrits de tailles 16×16). Cependant, l'algorithme est tout de même performant puisqu'on constate une nette amélioration des taux d'erreurs : ils passent (pour le même feature space) de plus de 25% d'erreur au taux honorable de 8.4% de chiffres classés de façon incorrecte lorsque l'on choisit de tirer 40 features. Cet algorithme donne donc des résultats nettement meilleur que celui de séparation linéaire sur les données(qui obtient une performance de 12% d'erreur) pour une complexité (lors de la phase de détection) qui lui est comparable ([LBBH98], paragraphe 3.3.1).

Enfin, le taux d'erreur obtenu en fin d'apprentissage pour un tirage de 100 features selon $\mathbb{P}_\infty$ en faisant collaborer 10 détecteurs par classe *via* un vote permet d'atteindre le taux moyen de 3.3% d'erreur. Ce calcul est réalisé en effectuant une moyenne sur les différents taux d'erreur obtenus par vote selon la probabilité $\mathbb{P}_\infty$. Afin d'évaluer les résultats de notre algorithme, voici les résultats mentionnés dans [Vap00] (chapitre 12) d'autres algorithmes existants sur cette base de données [USP].

$\text{SVM}_{Lin16\times16}$	8.9%
$\text{SVM}_{d^\circ=2,16\times16}$	4.7%
$\text{SVM}_{d^\circ=3,16\times16}$	4.0%
Arbre de décision	16.2%
Performance humaine	2.5%
Tangent Distance	2.7%
SVM & Feature Selection	3.3%

On constate donc ici que l'algorithme utilisé, bien qu'utilisant un feature space de données binaires, permet d'atteindre de bons résultats, d'autant plus que la complexité pour coder les données n'utilise donc en moyenne que $100 \times 10 \times 10$ bits, alors que l'utilisation par exemple d'un SVM sur les niveaux de gris utilise $10 \times 256 \times 16 \times 16/2$ bits soit environ 30 fois plus.

Néanmoins, il est difficile de comparer l'apport de notre méthode de proposition des variables par rapport à ces autres algorithmes puisque nous effectuons un vote de différents détecteurs pour mesurer la performance finale de détection alors que les performances obtenues dans les différents autres algorithmes ne font pas intervenir ce genre de vote.

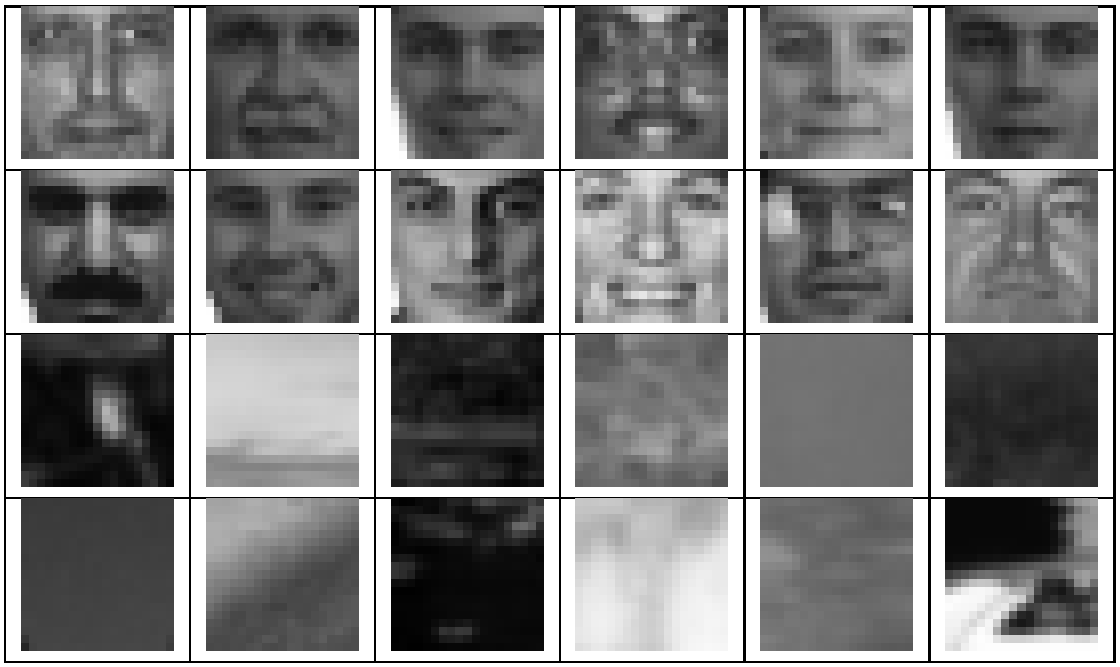
Pour effectuer une telle comparaison, nous pouvons examiner le taux moyen d'erreur utilisant un vote avant, et après l'apprentissage de notre probabilité d'extraction $\mathbb{P}_\infty$. Ces taux d'erreur passent, pour le même protocole de classification, de 7% d'erreur en moyenne à 3.3% d'erreur. Nous constatons donc que le gain apporté par cet apprentissage est réel lorsque la sélection des variables se fait par $\mathbb{P}_\infty$.

3.9 Détection de visages

3.9.1 Base de données

Le premier exemple d'application en traitement de l'image que l'on peut donner de notre algorithme de sélection de features est celui de la recherche d'indices importants dans des images contenant des visages. Nous étudions par ailleurs l'évolution du taux d'erreur de l'algorithme d'apprentissage sur la base de données de visages [MIT]. La base de données est constituée d'un Learning Set de 2429 images contenant des visages et 4548 images ne contenant pas de visages. Ces images sont disponibles au format .PGM en niveau de gris de taille 19×19.

Voici quelques échantillons des images issues de la base de données.

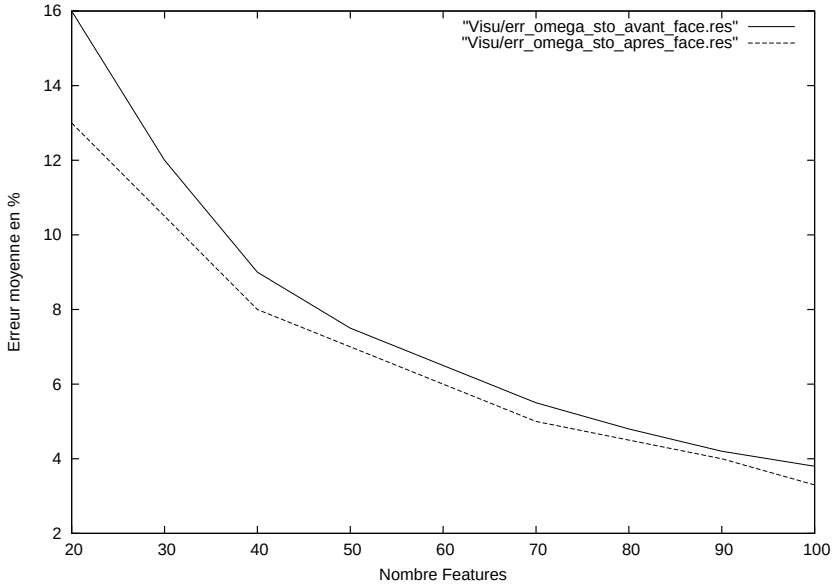


Comme nous l'avons mentionné au paragraphe 2.7.3.1, il n'est pas nécessaire de définir des features invariants par translation pour ces données, puisque les images à détecter (les visages) sont clairement centrées sur la grille des pixels.

3.9.2 Évolution de $\mathcal{E}$

On commence par extraire les features issus des détecteurs de bords que l'on garde, comme ce qui est mentionné dans l'annexe A. A l'issu de cette présélection des features, nous obtenons un ensemble de features comprenant environ 2000 éléments. Nous effectuons alors une descente de gradient approchée comme ce qui a été décrit dans le chapitre 3, en variables exponentielles (équation $(\mathbf{E_4})$) afin de ne pas avoir de contraintes sur α et β (3.13).

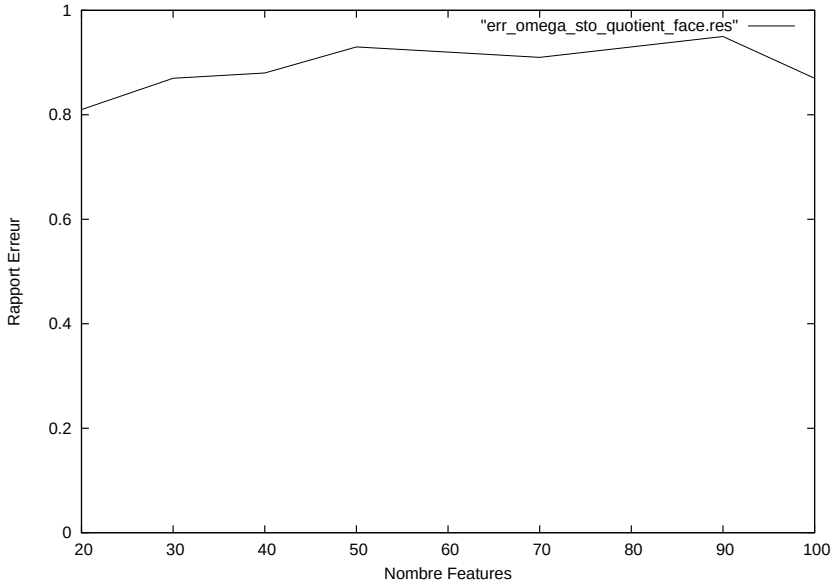
Les deux courbes suivantes représentent la variation de l'erreur moyenne g sur le test-set des images [MIT] en fonction du nombre de features tirés $|w|$ avant, et après l'apprentissage de $\mathbb{P}_\infty$.



Évolution du Taux d'erreur moyen avant et après apprentissage pour 10 *experts sur le Test-Set*

La courbe en traits pleins désigne la variation de l'erreur moyenne sur le Test-Set avec une distribution uniforme sur l'ensemble des features tirés, alors que la courbe en pointillés représente la courbe d'erreur lorsque les features sont tirés avec la probabilité apprise par l'algorithme d'apprentissage.

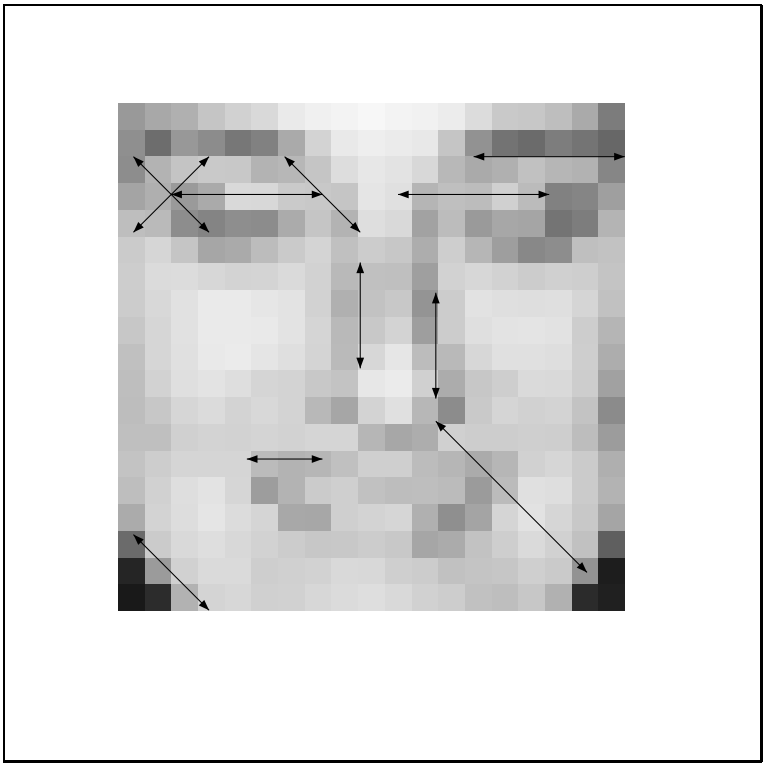
Ainsi, lorsque l'on tire 20 features, l'erreur moyenne $\mathbb{E}g$ diminue d'environ 1.5% après l'apprentissage. De même, si l'on tire 100 features, l'erreur diminue également (mais dans une moindre mesure) de 0.5%, atteignant alors le taux relativement bon de 3.5% de taux d'échec de détection de visages. On peut à nouveau représenter l'évolution du rapport entre les deux taux d'erreur obtenus en fonction de $|\omega|$.



Évolution du Taux d'erreur moyen avant et après apprentissage pour 10 *experts sur le Test-Set*

3.9.3 Localisation des features

On peut par ailleurs représenter les zones où sont localisés les tests les plus « intéressants » pour la tâche de détection des visages qui ont été sélectionnés par l'apprentissage de $\mathbb{P}_\infty$:



Les résultats sont relativement satisfaisants puisque les tests les plus probables pour $\mathbb{P}_\infty$ sont des détecteurs de bords qui sont bien localisés au niveau des yeux, du nez ainsi qu'au niveau de la bouche et des contours du visage.

3.9.4 Taux d'erreur

Afin d'obtenir le meilleur taux de classification possible pour la détection de visage, nous organisons une collaboration de détecteurs, votant à la majorité sur le test-set, constitué de 25% des données inutilisées lors de l'apprentissage. L'efficacité d'un tel « vote d'experts » a déjà été soulignée dans le paragraphe sur la classification de chiffres manuscrits [USP].

Nous obtenons un taux correct de classification lorsqu'on extrait 20 features en suivant $\mathbb{P}_\infty$ du feature space et qu'on organise un vote de détecteur puisque le taux d'erreur avant l'apprentissage qui était de 13.6% passe au taux honorable de 7.5%.

Le meilleur taux d'erreur ainsi obtenu (où l'apport de notre méthode de sélection de features est visible) lors des extractions de features permet de faire évoluer le taux d'erreur de 3.5% d'erreur à 2.3% d'erreur lorsque l'on décide de tire 100 features selon $\mathbb{P}_\infty$ et que l'algorithme de détection général utilisé $\mathbb{A}$ est l'algorithme de Support Vector Machine utilisant un noyau uniquement linéaire. L'augmentation des performances, par la sélection des variables ainsi décrites peut se résumer dans le tableau suivant :

Algorithme $\mathbb{A}$	Extraction par $\mathcal{U}$	Extraction par $\mathbb{P}_\infty$
SVM_{Lin}, $\lvert\omega\rvert = 20$	13.6%	7.5%
SVM_{Lin}, $\lvert\omega\rvert = 100$	3.5%	2.3%

3.10 Détection de SPAM

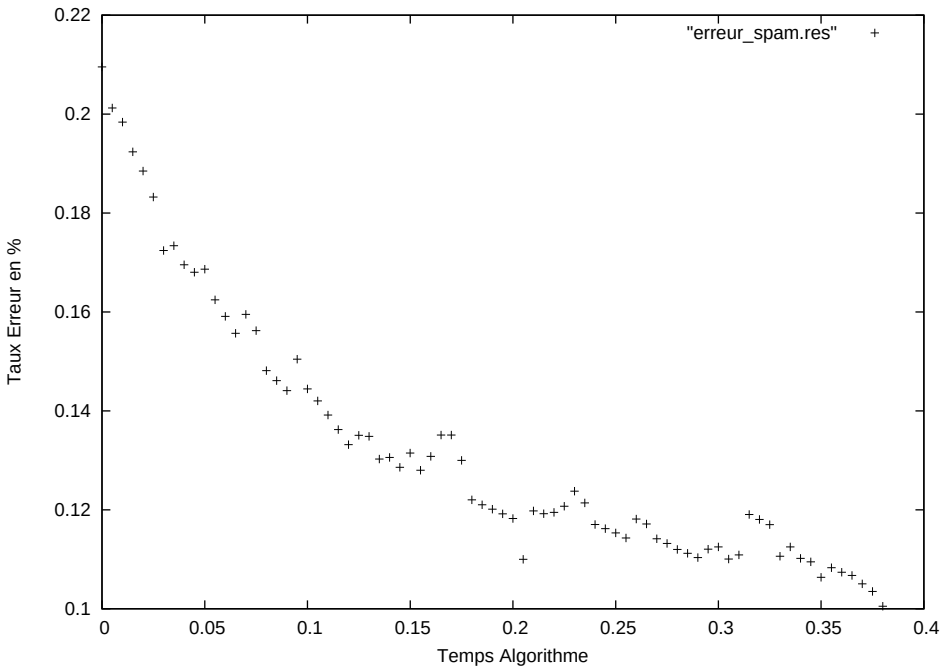
3.10.1 Évolution du taux d'erreur de classification

Voici l'évolution du taux d'erreur lors de notre approximation stochastique effectuée sur la base d'un 4-plus proche voisin en extrayant 20 coordonnées ($|\omega| = 20$) sur les 54 features possibles caractérisant les données. On constate là aussi une décroissance non négligeable de l'erreur de classification.

La liste de features originale a été légèrement modifiée : il y avait initialement 57 features qui étaient le pourcentage d'occurrences de 54 mots précis dans des emails ainsi que 3 entiers mesurant

- la longueur moyenne des mots écrits en lettres majuscules
- la longueur maximale d'un mot écrit en majuscule
- le nombre de mots écrits en majuscule

Nous avons choisi de ne garder que les 54 variables quantifiant les pourcentages d'occurrence de mots comme feature space. L'apprentissage a été effectué sur un learning-set comprenant 1813 emails de SPAM et 2788 emails « normaux ». La base de données [UCI] ne comportant pas de distinction entre un Learning-Set et un Test-Set, il a été procédé à un algorithme de Cross-Validation (pour des détails quand à l'efficacité d'un tel algorithme, on pourra se référer à [ET95]) pour l'estimation du taux d'erreur g.



3.10.2 Mots sélectionnés pour la détection de SPAM

Cette expérience nous permet, comme dans les paragraphes précédents, de mettre en valeur les features de $\mathcal{F}$ qui sont utiles à la tâche de classification, et ceux qui le sont moins. Les mots originaux auxquels nous nous intéressons sont répertoriés dans le tableau suivant :

make	address	all	3d	our
over	remove	internet	order	mail
receive	will	people	report	addresses
free	business	email	you	credit
your	font	000	money	hp
hpl	george	650	lab	labs
telnet	857	data	415	85
technology	1999	parts	pm	direct
cs	meeting	original	project	re
edu	table	conference	:	(
[	!	$	#	

Voici dans le tableau suivant les mots qui sont mis en valeurs par l'algorithme. Les mots de la colonne de gauche sont les mots qui sont plutôt présents lorsque le mail est du SPAM tandis que les mots de la colonne de droite sont les mots qui sont utilisés pour détecter les emails qui ne sont pas du SPAM.

Mots - SPAM	Fréquence	Mots -NON SPAM	Fréquence
report	8.8%	cs	5.4%
business	8.7%	857	4.6%
[	6%	415	4.4%
remove	5.9%	project	4.3%
receive	5.6%	table	4.2%
internet	4.4%	conference	4.2%
free	4.1%	lab	3.9%
people	3.7%	labs	3.2%
000	3.6%	edu	2.8%
direct	2.3%	650	2.7%
!	1.2%	85	2.5%
$	1%	george	1.6%

Les mots qui sont mis en valeur sont ici cohérents avec ceux qui ont été détectés dans [HTF01]. Les autres mots non-mentionnés se partagent alors les poucentages restants. On notera que le filtre anti-spam que l'on pourrait fabriquer à partir de ces données ne peut être qu'un filtre anti-spam personnel : la présence de chiffres comme « 857 »,« 415 », « 650 », « 85 » ou de mots comme « george » provient du fait que les emails constituants la base de données sont issus de la boite mail d'un ingénieur (George) de chez Hp-labs dont le numéro de téléphone professionnel est composé de certains chiffres précédents. Par ailleurs, les mots qui caractérisent le SPAM correspondent effectivement à ce qu'on pouvait attendre intuitivement.

On pourrait donc imaginer, à partir d'un tel algorithme, constituer une méthode de construction de filtres personnels anti-spam peu couteuse en stockage et en temps de calcul.

3.10.3 Vote de détecteurs

Pour obtenir le meilleur résultat de classification sur la base de données d'emails, nous avons fait collaborer des détecteurs basés sur l'algorithme des 4-NN, ces détecteurs ont alors été tirés là encore selon la loi apprise $\mathbb{P}_\infty$.

Les meilleurs résultats répertoriés par Hewlett-Packard sont d'approximativement 7% d'erreur de classification de SPAM tandis que lorsqu'on exige un taux de faux positif nul, il y a entre 20% et 25% de messages de SPAM qui ne sont pas filtrés. Le taux d'erreur que nous obtenons en organisant un vote de détecteurs issus de notre apprentissage est d'approximativement 7.5%. Nous pouvons mettre en avant la faible complexité algorithmique nécessaire à l'obtention d'un tel score par notre méthode, puisque peu de mots (10 ou 15) sont nécessaires à une telle performance.

3.11 Bilan

Nous avons donc pu constater que dans les différents exemples de tâches de reconnaissance de formes étudiés, notre algorithme d'apprentissage de $\mathbb{P}_\infty$ permet d'obtenir un bon taux de détection en préservant lors de la phase de détection une rapidité de traitement performante.

En effet, lors de la détection de SPAM par notre algorithme, le traitement d'un message dure environ 1 ms. En ce qui concerne la détection de chiffres manuscrits ou de visages, le test de séparation par hyperplan en utilisant 100 coordonnées dure 0.3 ms et l'exécution d'un vote de 10 détecteurs par classe permet un traitement de chaque image en 3 ms.

Nous allons dans la suite de ce mémoire travailler sur l'extraction et l'ajoût de features en formalisant très précisément un processus stochastique de sélection de variables. Cette méthode d'extraction s'inspirera de l'algorithme de descente de gradient étudié dans ce chapitre, en permettant alors la suppression de certaines variables « mal notées » par $\mathbb{P}$ et la formation de nouvelles à partir de variables « bien notées ».

Chapitre 4 - Processus de diffusion réfléchie

4.1 Introduction

Dans ce chapitre, nous allons construire dans un premier temps un processus de diffusion $(\mathbb{P}_t)_{t\in\mathbb{R}}$ réfléchi appartenant à tout instant à G, ensemble compact à bords $\mathcal{C}^\infty$ par morceaux de $\mathbb{R}^n$. Ce processus sera parfaitement continu en temps mais sa définition ainsi que son existence et son unicité ne sont pas évidentes *a priori*. Nous utiliserons pour cela la résolution d'une première équation différentielle stochastique ([Gad04]).

Dans un second temps, nous donnerons une application de l'existence et l'unicité d'un tel processus réfléchi pour l'étude de la descente de gradient de $\mathcal{E}_1$ (*cf* chapitre 3 paragraphe 3.2) sous les conditions d'appartenance à $\mathcal{S}_\mathcal{F}$. Ce processus permettra en définitive d'explorer l'ensemble des probabilités sur un ensemble fixé de variables en contournant les problèmes d'existence soulevés dans le paragraphe 3.3 pour la solution de l'équation $(\mathbf{E}-\mathbf{3})$.

4.2 Diffusion sous contraintes

Nous établissons donc dans un premier temps l'existence et l'unicité du processus de diffusion non dégénérée sur un G. Ce type de processus a été étudié longuement dans le cadre des réseaux de file d'attente ([HW92], [HR81] , [Rei84] [ABD01]) pour des applications en réseaux de télécommunications par exemple, avec des contraintes de type « cônique » ou « simplexe », mais aussi avec des contraintes plus diverses de type parabolique [FS03]. Ces processus sont construits la plupart du temps via l'application de Skorokhod Γ.

4.2.1 Application de Skorokhod

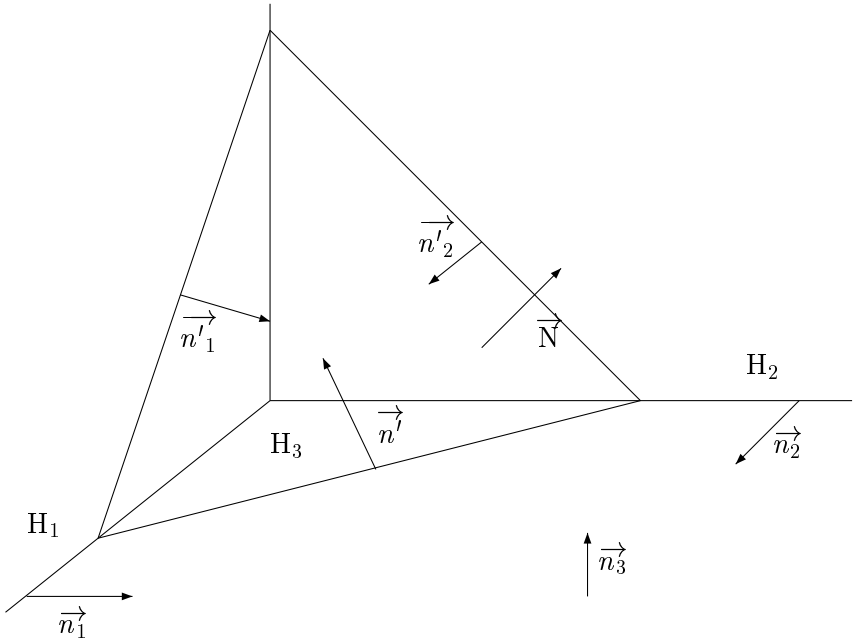
Dans ce paragraphe, nous allons définir une application Γ, lipschitzienne, qui permettra de construire précisément les diffusions sur G. Commençons tout d'abord par définir le problème de Skorokhod.

4.2.1.1 Problème de Skorokhod (SP) :

Le problème de Skorokhod fournit un outil très efficace pour la construction de processus sous contraintes. Initialement proposé pour la construction d'une solution d'équation différentielle stochastique sur $\mathbb{R}^+$ [Sko61] avec une condition de réflexion sur l'extrémité $x=0$, la construction peut se généraliser à un polyhèdre convexe G quelconque de $\mathbb{R}^n$ ([DR99]).

Nous supposerons désormais que G est une intersection finie de demi-espaces H_i^+ de $\mathbb{R}^n$, ce cas englobant bien entendu le cas particulier qui nous intéressera où $\mathrm{G}=\mathcal{S}_\mathcal{F}$.

L'énoncé du problème exact associé à G est le suivant. Posons $\mathcal{D}$ l'ensemble des applications de $[\,0\,;+\infty\,[$ dans $\mathbb{R}^n$ continues à droite ayant une limite à gauche (*càdlag*), et donnons-nous η une trajectoire de $\mathcal{D}$, $|\eta|(\mathrm{T})$ désigne la variation totale de η sur l'intervalle $[\,0\,;\mathrm{T}\,]$. On suppose de plus données des directions de réflexion associées aux faces de G. On peut donc représenter les données par le schéma suivant (le schéma illustre précisément le cas du simplexe $\mathrm{G} = \mathcal{S}_{\mathcal{F}}$ qui nous occupera par la suite) :



G est ici l'intersection de plusieurs demi-espaces H_i^+ avec l'hyperplan normal au vecteur unitaire $\overrightarrow{\mathrm{N}}$ noté $\mathcal{H}$ qui sont définis dans la définition suivante.

Définition 4.2.1 (Hyperplan $\mathcal{H}$ et H_i, vecteurs $\overrightarrow{\mathrm{N}}$ et $\overrightarrow{n_i}$)
Si f désigne le nombre d'éléments de $\mathcal{F}$ ($f = |\mathcal{F}|$), l'hyperplan $\mathcal{H}$ a pour équation

$$x_1 + \cdots + x_f = 1 \tag{4.23}$$

Le vecteur $\overrightarrow{\mathrm{N}}$ est alors donné par

$$\overrightarrow{\mathrm{N}} = \frac{1}{\sqrt{f}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \tag{4.24}$$

Enfin, nous définissons les vecteurs unitaires $\overrightarrow{n_i}$ par

$$\forall k \in \{1, \ldots |\mathcal{F}|\} \qquad \overrightarrow{n_i}(k) = \delta_{i,k}$$

où $\delta_{i,k}$ est le symbole de Kronecker. L'hyperplan H_i est l'hyperplan d'équation

$$x_i = 0$$

et $\overrightarrow{n_i}$ est alors normal à H_i.

On peut alors définir les vecteurs $\overrightarrow{n'_i}$ qui sont les projetés normés des vecteurs $\overrightarrow{n_i}$ sur l'hyperplan $\mathcal{H}$.

$$\forall i \qquad \overrightarrow{n'_i} = \frac{\overrightarrow{n_i} - (\overrightarrow{n_i} \mid \overrightarrow{\mathrm{N}})\overrightarrow{\mathrm{N}}}{\|\overrightarrow{n_i} - (\overrightarrow{n_i} \mid \overrightarrow{\mathrm{N}})\overrightarrow{\mathrm{N}}\|_2} \tag{4.25}$$

Ces vecteurs $\overrightarrow{n'_i}$ nous permettent alors de définir naturellement les directions de contraintes associées à l'ensemble G, décisions formalisées par la définition 4.2.2.

Définition 4.2.2 (Directions de réflexion)
Soit $x \in \partial\mathrm{G}$, on pose

$$\mathrm{I}(x) = \{i \quad | \quad x \in \mathrm{H}_i\}$$

Les directions de réflexion en x sont alors données par

$$d(x) = \left\{ \gamma = \sum_{i \in \mathrm{I}(x)} \alpha_i \overrightarrow{n'_i} \quad | \quad \alpha_i \geqslant 0 \right\}$$

Dans le cas général, on suppose données les directions de réflexion pour chaque point appartenant à la frontière $\partial\mathrm{G}$. Étant données de telles directions, on peut alors définir le problème de Skorokhod par ce qui suit.

Définition 4.2.3 (Problème de Skorokhod (SP))
Soit $\psi \in \mathcal{D}$ tel que $\psi(0) \in \mathrm{G}$ est donné ainsi que les directions d de réflexion. Le couple (ϕ, η) résout SP pour ψ si $\phi(0) = \psi(0)$ et pour tout $t \geqslant 0$

1. *ϕ et η sont deux trajectoires de $\mathcal{D}$*
2. $\forall t \in [\,0\,;+\infty\,[\qquad \phi(t) = \psi(t) + \eta(t)$
3. $\forall t \in [\,0\,;+\infty\,[\qquad \phi(t) \in \mathrm{G}$
4. $\forall \mathrm{T} \in [\,0\,;+\infty\,] \qquad |\eta|(\mathrm{T}) < +\infty$
5. $|\eta|(t) = \displaystyle\int_0^t \chi_{\phi(s) \in \partial\mathrm{G}} d|\eta|(s)$
6. *Il existe une trajectoire γ de $[\,0\,;+\infty\,[$ dans $\mathbb{R}^n$ telle que*

$$\forall s \geqslant 0 \qquad \gamma(s) \in d(\phi(s)) \qquad d|\eta| \text{ p.s.}$$

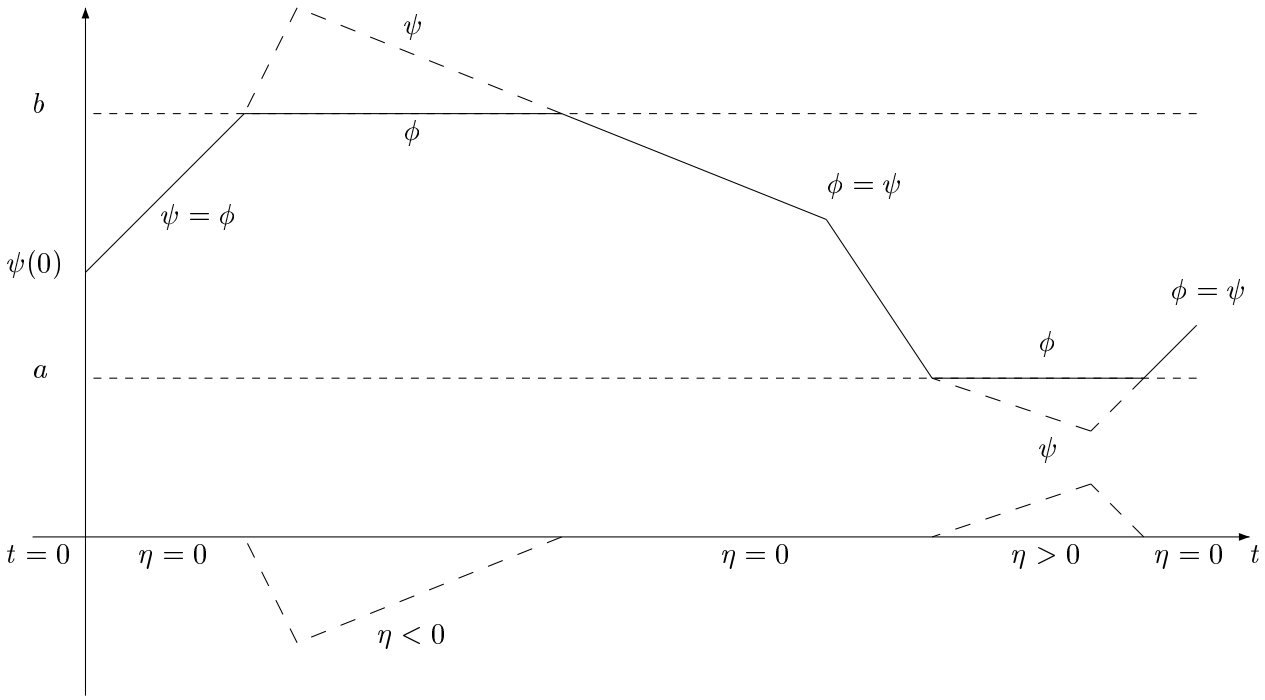
et

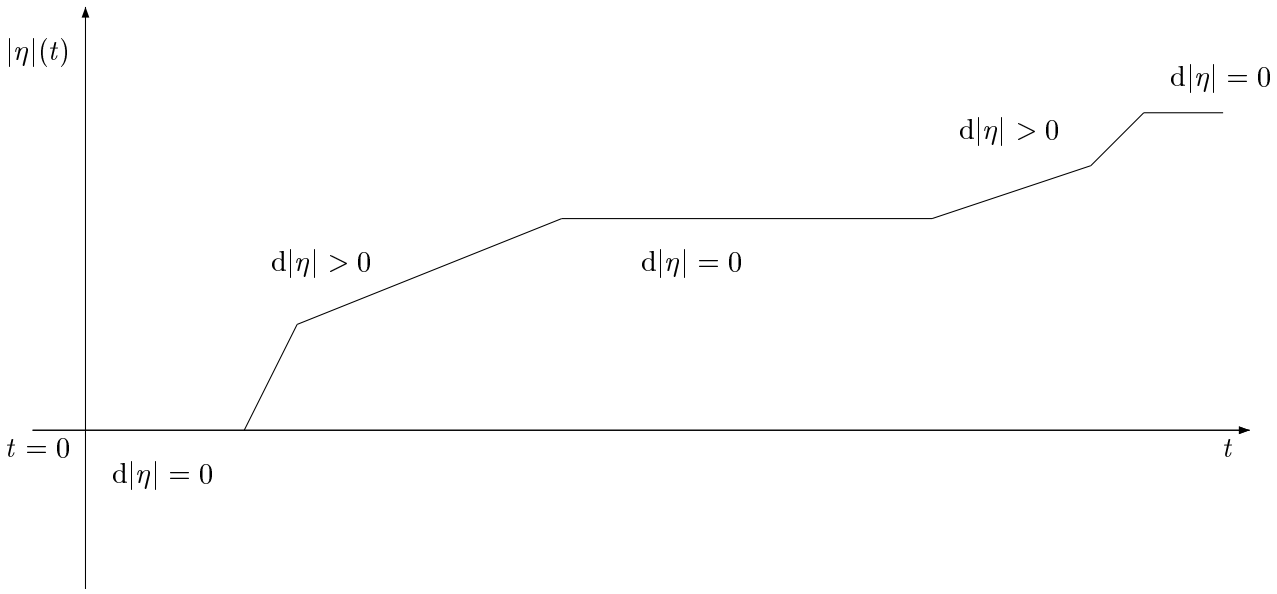
$$\forall t \geqslant 0 \qquad \eta(t) = \int_0^t \gamma(s) d|\eta|(s)$$

- ϕ ne quitte donc jamais le polyhèdre G et les valeurs de η ne sont modifiées que lorsque ϕ atteint la frontière $\partial\mathrm{G}$, puisque si $\phi(s) \notin \partial\mathrm{G}$, $d(\phi(s)) = 0_{\mathbb{R}^n}$.
- On a par ailleurs dans ce cas précis que η varie dans des directions appartenant à $d(\phi)$.

4.2.1.2 Illustration d'une solution du problème de Skorokhod en dimension 1

Nous pouvons représenter par un schéma (simple) à une dimension l'évolution de ϕ, ψ et η dans le cas où ϕ est contrainte d'évoluer dans un segment $[\,a\,;b\,]$ et ψ est affine par morceaux.



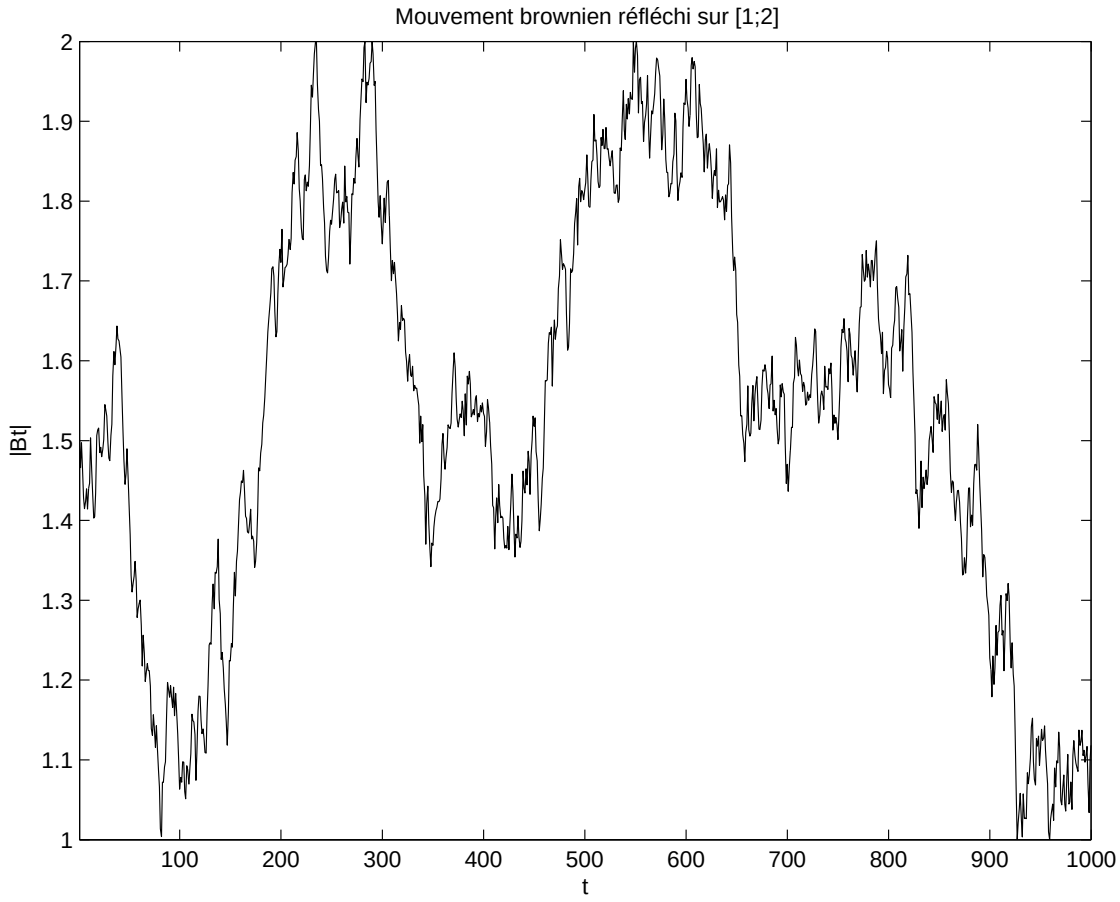


Dans l'évolution précédente, η est constante tant que ϕ n'appartient pas à $\{a; b\}$. Enfin, nous pouvons donner la formule reliant ϕ à ψ :

$$\phi(t) = \max\{a; \psi(t)\} + \min\{0; b - \psi(t)\}$$

et η s'obtient alors par soustraction.

Par ailleurs, pour l'étude des diffusions réfléchies, cet exemple n'est pas très représentatif de notre processus de diffusion. En réalité, dans le cas où l'on étudie une trajectoire ϕ qui est un mouvement Brownien réfléchi sur $[\,1\,;2\,]$, la trajectoire obtenue est :



L'accroissement $\mathrm{d}|\eta|$ n'a pas de densité par rapport à la mesure de Lebesgue et le processus de diffusion réfléchi ne reste pas sur la frontière du domaine (ici $\{1;2\}$). De même, la solution du problème de Skorokhod sur $\mathbb{R}^+$ est exactement $|\mathrm{B}_t|$ où B_t désigne le mouvement Brownien standard sur $\mathbb{R}$.

4.2.1.3 Application de Skorokhod

Afin de définir précisément les solutions des SP que nous manipulerons, nous utiliserons l'application Γ donnée par la définition 4.2.4 suivante.

Définition 4.2.4 (Application Γ)
Soit D *l'ensemble des trajectoires de* $\mathcal{D}$ *pour lesquelles il y a une unique solution à SP, et* ψ *élément de* D*, on définit l'application* Γ *par*

$$\Gamma(\psi) = \phi$$

où ϕ *est l'unique solution de SP.*

On se reportera à [DI91] pour des détails sur cette application Γ et les conditions de son existence en fonction de l'ensemble G. Chacune des trois conditions suivantes assure l'existence d'une telle application Γ

1. G est l'intersection de demi-espaces
$$\mathrm{G} = \bigcap \mathrm{E}_i$$
avec $\overrightarrow{n_i}$ vecteurs normaux unitaires des hyperplans H_i et
$$\mathrm{E}_i = \left\{ x \quad | \quad (\overrightarrow{n_i} \mid x) \geqslant 0 \right\}$$
Cet exemple traite du cas où G est un volume de $\mathbb{R}^n$ et pas exactement le cas d'une surface. Cependant, nous pouvons obtenir par exemple le cas des hyperplans en choisissant deux demi-espaces E_i avec des vecteurs $\overrightarrow{n_i}$ opposés.
2. Il existe des constantes a_i positives telles que
$$a_i(\overrightarrow{n_i} \mid \overrightarrow{n_i'}) > \sum_{j \neq i} a_j |(\overrightarrow{n_i} \mid \overrightarrow{n_j'})|$$
3. Il existe une projection Π de $\mathbb{R}^n$ dans G telle que

(a) $$\forall y \in \mathrm{G} \qquad \Pi(y) = y$$

(b) $$\forall y \notin \mathrm{G} \qquad \Pi(y) \in \partial \mathrm{G}$$

$$\forall y \notin \mathrm{G} \qquad y - \Pi(y) = \alpha\gamma \qquad \text{avec} \quad \alpha \leqslant 0 \quad \text{et} \quad \gamma \in d(\Pi(y))$$

On a alors le théorème 4.2.1 qui assure l'existence et l'unicité de solution à tout problème SP lorsque certaines conditions relativement générales sont vérifiées.

Théorème 4.2.1 (Existence et Unicité de Γ (Dupuis-Ishii))
Si l'on suppose que le polyhèdre G *est défini comme précédemment et satisfait les conditions 1, 2 ou 3, alors* Γ *est définie de façon unique sur* $\mathcal{D}$*, continue et Lipschitzienne. C'est-à-dire*

$$\exists \mathrm{K_G} < +\infty \quad \forall (\psi_1, \psi_2) \in \mathcal{D}^2 \qquad \operatorname*{Sup}_{0 \leqslant t < +\infty} |\Gamma(\phi_1)(t) - \Gamma(\phi_2)(t)| \leqslant \mathrm{K}_g \operatorname*{Sup}_{0 \leqslant t < +\infty} |\phi_1(t) - \phi_2(t)|$$

Ce théorème est fondamental pour la suite de notre mémoire. En effet, ce théorème assure l'existence d'une constante $\mathrm{K_G}$ décrivant la régularité de l'application de Skorokhod, cette régularité sera alors exploitée pour démontrer l'existence et l'unicité de certaines solutions d'équations différentielles stochastiques dans le paragraphe suivant, ainsi que dans le chapitre 5 . Ces résultats sur les équations différentielles seront établis en utilisant une méthode classique de points fixes, exploitant alors la régularité de Γ (*cf* Annexe D).

4.2.2 Existence de processus de diffusions sous contraintes dans G

Soit $(\Omega, \mathcal{T}, \mathrm{P})$ un espace probabilisé muni d'une filtration croissante $\mathcal{T}_t$, soit $(\mathrm{W}(t), \mathcal{T}_t)$ un mouvement brownien standard sur $\mathbb{R}^n$, on cherche à construire le processus X solution de l'équation

$$\mathrm{X}^x(t) = \Gamma\left(x + \int_0^{\cdot} \sigma(\mathrm{X}^x(s)) \mathrm{dW}(s) + \int_0^{\cdot} b(\mathrm{X}^x(s)) \mathrm{d}s \right)$$

où $\sigma : \mathrm{G} \mapsto \mathbb{R}^n$ et $b : \mathrm{G} \mapsto \mathrm{R}^n$ satisfont des conditions de type Lipschitz suivantes.
Il existe $\gamma \geqslant 0$ tel que :

- $$\forall (x,y) \in \mathrm{G}^2 \qquad |\sigma(x)-\sigma(y)| + |b(x)-b(y)| \leqslant \gamma |x-y| \qquad (\mathbf{C_1})$$
- $$\forall x \in \mathrm{G} \qquad |b(x)| \leqslant \gamma(1+|x|) \qquad (\mathbf{C_2})$$
- $$\forall x \in \mathrm{G} \qquad |\sigma(x)| \leqslant \gamma \qquad (\mathbf{C_3})$$

Il est tout d'abord nécessaire de s'assurer de l'existence de l'application Γ. On peut se référer au paragraphe 4.3.2 de [DR99] (cas A) pour s'assurer qu'une telle application existe.

Par la suite, on démontre alors [AO76, DI91] (en utilisant une méthode classique de point fixe de Picard) le théorème d'existence :

Théorème 4.2.2 (Existence de diffusions contraintes dans G)
Pour tout x de G, il existe un unique couple de processus $(\mathrm{X}^x(t), k(t))$ adapté à $\mathcal{T}_t$ ainsi que $\gamma(t)$ tel que

$$\forall t \geqslant 0 \qquad \mathrm{X}^x(t) \in \mathrm{G} \; p.s.$$

$$\forall t \geqslant 0 \qquad \mathrm{X}^x(t) = x + \int_0^t \sigma(\mathrm{X}^x(s))\mathrm{dW}(s) + \int_0^t b(\mathrm{X}^x(s))\mathrm{d}s + k(t) \; p.s.$$

$$\forall \mathrm{T} \geqslant 0 \qquad |k|(\mathrm{T}) < +\infty \; p.s.$$

$$\forall t \geqslant 0 \qquad |k|(t) = \int_0^t \chi_{\mathrm{X}^x(s) \in \partial \mathrm{G}} \, d|k|(s)$$

et
$$k(t) = \int_0^t \gamma(s) d|k|(s) \quad \text{avec} \quad \gamma(s) \in d(\mathrm{X}^x(s)) \, d|k| \; p.s.$$

Preuve : On pourra se reporter à l'annexe D. □.

Nous allons utiliser l'approche précédente pour définir la diffusion sous contraintes dans $\mathrm{G} = \mathcal{S}_\mathcal{F}$. Le vecteur $b(\mathrm{X})$ désignera une direction de descente de gradient. La matrice $\sigma(\mathrm{X})$ sera une matrice de covariance symétrique définie positive assurant que la diffusion dans G soit non-dégénérée.

4.3 Cas particulier où $\mathrm{G} = \mathcal{S}_\mathcal{F}$

Afin de pouvoir appliquer la théorie des équations différentielles stochastiques réfléchies du paragraphe précédent, il est nécessaire de définir des directions de réflexion pour l'espace des contraintes étudiées, c'est-à-dire donner l'ensemble des directions $d(x)$ pour $x \in \partial\mathcal{S}_\mathcal{F}$. C'est ce que nous faisons dans le prochain paragraphe.

4.3.1 Définition des directions de réflexion

Plutôt que d'exhiber précisément l'ensemble des directions de réflexion définissant le problème de Skorokhod sur le simplexe comme dans la section 4.3.2 de [DR99], on peut plutôt envisager de définir une projection π de $\mathcal{H}_\mathcal{F}$ sur $\mathcal{S}_\mathcal{F}$ et en déduire l'ensemble des directions de réflexion par le biais de la condition 3(b) précédente.

Définition 4.3.1 (Projection π)
Si X *est un point quelconque de l'hyperplan $\mathcal{H}_\mathcal{F}$, on définit $\pi(\mathrm{X})$ comme étant le point de $\mathcal{S}_\mathcal{F}$ le plus proche de* X *au sens de la norme euclidienne.*

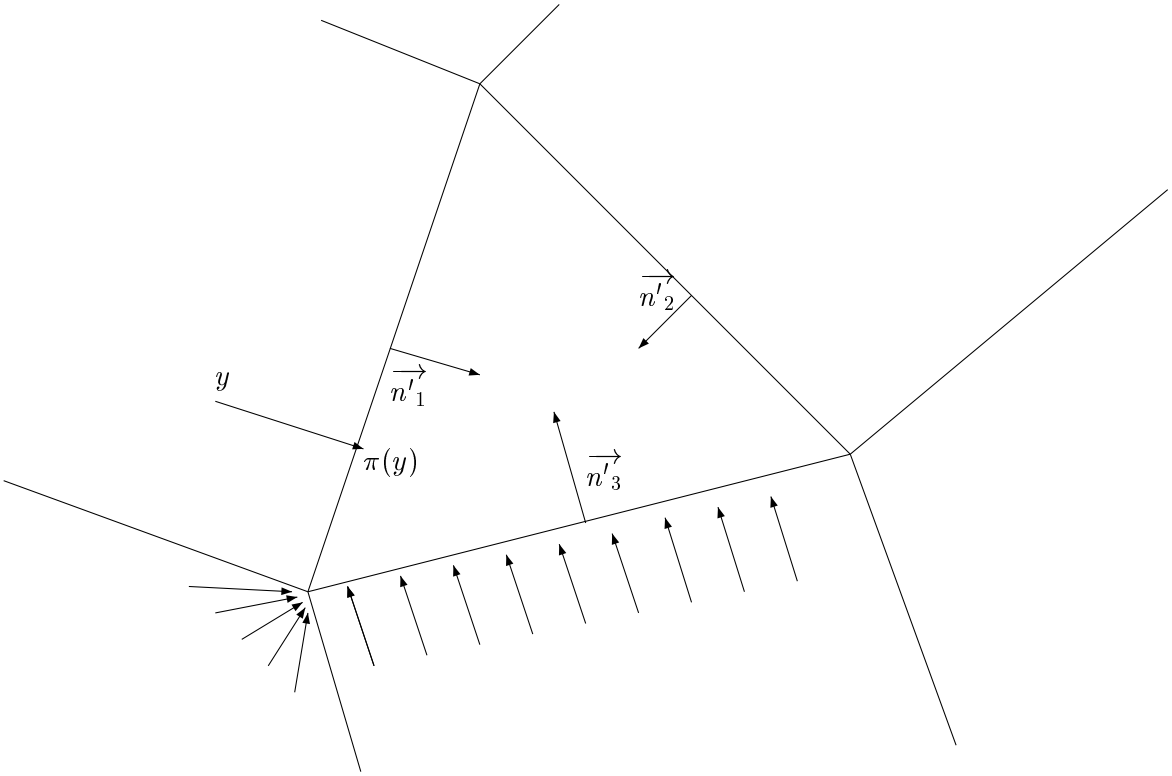
Cette projection $\pi(x)$ existe bien sans ambiguité pour tout x car c'est la projection d'un point x sur le convexe $\mathcal{S}_{\mathcal{F}}$. Nous définissons alors les directions de réflexion du problème de Skorokhod grâce à la définition qui suit.

Définition 4.3.2 (Directions de réflexion)
Soit X *un point de* $\partial\mathcal{S}_{\mathcal{F}}$, *on définit* $d(\mathrm{X})$ *l'ensemble des vecteurs de contraintes actives en* x *par*

$$d(\mathrm{X}) = \left\{ \overrightarrow{\gamma} \quad | \quad \|\gamma\|_2 = 1 \quad et \quad \exists\, y \in \mathcal{H}_{\mathcal{F}} \quad y - \pi(y) = \alpha\gamma \quad et \quad \alpha \leqslant 0 \quad et \quad \mathrm{X} = \pi(y) \right\}$$

On peut alors une nouvelle fois se référer à [DR99] (section 5.2) pour en déduire que la conclusion du théorème 4.2.1 reste vraie, avec ces nouvelles directions de réflexion : le problème de Skorokhod est à nouveau bien posé et l'application Γ est encore Lipschitzienne.

Nous pouvons décrire géométriquement l'ensemble de ces directions de contraintes à l'aide du schéma en deux dimensions suivant :

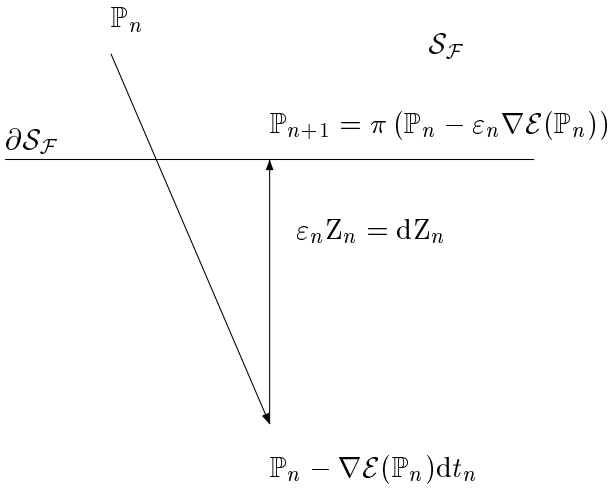


Dans le schéma suivant, les flêches représentent les directions de réflexion possibles en chaque point de la frontière $\partial\mathcal{S}_{\mathcal{F}}$. On constate donc que sur la plupart des points, il n'y a qu'une seule direction de contrainte alors que pour d'autres, il y a plusieurs choix pour les directions de réflexion.

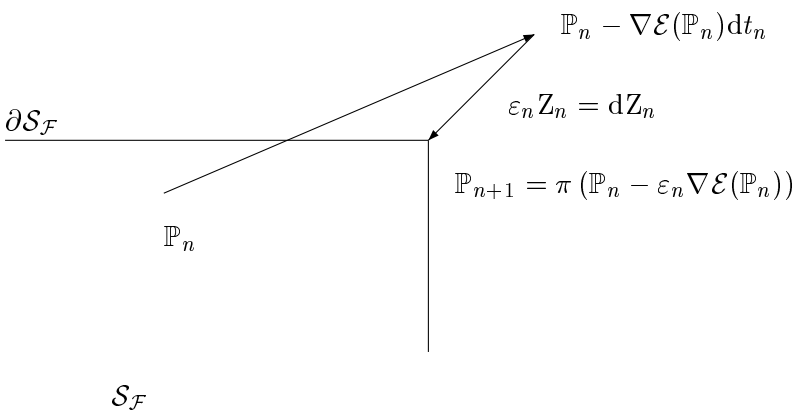
4.3.1.1 Justification des directions de contraintes

Enfin, nous pouvons apporter une autre justification à notre approche du problème de Skorokhod utilisant cette projection π plutôt que des directions de contraintes données *ad-hoc*. Au cours du chapitre 7, nous approcherons une diffusion sous-contraintes sur ce simplexe. Cette approximation stochastique utilisera cette projection π ; cela nous assurera alors facilement que le comportement asymptotique de l'algorithme stochastique approche bien la solution du problème de Skorokhod. En effet, les vecteurs de réflexion du processus approché appartiendront bien à l'ensemble des directions de réflexion du problème original par définition de π, et ceci, par construction de nos directions de réflexion du problème de Skorokhod.

Ce premier schéma représente l'effet qu'aura la projection π sur l'algorithme d'apprentissage, lors d'une projection sur le simplexe où une seule contrainte n'est pas respectée par notre algorithme.



Dans le cas où l'algorithme ne respecte pas plusieurs contraintes, voici le schéma représentant l'effet qu'aura la projection π sur l'algorithme d'apprentissage :



Dans les deux schémas précédents, $\mathbb{P}_n$ désigne l'état de l'algorithme à l'étape n, $\mathbb{P}_n - \nabla\mathcal{E}(\mathbb{P}_n)\mathrm{d}t_n$ l'état où serait virtuellement l'algorithme si aucune contrainte n'était imposée. Enfin, $\varepsilon_n \mathrm{Z}_n = \mathrm{d}\mathrm{Z}_n$ désigne le terme de rappel sur l'espace des contraintes, donné par la projection π. On voit donc que l'équation différentielle sous-jacente au processus est de la forme :

$$\mathrm{d}\mathbb{P}_{t_n} = -\nabla\mathcal{E}(\mathbb{P}_{t_n})\mathrm{d}t_n + \mathrm{d}\mathrm{Z}_{t_n}$$

avec $\mathrm{d}\mathrm{Z}_{t_n}$ dans l'espace des directions de réflexion associées au point $\mathbb{P}_{n+1}$.

On constate donc que la définition précise de SP pour $\mathcal{S}_\mathcal{F}$ revient donc à définir une projection sur $\mathcal{S}_\mathcal{F}$. C'est ce que nous faisons dans le paragraphe suivant en donnant un algorithme exact de projection sur $\mathcal{S}_\mathcal{F}$, où la projection est définie *via* la distance euclidienne dans $\mathbb{R}^f$.

4.3.1.2 Algorithme de projection sur $\mathcal{S}_\mathcal{F}$

La projection π sur le simplexe $\mathcal{S}_\mathcal{F}$ ne s'exprime pas directement en fonction des coordonnées $(\mathbb{P}_1, \dots \mathbb{P}_f)$ de $\mathbb{P}$. Il s'agit donc d'exhiber un algorithme pour implémenter cette projection. Donnons-nous donc un point P hors du simplexe $\mathcal{S}_\mathcal{F}$, la projection $\pi(\mathrm{P})$, plus proche point de P au sens de la norme euclidienne, existe donc puisque c'est l'unique point de $\mathcal{S}_\mathcal{F}$ réalisant la projection de P sur le convexe $\mathcal{S}_\mathcal{F}$. Il faut donc trouver le minimum

$$\operatorname*{Inf}_{(y_1,\dots y_f)\in\mathcal{S}_\mathcal{F}} \sum_{i=1}^{f} (\mathrm{P}_i - y_i)^2$$

Afin de contourner les contraintes de positivité, on peut paramétrer les y_i en w_i^2 et chercher le minimum de la fonction g donnée par

$$g(w) = \sum_{i=1}^{f} \left(\mathrm{P}_i - {w_i}^2\right)^2$$

sous la condition ${w_1}^2 + \dots + {w_f}^2 = 1$. En introduisant le multiplicateur de Lagrange λ, les valeurs w_i recherchées vérifient :

$$\begin{cases} \forall i \in [\![\, 1\,;\, f\,]\!] \qquad w_i({w_i}^2 - \mathrm{P}_i) = \lambda w_i \\ \\ \displaystyle\sum_{i=1}^{f} {w_i}^2 = 1 \end{cases}$$

Si p désigne le nombre de w_i non nuls, les conditions d'optimalité deviennent alors

$$\begin{cases} {w_i}^2 = \mathrm{P}_i + \lambda \qquad \text{si} \qquad w_i \neq 0 \qquad (\mathbf{O}) \\ \\ \lambda = \dfrac{1}{f-p}\left(1 - \displaystyle\sum_{i \quad | \, w_i \neq 0} \mathrm{P}_i\right) \end{cases}$$

Nous pouvons alors donner un algorithme récursif utilisant des projections successives sur les facettes du simplexe qui permet d'atteindre le point $\pi(\mathrm{P})$ vérifiant de telles contraintes en au plus f pas :

1. Si $\mathrm{P}^0 = \mathrm{P}$ n'appartient pas à $\mathcal{H}_\mathcal{F}$, projeter P sur $\mathcal{H}_\mathcal{F}$, on obtient ainsi P^1.
2. (a) Si P^k est dans $\mathcal{S}_\mathcal{F}$, terminer l'algorithme
   (b) Si P^k n'appartient pas à $\mathcal{S}_\mathcal{F}$, poser J_k l'ensemble des indices i tels que $\mathrm{P}_i^k \leqslant 0$. Définir alors P^{k+1} par

$$\begin{cases} \mathrm{P}_i^{k+1} = 0 \qquad \text{si} \qquad i \in \mathrm{J}_k \\ \\ \mathrm{P}_i^{k+1} = \mathrm{P}_i^k + \dfrac{1}{f - |\mathrm{J}_k|} \left(1 - \sum\limits_{j \notin \mathrm{J}_k} \mathrm{P}_j^k \right) \qquad \text{sinon} \end{cases}$$

On peut alors démontrer que la suite $\left(\mathrm{P}^k\right)_{k \in \mathbb{N}}$ est bien une suite de $\mathcal{H}_{\mathcal{F}}$ et qu'au bout de f itérations, l'algorithme est stationnaire, et vaut alors $\pi(\mathrm{P})$.

Preuve : On commence par remarquer que la suite de points construits appartient toujours à $\mathcal{H}_{\mathcal{F}}$. Ceci est évident puisque à l'issue de l'étape 2.b, la somme des $\left(\mathrm{P}_i^{k+1}\right)$ vaut précisément :

$$\sum_i \mathrm{P}_i^{k+1} = \underbrace{\sum_{i \in \mathrm{J}_k} \mathrm{P}_i^{k+1}}_{=0} + \sum_{i \notin \mathrm{J}_k} \mathrm{P}_i^{k+1}$$

Mais pour les indices i hors de J_k, l'expression de P^{k+1} donne alors

$$\sum_i \mathrm{P}_i^{k+1} = \sum_{i \notin \mathrm{J}_k} \mathrm{P}_i^k + \frac{1}{f - |\mathrm{J}_k|} \underbrace{\left(1 - \sum_{j \notin \mathrm{J}_k} \mathrm{P}_j^k \right)}_{\text{Indépendant de} j} \underbrace{(f - |\mathrm{J}_k|)}_{\text{Nbe termes}} = 1$$

Par conséquent, la suite des points $\left(\mathrm{P}^k\right)$ appartient bien à $\mathcal{H}_{\mathcal{F}}$.

Démontrons ensuite que l'algorithme est stationnaire au bout de f itérations. Pour cela, nous allons supposer (par l'absurde) que P^f n'est pas dans $\mathcal{S}_{\mathcal{F}}$ et étudier le nombre de coordonnées de P^{k+1} nulles. Nous montrons que si P^k n'appartient pas au simplexe, alors ce nombre est précisément supérieur ou égal à k.

Si $k = 1$, P^1 n'est pas dans $\mathcal{S}_{\mathcal{F}}$, il existe i tel que $\mathrm{P}_i^1 < 0$ et dans ce cas, $\mathrm{P}_i^2 = 0$, donc le nombre de coordonnées nulles de P^2 vaut donc au moins 1.

P^k n'est pas dans $\mathcal{S}_{\mathcal{F}}$ et possède au moins k coordonnées nulles, comme P^k n'est pas dans $\mathcal{S}_{\mathcal{F}}$, il possède une coordonnée strictement négative qui sera alors nulle pour l'étape $k+1$. Mais les coordonnées nulles de P^k sont également nulles pour P^{k+1}. On obtient donc $k+1$ coordonnées nulles pour P^{k+1}. C'est ce qu'il fallait démontrer.

Ainsi, si P^f n'était pas dans $\mathcal{S}_{\mathcal{F}}$, le nombre de coordonnées nulles de P^f vaudrait f, c'est absurde. En conclusion, P^f est dans $\mathcal{S}_{\mathcal{F}}$.

Enfin, montrons que le point atteint par $\mathcal{S}_{\mathcal{F}}$ est bien le projeté de P^1 sur $\mathcal{S}_{\mathcal{F}}$. Il suffit en réalité de remarquer que si P_i^f est non nul, alors par itération de notre algorithme, on a obtenu les termes P_i^f non nuls en ajoutant à chaque étape une même quantité, indépendamment de la coordonnée i étudiée dans l'ensemble des coordonnées non nulles de P^f. Ainsi, il existe λ tel que :

$$\begin{cases} \mathrm{P}_i^f = \mathrm{P}_i^0 + \lambda \qquad \text{si} \qquad \mathrm{P}_i^f \neq 0 \\ \\ \mathrm{P}_i^f = 0 \qquad \text{sinon} \end{cases}$$

Comme en plus P^f est dans $\mathcal{S}_{\mathcal{F}}$, et qu'on vérifie les conditions d'optimalité $(\mathbf{O})$, on en conclut que P^f est bien le projeté de P^0 cherché. $\square$.

4.3.2 Descente de gradient sous contraintes dans $\mathcal{S}_{\mathcal{F}}$

On cherche donc à utiliser un problème du type Skorokhod pour effectuer une descente de gradient définie correctement sur $\mathcal{S}_{\mathcal{F}}$ pour l'énergie

$$\mathcal{E}(\mathbb{P}) = \mathcal{E}_{err}(\mathbb{P}) = \sum_{\omega \in \mathcal{F}^p} g(\omega)\mathbb{P}(\omega)$$

On peut effectivement utiliser un problème SP basé sur une diffusion avec un terme de dérive $-\nabla\mathcal{E}(\mathbb{P})$ grâce aux définitions précédentes pour s'assurer dès lors de l'existence et de l'unicité de la diffusion sous contraintes :

$$\begin{cases} \mathrm{X}_t = \mathbb{P}_0 - \displaystyle\int_0^t \Pi_{\mathcal{H}_\mathcal{F}} \left(\nabla\mathcal{E}(\mathbb{P}_s)\right) \mathrm{d}s + \sigma \mathrm{dW}(s) \\ \mathbb{P}_t = \Gamma(\mathrm{X})_t \end{cases}$$

C'est-à-dire $\qquad \forall \mathbb{P} \in \mathcal{S}_\mathcal{F} \qquad b(\mathbb{P}) = -\Pi_{\mathcal{H}_\mathcal{F}} \left(\nabla\mathcal{E}(\mathbb{P})\right)$

où l'application $\Pi_{\mathcal{H}_\mathcal{F}}$ est la projection orthogonale sur l'hyperplan $\mathcal{H}_\mathcal{F}$. On munit donc l'espace $\mathcal{S}_\mathcal{F}$ de la norme euclidienne $\|.\|_2$.

Il s'agit alors de vérifier que les inégalités $(\mathbf{C_1})$, $(\mathbf{C_2})$ et $(\mathbf{C_3})$ sont bien vraies dans notre cas particulier.

- La condition $(\mathbf{C_2})$ provient du fait que $\mathcal{S}_\mathcal{F}$ est compact et b continue.
- Pour démontrer que $(\mathbf{C_1})$ est également vraie, on écrit

$$\begin{aligned} \|b(\mathbb{P}_1) - b(\mathbb{P}_2)\|_2^{\,2} &= \|\Pi_{\mathcal{H}_\mathcal{F}} \left(\nabla\mathcal{E}_{err}(\mathbb{P}_1) - \nabla\mathcal{E}_{err}(\mathbb{P}_2)\right) \|_2^{\,2} \\ &\leqslant \|\nabla\mathcal{E}_{err}(\mathbb{P}_1) - \nabla\mathcal{E}_{err}(\mathbb{P}_2)\|_2^{\,2} \\ &= \sum_{\delta \in \mathcal{F}} \left[\sum_{\omega \in \mathcal{F}^p} \mathrm{C}(\omega,\delta) g(\omega) (\mathbb{P}_1(\omega_{\diagdown\delta}) - \mathbb{P}_2(\omega_{\diagdown\delta})) \right]^2 \\ &\leqslant 2p^2 \sum_{\omega \in \mathcal{F}^{p-1}} \left[\mathbb{P}_1(\omega) - \mathbb{P}_2(\omega)\right]^2 \end{aligned}$$

En écrivant $\mathbb{P}_1 = \mathbb{P}_2 + h$, on peut alors écrire :

$$\begin{aligned} \frac{\|b(\mathbb{P}_1) - b(\mathbb{P}_2)\|_2^{\,2}}{\|h\|_2^{\,2}} &\leqslant 2p^2 \frac{\left[\displaystyle\sum_{i_1 \ldots i_{p-1}} (\mathbb{P}_2(i_1) + h(i_1)) \ldots (\mathbb{P}_2(i_{p-1}) + h(i_{p-1})) - \mathbb{P}_2(i_1) \ldots \mathbb{P}_2(i_{p-1}) \right]^2}{\|h\|_2^{\,2}} \\ &\leqslant 2p^2 \frac{\left[\displaystyle\sum_{i_1 \ldots i_{p-1}} \sum_{j=1}^{p-1} h(i_j) \left(\prod_{k \neq j} \mathbb{P}_2(i_k) + \mathrm{o}\left(\|h\|_2\right) \right) \right]^2}{\|h\|_2^{\,2}} \end{aligned}$$

Ainsi, l'application Φ définie par

$$\begin{cases} (\mathcal{S}_\mathcal{F} \times \mathcal{S}_\mathcal{F})_{\diagdown\Delta} \longrightarrow \mathbb{R} \\ (\mathbb{P}, \mathbb{Q}) \longmapsto \displaystyle\frac{\sum_{i_1 \ldots i_{p-1}} \left[\mathbb{P}(i_1 \ldots i_{p-1}) - \mathbb{Q}(i_1 \ldots i_{p-1})\right]^2}{\|\mathrm{P} - \mathrm{Q}\|_2^{\,2}} \end{cases}$$

où Δ est l'ensemble des couples $(\mathbb{P}, \mathbb{P})$ de ${\mathcal{S}_\mathcal{F}}^2$, est continue d'après l'inégalité précédente et se prolonge par continuité à ${\mathcal{S}_\mathcal{F}}^2$. Elle y est donc en particulier bornée par un réel M. Ce qui se traduit finalement par

$$\forall(\mathbb{P},\mathbb{Q}) \in \mathcal{S}_{\mathcal{F}}^2 \qquad \Phi(\mathbb{P},\mathbb{Q}) \leqslant \mathrm{M}$$

soit $\forall(\mathbb{P}_1,\mathbb{P}_2) \in \mathcal{S}_{\mathcal{F}}^2 \qquad \|b(\mathbb{P}_1) - b(\mathbb{P}_2)\|_2{}^2 \leqslant 2p^2\Phi(\mathbb{P}_1,\mathbb{P}_2)\|\mathbb{P}_1 - \mathbb{P}_2\|_2{}^2$

c'est-à-dire $\boxed{\forall(\mathbb{P}_1,\mathbb{P}_2) \in \mathcal{S}_{\mathcal{F}}^2 \qquad \|b(\mathbb{P}_1) - b(\mathbb{P}_2)\|_2{}^2 \leqslant 2p^2\mathrm{M}\|\mathbb{P}_1 - \mathbb{P}_2\|^2}$

La condition $(\mathbf{C_1})$ est donc également vérifiée.

- Enfin, on choisit de faire une diffusion brownienne standard sur $\mathcal{H}_{\mathcal{F}}$. Cette diffusion correspond donc au choix de σ constant, indépendant du temps et de la position du point dans $\mathcal{S}_{\mathcal{F}}$. Ce choix est détaillé dans [RW00], tome 2, page 114. En utilisant toujours $\overrightarrow{\mathrm{N}}$ le vecteur unitaire normal à $\mathcal{H}_{\mathcal{F}}$, on pose :

$$\sigma = \mathrm{Id}_{|\mathcal{F}|} - \overrightarrow{\mathrm{N}}\left(\overrightarrow{\mathrm{N}}\right)^t$$

σ est donc donné par la matrice symétrique positive valant

$$\sigma = \frac{1}{f}\begin{pmatrix} f-1 & -1 & \dots & -1 \\ -1 & f-1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & \dots & -1 & f-1 \end{pmatrix} \tag{4.26}$$

Les trois conditions $(\mathbf{C_1})$, $(\mathbf{C_2})$ et $(\mathbf{C_3})$ étant vraies, on peut donc établir le théorème 4.3.1.

Théorème 4.3.1 (Existence et Unicité des diffusions standard contraintes dans $\mathcal{S}_{\mathcal{F}}$) *Étant donnée $\mathbb{P}$ variable aléatoire $\mathcal{T}_0$ mesurable et $\mathrm{W}(t)$ le mouvement brownien standard sur $\mathbb{R}^{|\mathcal{F}|}$, il existe un unique couple de processus $(\mathbb{P}_t, k_t)$ adapté à la filtration $(\mathcal{T}_t)_{t\geqslant 0}$ vérifiant les conditions suivantes :*

1. $\mathbb{P}_0 = \mathbb{P}\ p.s$
2. $\mathrm{d}\mathbb{P}_t = -\Pi_{\mathcal{H}_{\mathcal{F}}}\left(\nabla\mathcal{E}(\mathbb{P}_t)\right)\mathrm{d}t + \sigma\mathrm{dW}(t) + \mathrm{d}k(t)$
3. $\forall \mathrm{T} \geqslant 0 \qquad |k|(\mathrm{T}) < +\infty$
4. $\forall t \in \mathbb{R}_+ \qquad \mathbb{P}_t \in \mathcal{S}_{\mathcal{F}}$

La diffusion sous contraintes ainsi construite a donc pour objectif de se substituer à la descente de gradient déterministe du chapitre précédent. Celle-ci ne se voit plus confrontée au problème du respect des contraintes :

$$\mathbb{P}(\delta) \geqslant 0$$

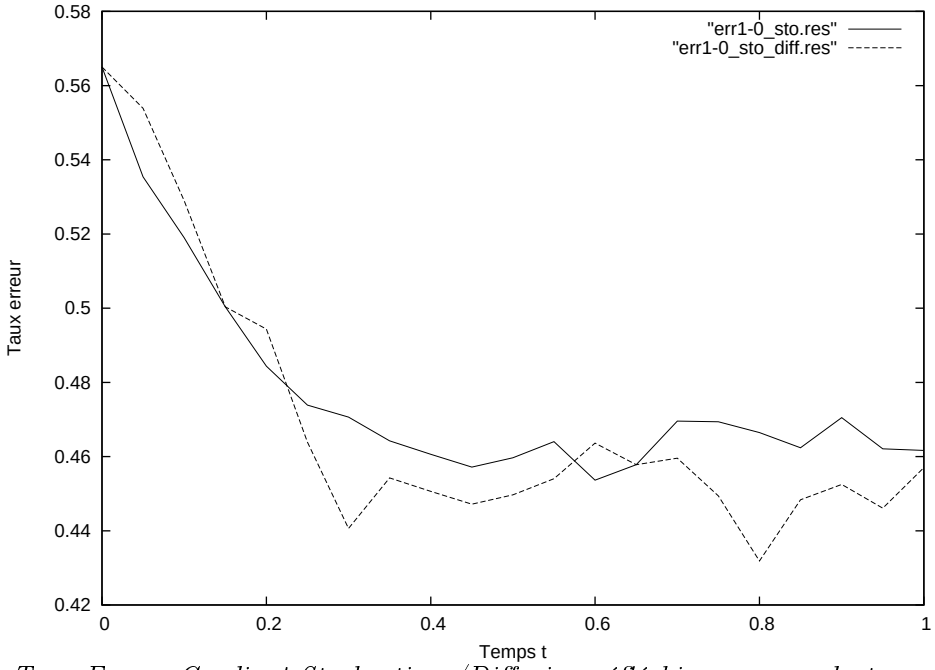
puisque ces contraintes sont alors imposées naturellement dans notre modèle.

L'objectif identique entre le modèle de diffusion de ce chapitre et celui du chapitre 3 est de parvenir à proposer des features ω de $\mathcal{F}^p$ performants pour la détection représentée par g. Mais le calcul de $\nabla\mathcal{E}$ étant toujours impossible au temps t, nous verrons au chapitre 7 comment simuler une suite de processus approchant au mieux la solution de l'équation différentielle stochastique précédente. Nous appliquerons par la suite ce nouvel algorithme sur les exemples du chapitre 3.

4.4 Expériences

4.4.1 Cadre synthétique

Nous pouvons comparer les résultat obtenus dans le cas des données synthétiques lorsqu'on utilise les deux méthodes présentées : équation différentielle ($\mathbf{E-3}$) ou diffusion sous contraintes dans $\mathcal{S}_{\mathcal{F}}$ donnée par le paragraphe 4.3.2. Voici les courbes d'erreur obtenues, pour la même temps horloge, sur l'exemple synthétique lorsque nous bruitons les données par une diffusion brownienne standard ou lorsque nous effectuons une descente de gradient donnée dans le chapitre 3.



Taux Erreur Gradient Stochastique/Diffusion réfléchie au cours du temps

On constate dans cette première expérience sur la diffusion réfléchie que les taux d'erreurs obtenus par notre modèle sont en tout point comparables à ceux obtenus dans le chapitre précédent. Il est également intéressant d'étudier aussi les variables sélectionnées lors de notre apprentissage de $\mathbb{P}$. Voici donc le poids moyen donné par $\mathbb{P}_t$ sur chacune des variables que nous considérons. La méthode de représentation des variables est identique à celle du chapitre 3 : plus la tâche est sombre, plus la variable est chargée, alors qu'inversement, plus la tâche est claire, moins la variable est chargée.

$\mathcal{F}$	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8	δ_9	δ_{10}	δ_{11}	δ_{12}	δ_{13}	δ_{14}	δ_{15}	δ_{16}	δ_{17}	δ_{18}	δ_{19}	δ_{20}
Gradient																				
Diffusion																				

Dans notre cadre synthétique, la sélection des variables par la méthode de la diffusion réfléchie est moins nette, comparée aux résultats du chapitre 3. On constate néanmoins que globalement, les variables qui sont retenues par notre méthode sont tout de même les variables qui recouvrent

les sources des données $\mathcal{F}_i^j$. Du point de vue de la richesse du vocabulaire de variables sélectionnées, les résultats sont tout de même meilleurs puisque l'ensemble des variables significativement chargées est plus conséquent qu'à l'issue de notre premier algorithme de descente de gradient. C'est ce type de résultat que nous souhaitons exploiter pour pouvoir composer des features plus complexes ultérieurement.

4.4.2 Détection de visages

Les résultats que nous obtenons dans le cadre de la détection des visages ne sont pas meilleurs en terme de performances lorsque nous procédons à un algorithme de diffusion réfléchie pour l'apprentissage de $\mathbb{P}_t$. Les améliorations ou baisses de performances ne sont pas spectaculaires, l'amélioration des performances peut s'interpréter dans certains cas comme la possibilité pour notre algorithme de diffusion sur $\mathcal{S}_\mathcal{F}$ d'éviter de rester piégée dans des minima locaux et d'en atteindre d'autres correspondant à des énergies $\mathcal{E}_{err}$ plus basses.

Nous pouvons étudier, comme dans l'approche du chapitre 3, les taux d'erreur obtenus en moyenne sur l'ensemble des variables extraites selon une probabilité $\mathbb{P}_t$ suivant un processus de diffusion réfléchie sur $\mathcal{S}_\mathcal{F}$. La limite ponctuelle « limite » de $\mathbb{P}_t$ n'ayant pas de sens dans notre situation, nous avons simplement choisi d'extraire les variables selon une probabilité $\mathbb{P}_t$ après plusieurs heures d'apprentissage. Nous avons alors recensé les taux d'erreur moyens obtenus sur les données de test après vote de 10 experts pour la détection de visages. Nous comparons alors les taux d'erreurs recensées aux taux d'erreurs obtenus lors de la simple descente de gradient du chapitre 3 sur l'ensemble de test.

$\lvert\omega\rvert$	Descente de gradient	Diffusion réfléchie
20	7.5%	8%
30	6.9%	6.9%
40	6.5%	6.7%
50	5.5%	4.5%
100	2.3%	2.7%

L'effet de la diffusion sur les performances de notre algorithme est double :

- Nous avons constaté numériquement que le temps nécessaire à l'obtention de bonnes performances pour notre algorithme était plus long que dans le cas d'une simple descente de gradient.
- Certaines performances obtenues lors de cette diffusion réfléchie ne sont pas meilleures que lors de la simple descente de gradient (taux d'erreur moyen de détection pour $|\omega| = 100$ par exemple) alors que d'autres taux d'erreurs moyens obtenus sont meilleurs. Ceci est très certainement dû aux instants que nous avons choisi pour stopper notre algorithme et mesurer ces taux sur l'ensemble de test. Ces instants de mesure sont en effet déterminés de manière arbitraire dans notre cas sans aucune considération statistique sur la stationnarité ou non du comportement de $\mathbb{P}_t$ au voisinage du temps de mesure.

En fin de compte, les performances de diffusion réfléchie pour l'étude de la détection de visages n'apporte pas d'amélioration tangible par rapport à la descente de gradient du chapitre 3. La vitesse d'un tel algorithme est plus lente que celle de la simple descente de gradient mais cette diffusion contraint plus facilement $\mathbb{P}_t$ dans $\mathcal{S}_\mathcal{F}$ et permet de proposer plus de variables de $\mathcal{F}$ *via* le mouvement Brownien standard sur $\mathcal{S}_\mathcal{F}$.

Par ailleurs, la réprésentation de la localisation des détecteurs de bords effectivement chargés par la probabilité a donné des résultats similaires à ceux mentionnés dans le paragraphe 3.9.3 concernant la descente de gradient.

Chapitre 5 - Processus de diffusion réfléchie avec sauts

5.1 Objectifs

L'objet de ce chapitre est de proposer une alternative pour sélectionner des features de $\mathcal{F}$ aux équations différentielles $(\mathbf{E}-\mathbf{3})$ et $(\mathbf{E}-\mathbf{4})$ proposées dans le chapitre précédent. Le but qui sera poursuivi est toujours de construire une méthode de sélection de variables, en s'autorisant cette fois à enrichir l'ensemble $\mathcal{F}$ de nouveaux features composés, créés à partir de variables appartenant elles aussi à $\mathcal{F}$. Nous utiliserons à ces fins les structures d'arbres et de forêts définies dans le chapitre 2. La modélisation du chapitre 3 peut alors être améliorée pour plusieurs raisons :

- il semble relativement artificiel de proposer une force de rappel vers la loi $\mathcal{U}_{\mathcal{F}}$, sachant que ce rappel est « aveugle » à l'apprentissage effectué lors de la minimisation de $\mathcal{E}_1$. Ce rappel peut donc annuler le gain effectué lors de la recherche des features performants pour la tâche de détection.
- Plus encore, la force de rappel s'accentue lorsqu'un feature n'apporte pas de bonnes performances à g puisque le terme $-\nabla\mathcal{E}_1$ aura certainement tendance à diminuer la probabilité de tirer un tel feature, mais le terme $-\nabla\mathcal{E}_2$ sera alors d'autant plus fort que la probabilité est faible ($\mathcal{U}_{\mathcal{F}} - \mathbb{P}(\delta)$ est grand quand $\mathbb{P}(\delta)$ est petit). La modélisation précédente aboutit donc à la construction d'un compromis qui peut paraître bancal entre la force de $\mathcal{E}_1$ et celle de $\mathcal{E}_2$ (on peut par exemple examiner précisément les conditions de stabilité dans $\mathcal{S}_{\mathcal{F}}$ théoriques établies précédemment) alors que justement, il serait souhaitable de vraiment positionner $\mathbb{P}(\delta)$ le plus proche possible de 0 lorsque l'apport de δ en terme de performance est médiocre.
- Enfin, on remarque dans l'approche du chapitre 3 que la modélisation fige l'ensemble des features $\mathcal{F}$ sans laisser une chance à $\mathcal{F}$ d'évoluer. Ceci est structurellement lié à notre modèle qui ne permet ni de créer de nouveaux éléments de $\mathcal{F}$, ni d'en supprimer. Il serait pourtant agréable de permettre de supprimer par exemple des features de $\mathcal{F}$ qui sont mal notés *via* $\mathbb{P}$ et d'ajouter des features composés apportant une information non négligeable au problème de détection.

Dans un premier temps, nous allons chercher à décrire très précisément un processus de diffusion réfléchie avec sauts qui couple à la fois une descente de gradient (intervalle de temps de diffusion), et la recherche d'un espace optimal lors de transitions brutales (instants de sauts).

Nous définissons là-encore ce processus comme la solution d'une équation différentielle stochastique comme dans le chapitre précédent. Nous établissons alors les conditions générales d'existence de diffusions réfléchies avec sauts.

Puis nous formalisons précisément l'exploration des forêts possibles $\mathcal{F}$ en donnant les règles de transition qui interviennent lors des sauts, ces règles suivront un comportement de type

MCMC et devront donc satisfaire des propriétés de réversibilité importantes pour la stabilité du processus.

Cela nous permettra finalement de construire dans un second temps un processus $((\mathcal{F}_t;\mathbb{P}_t))_{t\in\mathbb{R}^+}$ à dynamique markovienne $(\mathcal{F}_t)_{t\in\mathbb{R}^+}$ qui fera évoluer l'ensemble des features disponibles au temps t. Ce processus permettra de parcourir l'ensemble des forêts possibles, formées à partir des mots et des lettres élémentaires appartenant à $\mathcal{F}_0$, forêt des features élémentaires. La méthode utilisée est une méthode à sauts réversibles, technique la plus utilisée en exploration de modèles statistiques à dimension variable ([CRR03], [Gre95]). Ce processus sera guidé par une diffusion avec sauts distribués selon une loi de Poisson.

5.2 Diffusions réfléchies avec sauts

5.2.1 Formalisation

Nous noterons $\mathcal{A}^\star$ tous les arbres binaires issus de features élementaires de $\mathcal{F}_0$ et dont la définition est donnée au chapitre 2, paragraphe 2.7. Afin de pouvoir établir l'équation différentielle définissant notre processus, nous introduisons le vecteur X_t indicateur de la forêt $\mathcal{F}_t$ par les définitions ci-dessous.

Définition 5.2.1 (Vecteur X_t)
Le vecteur X_t est l'élément de $\{0;1\}^{|\mathcal{A}^\star|}$ tel que

$$\forall \mathcal{A}\in\mathcal{A}^\star \qquad \mathrm{X}_t(\mathcal{A})=1 \Leftrightarrow \mathcal{A}\in\mathcal{F}_t$$

Ainsi, $\mathrm{X}_t(\mathcal{A})$ vaut 1 si l'arbre $\mathcal{A}$ est présent au temps t et 0 sinon.

On définit alors l'application « réciproque » qui, à un vecteur X de $\{0;1\}^{|\mathcal{A}^\star|}$, associe la forêt qui lui correspond dans la définition 5.2.2.

Définition 5.2.2 (Application $\mathcal{F}$)
Si $\mathrm{X}\in\{0;1\}^{|\mathcal{A}^\star|}$, on définit $\mathcal{F}(\mathrm{X})$ comme étant la forêt qui satisfait :

$$\forall \mathcal{A}\in\mathcal{A}^\star \qquad \mathcal{A}\in\mathcal{F}(\mathrm{X}) \Leftrightarrow \mathrm{X}(\mathcal{A})=1$$

Nous considérons alors la probabilité $\mathbb{P}_t$ non pas comme un élément de $\mathcal{S}_{\mathcal{F}_t}$ mais comme une probabilité sur $\mathcal{A}^\star$. Nous plongeons donc chaque espace $\mathcal{S}_\mathcal{F}$ dans $\mathcal{S}_{\mathcal{A}^\star}$ en imposant des conditions de nullité sur les arbres n'appartenant pas à $\mathcal{F}$. Ainsi, si P est une probabilité sur $\mathcal{F}$, son image plongée $\overline{\mathrm{P}}$ est donnée par :

$$\forall \mathcal{A}\in\mathcal{F} \quad \forall \mathrm{P}\in\mathcal{S}_\mathcal{F} \qquad \overline{\mathrm{P}}(\mathcal{A})=\mathrm{P}(\mathcal{A})$$

et

$$\forall \mathcal{A}\notin\mathcal{F} \quad \forall \mathrm{P}\in\mathcal{S}_\mathcal{F} \qquad \overline{\mathrm{P}}(\mathcal{A})=0$$

Il y a donc parfaitement équivalence entre la connaissance du couple $(\mathbb{P}_t;\mathcal{F}_t)$ et la connaissance de $\left(\overline{\mathbb{P}_t};\mathrm{X}_t\right)$ en tout temps. Pour plus de lisibilité de la suite, nous confondrons systématiquement $\mathbb{P}_t$ et $\overline{\mathbb{P}_t}$.

L'équation d'évolution que nous allons étudier est alors du type :

$$(\mathbf{E-6})\begin{cases} \begin{pmatrix}\mathrm{Z}_t\\ \mathrm{X}_t\end{pmatrix} = \begin{pmatrix}\mathbb{P}_0\\ \mathrm{X}_0\end{pmatrix} - \displaystyle\int_0^t \mathrm{G}\begin{pmatrix}\mathbb{P}_s\\ \mathrm{X}_s\end{pmatrix}\mathrm{d}s + \int_0^t \Sigma\begin{pmatrix}\mathbb{P}_s\\ \mathrm{X}_s\end{pmatrix}\mathrm{dW}_s \\ \qquad\qquad + \displaystyle\int_0^t\int_{\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^{|\mathcal{A}^\star|}} q\left(\begin{pmatrix}\mathbb{P}_s\\ \mathrm{X}_s\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)\mathrm{N}\left(\mathrm{d}\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix};\mathrm{d}s\right) \\ \mathbb{P}_t=\Gamma_{\mathrm{X}_t}(\mathrm{Z}_t)\end{cases}$$

On peut par ailleurs reformuler cette équation $(\mathbf{E}-\mathbf{6})$ en équation différentielle stochastique comme ce qui est formulé dans [Kus00] et [Kus02] :

$$\begin{aligned} \mathrm{d}\begin{pmatrix} \mathrm{Z}_t \\ \mathrm{X}_t \end{pmatrix} &= -\mathrm{G}\begin{pmatrix} \mathbb{P}_t \\ \mathrm{X}_t \end{pmatrix}\mathrm{d}s + \Sigma\begin{pmatrix} \mathbb{P}_t \\ \mathrm{X}_t \end{pmatrix}\mathrm{dW}_s \\ &\quad + \int_{\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^{|\mathcal{A}^\star|}} q\left(\begin{pmatrix} \mathbb{P}_t \\ \mathrm{X}_t \end{pmatrix};\begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix}\right)\mathrm{N}\left(\mathrm{d}\begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix};\mathrm{d}t\right) \qquad (\mathbf{E}-\mathbf{7}) \\ &\quad + d(\mathrm{Z}-\mathbb{P})_t \end{aligned}$$

Ces deux dernières équations méritent quelques éclaircissements :

- L'application G est l'application qui donne le terme de « dérive » de l'équation de diffusion tandis que Σ est le terme de covariance du mouvement brownien standard.

  Plus tard, le terme de dérive $(-\mathrm{G})$ sera en réalité identifié à un terme de descente de gradient. Σ sera, elle, la matrice de covariance, définie positive sur $\mathcal{H}_{\mathcal{F}}$, qui bruitera le mouvement sur les particules sélectionnées (codées par le vecteur X).

  Enfin, X désigne en réalité les coordonnées libres de notre processus tandis que les coordonnées pour lesquelles $\mathrm{X}=0$ sont figées et nulles entre deux variations de X, c'est-à-dire entre deux sauts.

- Par ailleurs, $q(\mathrm{Z}_1;\mathrm{Z}_2)$ est la quantité à ajouter à l'état Z_1 pour passer à l'état Z_2 multiplié par la probabilité Q d'effectuer cette transition. La probabilité Q est en réalité celle qui sera définie dans le paragraphe suivant et qui régit les lois de parcours des forêts et donc des vecteurs X appartenant à $\mathcal{A}^\star$. C'est donc précisément q et N qui organisent la dynamique des sauts du processus.

- N désigne la mesure produit entre une mesure uniforme sur $\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^{|\mathcal{A}^\star|}$ et Poissonnienne en temps de paramètre λ.

- Enfin, Γ_{X_t} est l'application de Skorokhod décrite dans le paragraphe précédent associé au problème sous contraintes dans $\mathcal{S}_{\mathcal{F}(\mathrm{X}_t)}$, c'est-à-dire le simplexe quantifié par le vecteur X_t.

- $\mathrm{Z}-\mathbb{P}$ est le processus réfléchi qui permet de contraindre $\mathbb{P}$ dans $\mathcal{S}_{\mathcal{X}_t}$ en tout temps. Ce processus ne varie qu'aux points où $\mathbb{P}_t$ atteint la frontière $\partial\mathcal{S}_{\mathcal{X}_t}$. C'est précisément l'application de Skorokhod qui permet d'obtenir ce couple $(\mathbb{P};\mathrm{Z})$ entre deux instants où X ne varie pas, c'est-à-dire entre deux sauts distribués selon une loi de Poisson $\mathcal{P}(\lambda)$. Avec les définitions du chapitre 4 de SP, on a donc les correspondances :

$$\begin{cases} \mathbb{P} \longmapsto \phi \\ \mathrm{Z} \longmapsto \psi \\ \mathrm{Z}-\mathbb{P} \longmapsto \eta \end{cases}$$

- L'application Γ_{X} est construite pour chaque simplexe paramétré par X de manière identique à celle du chapitre précédent dans la section 4.2, ou en se référant à nouveau à [DR99].

Nous cherchons donc à construire un processus couplé $(\mathbb{P}_t;\mathrm{X}_t)$ tel que $\mathbb{P}_t$ suive une diffusion sous contraintes dictée par X_t entre deux instants de sauts et suivant des règles de transitions markoviennes lors des instants de sauts qui sont distribués suivant une loi de Poisson.

Nous ne donnerons un sens précis aux termes G et Σ mentionnés ci-dessus que dans le paragraphe 5.3, ceci afin de maintenir la généralité du pragraphe suivant. Enfin, si Q désigne la probabilité de transition d'une forêt à une autre mentionnée dans le paragraphe 5.3, on a alors la relation entre Q et q définie dans la définition suivante.

Définition 5.2.3 (Amplitudes des sauts q)
L'amplitude des sauts dans $(\mathbf{E}-\mathbf{7})$ *est donnée par :*

$$q\left(\begin{pmatrix}\mathbb{P}_1\\ \mathrm{X}_1\end{pmatrix};\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix}\right)=\begin{pmatrix}\mathbb{P}_2-\mathbb{P}_1\\ \mathrm{X}_2-\mathrm{X}_1\end{pmatrix}\mathrm{Q}\left[\begin{pmatrix}\mathbb{P}_1\\ \mathcal{F}(\mathrm{X}_1)\end{pmatrix};\begin{pmatrix}\mathbb{P}_2\\ \mathcal{F}(\mathrm{X}_2)\end{pmatrix}\right] \tag{5.27}$$

5.2.2 Existence des processus de diffusions réfléchies avec saut

On peut également construire, à l'image de ce qui est fait pour le théorème 4.2.2 du chapitre précédent, la solution unique d'une telle équation intégrale $(\mathbf{E}-\mathbf{6})$ en utilisant à nouveau un théorème de point fixe de Picard. Les conditions qu'il faut cette fois imposer sont à nouveau des conditions de type Lipschitz. On pourra se rapporter à [AB02] ou à [DI91] pour la démonstration de l'existence et l'unicité d'un tel processus sous réserve que les conditions suivantes soient satisfaites :

1. Le caractère Lipschitzien de la dérive :

$$\exists\,\theta_1>0 \qquad \left\|\mathrm{G}\begin{pmatrix}\mathbb{P}_1\\ \mathrm{X}_1\end{pmatrix}-\mathrm{G}\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix}\right\|\leqslant\theta_1\left\|\begin{pmatrix}\mathbb{P}_1-\mathbb{P}_2\\ \mathrm{X}_1-\mathrm{X}_2\end{pmatrix}\right\| \tag{$\mathbf{C_4}$}$$

   et de la covariance de la diffusion :

$$\exists\,\theta_2>0 \qquad \left\|\Sigma\begin{pmatrix}\mathbb{P}_1\\ \mathrm{X}_1\end{pmatrix}-\Sigma\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix}\right\|\leqslant\theta_2\left\|\begin{pmatrix}\mathbb{P}_1-\mathbb{P}_2\\ \mathrm{X}_1-\mathrm{X}_2\end{pmatrix}\right\| \tag{$\mathbf{C_5}$}$$

2. Le caractère Lipschitzien, uniformément en la deuxième variable, de la probabilité de transition Q :

$$\exists\,\theta_3>0 \qquad \left\|q\left(\begin{pmatrix}\mathbb{P}_1\\ \mathrm{X}_1\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)-q\left(\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)\right\|\leqslant\theta_3\left\|\begin{pmatrix}\mathbb{P}_1-\mathbb{P}_2\\ \mathrm{X}_1-\mathrm{X}_2\end{pmatrix}\right\| \tag{$\mathbf{C_6}$}$$

3. Le caractère Lipschitzien de l'application Γ_{X} :

$$\exists\,\theta_4 \qquad \sup_{t\geqslant0}\|\Gamma_{\mathrm{X}_1}(\mathrm{Z}_1)_t-\Gamma_{\mathrm{X}_2}(\mathrm{Z}_2)_t\|\leqslant\theta_4\sup_{t\geqslant0}\left\|\begin{pmatrix}\mathbb{Z}_1-\mathbb{Z}_2\\ \mathrm{X}_1-\mathrm{X}_2\end{pmatrix}_t\right\| \tag{$\mathbf{C_7}$}$$

4. Ainsi que des conditions de croissance (*cf* [AB02]) qui sont trivialement respectées dans notre cas du fait de la compacité de $\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^{|\mathcal{A}^\star|}$.

Avec ces quatre conditions, on peut alors énoncer le théorème 5.2.1.

Théorème 5.2.1 (Existence des diffusions sous contraintes avec saut)
En supposant que les conditions $(\mathbf{C_4})$, $(\mathbf{C_5})$, $(\mathbf{C_6})$ *et* $(\mathbf{C_7})$ *soient satisfaites, on a l'existence et l'unicité d'un processus couplé* $(\mathbb{P}_t;\mathrm{X}_t)_{t\geqslant0}$ *vérifiant l'équation intégrale* $(\mathbf{E}-\mathbf{6})$.

Preuve : Voir annexe D □

Autrement dit, si les conditions précédentes sont satisfaites, l'équation intégrale est satisfaite pour un processus et cette équation est bien posée.

5.3 Sauts dans l'espace des forêts

Nous allons donner dans ce paragraphe les règles formelles qui vont permettre d'implémenter un parcours de l'espace des forêts, parcours dépendant d'une matrice de transition Q homogène en temps. Par définition de l'indicateur X_t au temps t de la section précédente, on constate qu'il

est équivalent de donner les règles de transition Q pour $(\mathbb{P}, \mathrm{X})$ ou pour $(\mathbb{P}, \mathcal{F})$. Il s'agit dans ce paragraphe de permettre de faire évoluer l'ensemble des features $\mathcal{F}$. La dynamique de cet ensemble sera de type markovien, et l'ensemble des features au temps t sera donc noté $\mathcal{F}_t$. Il y a une double raison à la notation $\mathcal{F}_t$:

- c'est (bien entendu) l'ensemble des Features au temps t.
- cet ensemble peut également être considéré comme une « forêt » d'arbres binaires, définis comme dans le chapitre 2.

Dès lors, nous utiliserons autant le mot « forêt » que l'expression features pour désigner $\mathcal{F}_t$ et les instants de sauts seront les instants où la forêt sera modifiée. Ils seront notés génériquement t_s alors que $\mathcal{F}_{t_s}$ désignera l'état de la forêt avant le saut et $\mathcal{F}_{t_s+\mathrm{d}t}$ la forêt obtenue après la transition. Ces sauts vont alors nous permettre d'explorer totalement l'ensemble des forêts possibles. Cet ensemble de forêts $\mathcal{A}^\star$ est très grand (même s'il est fini puisque la répétition de lettres élémentaires est impossible dans un même mot) et il faudra établir une façon efficace de parcourir $\mathcal{A}^\star$ pour le but qui nous anime : trouver un ensemble d'arbres restreints permettant d'obtenir autant que possible des résultats de classification corrects.

Il s'agit donc, dans un premier temps, de préciser quelles peuvent être les différentes transitions de $\mathcal{F}_{t_s}$ à $\mathcal{F}_{t_s+\mathrm{d}t}$ ainsi que les probabilités associées à ces transitions markoviennes, ce sera précisément l'objet de ce 5.3.

Par la suite, il faudra également formaliser les nouvelles équations d'évolution du couple $(\mathcal{F}_t, \mathbb{P}_t)$ et ces équations d'évolutions seront très précisément décrites dans le 5.4.

Nous supposerons dans ce paragraphe et dans tout ce qui suivra que la concaténation de deux mots $(m_1; m_2) \longmapsto m_1 :: m_2$ est définie, ce qui sera le cas ultérieurement dans les différentes applications de notre algorithme sur l'analyse des données proposées.

5.4 Transitions de $\mathcal{F}_{t_s}$ à $\mathcal{F}_{t_s+\mathrm{d}t}$

En supposant que l'instant t_s soit un instant de saut d'un ensemble de features à un autre, nous allons détailler les différentes règles pour construire le nouvel ensemble $\mathcal{F}_{t_s+\mathrm{d}t}$ à partir de l'ancien $\mathcal{F}_{t_s}$. On souhaite pouvoir

- enrichir la forêt $\mathcal{F}$ de nouveaux arbres afin d'obtenir un pouvoir plus discriminant de ces nouveaux features sur l'ensemble des classes de données.
- enlever d'autres arbres de $\mathcal{F}$ qui sont eux moins efficaces pour classer correctement les données.

Pour des raisons qui seront précisées dans le paragraphe suivant, le principe fondamental qui devra régir la dynamique de création / suppression d'arbres est la réversibilité en « un coup » (ou réversibilité faible). Plus précisément, si l'on désigne par t_{s_1} et t_{s_2} deux instants consécutifs de saut, il faut imposer que si la transition

$$\mathcal{F}_{t_{s_1}} \xrightarrow[\text{Saut à l'instant } t_{s_1}]{} \mathcal{F}_{t_{s_2}}$$

est possible,

alors

$$\mathcal{F}_{t_{s_2}} \xrightarrow[\text{Saut à l'instant } t_{s_2}]{} \mathcal{F}_{t_{s_1}}$$

l'est également. Comme les transitions précédentes ne sont pas déterministes, ces conditions se réécrivent plutôt en

$$\mathcal{F}_{t_{s_1}+\mathrm{d}t} = \mathcal{F}_{t_{s_2}} \Longrightarrow \mathrm{P}(\mathcal{F}_{t_{s_2}+\mathrm{d}t} = \mathcal{F}_{t_{s_1}}) > 0$$

Cette condition assure en effet que la dynamique ne peut pas rester piégée dans certaines zones d'exploration des forêts. Mais nous lui donnerons une justification plus théorique dans le paragraphe suivant.

Les règles qui vont permettre de construire ces nouvelles forêts sont détaillées dans les paragraphes qui suivent.

5.4.1 Création de nouveaux arbres

À partir de deux arbres $\mathcal{A}_1$ et $\mathcal{A}_2$ de $\mathcal{F}_{t_{s_1}}$, on souhaite créer un néologisme qui fera partie de $\mathcal{F}_{t_{s_1}+\mathrm{d}t}$. Sa construction dépend du sens que l'on donne à la concaténation de deux mots $m_1 :: m_2$. Mais supposant cette concaténation définie, on peut donner la première règle de construction

$$\mathrm{T}_g\left(\begin{array}{ccc} m_1 & ; & m_2 \\ \mathcal{A}_1.\mathrm{g} \quad \mathcal{A}_1.\mathrm{d} & & \mathcal{A}_2.\mathrm{g} \quad \mathcal{A}_2.\mathrm{d} \end{array}\right) = \begin{array}{c} m_1 :: m_2 \\ m_1 \qquad m_2 \\ \mathcal{A}_1.\mathrm{g} \quad \mathcal{A}_1.\mathrm{d} \quad \mathcal{A}_2.\mathrm{g} \quad \mathcal{A}_2.\mathrm{d} \end{array}$$

La construction est donc « montante », mais l'évaluation de l'arbre sur un élément de la base de données n'est pas descendante, il suffit d'examiner le noeud principal de l'arbre formé pour évaluer le résultat d'un arbre sur les données. C'est un petit peu différent de ce qui est fait dans [FG01] puisque la construction est aussi montante, mais il est ensuite nécessaire de parcourir l'arbre en profondeur pour évaluer sa valeur sur un élément de la base de données. Il n'y a pas ici, dans notre cas, de parcours descendant des arbres, conditionné aux valeurs évaluées au fur et à mesure dans l'arbre.

Si on note donc T_g l'opération de $\mathcal{A}^\star \times \mathcal{A}^\star$ dans $\mathcal{A}^\star$ de construction d'un arbre greffé à partir de deux sous-arbres, et si l'on note $\widetilde{\mathrm{T}_g}$ l'opération qui lui correspond sur les forêts, l'opération sur la forêt se traduit en fait par

$$\mathcal{F}_{t_{s_1}+\mathrm{d}t} = \mathcal{F}_{t_{s_1}} \cup \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2) = \widetilde{\mathrm{T}_g}(\mathcal{F}_{t_{s_1}};\mathcal{A}_1;\mathcal{A}_2) \qquad (\mathbf{T_g})$$

avec

$$\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2) \notin \mathcal{F}_{t_{s_1}}$$

> Remarquons ici qu'il faut prendre garde au fait qu'on impose absolument le fait que l'arbre qui est issu de la greffe $(\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2))$ ne doit pas appartenir à $\mathcal{F}_{t_s}$. Ceci se justifiera ultérieurement. La raison principale est que si cette condition n'est pas imposée, il n'est alors pas possible d'expliciter clairement une règle de « retour » (inversion de cette greffe).

Nous définissons donc les règles de greffe de deux arbres $\mathcal{A}_1$ et $\mathcal{A}_2$ selon les trois modalités différentes que nous allons voir tour à tour.

5.4.1.1 Greffe de deux arbres sans coupe

La première règle est standard et correspond à une greffe sans modifier le contenu de la forêt précédant le saut. Cette sous-règle est en fait identique à l'opération $(\mathbf{T_g})$ précédente et sera notée de la même façon. Ainsi :

$$\widetilde{\mathrm{T}_g}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2) = \mathcal{F}_{t_s} \cup \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$$

5.4.1.2 Greffe de deux arbres avec suppression d'un des fils

La deuxième règle permet de modifier le contenu de la forêt en supprimant l'un des deux fils de l'arbre greffé obtenu. On autorise donc la greffe à partir de deux arbres $\mathcal{A}_1$ et $\mathcal{A}_2$ tout en permettant la suppression d'un des deux fils, on obtient les deux règles ($\mathbf{T_{g;sg}}$) (Greffe-Suppression fils Gauche) et ($\mathbf{T_{g;sd}}$) (Greffe-Suppression fils Droit) qui se résument en :

$$\widetilde{\mathrm{T}_{g;sg}}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2) = \mathcal{F}_{t_s} \cup \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2) \setminus \mathcal{A}_1$$

et

$$\widetilde{\mathrm{T}_{g;sd}}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2) = \mathcal{F}_{t_s} \cup \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2) \setminus \mathcal{A}_2$$

5.4.1.3 Greffe de deux arbres avec coupe des deux fils

Enfin, la dernière opération de greffe s'inspire de la greffe avec coupe d'un fils. On permet de greffer deux arbres dans $\mathcal{F}_{t_s+\mathrm{d}t}$ tout en supprimant les deux fils, ce qui permet de définir ($\mathbf{T_{g;sgd}}$) (greffe-suppressions fils gauche fils droit) :

$$\widetilde{\mathrm{T}_{g;sgf}}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2) = \mathcal{F}_{t_s} \cup \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2) \setminus (\mathcal{A}_1 \cup \mathcal{A}_2)$$

5.4.2 Coupe d'un arbre

La coupe d'un arbre est relativement simple ; il s'agit, étant donné un arbre $\mathcal{A}$, composé ou non, de le supprimer et d'ajouter dans $\mathcal{F}_{t_{s_1}+\mathrm{d}t}$ ses deux fils gauche et droit. Cette opération se traduit par la règle :

$$\mathrm{T}_c\left(\begin{array}{c} m \\ \widehat{\mathcal{A}.\mathrm{g}\quad \mathcal{A}.\mathrm{d}} \end{array}\right) = \mathcal{A}.g \cup \mathcal{A}.d$$

soit

$$\mathcal{F}_{t_{s_1}+\mathrm{d}t} = \left(\mathcal{F}_{t_{s_1}} \setminus \mathcal{A}\right) \cup \mathrm{T}_c(\mathcal{A}) = \widetilde{\mathrm{T}_c}(\mathcal{F}_{t_{s_1}};\mathcal{A}) \qquad (\mathbf{T_c})$$

en conservant la notation T_c pour l'opération sur les arbres et $\widetilde{\mathrm{T}_c}$ pour l'application sur les forêts.

On notera que si un (ou les deux) fils de $\mathcal{A}$ sont déjà présents dans $\mathcal{A}_{t_{s_1}}$, l'union précédente est « blanche » et T_c se traduit uniquement par la suppression de $\mathcal{A}$ de $\mathcal{F}_{t_{s_1}}$.

5.4.3 Renaissance d'arbres initiaux

On autorise enfin un ajout artificiel d'un arbre de $\mathcal{F}_0$, forêt initiale, qui n'est plus présent dans $\mathcal{F}_{t_{s_1}}$, si $\mathcal{A}$ est donc dans $\mathcal{F}_0 \setminus \mathcal{F}_{t_{s_1}}$:

$$\mathcal{F}_{t_{s_1}+\mathrm{d}t} = \mathcal{F}_{t_{s_1}} \cup \mathcal{A} = \widetilde{\mathrm{T}_i}(\mathcal{F}_{t_{s_1}};\mathcal{A}) \qquad (\mathbf{T_i})$$

5.4.4 Parcours de $\mathcal{F}^\star$

Avec ces trois règles, il est immédiat de voir que l'on peut construire toutes les forêts possibles. En effet, donnons-nous $\mathcal{F}$ une forêt quelconque, la règle ($\mathbf{T_g}$) permet de construire tous les arbres binaires (nécessairement complets dans notre définition) de longueur donnée : il s'agit d'itérer le processus de construction *via* ($\mathbf{T_g}$) un nombre suffisant de fois. Et la règle ($\mathbf{T_c}$) permet de supprimer tous les arbres indésirables de $\mathcal{F}$ qui auraient été formés lors de l'application de ($\mathbf{T_g}$). Suite à ces nouvelles définitions, on peut affirmer la propriété suivante.

Propriété 5.4.1 (Réversibilité faible des règles ($\mathbf{T_g}$)-($\mathbf{T_{g;sg}}$)-($\mathbf{T_{g;sd}}$)- ($\mathbf{T_{g;sgd}}$)-($\mathbf{T_c}$)-($\mathbf{T_i}$)) *Toute opération $\mathcal{F}_{t_s} \longrightarrow \mathcal{F}_{t_s+\mathrm{d}t}$ obtenue à partir des 6 règles précédentes peut être inversée également à partir de ces 6 mêmes règles en un coup.*

Preuve : Pour démontrer ce résultat, il suffit d'énumérer tous les cas possibles. Donnons-nous donc un instant de saut t_s.

- Si l'on effectue une greffe *via* ($\mathbf{T_g}$), ($\mathbf{T_{g;sg}}$), ($\mathbf{T_{g;sd}}$) ou ($\mathbf{T_{g;sgd}}$), sur deux arbres $\mathcal{A}_1$ et $\mathcal{A}_2$: l'arbre formé n'appartient pas à $\mathcal{F}_{t_s}$, l'opération « réciproque » correspond à la coupe de l'arbre $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$. En effet, que l'on décide ou non de supprimer les fils $\mathcal{A}_1$ et $\mathcal{A}_2$ lors du saut en t_s, la coupe de l'arbre $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$ reconstruit les deux fils $\mathcal{A}_1$ et $\mathcal{A}_2$. Les répétitions n'étant pas comptées dans l'énumération des arbres de la forêt, on a bien :
$$\mathrm{T}_c\left[\mathrm{T}_g\left(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2\right);\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)\right]=\mathcal{F}_{t_s}$$
- Si l'on effectue une coupe d'un arbre $\mathcal{A}$, différents cas se présentent :
  - L'arbre $\mathcal{A}$ est « élémentaire », c'est-à-dire $\mathcal{A}$ est réduit à son noeud principal et ses deux fils sont alors vides (ou bien encore $\mathcal{A}$ est de profondeur 1). Dans ce cas, la coupe de l'arbre correspond purement et simplement à la suppression de $\mathcal{A}$ de $\mathcal{F}_{t_s}$. Mais l'arbre $\mathcal{A}$ étant élémentaire, il est présent dans $\mathcal{F}_0$. Ainsi, on peut reconstituer $\mathcal{F}_{t_s}$ grâce à ($\mathbf{T_i}$) :
$$\forall\mathcal{A}\in\mathcal{F}_{t_s}\quad |\mathcal{A}|=1\qquad \mathrm{T}_i\left[\mathrm{T}_c\left(\mathcal{F}_{t_s};\mathcal{A}\right);\mathcal{A}\right]=\mathcal{F}_{t_s}$$
  - L'arbre $\mathcal{A}$ est composé et ses deux fils appartiennent à $\mathcal{F}_{t_s}$. Dans ce cas, l'opération T_g permet directement d'annuler la coupe de cet arbre :
$$\forall \mathrm{A}\in\mathcal{F}_{t_s}\quad|\quad \mathcal{A}=\begin{array}{c}\mathcal{A}.\mathrm{m}\\ \mathcal{A}.\mathrm{g}\quad\mathcal{A}.\mathrm{d}\end{array}\qquad\text{et}\quad \mathcal{A}.g\in\mathcal{F}_{t_s}\quad\text{et}\quad \mathcal{A}.f\in\mathcal{F}_{t_s}$$
    on a alors $\mathrm{T}_g(\mathrm{T}_c(\mathcal{F}_{t_s};\mathcal{A});\mathcal{A}.g;\mathcal{A}.d)=\mathcal{F}_{t_s}$
  - L'arbre est composé d'un fils appartenant à $\mathcal{F}_{t_s}$ et d'un fils n'y appartenant pas, cela s'écrit alors :
$$\mathcal{A}\in\mathcal{F}_{t_s}\quad|\quad \mathcal{A}=\begin{array}{c}\mathcal{A}.\mathrm{m}\\ \mathcal{A}.\mathrm{g}\quad\mathcal{A}.\mathrm{d}\end{array}\qquad\text{et}\quad \mathcal{A}.g\notin\mathcal{F}_{t_s}\quad\text{et}\quad \mathcal{A}.d\in\mathcal{F}_{t_s}$$
    Dans ce cas $\mathrm{T}_{g;sg}\left(\mathrm{T}_c\left(\mathcal{F}_{t_s};\mathcal{A}\right);\mathcal{A}.g;\mathcal{A}.d\right)=\mathcal{F}_{t_s}$
    De la même façon, si
$$\mathcal{A}\in\mathcal{F}_{t_s}\quad|\quad \mathcal{A}=\begin{array}{c}\mathcal{A}.\mathrm{m}\\ \mathcal{A}.\mathrm{g}\quad\mathcal{A}.\mathrm{d}\end{array}\qquad\text{et}\quad \mathcal{A}.g\in\mathcal{F}_{t_s}\quad\text{et}\quad \mathcal{A}.d\notin\mathcal{F}_{t_s}$$
    Dans ce cas $\mathrm{T}_{g;sd}\left(\mathrm{T}_c\left(\mathcal{F}_{t_s};\mathcal{A}\right);\mathcal{A}.g;\mathcal{A}.d\right)=\mathcal{F}_{t_s}$
  - Enfin, si l'arbre est composé de deux fils qui n'appartiennent pas à $\mathcal{F}_{t_s}$, on a
$$\mathcal{A}\in\mathcal{F}_{t_s}\quad|\quad \mathcal{A}=\begin{array}{c}\mathcal{A}.\mathrm{m}\\ \mathcal{A}.\mathrm{g}\quad\mathcal{A}.\mathrm{d}\end{array}\qquad\text{et}\quad \mathcal{A}.g\notin\mathcal{F}_{t_s}\quad\text{et}\quad \mathcal{A}.d\notin\mathcal{F}_{t_s}$$
    puis $\mathrm{T}_{g;sgd}\left[\mathrm{T}_c\left(\mathcal{F}_{t_s};\mathcal{A}\right);\mathcal{A}.g;\mathcal{A}.d\right]=\mathcal{F}_{t_s}$

- Si l'on effectue un ajout d'un arbre élémentaire $\mathcal{A}$ de $\mathcal{F}_0$

$$|\mathcal{A}| = 1$$

dans $\mathcal{F}_{t_s}$, on constate immédiatement que la coupe de ce même arbre par T_c permet d'inverser l'opération :

$$\mathrm{T}_c\left(\mathrm{T}_i\left(\mathcal{F}_{t_s}; \mathcal{A}\right)\right) = \mathcal{F}_{t_s}$$

Ainsi, quelle que soit la règle choisie pour effectuer le saut à l'instant t_s, il est possible en utilisant $(\mathbf{T_g})$, $(\mathbf{T_{g;sg}})$, $(\mathbf{T_{g;sd}})$, $(\mathbf{T_{g;sgd}})$, $(\mathbf{T_c})$,$(\mathbf{T_i})$ ou $(\mathbf{T_{id}})$ d'inverser cette règle. □

Cette propriété assure donc que les règles de transition ainsi définies permettent d'explorer l'ensemble des forêts $\mathcal{F}^\star$ de façon complètement réversible « en un coup ».

5.4.5 Non réversibilité faible des règles $(\mathbf{T_g})$-$(\mathbf{T_c})$-$(\mathbf{T_i})$

Nous pouvons justifier l'emploi des règles $(\mathbf{T_{g;sg}})$, $(\mathbf{T_{g;sd}})$ et $(\mathbf{T_{g;sgd}})$ qui paraissent artificielles en remarquant qu'elles sont absolument nécessaires à la dynamique type « Métropolis » ([MRR+53]). Il y a en effet un problème de réversibilité faible si on laisse en l'état les seules trois règles $(\mathbf{T_g})$-$(\mathbf{T_c})$-$(\mathbf{T_i})$ précédentes pour construire $\mathcal{F}_{t_{s_1}+\mathrm{d}t}$ à partir de $\mathcal{F}_{t_{s_1}}$. La proposition suivante explique en quoi la donnée des trois règles précédentes n'est pas « à sauts réversibles » en un coup.

Propriété 5.4.2 (Non réversibilité faible des règles $(\mathbf{T_g})$-$(\mathbf{T_c})$-$(\mathbf{T_i})$ « en un coup »)
On peut trouver une forêt $\mathcal{F}_{t_s}$ et un saut construisant $\mathcal{F}_{t_s+\mathrm{d}t}$ tels qu'il est impossible de reformer $\mathcal{F}_{t_s}$ à partir de $\mathcal{F}_{t_s+\mathrm{d}t}$ via $(\mathbf{T_g})$-$(\mathbf{T_c})$-$(\mathbf{T_i})$ « en un coup ».

Preuve : Pour démontrer ce résultat, on peut considérer n'importe quelle forêt $\mathcal{F}_{t_s}$, même correspondant à un cas pathologique vis-à-vis des forêts construites lors de nos algorithmes d'apprentissage ultérieurs. Considérons le cas d'une forêt $\mathcal{F}_{t_s}$ telle qu'elle ne contienne pas le fils gauche d'un de ses arbres $\mathcal{A}$ de profondeur maximale. On a alors

$$\mathcal{A} = \quad \mathcal{A}.\mathrm{m} \quad \in \mathcal{F}_{t_s}$$
$$\mathcal{A}.\mathrm{g} \quad \mathcal{A}.\mathrm{d}$$

tandis que $$\mathcal{A}.g \notin \mathcal{F}_{t_s} \quad \text{et} \quad \mathcal{A}.d \in \mathcal{F}_{t_s}$$

Si l'on choisit d'appliquer $(\mathbf{T_c})$ à $\mathcal{F}_{t_s}$ et à $\mathcal{A}$, on obtient dans $\mathcal{F}_{t_s+\mathrm{d}t}$:

$$\mathcal{A}.g \in \mathcal{F}_{t_s+\mathrm{d}t} \quad \text{et} \quad \mathcal{A}.d \in \mathcal{F}_{t_s+\mathrm{d}t} \quad \text{et} \quad \mathcal{A} \notin \mathcal{F}_{t_s+\mathrm{d}t}$$

Il est alors impossible de revenir en arrière en un coup et de faire donc $\mathcal{F}_{t_s+\mathrm{d}t} \longrightarrow \mathcal{F}_{t_s}$ en utilisant uniquement une des trois règles précédentes puisque pour recréer $\mathcal{F}_{t_s}$, il est nécessaire de

- reformer $\mathcal{A}$, soit par une greffe, soit par une coupe d'un arbre dont $\mathcal{A}$ serait un fils gauche ou droit.
- supprimer le fils gauche $\mathcal{A}.g$.

Pour obtenir à nouveau $\mathcal{A}$, on ne peut utiliser que la règle $(\mathbf{T_g})$ puisque cet arbre est de profondeur maximale et n'est donc pas le sous-arbre d'un autre arbre de $\mathcal{F}_{t_s}$. Mais la règle $(\mathbf{T_g})$ ne permet pas de supprimer $\mathcal{A}.g$ de $\mathcal{F}_{t_s+\mathrm{d}t}$.

Pour supprimer $\mathcal{A}.g$, on ne peut utiliser que $(\mathbf{T_c})$, mais cette règle ne permet pas, quant à elle, de construire le père $\mathcal{A}$.

Il est donc impossible en appliquant une seule des trois règles d'effectuer la transition $\mathcal{F}_{t_s+\mathrm{d}t} \longrightarrow \mathcal{F}_{t_s}$. □

La construction du contre-exemple précédent permet donc de montrer la nécessité de $(\mathbf{T_g})$, $(\mathbf{T_{g;sg}})$, $(\mathbf{T_{g;sd}})$ dans le parcours de $\mathcal{F}^\star$ précédent.

Pour pouvoir conclure la description totale de la dynamique de notre algorithme, il s'agit enfin de détailler les probabilités de transition données par Q définie dans le 5.2, mais non explicitée jusqu'ici.

5.5 Probabilités des propositions de transitions de $\mathcal{F}_{t_s}$ à $\mathcal{F}_{t_s+\mathrm{d}t}$

Dès lors que les règles possibles pour passer de $\mathcal{F}_{t_s}$ à $\mathcal{F}_{t_s+\mathrm{d}t}$ sont établies précisément, nous allons établir les probabilités de transition associées à chacune de ces règles. Ces probabilités de transition respecteront plusieurs propriétés décrites ci-après :

1. Le choix d'effectuer une des 6 règles $(\mathbf{T_g})$, $(\mathbf{T_{g;sg}})$, $(\mathbf{T_{g;sd}})$, $(\mathbf{T_{g;sgd}})$, $(\mathbf{T_c})$ ou$(\mathbf{T_i})$ ne dépend pas de l'instant du saut t_s et de la nature de la composition de $\mathcal{F}_{t_s}$.
2. Une fois la règle $(\mathbf{T})$ choisie pour effectuer le saut au temps t_s, la loi de probabilité pour proposer un saut de $\mathcal{F}_{t_s}$ à $\mathcal{F}_{t_s+\mathrm{d}t}$ peut être prise uniforme, pour l'instant, sur tous les sauts possibles respectant la règle $(\mathbf{T})$. Cependant, dans l'ensemble de nos expériences pratiques, nous avons choisis de privilégier certains arbres pour la greffe ou la coupe, ceci améliorant très nettement les résultats de nos tests.

On constate donc que le choix du saut à l'instant t_s comprend deux phases :

- le choix de la décision $(\mathbf{T})$ qui sera appelée étape $(\mathbf{R1})$
- le choix des arbres qui se verront appliquer cette règle $(\mathbf{T})$ qui sera appelée étape $(\mathbf{R2})$

Si l'on note $p_g, p_{g;sg}, p_{g;sd}, p_{g,sgd}, p_c$ et p_i les probabilités de choisir d'effectuer la règle associée à chaque instant de saut t_s lors de l'étape $(\mathbf{R1})$, on doit donc avoir la condition

$$p_g + p_{g;sg} + p_{g;sd} + p_{g,sgd} + p_c + p_i = 1$$

De plus, ces probabilités sont indépendantes de t_s et $\mathcal{F}_{t_s}$, ce sont donc des constantes fixées par l'utilisateur à l'initialisation de l'algorithme de construction des forêts.

Enfin, on peut donner explicitement la probabilité d'effectuer tel ou tel saut à l'instant t_s lors de l'étape $(\mathbf{R2})$. Pour une meilleure lisibilité, nous renvoyons à l'annexe C pour les détails des formules de transitions correspondant à $(\mathbf{R2})$.

Nous retiendrons cependant que les règles de proposition des transitions vérifient les deux règles qui suivent.

- Lorsque $(\mathbf{R_1})$ choisit comme transition une greffe, on proposera *via* $(\mathbf{R2})$ plus souvent des greffes à partir d'arbres qui ont une probabilité $\mathbb{P}$ importante à l'instant du saut.
- Lorsque $(\mathbf{R_1})$ choisit comme transition une coupe, on proposera plus souvent *via* $(\mathbf{R2})$ une coupe à partir d'arbres qui ont une probabilité $\mathbb{P}$ faible à l'instant du saut.

5.6 Dynamique markovienne des sauts

L'approche que nous allons adopter va consister à minimiser une énergie dépendant de :

1. l'ensemble des features $\mathcal{F}_t$ sélectionnés
2. une probabilité de tirage d'un arbre $\mathbb{P}$ sur cet ensemble $\mathcal{F}_t$.

L'expression de cette énergie est de la forme suivante :

$$\boxed{\mathcal{E}(\mathcal{F};\mathbb{P}) = \mathcal{E}_{\mathrm{L}g}(\mathcal{F}) + \mathcal{E}_{\rho}(\mathcal{F}) + \mathcal{E}_{err}(\mathcal{F};\mathbb{P})}$$

La mesure de Gibbs invariante associée à cette énergie s'exprime alors en

$$\mu(\mathcal{F};\mathbb{P}) = \frac{e^{-\mathcal{E}(\mathcal{F};\mathbb{P})}}{\mathrm{Z}} \tag{5.28}$$

Même si cette mesure μ ne sera pas destinée à effectuer un algorithme de Recuit-Simulé, elle nous sera utile par la suite pour caractériser l'évolution des processus que nous construirons. Nous verrons plutôt cette mesure comme une loi invariante d'un processus stochastique $(\mathcal{F}_t;\mathbb{P}_t)$ que nous allons chercher à simuler, cette approche est donc comparable à [GM94].

Nous nous intéressons dans ce paragraphe aux sauts effectués dans l'algorithme. Ces sauts correspondent en réalité à la modification de $\mathcal{F}_t$. Afin de rechercher un comportement de loi invariante μ, on peut s'inspirer des résultats classiques sur les algorithmes de Métropolis pour décrire de façon très précise les probabilités de sauts.

5.6.1 Acceptation des sauts par un algorithme de Métropolis-Hastings

Afin de simuler les sauts, nous utilisons une dynamique markovienne, c'est-à-dire, les sauts sont représentés par un processus de Markov avec une probabilité de transition R(.,.). Si $\mathcal{F}_{t_s}$ et $\mathbb{P}_{t_s}$ sont les états du processus $(\mathcal{F};\mathbb{P})$ à l'instant d'un saut et si $\mathcal{F}_{t_s+\mathrm{d}t}$ et $\mathbb{P}_{t_s+\mathrm{d}t}$ représentent l'état du processus après ce saut, l'équation d'équilibre de l'évolution régie par une dynamique Métropolis-Hastings ([MRR$^+$53], [Has70]) est

$$\mu\left(\mathcal{F}_{t_s};\mathbb{P}_{t_s}\right)\mathrm{R}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right] = \mu\left(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t}\right)\mathrm{R}\left[\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix}\right]$$

Par ailleurs, on impose que la probabilité de transition s'écrive :

$$\mathrm{R}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right] = \underbrace{\mathrm{P}\left[\mathcal{F}_{t_s+\mathrm{d}t}|\mathcal{F}_{t_s}\right]}_{\text{probabilité issue de (\textbf{R1}) et (\textbf{R2})}}\mathrm{Q}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right]$$

où P est la probabilité conditionnelle du paragraphe précédent et Q est le taux d'acceptation du saut correspondant. Ainsi, le taux d'acceptation Q vérifie l'égalité

$$\mathrm{Q}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right] = \mathrm{Q}\left[\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix}\right]\underbrace{\frac{\mathrm{P}\left[\mathcal{F}_{t_s}|\mathcal{F}_{t_s+\mathrm{d}t}\right]}{\mathrm{P}\left[\mathcal{F}_{t_s+\mathrm{d}t}|\mathcal{F}_{t_s}\right]}}_{\text{terme issu de (\textbf{R1}) et (\textbf{R2})}}\frac{\mu\left(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t}\right)}{\mu\left(\mathcal{F}_{t_s};\mathbb{P}_{t_s}\right)}$$

La méthode classique pour accepter de tels sauts revient alors à poser :

$$\mathrm{Q}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right] = \mathrm{Min}\left[1; \underbrace{\frac{\mathrm{P}\left[\mathcal{F}_{t_s+\mathrm{d}t}|\mathcal{F}_{t_s}\right]}{\mathrm{P}\left[\mathcal{F}_{t_s}|\mathcal{F}_{t_s+\mathrm{d}t}\right]}\frac{\mu\left(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t}\right)}{\mu\left(\mathcal{F}_{t_s};\mathbb{P}_{t_s}\right)}}_{=\tau\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right]}\right]$$

De ce fait, la simulation lors d'un saut se déroule ainsi :

1. Étant données une forêt $\mathcal{F}_{t_s}$ et une probabilité $\mathbb{P}_{t_s}$ sur cette forêt, tirer selon $\mathrm{P}\left[.|\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix}\right]$ une nouvelle forêt possible $\mathcal{F}_{t_s+\mathrm{d}t}$ avec une nouvelle probabilité définie par $\mathbb{P}_{t_s+\mathrm{d}t}$.
2. Calculer le rapport
$$\tau = \frac{\mathrm{P}\left[\mathcal{F}_{t_s+\mathrm{d}t}|\mathcal{F}_{t_s}\right]}{\mathrm{P}\left[\mathcal{F}_{t_s}|\mathcal{F}_{t_s+\mathrm{d}t}\right]}\frac{\mu\left(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t}\right)}{\mu\left(\mathcal{F}_{t_s};\mathbb{P}_{t_s}\right)}$$
3. - Si $\tau > 1$, accepter le saut, c'est-à-dire valider $(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t})$ comme nouvel état.
   - Si $\tau \leqslant 1$, accepter le saut avec la probabilité τ.

On démontre classiquement qu'étant données de telles règles de transition, la loi μ est bien une mesure invariante associée à la chaîne de Markov de probabilité de transition $\mathrm{R}[.;.]$. Il suffit en effet pour cela de voir que l'on a :

$$\mu(\mathrm{X})\mathrm{R}[\mathrm{X};\mathrm{Y}] = \mu(\mathrm{Y})\mathrm{R}[\mathrm{Y};\mathrm{X}]$$

On peut de plus justifier de la nécessité de la réversibilité « en un coup » de la chaîne de Markov basée sur la probabilité de transition $\mathrm{P}[.|.]$. On constate en effet que s'il est impossible de revenir en arrière en un coup, la formule de stationnarité

$$\mu\left(\mathcal{F}_{t_s};\mathbb{P}_{t_s}\right)\mathrm{R}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right] = \mu\left(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t}\right)\mathrm{R}\left[\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix}\right]$$

impose que

$$\mathrm{R}\left[\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix}\right] = 0 \Longleftrightarrow \mathrm{R}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix};\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right] = 0$$

De l'expression de R, on tire alors que

$$\mathrm{P}\left[\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix}\Big|\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\right] = 0 \Longleftrightarrow \mathrm{P}\left[\begin{pmatrix}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{pmatrix}\Big|\begin{pmatrix}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{pmatrix}\right] = 0$$

5.6.2 Détermination de $\mathbb{P}_{t_s+\mathrm{d}t}$

Dans le paragraphe précédent, il n'a pas été précisé ce que valait exactement $\mathbb{P}_{t_s+\mathrm{d}t}$ en fonction du saut qui a été choisi depuis $(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$. En réalité, il est nécessaire de donner une définition à cette nouvelle probabilité après le saut puisque son expression est nécessaire dans le calcul de τ, quantité conditionnant l'acceptation du saut.

5.6.2.1 Cas de $(\mathbf{T_g})$

Si l'on suppose que le saut est obtenu *via* une greffe sans coupe de deux arbres $\mathcal{A}_1$ et $\mathcal{A}_2$, il s'agit donc de définir $\mathbb{P}_{t_s+\mathrm{d}t}$ notamment en $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$. Nous avons choisi d'imposer que

$$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)\right)=0$$

tandis que $$\forall \mathcal{A}\in\mathcal{F}_{t_s} \qquad \mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}\right)=\mathbb{P}_{t_s}\left(\mathcal{A}\right)$$

5.6.2.2 Cas de $(\mathbf{T_{g;sg}})$ ou $(\mathbf{T_{g;sd}})$

Ici, nous gérons le saut sur $\mathbb{P}$ de façon quasi-similaire, sauf qu'il faut en plus renormaliser $\mathbb{P}_{t_s}$.

- Si on applique $(\mathbf{T_{g;sg}})$ sur $\mathcal{F}_{t_s}$ avec le couple $(\mathcal{A}_1;\mathcal{A}_2)$, alors

$$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathrm{T}_{g;sg}(\mathcal{A}_1;\mathcal{A}_2)\right)=0$$

  puis $$\forall \mathcal{A}\in\mathcal{F}_{t_s+\mathrm{d}t}\cap\mathcal{F}_{t_s} \qquad \mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}\right)=\frac{\mathbb{P}_{t_s}\left(\mathcal{A}\right)}{1-\mathbb{P}_{t_s}\left(\mathcal{A}_1\right)}$$

  tandis que $$\mathcal{A}_1\notin\mathcal{F}_{t_s+\mathrm{d}t}$$

- Si on applique $(\mathbf{T_{g;sd}})$ sur $\mathcal{F}_{t_s}$ avec le couple $(\mathcal{A}_1;\mathcal{A}_2)$, alors

$$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathrm{T}_{g;sg}(\mathcal{A}_1;\mathcal{A}_2)\right)=0$$

  puis $$\forall \mathcal{A}\in\mathcal{F}_{t_s+\mathrm{d}t}\cap\mathcal{F}_{t_s} \qquad \mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}\right)=\frac{\mathbb{P}_{t_s}\left(\mathcal{A}\right)}{1-\mathbb{P}_{t_s}\left(\mathcal{A}_2\right)}$$

  tandis que $$\mathcal{A}_2\notin\mathcal{F}_{t_s+\mathrm{d}t}$$

5.6.2.3 Cas de $(\mathbf{T_{g;sgd}})$

Là encore, il suffit de renormaliser les probabilités des arbres communs aux forêts $\mathcal{F}_{t_s+\mathrm{d}t}$ et $\mathcal{F}_{t_s}$ tout en annulant la probabilité du nouvel arbre greffé issu de $\mathcal{A}_1$ et $\mathcal{A}_2$.

Donc $$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathrm{T}_{g;sg}(\mathcal{A}_1;\mathcal{A}_2)\right)=0$$

puis $$\forall \mathcal{A}\in\mathcal{F}_{t_s+\mathrm{d}t}\cap\mathcal{F}_{t_s} \qquad \mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}\right)=\frac{\mathbb{P}_{t_s}\left(\mathcal{A}\right)}{1-\mathbb{P}_{t_s}(\mathcal{A}_1)-\mathbb{P}_{t_s}(\mathcal{A}_2)}$$

tandis que $$\mathcal{A}_1\notin\mathcal{F}_{t_s+\mathrm{d}t} \quad \text{et} \quad \mathcal{A}_2\notin\mathcal{F}_{t_s+\mathrm{d}t}$$

5.6.2.4 Cas de $(\mathbf{T_c})$

Si l'on décide de couper $\mathcal{A}_0$ pour obtenir $\mathcal{F}_{t_s+\mathrm{d}t}$, on impose que

$$\forall \mathcal{A}\in\mathcal{F}_{t_s+\mathrm{d}t}\cap\mathcal{F}_{t_s} \qquad \mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}\right)=\frac{\mathbb{P}_{t_s}\left(\mathcal{A}\right)}{1-\mathbb{P}_{t_s}(\mathcal{A}_0)}$$

Enfin, si $\mathcal{A}_0.g\notin\mathcal{F}_{t_s}$, on a $$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}_0.g\right)=0$$

tandis que dans le cas où le fils gauche est déjà dans $\mathcal{F}_{t_s}$, on renormalise simplement la probabilité de tirer cet arbre comme les autres de $\mathcal{F}_{t_s}$:

si $\mathcal{A}_0.g \in \mathcal{F}_{t_s}$ $$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}_0.g\right) = \frac{\mathbb{P}_{t_s}\left(\mathcal{A}_0.g\right)}{1-\mathbb{P}_{t_s}\left(\mathcal{A}_0\right)}$$

Le cas du fils droit se traite de manière exactement identique :

si $\mathcal{A}_0.d \notin \mathcal{F}_{t_s}$ $$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}_0.d\right) = 0$$

sinon $$\mathbb{P}_{t_s+\mathrm{d}t}\left(\mathcal{A}_0.d\right) = \frac{\mathbb{P}_{t_s}\left(\mathcal{A}_0.d\right)}{1-\mathbb{P}_{t_s}\left(\mathcal{A}_0\right)}$$

Et bien entendu $$\mathcal{A}_0 \notin \mathcal{F}_{t_s+\mathrm{d}t}$$

5.6.2.5 Cas de $(\mathbf{T_i})$

Si l'on décide de mettre à jour $\mathcal{F}_{t_s}$ en recréant un arbre élémentaire $\mathcal{A}_i$, la mise à jour de $\mathbb{P}_{t_s}$ se fait en

$$\forall \mathcal{A} \in \mathcal{F}_{t_s+\mathrm{d}t} \cap \mathcal{F}_{t_s} \qquad \mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{A}) = \mathbb{P}_{t_s}(\mathcal{A})$$

alors que $$\mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{A}_i) = 0$$

5.6.3 Calcul de τ

Pour pouvoir interpréter facilement l'acceptation ou le rejet d'un saut, on doit calculer précisément le rapport τ en fonction des données de l'algorithme au temps du saut.

$$\tau = \underbrace{\frac{\mathrm{P}\left[\mathcal{F}_{t_s+\mathrm{d}t}|\mathcal{F}_{t_s}\right]}{\mathrm{P}\left[\mathcal{F}_{t_s}|\mathcal{F}_{t_s+\mathrm{d}t}\right]}}_{=\tau_1} \underbrace{\frac{\mu\left(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t}\right)}{\mu\left(\mathcal{F}_{t_s};\mathbb{P}_{t_s}\right)}}_{=\tau_2}$$

En réalité, ce rapport est constitué de deux termes.

1. Le premier terme est le rapport des probabilités de proposer les sauts *via* P :

   $$\tau_1 = \frac{\mathrm{P}\left[\mathcal{F}_{t_s+\mathrm{d}t}|\mathcal{F}_{t_s}\right]}{\mathrm{P}\left[\mathcal{F}_{t_s}|\mathcal{F}_{t_s+\mathrm{d}t}\right]}$$

   Ce terme ne dépend que de l'état de $\mathcal{F}_{t_s}$ et $\mathcal{F}_{t_s+\mathrm{d}t}$. Il se calcule donc facilement grâce aux résultats de la section 5.4.

2. Le second terme est plus délicat à calculer puisqu'il fait intervenir la distribution stationnaire associée au champ de Gibbs défini en début de section 5.4 :

   $$\tau_2 = \frac{\mu\left(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t}\right)}{\mu\left(\mathcal{F}_{t_s};\mathbb{P}_{t_s}\right)} = e^{-\mathcal{E}(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t})+\mathcal{E}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})} = e^{-\Delta\mathcal{E}[(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\longrightarrow(\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t})]}$$

   Ainsi, ce terme est d'autant plus grand qu'il y a une diminution d'énergie $\Delta\mathcal{E}$ lors du saut. Autrement dit, plus l'énergie après le saut proposé est faible, plus le rapport τ est grand et donc plus il est probable d'accepter le saut. Cette dernière remarque va donc assurer le fait que les sauts permettront de faire diminuer le niveau énergétique du système.

5.6.3.1 Calcul de τ_1

τ_1 ne dépend donc que des règles de transition conditionnelles établies dans la section 5.3. Il s'agit d'énumérer en fonction des sauts proposés la valeur de ce rapport. Là encore, nous renvoyons à l'annexe C pour le détail et la justification des calculs.

5.6.3.2 Calcul de τ_2

Ce calcul est conditionné à l'expression qu'on impose sur $\mathcal{E}$, et notamment celle de $\mathcal{E}_{\mathrm{L}g}$ et $\mathcal{E}_\rho$. Ces deux termes dépendant du type de données que l'on traite, on ne s'intéressera dans un premier temps qu'au dernier terme, celui issu de $\mathcal{E}_{err}$. L'expression de ce terme de l'énergie est en fait celui qui a été étudié au chapitre 3, à savoir :

$$\mathcal{E}_{err}(\mathcal{F};\mathbb{P}) = \sum_{\omega\in\mathcal{F}^p} g(\omega)\mathbb{P}(\omega)$$

Les autres termes d'énergie seront définis dans le cadre de certaines bases de données ultérieurement, et pour l'instant, l'énergie totale du système est donc réduite à son terme d'erreur. Ce terme $\Delta\mathcal{E}_{err}$ vaut alors

$$\Delta\mathcal{E}_{err}\left[(\mathcal{F}_{t_s};\mathbb{P}_{t_s}) \longrightarrow (\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t})\right] = \sum_{\omega\in(\mathcal{F}_{t_s+\mathrm{d}t})^p} g(\omega)\mathbb{P}_{t_s+\mathrm{d}t}(\omega) - \sum_{\omega\in(\mathcal{F}_{t_s})^p} g(\omega)\mathbb{P}_{t_s}(\omega)$$

Le calcul dépend donc du saut qui a été choisi à l'instant t_s.

5.6.3.3 Cas d'une greffe ($\mathbf{T_g}$)

Dans ce cas précis, on a

$$\forall\mathcal{A}\in\mathcal{F}_{t_s} \qquad \mathbb{P}_{t_s}(\mathcal{A}) = \mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{A})$$

Le nouvel arbre étant de probabilité nulle sous $\mathbb{P}_{t_s+\mathrm{d}t}$, on en déduit immédiatement la propriété suivante.

Propriété 5.6.1 (Calcul de $\Delta\mathcal{E}_{err}$ pour ($\mathbf{T_g}$))
S'il est proposé une greffe ($\mathbf{T_g}$) *à l'instant du saut, alors*

$$\Delta\mathcal{E}_{err}\left[(\mathcal{F}_{t_s};\mathbb{P}_{t_s}) \longrightarrow (\mathcal{F}_{t_s+\mathrm{d}t};\mathbb{P}_{t_s+\mathrm{d}t})\right] = 0$$

5.6.3.4 Cas d'une greffe ($\mathbf{T_{g;sg}}$) ou ($\mathbf{T_{g;sd}}$)

Ce cas est quelque peu différent puisque la probabilité $\mathbb{P}_{t_s+\mathrm{d}t}$ est modifiée. Notons $\mathcal{A}$ le mot que l'on supprime, et $\mathcal{B}$ le mot qui est créé à la suite de la greffe, on a alors la nouvelle propriété :

Propriété 5.6.2 (Calcul de $\Delta\mathcal{E}_{err}$ pour ($\mathbf{T_{g;sg}}$) ou ($\mathbf{T_{g;sd}}$))
Si l'on note t la quantité

$$t = \frac{\mathbb{E}\left[g(\omega)\,|\,\mathcal{A}\in\omega\right]}{\mathbb{E}\left[g(\omega)\right]}$$

alors le signe de $\Delta\mathcal{E}_{err}$ dépend de ce rapport t. Plus précisément,

- *Si $t>1$, $\Delta\mathcal{E}_{err}$ est négatif.*
- *Si $t<1$, il existe p_0 tel que*

$$\begin{cases}\mathbb{P}_{t_s}(\mathcal{A}) < p_0 \Longrightarrow \Delta\mathcal{E}_{err} > 0\\ \mathbb{P}_{t_s}(\mathcal{A}) > p_0 \Longrightarrow \Delta\mathcal{E}_{err} < 0\end{cases}$$

Cette proposition nous donne donc l'influence de la suppression d'un arbre $\mathcal{A}$ de $\mathcal{F}_{t_s}$.
Preuve : Nous renvoyons à l'annexe C, section C-3 pour la démonstration complète. □

En définissant les quantités $\mathcal{E}_{err,\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$ par ce qui suit :

Définition 5.6.1 (Erreur relative à $\mathcal{A}$: $\mathcal{E}_{err,\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$)
La quantité $\mathcal{E}_{err,\mathcal{A}}(\mathcal{F};\mathbb{P})$ est définie par

$$\mathcal{E}_{err,\mathcal{A}}(\mathcal{F};\mathbb{P}) = \sum_{\omega\in\mathcal{F}^{p-1}} g(\omega;\mathcal{A})\mathbb{P}(\omega)$$

Cette quantité désigne donc l'erreur moyenne g commise par le classifieur sachant que les features tirés contiennent $\mathcal{A}$ puisque :

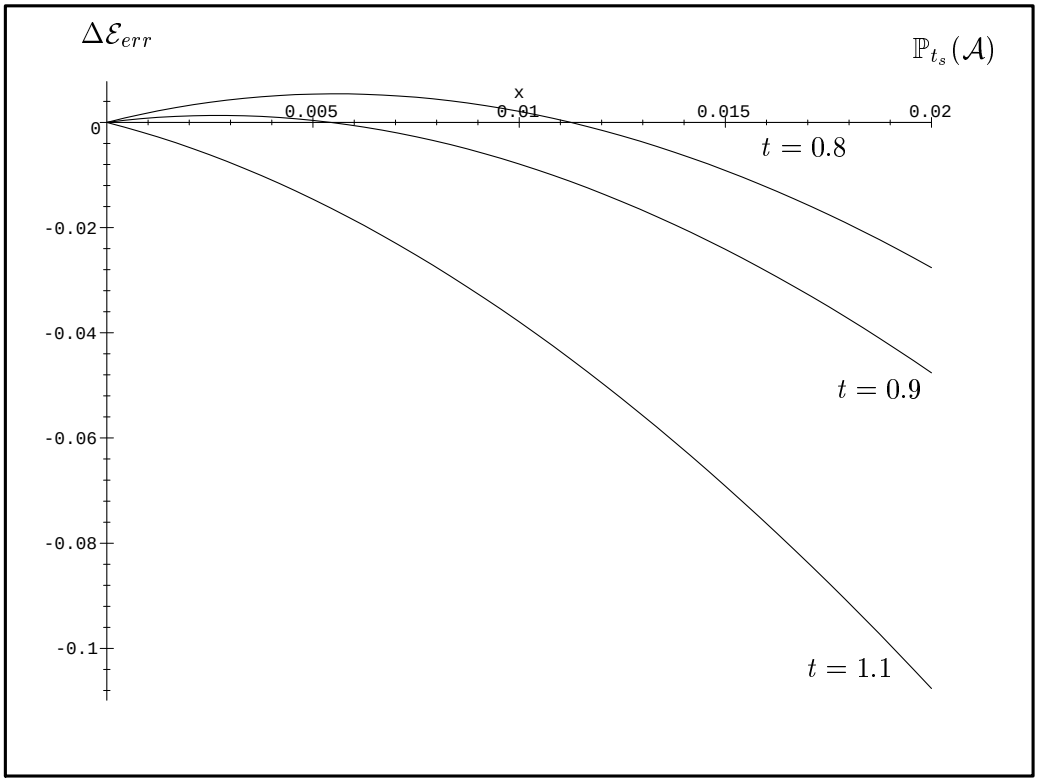
$$\mathcal{E}_{err,\mathcal{A}}(\mathcal{F};\mathbb{P}) = \mathbb{E}\left[g(\omega)\,|\,\mathcal{A}\in\omega\right]$$

on peut donc conclure que

- Si l'erreur $\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$ est supérieure à $\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$, on est dans le cas où $t>1$ et le terme d'énergie d'erreur dans τ_2 est strictement plus grand que 1 (le saut est favorisé).

- Si l'erreur $\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$ est inférieure à $\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$, on est dans le cas où $t<1$ et le terme d'énergie d'erreur dans τ_2 est strictement inférieur à 1, sauf pour de grandes probabilités $\mathbb{P}_{t_s}(\mathcal{A})$. Le saut est donc défavorisé sauf dans le cas où $\mathbb{P}_{t_s}(\mathcal{A})$ est trop grand. On verra dans l'annexe C comment s'affranchir d'avoir à manipuler des coupes avec des arbres ayant une trop grande probabilité $\mathbb{P}_{t_s}(\mathcal{A})$. On peut constater également que l'ordre de grandeur de $\mathbb{P}_{t_s}(\mathcal{A})$ indique que l'on est malgré tout dans la zone où le signe de $\Delta\mathcal{E}_{err}$ est positif.

Voici une courbe représentant l'évolution du différentiel énergétique en fonction de $\mathbb{P}_{t_s}(\mathcal{A})$ dans différents cas $t<1$ et $t>1$:



5.6.3.5 Cas d'une greffe ($\mathbf{T_{g;sgd}}$)

Ce cas se traite de façon relativement similaire au cas précédent. Si $\mathcal{A}_1$ et $\mathcal{A}_2$ désignent les deux fils qui engendrent la création de $\mathcal{B} = \mathrm{T}_g(\mathcal{A}_1; \mathcal{A}_2)$ et qui sont supprimés à l'instant du saut t_s, le différentiel énergétique du terme d'erreur ne prend finalement en compte que la suppression de $\mathcal{A}_1$ et $\mathcal{A}_2$ puisque

$$\mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{B}) = 0$$

Si l'on pose là encore formellement que

$$\begin{cases} \mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{A}_1) = 0 \\ \mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{A}_2) = 0 \end{cases}$$

la valeur de $\mathcal{E}_{err}(\mathcal{F}_{t_s+\mathrm{d}t}; \mathbb{P}_{t_s+\mathrm{d}t})$ est à nouveau inchangée et l'on peut alors répéter les calculs effectués dans le cas de ($\mathbf{T_{g;sg}}$) et ($\mathbf{T_{g;sd}}$). On définit l'erreur relative à $\mathcal{A}_1$ et $\mathcal{A}_2$ selon la définition 5.6.2.

Définition 5.6.2 (Erreur relative à $\mathcal{A}_1$ et $\mathcal{A}_2$: $\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}(\mathcal{F};\mathbb{P})$:)
La quantité $\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}(\mathcal{F};\mathbb{P})$ est définie par

$$\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}(\mathcal{F};\mathbb{P}) = \sum_{\omega \in (\mathcal{F})^{p-2}} g(\omega; \mathcal{A}_1; \mathcal{A}_2)\mathbb{P}(\omega)$$

Cette quantité désigne, là aussi, l'erreur moyenne g commise par le classifieur sachant que les features tirés contiennent $\mathcal{A}_1$ et $\mathcal{A}_2$ puisque :

$$\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}(\mathcal{F};\mathbb{P}) = \mathbb{E}[g(\omega)|(\mathcal{A}_1; \mathcal{A}_2) \in \omega]$$

La définition précédente permet d'expliciter le différentiel énergétique $\Delta\mathcal{E}_{err}$ dans la propriété 5.6.3.

Propriété 5.6.3 (Calcul de $\Delta\mathcal{E}_{err}$ pour ($\mathbf{T_{g;sgd}}$) :)
$\Delta\mathcal{E}_{err}$ est du signe de

$$\mathcal{E}_{err}(\mathcal{F}_{t_s}; \mathbb{P}_{t_s})\,[1-(1-s)^p] - ps\{\mathcal{E}_{err;\mathcal{A}_1} + \mathcal{E}_{err;\mathcal{A}_2}\} + p(p-1)q\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}$$

où

$$\begin{cases} s = \mathbb{P}_{t_s}(\mathcal{A}_1) + \mathbb{P}_{t_s}(\mathcal{A}_2) \\ q = \mathbb{P}_{t_s}(\mathcal{A}_1)\mathbb{P}_{t_s}(\mathcal{A}_2) \end{cases}$$

et

$$\{\mathcal{E}_{err;\mathcal{A}_1} - \mathcal{E}_{err;\mathcal{A}_2}\} = \frac{\mathbb{P}_{t_s}(\mathcal{A}_1)\mathcal{E}_{err;\mathcal{A}_1} + \mathbb{P}_{t_s}(\mathcal{A}_2)\mathcal{E}_{err;\mathcal{A}_2}}{s}$$

Preuve : Nous renvoyons à l'appendice C paragraphe 3 pour le calcul d'un tel différentiel énergétique. □

5.6.3.6 Cas d'une coupe ($\mathbf{T_c}$)

Ce cas se traite exactement de la même façon que celui de ($\mathbf{T_{g;sg}}$) ou ($\mathbf{T_{g;sd}}$). Si $\mathcal{A}$ est l'arbre que l'on souhaite couper, on a donc :

Propriété 5.6.4 (Calcul de $\Delta\mathcal{E}_{err}$ pour ($\mathbf{T_c}$) :)
Dans le cas d'une coupe, le calcul du différentiel énergétique du terme d'erreur donne :

$$\Delta\mathcal{E}_{err}\left[\begin{pmatrix} \mathcal{F}_{t_s} \\ \mathbb{P}_{t_s} \end{pmatrix} \longrightarrow \begin{pmatrix} \mathcal{F}_{t_s+\mathrm{d}t} \\ \mathbb{P}_{t_s+\mathrm{d}t} \end{pmatrix}\right] \equiv [1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p] - p\mathbb{P}_{t_s}(\mathcal{A})t$$

5.6.3.7 Cas d'une renaissance ($\mathbf{T_i}$)

Dans ce cas, on a immédiatement la propriété 5.6.5.

Propriété 5.6.5 (Calcul de $\Delta\mathcal{E}_{err}$ pour ($\mathbf{T_i}$) :)
Le différentiel énergétique $\Delta\mathcal{E}_{err}$ vaut :

$$\Delta\mathcal{E}_{err}\left[\begin{array}{ccc}\mathcal{F}_{t_s} & \longrightarrow & \mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s} & & \mathbb{P}_{t_s+\mathrm{d}t}\end{array}\right]=0$$

5.6.4 Définition de $\mathcal{E}(\mathcal{F};\mathbb{P})$

Pour achever complètement le calcul de τ, il est nécessaire de calculer les différentiels énergétiques associés aux énergies $\mathcal{E}_{\mathrm{L}g}$ et $\mathcal{E}_\rho$. En réalité, le choix est laissé à l'utilisateur d'imposer telle ou telle définition pour ces énergies. L'idée directrice est de privilégier les forêts ayant un nombre d'arbres plutôt restreint mais étant si possible composés. Les travaux [FG01] ont permis de mettre en avant une construction de tests binaires de plus en plus complexes utilisant un critère de corrélation pour agréger des détecteurs. Nous utiliserons donc deux termes dans $\mathcal{E}$ pour une telle construction :

- Un terme $\mathcal{E}_{\mathrm{L}g}$ qui est un terme du type MDL ([Ris89], [Ris83])

  par exemple $$\mathcal{E}_{\mathrm{L}g}(\mathcal{F};\mathbb{P})=\sum_{\mathcal{A}\in\mathcal{F}}|\mathcal{A}|$$

- Un terme de mesure d'information sur chacun des arbres $\mathcal{A}$ de $\mathcal{F}$ relativement à leurs deux fils :
  - Dans le cas de variables binaires, on peut par exemple choisir :
  $$\mathcal{E}_\rho(\mathcal{F};\mathbb{P})=-\sum_{\mathcal{A}\in\mathcal{F}}\rho\left(\mathcal{A}.g;\mathcal{A}.d\right)$$
  - Si les variables sont quelconques, on peut choisir également
  $$\mathcal{E}_\rho(\mathcal{F};\mathbb{P})=-\sum_{\mathcal{A}\in\mathcal{F}}\mathrm{I}(\mathcal{A}.g;\mathcal{A}.d)$$

Pour de telles définitions d'énergie, on constate en réalité que seule la « structure » de la forêt intervient dans la méthode de calcul et pas la probabilité $\mathbb{P}$. Ainsi, le terme τ_2 sera :

$$\tau_2=e^{-\Delta\mathcal{E}_{err}-\Delta\mathcal{E}_{\mathrm{L}g}-\Delta\mathcal{E}_\rho}$$

avec $$e^{-\Delta\mathcal{E}_{\mathrm{L}g}}=e^{\sum_{\mathcal{A}\text{ enlevé}}|\mathcal{A}|-\sum_{\mathcal{A}\text{ ajouté}}|\mathcal{A}|}$$

et $$e^{-\Delta\mathcal{E}_\rho}=e^{\sum_{\mathcal{A}\text{ ajouté}}\mathrm{I}(\mathcal{A}.g;\mathcal{A}.d)-\sum_{\mathcal{A}\text{ enlevé}}\mathrm{I}(\mathcal{A}.g;\mathcal{A}.d)}$$

Les commentaires que l'on peut apporter à de telles expressions sont relativement simples :

- l'augmentation de la somme des longueurs des arbres implique une diminution de τ_2, et l'ajout d'un arbre à $\mathcal{F}$ sans suppression (par exemple ($\mathbf{T_i}$) ou ($\mathbf{T_g}$)) ne sera pas favorisé par ce terme, alors que la greffe avec suppression d'un ou des deux fils *via* ($\mathbf{T_{g;sg}}$), ($\mathbf{T_{g;sd}}$) ou ($\mathbf{T_{g;sgd}}$) sera plutôt favorisée.

- la greffe d'un arbre tel que ses deux fils ne possèdent pas d'information commune I notable ne sera pas favorisée, alors que la probabilité d'accepter l'ajout d'un arbre composé de deux fils corrélés sera plus importante.

Ces deux termes d'énergie sont motivés par une nécessaire réduction de la complexité du problème (maîtrise du nombre d'arbres présents dans les forêts) et par des critères provenant de la théorie de l'information (information relative et minimisation de la variance du système par le principe de MDL [Ris89]).

On peut cependant privilégier une approche un peu différente en ce qui concerne le terme de coût issu de $\mathcal{E}_{\mathrm{L}g}$. En effet, on peut décider de privilégier la redondance de certains arbres dans d'autres sous-arbres de la forêt, car le temps de calcul total de la forêt sur la base de données est alors réduite. Plus il y a de redondances d'arbres dans d'autres sous-arbres de la forêt, plus il y a de fragments d'information réutilisables, ce qui apporte un avantage notable en terme de complexité [KGA02].

Nous renvoyons enfin à l'annexe C pour un récapitulatif complet des quantités permettant de définir totalement les transitions lors d'un saut.

En fin de compte, il ne reste plus qu'à vérifier que les conditions d'existence de processus de diffusion avec sauts sont vérifiées lorsqu'on définit ainsi les probabilités Q de transition. Nous rappelons que ces conditions sont données dans la section 5.3 de ce chapitre.

5.7 Existence et unicité du processus de diffusion réfléchie avec sauts entre les forêts

Il s'agit donc comme nous l'avons vu au paragraphe 5.2 de vérifier les conditions ($\mathbf{C_{4,5,6,7}}$). pour conclure enfin à l'existence de tels processus. C'est précisément ce que nous établissons dans ce paragraphe.

5.7.1 Terme de dérive G et covariance Σ

Nous cherchons donc à construire un processus couplé $(\mathbb{P}_t; \mathrm{X}_t)$ tel que $\mathbb{P}_t$ suive une diffusion sous contraintes dictée par X_t entre deux instants de sauts et suivant des règles de transitions markoviennes lors des instants de sauts qui sont distribués suivant une loi de Poisson.

Il s'agit désormais de donner un sens précis aux termes mentionnés ci-dessus. On peut définir le terme de dérive G par

Définition 5.7.1 (Terme de dérive G)
Afin d'effectuer une recherche de minimum de la fonction $\mathcal{E}$, nous utiliserons comme vecteur de dérive G *:*

$$\mathrm{G}\begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix} = \begin{pmatrix} \Pi_{\mathcal{H}_{\mathcal{F}(\mathrm{X})}} \left(\nabla \mathcal{E}_{err}(\mathbb{P})\right) \\ 0 \end{pmatrix} \tag{5.29}$$

Seuls les éléments présents dans le support de $\mathbb{P}$ interviennent dans $\mathcal{E}_{err}$. On a donc en réalité

$$\mathcal{E}_{err}\left(\mathbb{P}; \mathcal{F}\right) = \mathcal{E}_{err}\left(\mathbb{P}\right)$$

La notation $\nabla \mathcal{E}_{err}(\mathbb{P})$ a donc bien un sens, et il en est de même de l'application G*. L'hyperplan $\mathcal{H}$ sur lequel on projette le vecteur gradient est celui qui correspond à (4.23) pour les coordonnées données par* X*.*

De façon similaire, on a la définition de la matrice Σ :

Définition 5.7.2 (Matrice Σ)
On définit la matrice de covariance du mouvement brownien par

$$\Sigma \begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix} = \begin{pmatrix} \begin{matrix} & & \sigma(\mathrm{X}) & & \\ 0 & \dots & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{matrix} \end{pmatrix} \tag{5.30}$$

La diffusion sous contraintes utilise donc un mouvement brownien standard sur l'hyperplan $\mathcal{H}_{\mathcal{F}}$ entre les sauts (covariance égale à (4.26) sur la restriction à $\mathcal{S}_{\mathcal{F}(\mathrm{X})}$ et nulle en dehors) et le terme brownien n'intervient pas pour X ainsi que pour les coordonnées de P n'étant pas dans $\mathcal{F}(\mathrm{X})$. Les quantités $\mathbb{P}_t(\mathcal{A})$ sont donc figées et nulles lorsque $\mathrm{X}_t(\mathcal{A}) = 0$.

5.7.2 Conditions $(\mathbf{C_4})$, $(\mathbf{C_5})$, $(\mathbf{C_6})$ et $(\mathbf{C_7})$ dans notre modèle

Il est tout d'abord nécessaire de munir l'espace dans lequel « vit » le processus $(\mathbb{P}, \mathrm{X})$ d'une norme (certes toutes équivalentes) pour donner un sens précis aux conditions $(\mathbf{C})$. Ceci est fait grâce à la définition qui suit.

Définition 5.7.3 (Norme sur $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^{|\mathcal{A}^\star|}$)
Nous munissons l'espace produit entre le simplexe des probabilités sur tous les arbres possibles $(\mathcal{A}^\star)$ et les vecteurs indicateurs des arbres de la norme :

$$\forall (\mathbb{P}; \mathrm{X}) \in \mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^{|\mathcal{A}^\star|} \qquad \left\| \begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix} \right\| = \|\mathbb{P}\|_2 + \|\mathrm{X}\|_\infty \tag{5.31}$$

Même si cette application ne définit pas exactement un espace vectoriel normé sur $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^{|\mathcal{A}^\star|}$ mais sur $\mathbb{R}^{|\mathcal{A}^\star|} \times \mathbb{R}^{|\mathcal{A}^\star|}$, on peut tout de même vérifier que suivant cette norme, les conditions évoquées précédemment sont vraies.

Lemme 5.7.1 (Vérification de $(\mathbf{C_4})$)
Le terme de dérive correspondant à (5.29) vérifie bien $(\mathbf{C_4})$.

Preuve :

- Si $\mathrm{X}_1 = \mathrm{X}_2$, on peut utiliser (5.29) et (5.31), ainsi que l'expression de $\nabla \mathcal{E}_{err}$ issue de (3.5), on obtient :

$$\begin{aligned} \left\| \mathrm{G} \begin{pmatrix} \mathbb{P}_1 \\ \mathrm{X}_1 \end{pmatrix} - \mathrm{G} \begin{pmatrix} \mathbb{P}_2 \\ \mathrm{X}_2 \end{pmatrix} \right\| &= \left\| \begin{pmatrix} \Pi_{\mathcal{H}_1} \left(\nabla \mathcal{E}_{err}(\mathbb{P}_1) - \nabla \mathcal{E}_{err}(\mathbb{P}_2) \right) \\ 0 \end{pmatrix} \right\| \\ &\leqslant \left\| \begin{pmatrix} \nabla \mathcal{E}_{err}(\mathbb{P}_1) - \nabla \mathcal{E}_{err}(\mathbb{P}_2) \\ 0 \end{pmatrix} \right\| \\ &= \|\nabla \mathcal{E}_{err}(\mathbb{P}_1) - \nabla \mathcal{E}_{err}(\mathbb{P}_2)\|_2 \end{aligned}$$

  On peut alors appliquer le résultat du paragraphe 4.2.2 (condition $(\mathbf{C_1})$) pour en déduire que :

$$\left\| \mathrm{G} \begin{pmatrix} \mathbb{P}_1 \\ \mathrm{X}_1 \end{pmatrix} - \mathrm{G} \begin{pmatrix} \mathbb{P}_2 \\ \mathrm{X}_2 \end{pmatrix} \right\|^2 \leqslant 2p^2 \mathrm{M} \|\mathbb{P}_1 - \mathbb{P}_2\|_2{}^2$$

- Si $X_1 \neq X_2$, on a

$$\|X_1 - X_2\|_\infty \geqslant 1$$

Par ailleurs, b est une fonction continue sur le compact $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$. Elle est donc bornée, et G est également borné en norme par K, ce qui finalement se traduit par :

$$\left\| G\begin{pmatrix} \mathbb{P}_1 \\ X_1 \end{pmatrix} - G\begin{pmatrix} \mathbb{P}_2 \\ X_2 \end{pmatrix} \right\| \leqslant K\|X_1 - X_2\|_\infty \leqslant K \left\| \begin{pmatrix} \mathbb{P}_1 - \mathbb{P}_2 \\ X_1 - X_2 \end{pmatrix} \right\|$$

En fin de compte, on obtient la première condition :

$$\left\| G\begin{pmatrix} \mathbb{P}_1 \\ X_1 \end{pmatrix} - G\begin{pmatrix} \mathbb{P}_2 \\ X_2 \end{pmatrix} \right\| \leqslant \theta_4 \left\| \begin{pmatrix} \mathbb{P}_1 - \mathbb{P}_2 \\ X_1 - X_2 \end{pmatrix} \right\|$$

Ceci ce réécrit bien en $(\mathbf{C_4})$ en posant $\theta_4 = \text{Max}\{K; p\sqrt{2M}\}$. □

Lemme 5.7.2 (Vérification de $(\mathbf{C_5})$)
L'expression de Σ (5.30) assure que la condition $(\mathbf{C_5})$ est vraie.

Preuve : Donnons-nous deux probabilités $\mathbb{P}_1$ et $\mathbb{P}_2$ dans $\mathcal{S}_{\mathcal{A}^\star}$ ainsi que deux vecteurs indicateurs X_1 et X_2 de deux forêts. L'expression de (5.30) ne dépend pas des probabilités $\mathbb{P}_1$ et $\mathbb{P}_2$ et

$$\Sigma\begin{pmatrix} \mathbb{P}_1 \\ X_1 \end{pmatrix} - \Sigma\begin{pmatrix} \mathbb{P}_2 \\ X_2 \end{pmatrix} = \begin{pmatrix} & \sigma(X_1) - \sigma(X_2) & & & \\ \begin{pmatrix} 0 & \dots & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix} & & & & \end{pmatrix}$$

Ainsi, la norme de la matrice précédente vaut

$$\left\| \Sigma\begin{pmatrix} \mathbb{P}_1 \\ X_1 \end{pmatrix} - \Sigma\begin{pmatrix} \mathbb{P}_2 \\ X_2 \end{pmatrix} \right\|^2 = \|\sigma(X_1) - \sigma(X_2)\|_2{}^2$$

On étudie deux cas comme dans la démonstration précédente :

- $X_1 = X_2$ Dans ce cas, la différence est nulle et n'importe quelle constante convient pour θ_2.
- $X_1 \neq X_2$ La quantité précédente est maximale lorsque $\{X_1 = 1\} \cap \{X_2 = 1\} = \varnothing$. Dans ce cas là, cette norme vaut alors :

$$\begin{aligned} \|\sigma(X_1) - \sigma(X_2)\|_2{}^2 &= \sum_{i,j=1\dots n} (\sigma(X_1)_{i,j} - \sigma(X_2)_{i,j})^2 \\ &= \sum_{i=1}^n \underbrace{(\sigma(X_1)_{i,i} - \sigma(X_2)_{i,i})^2}_{\leqslant \frac{(f-1)^2}{f^2}} + \sum_{i=1}^n \sum_{j\neq i} \underbrace{(\sigma(X_1)_{i,j} - \sigma(X_2)_{i,j})^2}_{\leqslant \frac{1}{f^2}} \end{aligned}$$

Finalement $\|\sigma(X_1) - \sigma(X_2)\|_2{}^2 \leqslant \dfrac{(f-1)^2}{f} + f(f-1)\dfrac{1}{f^2} = f - 1$

En utilisant le fait que $\|X_1 - X_2\|_\infty \geqslant 1$, on en déduit que :

$$\|\sigma(X_1) - \sigma(X_2)\|_2{}^2 \leqslant (f-1)\|X_1 - X_2\|_\infty^2 \leqslant (f-1)\left\| \begin{pmatrix} \mathbb{P}_1 - \mathbb{P}_2 \\ X_1 - X_2 \end{pmatrix} \right\|^2$$

La condition $(\mathbf{C_5})$ est donc établie en posant $\theta_2 = \sqrt{f-1}$. □

Vient ensuite la vérification de la condition $(\mathbf{C_6})$.

Lemme 5.7.3 (Vérification de $(\mathbf{C_6})$)
La fonction q définie en (5.27) et telle que la probabilité de transition Q *est donnée par les résultats du paragraphe 5.6 vérifie la condition* $(\mathbf{C_6})$.

Preuve : Pour démontrer ce point, donnons-nous deux états du processus $(\mathbb{P}_1; \mathrm{X}_1)$ et $(\mathbb{P}_2; \mathrm{X}_2)$. Il faut alors étudier la différence

$$\mathrm{D} = \underbrace{\mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_1\\ \mathrm{X}_1\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)}_{\text{Probabilité d'acceptation du saut}} \underbrace{\begin{pmatrix}\mathbb{P}-\mathbb{P}_1\\ \mathrm{X}-\mathrm{X}_1\end{pmatrix}}_{\text{Quantité à ajouter}} - \underbrace{\mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)}_{\text{Probabilité d'acceptation du saut}} \underbrace{\begin{pmatrix}\mathbb{P}-\mathbb{P}_2\\ \mathrm{X}-\mathrm{X}_2\end{pmatrix}}_{\text{Quantité à ajouter}}$$

où $(\mathbb{P}; \mathrm{X})$ désigne un état quelconque possible du processus après une transition décrite au chapitre 5.

1. Si la transition $(\mathbb{P}_1; \mathrm{X}_1) \longmapsto (\mathbb{P}; \mathrm{X})$ s'effectue sans coupe d'arbres $((\mathbf{T_c}))$ ou suppression $((\mathbf{T_{g;sg}})$, $(\mathbf{T_{g;sd}})$ ou $(\mathbf{T_{g;sgd}}))$, la probabilité $\mathbb{P}_1$ n'est alors pas modifiée et on a :
$$\mathbb{P} - \mathbb{P}_1 = 0 \tag{5.32}$$
2. Si la transition $(\mathbb{P}_1; \mathrm{X}_1) \longmapsto (\mathbb{P}; \mathrm{X})$ s'effectue avec une coupe ou une suppression d'un ou plusieurs arbres, on pose $p_{1,1}$ la probabilité de l'arbre supprimé $\mathbb{P}_1(\mu_{1,1})$ (on a également $p_{1,2}$ la probabilité du second arbre supprimé $\mathbb{P}_1(\mu_{1,2})$ s'il y a suppression de deux arbres). On a alors :
$$\begin{cases} \mathbb{P}(\delta) = \mathbb{P}_1(\delta)\dfrac{1}{1-p} & \text{si} \quad \delta \notin \{\mu_{1,1}; \mu_{1,2}\} \\ \mathbb{P}(\mu_{1,1}) = 0 \\ \mathbb{P}(\mu_{1,2}) = 0 \end{cases}$$
avec $p = p_{1,1} + p_{1,2}$. La quantité à ajouter pour passer de $\mathbb{P}_1$ à $\mathbb{P}$ vaut donc :
$$\Delta\mathbb{P}_1(\delta) = \begin{cases} \mathbb{P}_1(\delta)\dfrac{p}{1-p} & \text{si} \quad \delta \notin \{\mu_{1,1}; \mu_{1,2}\} \\ -p_{1,1} & \text{si} \quad \delta = \mu_{1,1} \\ -p_{1,2} & \text{si} \quad \delta = \mu_{1,2} \end{cases} \tag{5.33}$$

On constate donc que quitte à poser $p_{1,1} = p_{1,2} = p = 0$ en cas de non suppression ou coupe, les formules précédentes coïncident avec (5.32). Ainsi, quelle que soit l'opération effectuée, les formules (5.33) sont vraies pour la différence $\mathbb{P} - \mathbb{P}_1$.

En revenant au calcul de D, on peut réécrire son expression en

$$\mathrm{D} = \underbrace{\left[\mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_1\\ \mathrm{X}_1\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right) - \mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)\right]}_{\text{Uniformément Lipschitzien en}(\mathbb{P};\mathrm{X})?} \underbrace{\begin{pmatrix}\mathbb{P}-\mathbb{P}_1\\ \mathrm{X}-\mathrm{X}_1\end{pmatrix}}_{\text{borné d'après(5.33)}}$$

$$+ \underbrace{\mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)}_{\text{borné car } \min\{1;\tau\}\leqslant 1} \underbrace{\begin{pmatrix}\mathbb{P}_2-\mathbb{P}_1\\ \mathrm{X}_2-\mathrm{X}_1\end{pmatrix}}_{\text{Lipschitzien}}$$

Il suffit donc d'étudier le caractère Lipschitzien de la probabilité d'acceptation du saut Q, uniformément en la seconde variable pour établir la condition ($\mathbf{C_6}$). La différence à étudier est donc

$$\mathrm{D}' = \mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_1\\ \mathrm{X}_1\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right) - \mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_2\\ \mathrm{X}_2\end{pmatrix};\begin{pmatrix}\mathbb{P}\\ \mathrm{X}\end{pmatrix}\right)$$

$$= \min\{1;\tau_1\} - \min\{1;\tau_2\}$$

et les termes τ_1 et τ_2 sont donnés par :

$$\begin{cases}\tau_1 = \dfrac{\mathrm{P}[\mathrm{X}|\mathrm{X}_1]}{\mathrm{P}[\mathrm{X}_1|\mathrm{X}]} e^{\mathcal{E}(\mathbb{P}_1;\mathrm{X}_1)-\mathcal{E}(\mathbb{P};\mathrm{X})}\\ \tau_2 = \dfrac{\mathrm{P}[\mathrm{X}|\mathrm{X}_2]}{\mathrm{P}[\mathrm{X}_2|\mathrm{X}]} e^{\mathcal{E}(\mathbb{P}_2;\mathrm{X}_2)-\mathcal{E}(\mathbb{P};\mathrm{X})}\end{cases}$$

On procède alors à une disjonction des différents cas possibles :

1. Si $\mathrm{X}_1 = \mathrm{X}_2$: les termes issus des probabilités de transition $\mathrm{P}[.|.]$ sont donc égaux dans τ_1 et τ_2. On effectue une seconde disjonction des cas :

   (a) Si $\tau_1 \geqslant 1$ et $\tau_2 \geqslant 1$ la quantité D' est alors nulle :

   $$\mathrm{D}' = 0$$

   (b) Si $\tau_1 < 1$ et $\tau_2 < 1$ la quantité D' se simplifie en

   $$\mathrm{D}' = \underbrace{\frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_1]}{\mathrm{P}[\mathrm{X}_1|\mathrm{X}]}}_{=\frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_2]}{\mathrm{P}[\mathrm{X}_2|\mathrm{X}]}} \underbrace{e^{-\mathcal{E}(\mathbb{P};\mathrm{X})}}_{\text{borné par 1}} \left[e^{\mathcal{E}(\mathbb{P}_1;\mathrm{X}_1)} - e^{\mathcal{E}(\mathbb{P}_2;\mathrm{X}_2)}\right]$$

   mais $$\mathcal{E}(\mathbb{P}_1;\mathrm{X}_1) = \mathcal{E}_{err}(\mathbb{P}_1) + \mathcal{E}_{\mathrm{L}g}(\mathrm{X}_1) + \mathcal{E}_{\rho}(\mathrm{X}_1)$$

   d'où $$\mathrm{D}' = \frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_1]}{\mathrm{P}[\mathrm{X}_1|\mathrm{X}]} e^{\mathcal{E}_{\mathrm{L}g}(\mathrm{X}_1)+\mathcal{E}_{\rho}(\mathrm{X}_1)} \left[e^{\mathcal{E}_{err}(\mathbb{P}_1)} - e^{\mathcal{E}_{err}(\mathbb{P}_2)}\right]$$

   L'ensemble des forêts $\mathcal{A}^\star$ étant fini, X et X_1 varient dans un ensemble fini et d'après les règles de transition du chapitre 5, on sait que si $\mathrm{P}[\mathrm{X}|\mathrm{X}_1] > 0$, alors $\mathrm{P}[\mathrm{X}_1|\mathrm{X}] > 0$.

   Ainsi $$\frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_1]}{\mathrm{P}[\mathrm{X}_1|\mathrm{X}]} \leqslant \underset{\mathrm{X},\mathrm{Y}\in\mathcal{A}^\star}{\mathrm{Max}} \frac{\mathrm{P}[\mathrm{X}|\mathrm{Y}]}{\mathrm{P}[\mathrm{Y}|\mathrm{X}]}$$

   Par ailleurs, la fonction $\mathcal{E}_{err}$ est différentiable sur $\mathcal{S}_{\mathcal{A}^\star}$ qui est compact, donc $e^{\mathcal{E}_{err}}$ est également différentiable. Finalement $\nabla e^{\mathcal{E}_{err}}$ est borné sur $\mathcal{S}_{\mathcal{A}^\star}$ et l'inégalité des accroissements finis implique que

   $$\exists\, \mathrm{K}_1 \qquad \left|e^{\mathcal{E}_{err}(\mathbb{P}_1)} - e^{\mathcal{E}_{err}(\mathbb{P}_2)}\right| \leqslant \mathrm{K}_1 \|\mathbb{P}_1 - \mathbb{P}_2\|_2$$

   On peut donc conclure que

   $$\mathrm{D}' \leqslant \mathrm{K}_1 \underset{\mathrm{X},\mathrm{Y}\in\mathcal{A}^\star}{\mathrm{Max}} \frac{\mathrm{P}[\mathrm{X}|\mathrm{Y}]}{\mathrm{P}[\mathrm{Y}|\mathrm{X}]} \underset{\mathrm{X}\in\mathcal{A}^\star}{\mathrm{Max}} e^{\mathcal{E}_{\mathrm{L}g}(\mathrm{X})+\mathcal{E}_{\rho}(\mathrm{X})} \|\mathbb{P}_1 - \mathbb{P}_2\|_2$$

(c) Si $\tau_1 \geqslant 1$ et $\tau_2 < 1$ Dans ce cas, D' vaut

$$\mathrm{D}' = 1 - \frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_2]}{\mathrm{P}[\mathrm{X}_2|\mathrm{X}]} e^{\mathcal{E}_{err}(\mathbb{P}_2) - \mathcal{E}_{err}(\mathbb{P})} \tag{5.34}$$

Les forêts indexées par X_1 et X_2 étant égales, on a toujours

$$\frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_2]}{\mathrm{P}[\mathrm{X}_2|\mathrm{X}]} = \frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_1]}{\mathrm{P}[\mathrm{X}_1|\mathrm{X}]} \tag{5.35}$$

En outre, comme $\tau_1 \geqslant 1$, on sait que :

$$\frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_1]}{\mathrm{P}[\mathrm{X}_1|\mathrm{X}]} e^{\mathcal{E}(\mathbb{P}_1,\mathrm{X}_1) - \mathcal{E}(\mathbb{P},\mathrm{X})} \geqslant 1$$

soit encore

$$-\frac{\mathrm{P}[\mathrm{X}|\mathrm{X}_1]}{\mathrm{P}[\mathrm{X}_1|\mathrm{X}]} \leqslant e^{\mathcal{E}(\mathbb{P},\mathrm{X}) - \mathcal{E}(\mathbb{P}_1,\mathrm{X}_1)} \tag{5.36}$$

En utilisant alors (5.36) et (5.35) dans (5.34), on obtient

$$\mathrm{D}' \leqslant 1 - \frac{e^{\mathcal{E}(\mathbb{P}_2,\mathrm{X}_2) - \mathcal{E}(\mathbb{P},\mathrm{X})}}{e^{\mathcal{E}(\mathbb{P}_1,\mathrm{X}_1) - \mathcal{E}(\mathbb{P},\mathrm{X})}} = 1 - e^{\mathcal{E}(\mathbb{P}_2,\mathrm{X}_2) - \mathcal{E}(\mathbb{P}_1,\mathrm{X}_1)} = 1 - e^{\mathcal{E}_{err}(\mathbb{P}_2) - \mathcal{E}_{err}(\mathbb{P}_1)}$$

Par ailleurs, D' est positif au vu des hypothèses sur τ_1 et τ_2, donc

$$0 \leqslant \mathrm{D}' \leqslant 1 - e^{\mathcal{E}_{err}(\mathbb{P}_2) - \mathcal{E}_{err}(\mathbb{P}_1)}$$

Cette dernière inégalité implique en particulier que

$$t = \mathcal{E}_{err}(\mathbb{P}_2) - \mathcal{E}_{err}(\mathbb{P}_1) \leqslant 0$$

Ainsi, puisque $t \leqslant 0$ $\qquad 1 - e^t \leqslant -t$

d'où $\qquad 0 \leqslant \mathrm{D}' \leqslant \mathcal{E}_{err}(\mathbb{P}_1) - \mathcal{E}_{err}(\mathbb{P}_2)$

En utilisant alors également la différentiabilité de $\mathcal{E}_{err}$ sur le compact $\mathcal{S}_{\mathcal{A}^\star}$, on en conclut que

$$\exists\, \mathrm{K}_2 \qquad |\mathrm{D}'| \leqslant \mathrm{K}_2 \|\mathbb{P}_1 - \mathbb{P}_2\|$$

2. Si $\mathrm{X}_1 \neq \mathrm{X}_2$: la situation est nettement plus facile puisque dans ce cas :

$$\|\mathrm{X}_1 - \mathrm{X}_2\|_\infty \geqslant 1$$

donc d'après (5.31) $\qquad \left\| \begin{pmatrix} \mathbb{P}_1 - \mathbb{P}_2 \\ \mathrm{X}_1 - \mathrm{X}_2 \end{pmatrix} \right\| \geqslant 1$

Par ailleurs, Q est une probabilité de transition et donc

$$\left| \mathrm{Q}\left(\begin{pmatrix} \mathbb{P}_2 \\ \mathrm{X}_2 \end{pmatrix}; \begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix} \right) - \mathrm{Q}\left(\begin{pmatrix} \mathbb{P}_1 \\ \mathrm{X}_1 \end{pmatrix}; \begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix} \right) \right| \leqslant 2 \leqslant 2 \left\| \begin{pmatrix} \mathbb{P}_1 - \mathbb{P}_2 \\ \mathrm{X}_1 - \mathrm{X}_2 \end{pmatrix} \right\|$$

Dans tous les cas envisagés, nous avons vu que D' est une fonction Lipschitzienne de sa première variable uniformément en sa deuxième variable. Il en est donc de même de D, ce qui assure bien que $(\mathbf{C_6})$ est vraie. $\square$

Il reste enfin à établir $(\mathbf{C_7})$:

Lemme 5.7.4 (Vérification de $(\mathbf{C_7})$)
Si le processus réfléchi respecte $(\mathbf{E-6})$ *ainsi que les contraintes pour chaque simplexe définies comme dans [DR99] (paragraphe 4.3.2, cas A), alors* $(\mathbf{C_7})$ *est vraie.*

Preuve : La démonstration est également basée sur une disjonction de cas. X est constant par morceaux au cours du temps. Supposons donc que t varie entre deux instants de sauts.

1. Supposons que $\mathrm{X}_1 = \mathrm{X}_2$, alors les applications de Skorokhod Γ_{X_1} et Γ_{X_2} pour le simplexe $\mathcal{S}_{\mathcal{F}(\mathrm{X}_1)}$ et $\mathcal{S}_{\mathcal{F}(\mathrm{X}_1)}$ sont identiques. Elles sont par ailleurs $\mathrm{K_X}$ Lipschitziennes d'après [DR99].

   Ainsi $$\underset{t\in]\,t_1\,;t_2\,]}{\mathrm{Sup}} \|\Gamma_{\mathrm{X}_1}(\mathrm{Z}_1)_t - \Gamma_{\mathrm{X}_2}(\mathrm{Z}_2)_t\| \leqslant \mathrm{K_X} \underset{t\in]\,t_1\,;t_2\,]}{\mathrm{Sup}} \left\| \begin{pmatrix} \mathbb{Z}_1 - \mathbb{Z}_2 \\ \mathrm{X}_1 - \mathrm{X}_2 \end{pmatrix}_t \right\|$$

2. Supposons que $\mathrm{X}_1 \neq \mathrm{X}_2$, on sait que

   $$\Gamma_{\mathrm{X}_1}(\mathrm{Z}_1)_t \in \mathcal{S}_{\mathcal{F}(\mathrm{X}_1)}$$

   et $$\Gamma_{\mathrm{X}_2}(\mathrm{Z}_2)_t \in \mathcal{S}_{\mathcal{F}(\mathrm{X}_2)}$$

   Comme X_1 et X_2 sont distincts, on a donc :

   $$\|\mathrm{X}_1 - \mathrm{X}_2\|_\infty \geqslant 1$$

   D'où $$\|\Gamma_{\mathrm{X}_1}(\mathrm{Z}_1)_t - \Gamma_{\mathrm{X}_2}(\mathrm{Z}_2)_t\|_2^{\,2} \leqslant |\mathcal{A}^\star| \leqslant |\mathcal{A}^\star| \|\mathrm{X}_1 - \mathrm{X}_2\|_\infty$$

   soit $$\underset{t\in]\,t_1\,;t_2\,]}{\mathrm{Sup}} \|\Gamma_{\mathrm{X}_1}(\mathrm{Z}_1)_t - \Gamma_{\mathrm{X}_2}(\mathrm{Z}_2)_t\| \leqslant \sqrt{|\mathcal{A}^\star|} \underset{t\in]\,t_1\,;t_2\,]}{\mathrm{Sup}} \left\| \begin{pmatrix} \mathbb{Z}_1 - \mathbb{Z}_2 \\ \mathrm{X}_1 - \mathrm{X}_2 \end{pmatrix}_t \right\|$$

En choisissant enfin la constante θ_4 étant égale à :

$$\theta_4 = \mathrm{Max}\left\{ \sqrt{|\mathcal{A}^\star|}; \underset{\mathrm{X}\in\mathcal{A}^\star}{\mathrm{Max}}\, \mathrm{K_X} \right\}$$

on obtient alors, en faisant la réunion de tous les intervalles de sauts, que :

$$\underset{t\in\mathbb{R}^+}{\mathrm{Sup}} \|\Gamma_{\mathrm{X}_1}(\mathrm{Z}_1)_t - \Gamma_{\mathrm{X}_2}(\mathrm{Z}_2)_t\| \leqslant \theta_4 \left\| \begin{pmatrix} \mathbb{Z}_1 - \mathbb{Z}_2 \\ \mathrm{X}_1 - \mathrm{X}_2 \end{pmatrix}_t \right\|$$

Cette dernière inégalité assure bien $(\mathbf{C_7})$. □

5.7.3 Bilan

Les conditions ($\mathbf{C_{4,5,6,7}}$) étant désormais satisfaites, il ne reste plus qu'à conclure. Nous pouvons donc énoncer le théorème 5.7.1 de diffusions réfléchies avec sauts sur $\mathcal{S}_{\mathcal{A}^\star}$ qui conclut ce chapitre.

Théorème 5.7.1 (Diffusions réfléchies avec saut sur $\mathcal{S}_{\mathcal{A}^\star}$)
Il existe un unique processus $(\mathbb{P}_t; \mathrm{X}_t; \mathrm{Z}_t)_{t\in\mathbb{R}}$ *tel que*

$$(\mathbf{E-8})\begin{cases}
\mathbb{P}_0 = \mathcal{U}_{\mathcal{F}_0} \quad et \quad \mathcal{F}(\mathrm{X}_0) = \mathcal{F}_0 \\
(t_i)_{i\in\mathbb{N}} \text{ suit une loi de poisson de paramètre } \quad \lambda > 0 \\
\forall t \in [\, t_i \,;\, t_{i+1} \,[\qquad \mathrm{X}_t \quad \text{constant} \\
\forall t \in [\, t_i \,;\, t_{i+1} \,[\qquad \mathrm{d}\mathbb{P}_t = -\Pi_{\mathcal{H}_t}\left(\nabla \mathcal{E}_{err}(\mathbb{P}_t)\right)\mathrm{d}t + \Sigma(\mathrm{X}_t)\mathrm{dW}_t + \mathrm{dZ}_t \\
\forall i \in \mathbb{N} \qquad |\mathrm{Z}|(t_i \mapsto t_{i+1}) < +\infty \qquad p.s. \\
\forall i \in \mathbb{N} \quad \forall t \in [\, t_i \,;\, t_{i+1} \,[\qquad |\mathrm{Z}|(t_i \mapsto t) = \displaystyle\int_{t_i}^{t_{i+1}} \chi_{\mathbb{P}_s \in \partial \mathcal{S}_{\mathcal{F}(\mathrm{X}_i)}} \mathrm{d}|\mathrm{Z}|(s) \\
\forall i \in \mathbb{N} \quad \forall t \in [\, t_i \,;\, t_{i+1} \,[\qquad \mathrm{Z}(t) = \mathrm{Z}(t_i) + \displaystyle\int_{t_i}^{t} \gamma(s)\mathrm{d}|\mathrm{Z}|(s) \\
\forall i \in \mathbb{N} \quad \forall t \in [\, t_i \,;\, t_{i+1} \,[\qquad \gamma(s) \in d_{\mathrm{X}_i}(\mathbb{P}_s) \\
\forall i \in \mathbb{N} \qquad \mathrm{d}\begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix}_{t=t_i} \text{ suit la loi } Q \text{ conditionnée à } \begin{pmatrix} \mathbb{P}_{t_i} \\ \mathrm{X}_{t_i} \end{pmatrix}
\end{cases}$$

$\mathcal{H}_t$ désigne alors l'hyperplan d'appui du simplexe défini par $\mathcal{F}(\mathrm{X}_t)$.

Nous avons donc construit un processus réfléchi avec sauts, couplant à la fois descentes de gradient avec diffusion et sauts d'un espace vers un autre. Les sauts sont conditionnés à une fonction de coût prenant à la fois en compte :

- les performances gagnées ou perdues lors de la suppression d'arbres dans $\mathcal{F}_{t_s}$ *via* le terme $\Delta\mathcal{E}_{err}$.
- la structure même de la forêt $\mathcal{F}_{t_s+\mathrm{d}t}$ par rapport à celle de $\mathcal{F}_{t_s}$: comparaison des longueurs des forêts, des quantités d'informations apportées dans la structure de chaque arbre.

Ces mêmes sauts sont faiblement réversibles, ce qui assure que finalement toute coupe ou greffe d'un arbre de $\mathcal{F}$ peut être « annulée » au saut suivant.

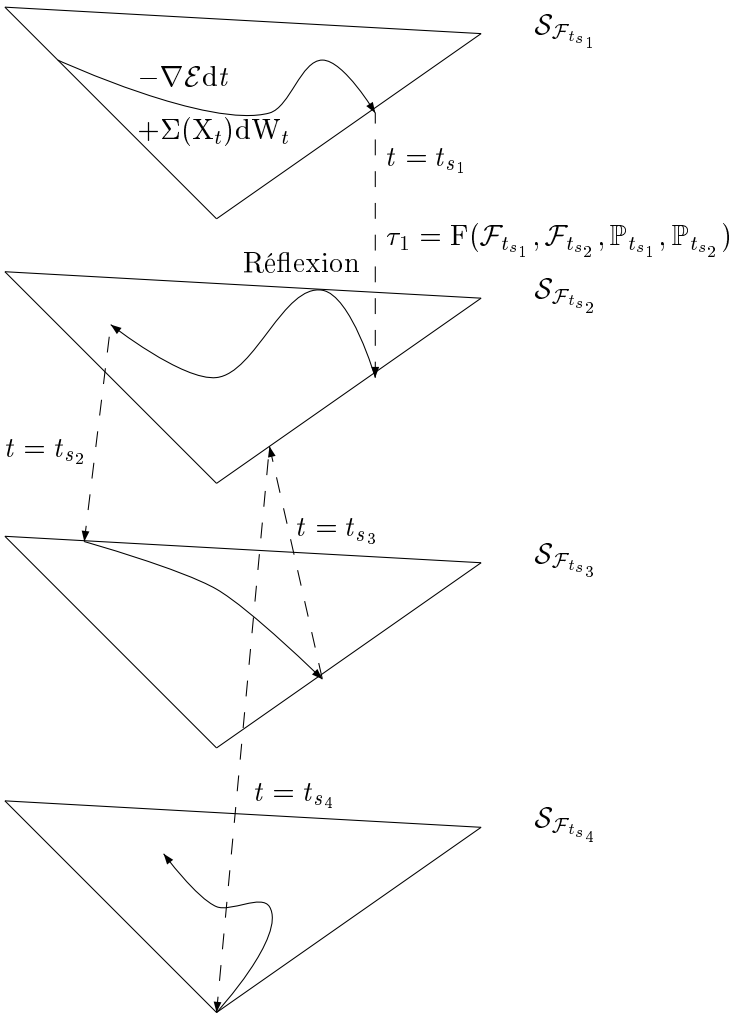
Le processus de diffusion, peut alors être vu comme une juxtaposition, au fil du temps, de plusieurs processus de diffusions réfléchies entre deux instants de sauts.

En fin de compte, la recherche de l'espace optimal avec la distribution de probabilité optimale se fait en utilisant le couplage des sauts et de la diffusion, chaque étape de cette recherche ne peut se faire indépendamment l'une de l'autre. Les sauts dépendent en fait de l'état de $\mathbb{P}$ aux temps de sauts tandis que la diffusion est fonction de l'espace dans lequel se trouve $\mathbb{P}$ (le bruit $\Sigma\mathrm{dW}_t$ dépend de la forêt au temps t).

On pourra néanmoins permettre à l'algorithme d'avoir différents types de comportements et se libérer de sa rigidité apparente.

- Si l'on souhaite obtenir plus de sauts, il suffit de proposer un paramètre λ pour la distribution Poissonnienne des sauts qui soit petit.
- Si l'on souhaite obtenir une forêt de cardinal petit, il suffit de proposer un gros coefficient devant le terme $\mathcal{E}_{\mathrm{L}g}$.
- Enfin, la structure de la forêt elle-même peut être influencée par des termes de redondance d'arbres , de complexité d'arbres etc. Un processus utilisant de tels termes est alors aisément implémentable et finalement assez peu couteux en temps de calcul pour chaque sauts puisqu'il n'y a que des différences d'énergie à calculer, et que ces différences ne font en général pas intervenir la structure totale de la forêt mais uniquement un arbre particulier.

Enfin, en guise d'illustration, nous pouvons représenter le type d'évolution possible de notre processus de diffusions avec sauts entre les espaces $\mathcal{S}_{\mathcal{F}(\mathrm{X})}$.



Lorsque le processus est en phase de diffusion, la trajectoire $(\mathbb{P}_t)_t$ est parfaitement continue dans chaque sous-espace $\mathcal{S}_{\mathcal{F}(\mathrm{X})}$. En revanche, lors des instants de sauts, le processus opère alors

des transitions brutales. Ces transitions sont alors conditionnées à un seuil d'acceptation de ces sauts donnés par les réels τ_{t_s} du paragraphe 5.6.3.

Chapitre 6 - Asymptotique du processus de diffusion réfléchie avec sauts

Dans ce chapitre, nous étudions précisément l'évolution infinitésimale du processus (son générateur Markovien du processus) et son évolution asymptotique.

Nous remarquons tout d'abord que le processus en question rentre parfaitement dans la théorie des processus Markoviens et nous rappelons alors quelques définitions sur ces processus.

Nous calculons ensuite le générateur infinitésimal du processus sur une classe réduite de fonctions vérifiant certaines conditions de type Neumann sur les frontières des simplexes $\mathcal{S}_{\mathcal{F}}$.

Enfin, nous démontrons que le processus suit un comportement « chain-recurrent », ce qui assure finalement que le processus ne reste pas confiné dans certaines zones d'exploration de $\mathcal{S}_{\mathcal{A}^\star}$, avant d'exhiber la mesure invariante de ce processus stochastique.

6.1 Étude infinitésimale sur le processus de diffusion sous contraintes avec sauts

6.1.1 Description du processus

Le théorème précédent (formules $(\mathbf{E}-\mathbf{8})$) définit un processus fortement markovien. Ce processus permet de :

- Simuler une diffusion réfléchie sur un espace $\mathcal{F}(\mathrm{X}_{t_i})$ dans lequel on cherche à faire baisser la valeur d'une énergie $\mathcal{E}$:

$$\forall t \in [\, t_i \,;\, t_{i+1} \,[\qquad \mathrm{d}\mathbb{P}_t = -\frac{\partial \mathcal{E}}{\partial \mathbb{P}}\mathrm{d}t + \Sigma(\mathrm{X}_t)\mathrm{dW}_t + \mathrm{dZ}_t \tag{6.37}$$

  Comme le terme d'énergie $\mathcal{E}$ qui fait intervenir $\mathbb{P}$ est restreint à $\mathcal{E}_{err}$, (6.37) devient alors :

$$\forall t \in [\, t_i \,;\, t_{i+1} \,[\qquad \mathrm{d}\mathbb{P}_t = -\Pi_{\mathcal{H}_t}\left(\nabla \mathcal{E}_{err}(\mathbb{P}_t)\right)\mathrm{d}t + \Sigma(\mathrm{X}_t)\mathrm{dW}_t + \mathrm{dZ}_t \tag{6.38}$$

- Changer l'espace $\mathcal{F}(\mathrm{X}_{t_i})$ en un nouvel espace $\mathcal{F}(\mathrm{X}_{t_{i+1}})$ afin de baisser le plus possible l'énergie $\mathcal{E}$ en utilisant les processus de sauts distribués en temps selon un processus de Poisson ([Bou00]) :

$$\forall i \in \mathbb{N} \qquad \mathrm{P}[t_{i+1} = t | t_i = s] = e^{-\lambda(t-s)} \tag{6.39}$$

  et un seuil d'acceptation du saut basée sur la diminution d'énergie $\Delta\mathcal{E}$ occasionnée par ce changement d'espace, l'acceptation d'un tel saut a alors pour probabilité :

$$\mathrm{Q}[\mathrm{X}_{t_{i+1}} = \mathrm{X} | \mathrm{X}_{t_i}] = \mathrm{Min}\left\{1; \tau_{\mathrm{X}_{t_i} \mapsto \mathrm{X}_{t_{i+1}}} e^{-\Delta\mathcal{E}(\mathrm{X}_{t_i} \mapsto \mathrm{X}_{t_{i+1}})}\right\} \tag{6.40}$$

Ce processus est donc parfaitement continu entre les sauts. Il suit une loi de diffusion standard lorsque le processus $\mathbb{P}$ n'est pas sur la frontière du $i^{\text{ème}}$ simplexe défini par X_{t_i+}, tandis que le processus de réflexion Z qui permet de restreindre $\mathbb{P}$ dans ce même simplexe n'augmente que lorsque $\mathbb{P}$ atteint une frontière. Lors des sauts, en revanche, le processus $\mathbb{P}$ peut présenter des discontinuités (coupe ou suppression d'arbres de probabilités strictement positives). Toujours lors de ces sauts (6.39), X est également discontinu, sauf si le saut n'est pas accepté (6.40).

6.1.2 Généralité sur les processus Markoviens

Le processus précédent $(\mathbb{P}_t; \mathrm{X}_t)_{t\in\mathbb{R}}$ est un processus Markovien défini sur Ω espace $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^{|\mathcal{A}^\star|}$ et muni de la filtration $\{\mathcal{G}_t\}$ pour laquelle $(\mathbb{P}_t; \mathrm{X}_t)$ est $\mathcal{G}_t$-mesurable en tout temps. Pour plus de simplicité, on notera dans un premier temps le processus Y, à valeurs dans E. Pour une définition précise des processus homogènes de Markov, on pourra se reporter à [RW00] ou à [SV79]. On note alors Q_y la loi associée au processus initialisé en y et le processus Y vérifie l'égalité :

$$\mathrm{Q}_y\left(\mathrm{Y}_{t+h} \in \Gamma | \mathcal{G}_t\right) = \mathrm{Q}_{\mathrm{Y}_t}\left(\mathrm{Y}_h \in \Gamma\right) \qquad \mathrm{Q}_y \quad p.s.$$

Si l'on note $\mathbb{E}_y$ l'espérance sous Q_y, on a alors pour toute fonction f appartenant à $\mathcal{E}$ (fonctions de E dans $\mathbb{R}$) :

$$\mathbb{E}_y\left[f(\mathrm{Y}_{t+h}) | \mathcal{G}_t\right] = \mathbb{E}_{\mathrm{Y}_t}\left[f(\mathrm{Y}_h)\right] \qquad \mathrm{Q}_y \quad p.s.$$

La définition 6.1.1 rappelle l'expression classique des fonctions de transitions des processus markoviens.

Définition 6.1.1 (Fonction de transition du processus)
On pose alors pour f dans $\mathcal{E}$:

$$\mathrm{P}_t f(y) = \mathbb{E}_y\left[f(\mathrm{Y}_t)\right] \tag{6.41}$$

On définit également pour $\Gamma \in \mathrm{E}$ et $y \in \mathrm{E}$:

$$\mathrm{Q}_t(y, \Gamma) = \mathrm{Q}_y(\mathrm{Y}_t \in \Gamma) \tag{6.42}$$

La famille $(\mathrm{Q}_t(y, dy), t, y, dy)$ s'appelle fonction de transition de Y et la proposition suivante assure que la connaissance d'une telle famille suffit à déterminer totalement la loi de Y :

Propriété 6.1.1
Pour tout y de E, tous temps $t_1 \leqslant \ldots \leqslant t_n$ et toute f de E^n dans $\mathbb{R}$:

$$\mathbb{E}_y\left[f(\mathrm{Y}_{t_1}, \ldots \mathrm{Y}_{t_n})\right] = \int \mathrm{Q}_{t_1}(y, \mathrm{d}y_1) \ldots \mathrm{Q}_{t_n}(y_{n-1}, \mathrm{d}y_n) f(y_1, \ldots y_n)$$

Cette propriété implique alors la relation fondamentale sur les fonctions de transitions P_t :

$$\mathrm{P}_{t+s} = \mathrm{P}_s \mathrm{P}_t = \mathrm{P}_t \mathrm{P}_s \tag{6.43}$$

Cette relation fait de $(\mathrm{P}_t, t \geqslant 0)$ un semi-groupe de Feller ([RW00],[EK86]) puisque si $\mathcal{C}_0(\mathrm{E}, \mathbb{R})$ désigne l'espace des fonctions continues de E tendant vers 0 à l'infini, alors on a la relation de continuité :

$$\forall f \in \mathcal{C}_0\left(\mathrm{E}, \mathbb{R}\right) \qquad \lim_{t\mapsto 0} \|\mathrm{P}_t f - f\| = 0 \tag{6.44}$$

où $\|.\|$ désigne la norme infinie sur E. On introduit alors le générateur infinitésimal du processus de Markov Y donné par la définition 6.1.2.

Définition 6.1.2 (Générateur infinitésimal)
Soit $(\mathrm{P}_t, t \geqslant 0)$ *un semi-groupe de Feller, on dit que* $f \in \mathcal{D}_{\mathrm{A}}$ *et que* $\mathrm{A}f = g$ *s'il existe* g *dans* $\mathcal{C}_0(\mathrm{E}, \mathbb{R})$ *telle que*

$$\lim_{t \mapsto 0} \left\| \frac{\mathrm{P}_t f - f}{t} - g \right\| = 0 \tag{6.45}$$

L'opérateur $(\mathcal{D}_{\mathrm{A}}, \mathrm{A})$ *est le générateur infinitésimal de* P_t.

Ce générateur du processus vérifie alors la relation qui relie l'évolution du processus Y à l'effet de son générateur A.

Propriété 6.1.2
Si $f \in \mathcal{D}_{\mathrm{A}}$, *alors*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{P}_t f = \mathrm{P}_t \mathrm{A} f$$

et

$$\mathrm{P}_t f(y) - f(y) = \int_0^t \mathrm{P}_s \mathrm{A} f(y) \mathrm{d}s$$

L'intérêt du générateur infinitésimal est donc qu'il donne une idée précise de la variation locale du processus Y. Son autre grand intérêt est qu'il permet de caractériser là ou les mesures invariantes du processus Y. Les définitions ainsi que le théorème illustrant cette idée seront mentionnés au paragraphe 6.5. Par ailleurs, on peut formuler le problème de la recherche du générateur infinitésimal en utilisant également la formulation du problème Martingale de Strook et Varadhan [SV79] de la définition 6.1.3.

Définition 6.1.3 (Problème Martingale pour (A, μ)**)**
Le processus $(\mathrm{Y}_t)_{t \geqslant 0}$ *est solution du problème Martingale pour* $(f, \mathrm{A}f = g)$ *s'il existe une filtration continue* $(\mathcal{G}_t)_{t \geqslant 0}$ *telle que*

$$f(\mathrm{Y}_t) - f(\mathrm{Y}_0) - \int_0^t g(\mathrm{Y}_s) \mathrm{d}s \tag{6.46}$$

est une $\mathcal{G}_t$*-martingale et si* Y_0 *a pour loi initiale* μ.

Nous introduisons par ailleurs la notion de problème bien posé pour la notion de problème martingale associé au générateur du processus.

Définition 6.1.4 (Problème Martingale bien posé)
Le problème martingale définit précédemment est dit bien posé pour un processus de générateur A *et de loi initiale* μ *sur un ensemble de fonctions* $\mathcal{Z}$ *si il existe un processus Markovien* $(\mathrm{Y}_t)_{t \geqslant 0}$ *vérifiant :*

$$f(\mathrm{Y}_t) - f(\mathrm{Y}_0) - \int_0^t g(\mathrm{Y}_s) \mathrm{d}s \tag{6.47}$$

est une $\mathcal{G}_s$*-martingale pour toute fonction* f *dans* $\mathcal{Z}$ *avec* $g = \mathrm{A}f$ *et si la loi du processus* $(\mathrm{Y}_t)_{t \geqslant 0}$ *est alors définie de façon unique.*

> Il s'agit de remarquer que la notion de « problème bien posé » dépend fortement de l'ensemble de fonctions tests $\mathcal{Z}$ auxquelles il est relatif. En effet, plus l'ensemble des fonctions $\mathcal{Z}$ est petit, plus il est difficile d'obtenir la propriété d'unicité en loi du processus solu-

tion, et donc le fait que le problème soit bien posé. Pour obtenir une telle propriété, il faudra donc veiller à ce que l'ensemble $\mathcal{Z}$ des fonctions caractérisant (6.46) soit suffisament riche pour assurer l'unicité en loi du processus solution. L'ensemble de fonctions $\mathcal{Z}$ « maximal » suffisant pour caractériser un tel processus en loi est bien entendu l'ensemble des fonctions mesurables. Ultérieurement, nous prendrons un sous-ensemble de fonctions mesurables sur lequel il sera plus aisé d'établir des propriétés sur le comportement infinitésimal du processus, nous étendrons alors ces propriétés en utilisant un raisonnement par densité.

Cette martingale permet en réalité de donner une autre définition du générateur infinitésimal : le générateur $(\mathcal{D}_{\mathrm{A}}, \mathrm{A})$ est aussi l'ensemble des fonctions f et g telles que (6.46) est une $\mathcal{G}_t$-martingale. Nous allons voir comment à partir d'une équation différentielle stochastique comme $(\mathbf{E}-\mathbf{8})$ il est possible de déterminer le générateur infinitésimal du processus.

6.1.3 Générateur du processus de diffusion sous contraintes

Le processus $\mathbb{P}$ (entre les sauts, seul $\mathbb{P}$ varie) appartient à un hyperplan $\mathcal{H}_{\mathrm{X}_t}$ et cet hyperplan ne variant pas au cours du temps entre deux sauts successifs, nous le noterons dans ce paragraphe plus simplement $\mathcal{H}$. Nous allons tout d'abord munir cet espace $\mathcal{H}$ d'opérateurs différentiels qui permettront de décrire plus facilement le générateur du processus sans saut dans cet espace. Nous noterons toujours $f = \dim \mathcal{H} + 1$.

6.1.3.1 Opérateurs différentiels sur $\mathcal{H}$

Nous munissons cet hyperplan de la métrique euclidienne standard et notons $\nabla^{\mathcal{H}}$ l'opérateur de gradient sur cet espace. On a la relation fondamentale donnée par la propriété 6.1.3.

Propriété 6.1.3 (Relation entre ∇ et $\nabla^{\mathcal{H}}$)
Si ∇ désigne le gradient standard dans $\mathbb{R}^f$ et $\Pi_{\mathcal{H}}$ la projection orthogonale sur $\mathcal{H}$, on a :

$$\nabla^{\mathcal{H}} = \Pi_{\mathcal{H}}\left(\nabla\right)$$

Preuve : Si f est une fonction $\mathcal{C}^1\left(\mathcal{H}, \mathbb{R}\right)$, $\nabla^{\mathcal{H}}$ est l'unique vecteur de $\overrightarrow{\mathcal{H}}$ tel que :

$$\forall \mathrm{X} \in \mathcal{H} \quad \forall \overrightarrow{u} \in \overrightarrow{\mathcal{H}} \qquad (\nabla^{\mathcal{H}} f(\mathrm{X}) \mid \overrightarrow{u}) = \lim_{\varepsilon \mapsto 0} \frac{f(\mathrm{X} + \varepsilon \overrightarrow{u}) - f(\mathrm{X})}{\varepsilon}$$

De plus $$\lim_{\varepsilon \mapsto 0} \frac{f(\mathrm{X} + \varepsilon \overrightarrow{u}) - f(\mathrm{X})}{\varepsilon} = (\nabla f(\mathrm{X}) \mid \overrightarrow{u})$$

Comme $\nabla f(\mathrm{X})$ se décompose en :

$$\nabla f(\mathrm{X}) = \underbrace{\Pi_{\mathcal{H}}(\nabla f(\mathrm{X}))}_{\in \overrightarrow{\mathcal{H}}} + \underbrace{(\nabla f(\mathrm{X}) \mid \overrightarrow{\mathrm{N}})\overrightarrow{\mathrm{N}}}_{\in \mathcal{H}^{\perp}}$$

On obtient donc

$$\forall \mathrm{X} \in \mathcal{H} \quad \forall \overrightarrow{u} \in \overrightarrow{\mathcal{H}} \qquad (\nabla f(\mathrm{X}) \mid \overrightarrow{u}) = (\Pi_{\mathcal{H}}(\nabla f(\mathrm{X})) \mid \overrightarrow{u}) + \underbrace{((\nabla f(\mathrm{X}) \mid \overrightarrow{\mathrm{N}})\overrightarrow{\mathrm{N}} \mid \overrightarrow{u})}_{=0 \quad \text{car} \quad (\overrightarrow{\mathrm{N}} \mid \overrightarrow{u}) = 0}$$

En conclusion $$\nabla^{\mathcal{H}} = \Pi_{\mathcal{H}}\left(\nabla\right)$$ □

Définition 6.1.5 (Divergence et Laplacien sur $\mathcal{H}$)
On définit également les opérateurs $\mathrm{div}^{\mathcal{H}}$ *et* $\Delta^{\mathcal{H}}$ *sur* $\mathcal{H}$ *par :*

$$\forall \mathrm{X} \in \mathcal{H} \quad \forall \mathrm{F} \in \mathcal{C}^0\left(\mathcal{H}, \overrightarrow{\mathcal{H}}\right) \qquad \mathrm{div}^{\mathcal{H}}(\mathrm{F}(\mathrm{X})) = \lim_{\mathrm{V} \mapsto 0 \quad \mathrm{X} \in \mathrm{V}} \frac{\displaystyle\int_{\partial \mathrm{V}} (\mathrm{F} \mid \mathrm{d}a)}{|\mathrm{V}|}$$

et

$$\Delta^{\mathcal{H}} = \mathrm{div}\left(\nabla^{\mathcal{H}}\right)$$

On peut alors exprimer cet opérateur en utilisant les coefficients de Beltrami [Hsu04], mais ce que nous retiendrons est que l'opérateur $\Delta^{\mathcal{H}}$ est l'opérateur du générateur infinitésimal d'une diffusion brownienne standard sur $\mathcal{H}$. Cette diffusion brownienne standard sur $\mathcal{H}$ est alors définie par l'équation différentielle stochastique [RW00] (V.30 tome 2) :

$$\mathrm{d}\mathrm{X}_t = \Sigma^{\mathcal{H}} \mathrm{d}\mathrm{W}_t$$

Ces résultats se résument dans notre cas particulier en la propriété 5.4.4 qui donne l'expression du générateur infinitésimal d'une diffusion brownienne standard sur $\mathcal{H}$.

Propriété 6.1.4 (Diffusion brownienne standard sur $\mathcal{H}$)
Si l'on désigne par $\Sigma^{\mathcal{H}}$ *la matrice donnée par (4.26), la solution de l'équation différentielle stochastique*

$$\mathrm{d}\mathrm{X}_t = \Sigma^{\mathcal{H}} \mathrm{d}\mathrm{W}_t \tag{6.48}$$

avec $\mathrm{X}_0 \in \mathcal{H}$ *est une diffusion brownienne standard sur* $\mathcal{H}$*. Son générateur est alors*

$$\mathrm{A}f = \frac{1}{2}\Delta^{\mathcal{H}} f \tag{6.49}$$

Preuve : En utilisant [RW00] (tome 2, page 114) et le fait que $\mathcal{H}$ est à courbure constante, on en déduit que $\mathrm{div}\ \overrightarrow{\mathrm{N}}(\mathrm{X}) = 0$.

L'équation différentielle (7.64) est donc l'équation d'une diffusion standard sur $\mathcal{H}$. Si $(\mathcal{D}_{\mathrm{A}}, \mathrm{A})$ désigne le générateur infinitésimal de X, on peut se référer à [Hsu04] ou [RW00] (tome 2, page 185) pour avoir que $\mathcal{D}_{\mathrm{A}} = \mathcal{C}^2(\mathcal{H}, \mathbb{R})$ et

$$\forall f \in \mathcal{D}_{\mathrm{A}} \qquad \mathrm{A}f = \frac{1}{2}\Delta^{\mathcal{H}} f \qquad \square$$

6.1.3.2 Calcul du générateur du processus de diffusion réfléchie

On numérote les arbres $\mathcal{A}$ possibles de $\mathcal{A}^\star$ de 1 à n et on renomme les variables $\mathbb{P}(\mathcal{A})$ pour des facilités de notation en $x_1, \ldots x_n$, ces variables décrivent donc $\mathcal{S}_{\mathcal{A}^\star}$ où $n = |\mathcal{A}^\star|$. On étudie tout d'abord le générateur du processus (entre les instants de sauts) défini par $(\mathbf{E}-\mathbf{8})$ en découpant l'équation

$$\mathrm{d}\mathbb{P}_t = \underbrace{-\Pi_{\mathcal{H}}\left(\nabla \mathcal{E}_{err}(\mathbb{P}_t)\right)\mathrm{d}t}_{=\mathrm{A}_2} + \underbrace{\Sigma(\mathrm{X}_t)\mathrm{d}\mathrm{W}_t}_{=\mathrm{A}_1} + \underbrace{\mathrm{d}\mathrm{Z}_t}_{=\mathrm{A}_3}$$

en trois termes $\mathrm{A}_1, \mathrm{A}_2$ et A_3.

6.1.3.3 Termes A_1 et A_2 :

Propriété 6.1.5 (Générateur du terme de diffusion réfléchie)
Le générateur du terme issu de $A_1 + A_2$ *est le générateur classique des diffusions sans contrainte dans* $\mathcal{H}_{t_i}$ *où* $\mathcal{H}_{t_i}$ *est l'hyperplan désigné par la quantité* X_t *constante entre deux sauts,* $t \in [\, t_i \,;\, t_{i+1} \,[$.

Preuve : Le générateur de la partie donnée par $A_1 + A_2$ se calcule en appliquant la formule d'Itô pour f de classe $\mathcal{C}^2$ sur $\mathcal{H}$ [RY94], [EK86] au processus Y solution de l'équation différentielle stochastique :

$$dY_t = -\nabla^{\mathcal{H}} \mathcal{E}_{err}(Y_t)dt + \Sigma(X_t)dW_t \tag{6.50}$$

Σ étant constante entre deux sauts, réduite à (4.26) sur l'espace $\mathcal{F}(X_{t_i+})$, il suffit d'utiliser la formule d'Itô pour $f \in \mathcal{C}^2(\mathcal{H}, \mathbb{R})$

$$f(Y_t) - f(Y_{t_i}) = \int_{t_i}^{t} (\nabla f(Y_s) \mid dY_s) + \frac{1}{2} \sum_{i,j} \int_0^t \frac{\partial^2 f}{\partial x_i \partial x_j}(Y_s) \sum_k \Sigma_{i,k} \Sigma_{k,j} ds$$

En utilisant la propriété 6.1.4, on sait alors que :

$$\lim_{t \mapsto t_i} \frac{1}{2} \frac{1}{t - t_i} \mathbb{E}\left[\frac{1}{2} \sum_{i,j} \int_{t_i}^t \frac{\partial^2 f}{\partial x_i \partial x_j}(Y_s) \sum_k \Sigma_{i,k} \Sigma_{k,j} ds \right] = \frac{1}{2} \Delta^{\mathcal{H}} f(Y_{t_i})$$

puisque cette limite correspond en fait au calcul du générateur infinitésimal d'un processus de diffusion standard sur $\mathcal{H}$ sans terme de dérive.

Par ailleurs, on a :

$$\int_{t_i}^t \nabla f(Y_s) dY_s = \int_{t_i}^t - (\nabla f(Y_s) \mid \nabla^{\mathcal{H}} \mathcal{E}_{err}(Y_s)) ds + \int_{t_i}^t (\nabla f(Y_s) \mid \Sigma dW_s)$$

Comme $$(\nabla f(Y_s) \mid \nabla^{\mathcal{H}} \mathcal{E}_{err}(Y_s)) = (\nabla^{\mathcal{H}} f(Y_s) \mid \nabla^{\mathcal{H}} \mathcal{E}_{err}(Y_s))$$

alors

$$\begin{aligned} \mathbb{E}\left[\int_{t_i}^t (\nabla f(Y_s) \mid \Sigma dW_s) \right] &= \mathbb{E}\left[\int_{t_i}^t \mathbb{E}[(\nabla f(Y_s) \mid \Sigma dW_s) | \mathcal{G}_s] \right] \\ &= \mathbb{E}\left[\int_{t_i}^t (\nabla f(Y_s) \mid \mathbb{E}[\Sigma dW_s | \mathcal{G}_s]) \right] \\ \mathbb{E}\left[\int_{t_i}^t (\nabla f(Y_s) \mid \Sigma dW_s) \right] &= 0 \end{aligned}$$

En prenant la limite lorsque t tend vers t_i du terme

$$\frac{\mathbb{E}[f(Y_t) - f(Y_{t_i})]}{t - t_i}$$

on obtient $$\underbrace{(A_1 + A_2)}_{=\mathcal{A}} f = \frac{1}{2} \Delta^{\mathcal{H}} f - (\nabla^{\mathcal{H}} \mathcal{E}_e rr \mid \nabla^{\mathcal{H}} f)$$ □

6.1.3.4 Terme A_3

Le générateur de la partie A_3 se calcule comme dans [KS01]. Le générateur d'un tel processus est, toujours, en utilisant la linéarité de $(\mathbf{E}-\mathbf{8})$, le générateur du terme de diffusion plus le générateur du processus de réflexion correspondant au processus $(Z_t)_{t\in\mathbb{R}}$, mais ce générateur s'exprime en réalité sur le couple $(\mathbb{P}_t, Z_t)_{t\in\mathbb{R}}$. Nous pouvons envisager deux méthodes pour traiter ce terme A_3.

1. On peut étudier le générateur du terme de réfléxion sur l'ensemble des fonctions f de classe $\mathcal{C}^2$ de $\mathcal{H}$ dans $\mathbb{R}$. En général, ce terme fait alors intervenir le temps local du processus $(\mathbb{P}_t)_{t\geqslant 0}$ sur la frontière du domaine [Abr00].
2. On peut étudier le générateur sur un domaine plus petit, de fonctions $\mathcal{C}^2$ de $\mathcal{S}_\mathcal{H}$ dans $\mathbb{R}$ vérifiant des conditions de Neumann sur $\partial\mathcal{S}_\mathcal{H}$

Dans le cas où l'on cherche à calculer le générateur sur le domaine des fonctions de $\mathcal{C}^2(\mathcal{H},\mathbb{R})$, le générateur est relativement difficile à exhiber par une formule générale dans notre situation précise au vu des travaux de [KS01]. Si le terme de réflexion dépend continûment du point sur $\partial\mathcal{S}_\mathcal{H}$ et que l'équation $(\mathbf{E}-\mathbf{8})$ se met sous la forme

$$\mathrm{d}\mathbb{P}_t = \underbrace{-\nabla^\mathcal{H}\mathcal{E}_{err}(\mathbb{P}_t) + \Sigma(\mathcal{H})\mathrm{dW}_t}_{\text{générateur } \mathcal{A}} + \underbrace{m(\mathbb{P}_{t^-})\mathrm{d}\xi_t}_{\text{générateur } \mathcal{B}} \tag{6.51}$$

où $$m \in \mathcal{C}^0\left(\partial\mathcal{S}_\mathcal{H}, \overrightarrow{\mathcal{S}_\mathcal{H}}\right)$$

avec ξ un processus croissant, augmentant uniquement quand $\mathbb{P}$ atteint $\partial\mathcal{S}_\mathcal{H}$, le générateur s'exprime alors en

$$\forall f \in \mathcal{D}(\mathcal{B}) \qquad \mathcal{B}f = (\nabla^\mathcal{H} f \mid m)$$

Mais ici, la direction de réflexion dépend, en réalité, de la direction dans laquelle est heurtée la frontière ([KS01]). Ce mode de réflexion fait donc intervenir un paramètre de contrôle dans l'équation différentielle stochastique (6.51) qui devient :

$$\mathrm{d}\mathbb{P}_t = \underbrace{-\nabla^\mathcal{H}\mathcal{E}_{err}(\mathbb{P}_t) + \Sigma(\mathcal{H})\mathrm{dW}_t}_{\text{générateur } \mathcal{A}} + \underbrace{m(\mathbb{P}_{t^-}, u_t)\mathrm{d}\xi_t}_{\text{générateur } \mathcal{B}}$$

Le générateur du processus est alors plus difficile à expliciter ([KS01] paragraphe 1.1) mais nous retiendrons que si $\mathcal{B}$ est le générateur du processus, on a

$$\forall f \in \mathcal{D}(\mathcal{B}) \quad \nabla f_{\big|\partial\mathcal{S}_\mathcal{H}} = 0 \qquad \mathcal{B}(f) = 0$$

Ceci nous amène donc à considérer un domaine plus petit pour le générateur du processus, constitué de fonctions $\mathcal{C}^2$ de $\mathcal{S}_\mathcal{H}$ dans $\mathbb{R}$ satisfiant des conditions de Neumann aux frontières du simplexe. Autrement dit, le générateur du processus total entre les sauts coïncide exactement avec le générateur du processus de diffusion sans réflexion (diffusion libre) sur l'ensemble des fonctions de classe $\mathcal{C}^2(\mathcal{S}_\mathcal{H},\mathbb{R})$, avec une condition de Neumann $\nabla^\mathcal{H} f = 0$ sur la frontière du simplexe.

Propriété 6.1.6 (Générateur du processus réfléchi sans saut)
Si $\mathcal{L}_1$ désigne le générateur infinitésimal du processus $(\mathbb{P}_t)_{t\in\mathbb{R}}$ entre deux sauts consécutifs,

$$\forall f \in \mathcal{C}^2(\mathcal{S}_\mathcal{H},\mathbb{R}) \cap \{\nabla^\mathcal{H} f_{\big|\partial\mathcal{S}_\mathcal{H}} = 0\} \qquad \mathcal{L}_1 f = -(\nabla^\mathcal{H}\mathcal{E}_{err} \mid \nabla^\mathcal{H} f) + \frac{1}{2}\Delta^\mathcal{H}$$

6.1.4 Générateur du processus de diffusions sous contraintes avec sauts

Le calcul du générateur du processus de diffusions sous contraintes avec sauts est issu du calcul du générateur infinitésimal $\mathcal{L}$ précédent, auquel on rajoute le terme correspondant aux sauts. Il suffit donc de calculer le générateur du processus de saut pour obtenir le générateur du processus total :

Propriété 6.1.7 (Générateur du processus de saut)
Si $\mathcal{L}_2$ désigne le générateur infinitésimal du processus de sauts basé sur la probabilité de transition $\mathrm{Q}(.|.)$ et $\mathcal{D}(\mathcal{L}_2)$ son domaine de définition. Si de plus les simplexes $\mathcal{S}_{\mathcal{H}}$ sont numérotés en $\mathcal{S}_1, \dots \mathcal{S}_f$, alors

$$\mathcal{D}\left(\mathcal{L}_2\right) = \left\{ \begin{array}{ll} f \in \mathcal{F}\left(\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f, \mathbb{R}\right) & \left| \; f\left(\mathbb{P}, \mathrm{X}\right) = \sum_{i=1}^{f} \chi_{\mathcal{H}(\mathrm{X})=\mathcal{H}_i} f_i\left(\mathbb{P}, \mathrm{X}\right) \right. \\ & \\ & \forall i \in [\![1\,;\, f]\!] \qquad f_i \in \mathcal{C}^2\left(\mathcal{S}_{\mathcal{H}_i} \times \{0;1\}^f, \mathbb{R}\right) \end{array} \right\}$$

Le processus de saut basé sur la probabilité de transition $\mathrm{Q}(.|.)$ vaut alors :

$$\forall f \in \mathcal{D}(\mathcal{L}_2) \quad \forall \left(\mathbb{P}, \mathrm{X}\right) \in \mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f \qquad \mathcal{L}_2 f\left(\mathbb{P}, \mathrm{X}\right) = \int_{\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f} \left[f(y) - f\left(\mathbb{P}, \mathrm{X}\right)\right] \mathrm{Q}\left(\left(\mathbb{P}, \mathrm{X}\right) | \mathrm{d}y\right)$$

Preuve : On pourra se référer à [EK86] (page 266) et [GM94] section 3. □

En utilisant les propriétés 6.4.6 et 6.4.7, et par linéarité du calcul du générateur infinitésimal, on en déduit le théorème 6.1.1.

Théorème 6.1.1 (Générateur de $(\mathbf{E}-\mathbf{8})$)
Supposons que $(\mathbb{P}_t, \mathrm{X}_t)_{t\in\mathbb{R}}$ satisfait le système $(\mathbf{E}-\mathbf{8})$, le générateur infinitésimal du processus restreint à l'espace $\mathbb{D}$ donné par :

$$\mathbb{D} = \left\{ \begin{array}{ll} f \in \mathcal{F}\left(\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f, \mathbb{R}\right) & \left| \; f\left(\mathbb{P}, \mathrm{X}\right) = \sum_{i=1}^{f} \chi_{\mathcal{H}(\mathrm{X})=\mathcal{H}_i} f_i\left(\mathbb{P}, \mathrm{X}\right) \right. \\ & \\ & \forall i \in [\![1\,;\, f]\!] \qquad f_i \in \mathcal{C}^2\left(\mathcal{S}_{\mathcal{H}_i} \times \{0;1\}^f, \mathbb{R}\right) \\ & \\ & \forall i \in [\![1\,;\, f]\!] \qquad \nabla^{\mathcal{H}_i} f_i\big|_{\partial\mathcal{S}_{\mathcal{H}_i}} = 0 \end{array} \right\}$$

vaut

$$\mathcal{L}(f) = \sum_{i=1}^{f} \mathcal{L}_1(f_i) + \mathcal{L}_2(f)$$

6.2 Dynamique du processus

Nous allons enfin brièvement étudier le comportement dynamique asymptotique du processus défini par les équations $(\mathbf{E}-\mathbf{8})$.

Dans un premier temps, nous étudierons uniquement la dynamique d'un processus de diffusion sous contraintes (processus réfléchi) dans un seul simplexe ; il n'y aura donc pas de termes de sauts envisagés dans ce cas là. Puis, les termes de sauts seront pris en compte pour démontrer finalement que le processus défini en $(\mathbf{E}-\mathbf{8})$ est récurrent.

Dans un second temps, nous donnerons l'existence et l'unicité de la mesure invariante d'un tel processus avant de finalement exprimer cette mesure comme le champ de Gibbs associé à l'énergie initiale $\mathcal{E}$.

6.2.1 Processus markovien récurrent

Un processus markovien récurrent est un processus qui peut atteindre n'importe quel ensemble de mesure de Lebesgue strictement positive depuis un état quelconque au temps t. Cela se traduit alors par la définition suivante.

Définition 6.2.1 (Processus Récurrent, Récurrent positif)
Un processus $(\mathrm{Y}_t)_{t\in\mathbb{R}}$ *à valeurs dans* G *est récurrent si pour tout compact* S *tel que* $\lambda(\mathrm{S}) > 0$ *(où* λ *mesure de Lebesgue sur* G*), on a :*

$$\tau_{\mathrm{S}} < +\infty \quad p.s.$$

où τ_{S} *est le temps d'atteinte de* S *par le processus* Y*. Il est récurrent positif si l'on a en plus*

$$\mathbb{E}_y\left[\tau_{\mathrm{S}}\right] < +\infty$$

Enfin, nous dirons que le processus Y *est uniformément récurrent positif si*

$$\underset{y\in\mathrm{G}}{\mathrm{Sup}}\ \mathbb{E}_y\left[\tau_{\mathrm{S}}\right] < +\infty$$

Nous allons établir que le processus défini par $(\mathbf{E}-\mathbf{6})$ vérifie de telles propriétés. Le résultat fondateur qui va permettre d'exprimer un tel résultat apparaît dans [HW92] puis a été utilisé dans [ABD01] pour démontrer la récurrence d'un processus réfléchi sous des conditions géométriques générales. Dans notre situation, les deux arguments qui permettront de conclure la récurrence du processus sans saut sont basés sur le caractère défini positif sur $\mathcal{S}_{\mathcal{H}}$ de la diffusion, et sur la compacité du simplexe dans lequel vit le processus $\mathbb{P}$.

6.2.1.1 Étude du processus sans saut

On se place donc dans le cas où le processus est confiné à un seul simplexe $\mathcal{S}_{\mathcal{H}}$. L'équation différentielle est donc (6.50). On remarque tout d'abord que l'opérateur du second ordre de la diffusion est uniformément non-dégénéré sur $\mathcal{S}_{\mathcal{H}}$. En effet, si $\overrightarrow{\mathrm{N}_{\mathcal{H}}}$ désigne le vecteur normal à $\mathcal{H}$, d'après la définition de σ en (4.26) :

$$\begin{aligned}
\forall \mathbb{P}\in\mathcal{S}_{\mathcal{H}} \qquad \mathbb{P}^t\sigma(\mathcal{H})\mathbb{P} &= \mathbb{P}^t\left(\mathrm{Id}\ -\overrightarrow{\mathrm{N}_{\mathcal{H}}}\overrightarrow{\mathrm{N}_{\mathcal{H}}}^t\right)\mathbb{P} \\
&= \mathbb{P}^t\left(\mathbb{P} - \overrightarrow{\mathrm{N}_{\mathcal{H}}}(\overrightarrow{\mathrm{N}_{\mathcal{H}}}\mid\mathbb{P})\right) \\
&= \|\mathbb{P}\|_2{}^2 - \underbrace{(\mathbb{P}\mid\overrightarrow{\mathrm{N}_{\mathcal{H}}})^2}_{=0 \quad \text{car} \quad \mathbb{P}\in\mathcal{S}_{\mathcal{H}}}
\end{aligned}$$

Finalement, on démontre la propriété :

Propriété 6.2.1 (Non-dégénérescence de la diffusion sans saut sur $\mathcal{S}_{\mathcal{H}}$)
Le processus défini par (6.50) est non-dégénéré sur $\mathcal{S}_{\mathcal{H}}$ *et de plus :*

$$\forall \mathbb{P}\in\mathcal{S}_{\mathcal{H}} \qquad \mathbb{P}^t\sigma(\mathcal{H})\mathbb{P} = \|\mathbb{P}\|_2{}^2$$

Une telle propriété peut s'interpréter également plus simplement comme le caractère elliptique de l'opérateur $\Delta^{\mathcal{H}}$ sur $\mathcal{H}$ et $\mathcal{S}_{\mathcal{H}}$.

Moyennant une telle hypothèse sur $\sigma(\mathcal{H})$, on peut alors énoncer le théorème :

Théorème 6.2.1 (Temps d'atteinte d'un compact de $(\mathbb{P}_t)_{t\geqslant 0}$, (Harrison-Williams))
Si S *est un compact de* $\mathcal{S}_{\mathcal{H}}$ *de mesure de Lebesgue strictement positive*

$$\lambda(\mathrm{S}) > 0$$

et si τ_{S} *désigne le temps d'atteinte (du processus donné par (6.50)) de* S*, on a alors pour tout compact* K *de* $\mathcal{S}_{\mathcal{H}}$ *:*

$$\underset{x\in \mathrm{K}}{\mathrm{Inf}}\, \mathrm{P}_x\left(\tau_{\mathrm{S}} \leqslant 1\right) > 0$$

Preuve : On se référera à l'article [HW92], partie 7, théorème 1 ainsi qu'à [ABD01] théorème 2.2. □.

Le résultat précédent peut être également substitué par

$$\underset{x\in \mathrm{K}}{\mathrm{Inf}}\, \mathrm{P}_x\left(\mathbb{P}(1) \in \mathrm{S}\right) > 0$$

c'est-à-dire : la probabilité d'atteindre un compact K quelconque de mesure de Lebesgue strictement positive en $t = 1$ est strictement positive. C'est précisément cette formulation que nous allons utiliser par la suite.

Ce résultat nous permet alors d'établir le théorème qui assure la récurrence du processus $\mathbb{P}$ donné par (6.50) dans $\mathcal{S}_{\mathcal{H}}$. Ce résultat utilise des outils classiques tels que la propriété de Markov forte sur le processus $\mathbb{P}$. La démonstration est très largement inspirée du théorème 2.2 de [ABD01] mais est ici présentée dans un cadre plus simple grâce à la compacité de $\mathcal{S}_{\mathcal{H}}$.

Théorème 6.2.2 (Récurrence du processus $\mathbb{P}$)
Le processus $\mathbb{P}$*, unique solution du système couplé d'équations différentielles stochastiques*

$$\begin{cases} \mathrm{Z}_t = -\displaystyle\int_0^t \nabla^{\mathcal{H}} \mathcal{E}_{err}(\mathbb{P}_s)\mathrm{d}s + \int_0^t \sigma(\mathcal{H})\mathrm{dW}_s \\ \mathbb{P}_t = \Gamma_{\mathcal{S}_{\mathcal{H}}}(\mathrm{Z}_t) \end{cases}$$

est un processus uniformément récurrent positif dans $\mathcal{S}_{\mathcal{H}}$*.*

Preuve : Donnons-nous S un compact de $\mathcal{S}_{\mathcal{H}}$ de mesure de Lebesgue strictement positive, il s'agit de démontrer que

$$\underset{x\in \mathcal{S}_{\mathcal{H}}}{\mathrm{Sup}}\, \mathbb{E}_x\left[\tau_{\mathrm{S}}\right] < +\infty \tag{6.52}$$

D'après le théorème 6.2.1, et par compacité de $\mathcal{S}_{\mathcal{H}}$, on a

$$\underset{x\in \mathcal{S}_{\mathcal{H}}}{\mathrm{Inf}}\, \mathrm{P}_x(\mathbb{P}(1) \in \mathrm{S}) > 0$$

On pose alors

$$p(\mathrm{S}) = \underset{x\in \mathcal{S}_{\mathcal{H}}}{\mathrm{Inf}}\, \mathrm{P}_x(\mathbb{P}(1) \in \mathrm{S}) > 0 \tag{6.53}$$

Notons $\widetilde{\tau} = \mathrm{Min}\left\{1; \tau_{\mathrm{S}}\right\}$, la propriété de Markov forte implique alors que

$$\forall x \in \mathcal{S}_{\mathcal{H}} \qquad \mathbb{E}_x\left[\tau_{\mathrm{S}}\right] = \mathbb{E}_x\left[\mathbb{E}_x\left[\tau_{\mathrm{S}} | \mathcal{G}_{\widetilde{\tau}}\right]\right]$$

$$\leqslant \mathbb{E}_x\left[\widetilde{\tau} + \mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right]\right]$$

Comme $\widetilde{\tau}$ est inférieur ou égal à 1 par définition, on a donc

$$\mathbb{E}_x\left[\tau_{\mathrm{S}}\right] \leqslant 1 + \mathbb{E}_x\left[\mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right]\right] \tag{6.54}$$

Si l'on note Λ l'ensemble des trajectoires continues dans $\mathcal{S}_{\mathcal{H}}$ atteignant S au temps 1, on a donc :

$$\Lambda = \{\mathrm{Y} \in \mathcal{C}([\,0\,;+\infty\,[\,, \mathcal{S}_{\mathcal{H}}) \quad | \quad \mathrm{Y}(1) \in \mathrm{S}\}$$

Étudions dès lors les deux cas :

- Si $\mathbb{P} \in \Lambda$, on a $\tau_{\mathrm{S}} \leqslant 1$ et $\widetilde{\tau} = \tau_{\mathrm{S}}$.

  Ainsi $$\mathbb{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right] = 0$$

- Sinon $\mathbb{P} \notin \Lambda$ et

$$\mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right] = \chi_{\mathbb{P} \notin \Lambda} \mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right]$$

On a donc finalement :

$$\mathbb{E}_x\left[\mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right]\right] = \mathbb{E}_x\left[\chi_{\mathbb{P} \notin \Lambda} \mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right]\right] \tag{6.55}$$

Mais

$$\mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right] \leqslant \operatorname*{Sup}_{x \in \mathcal{S}_{\mathcal{H}}} \mathrm{E}_x\left[\tau_{\mathrm{S}}\right] \qquad p.s. \tag{6.56}$$

(6.55) et (6.56) impliquent alors que

$$\mathbb{E}_x\left[\mathrm{E}_{\mathbb{P}_{\widetilde{\tau}}}\left[\tau_{\mathrm{S}}\right]\right] \leqslant \mathrm{P}\left[\mathbb{P} \notin \Lambda\right] \operatorname*{Sup}_{x \in \mathcal{S}_{\mathcal{H}}} \mathrm{E}_x\left[\tau_{\mathrm{S}}\right] \tag{6.57}$$

Par ailleurs, au vu de la définition (6.53) et de celle de Λ, on a

$$\mathrm{P}_x\left(\mathbb{P} \in \Lambda\right) \geqslant p(\mathrm{S})$$

D'où $$\mathrm{P}\left[\mathbb{P} \notin \Lambda\right] \leqslant 1 - p(\mathrm{S})$$

En utilisant alors (6.54) ainsi que (6.57), on obtient

$$\mathbb{E}_x\left[\tau_{\mathrm{S}}\right] \leqslant 1 + (1 - p(\mathrm{S})) \operatorname*{Sup}_{y \in \mathcal{S}_{\mathcal{H}}} \mathrm{E}_y\left[\tau_{\mathrm{S}}\right]$$

En prenant alors le Sup sur x dans l'inégalité précédente, on en déduira

$$\operatorname*{Sup}_{x \in \mathcal{S}_{\mathcal{H}}} \mathbb{E}_x\left[\tau_{\mathrm{S}}\right] \leqslant 1 + (1 - p(\mathrm{S})) \operatorname*{Sup}_{x \in \mathcal{S}_{\mathcal{H}}} \mathrm{E}_x\left[\tau_{\mathrm{S}}\right]$$

Ce qui se réécrit en

$$\operatorname*{Sup}_{x \in \mathcal{S}_{\mathcal{H}}} \mathbb{E}_x\left[\tau_{\mathrm{S}}\right] \leqslant \frac{1}{p(\mathrm{S})} \leqslant +\infty \qquad \square$$

De ce résultat fondamental sur le comportement du processus au sein de chaque simplexe $\mathcal{S}_{\mathcal{H}}$, on en déduit le résultat global concernant alors le processus avec sauts sur $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$.

6.2.1.2 Étude du processus avec sauts

Le comportement du processus couplé $(\mathbb{P}_t, \mathrm{X}_t)_{t\geqslant 0}$ se déduit du paragraphe précédent. En effet, nous avons vu au chapitre 5 que la dynamique des sauts permettait de couvrir l'ensemble des forêts possibles. Il en est donc de même de la dynamique de la chaîne du processus $(\mathrm{X}_{t_i})_{i\in\mathbb{N}}$, t_i désignant alors l'ensemble des instants de sauts du processus. Nous allons baser notre démonstration sur une minoration du type (6.53) :

$$\inf_{(\mathbb{P},\mathrm{X})\in\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^f} \mathrm{P}_{(\mathbb{P},\mathrm{X})}\left[\tau_{\mathrm{S}} \leqslant 1\right] > 0 \tag{6.58}$$

Il est alors nécessaire de définir les quantités :

Définition 6.2.2 (Minorant du temps d'atteinte dans un simplexe)
Pour $\mathcal{S}(\mathrm{X})$ un simplexe quelconque issu de la donnée d'un vecteur X de $\{0;1\}^f$, et pour T strictement positif, on définit pour tout compact S de $\mathcal{S}_{\mathcal{X}}$:

$$p_{\mathrm{X},\mathrm{T}}(\mathrm{S}) = \inf_{\mathbb{P}\in\mathcal{S}(\mathrm{X})} \mathrm{P}_{\mathbb{P}}\left[\tau_{\mathrm{S}} = \mathrm{T}\right] \tag{6.59}$$

Nous savons, d'après le théorème 6.2.1 ([HW92]), que les quantités $p_{\mathrm{X},\mathrm{T}}(\mathrm{S})$ sont strictement positives, dès que la mesure de Lebesgue dans $\mathcal{S}(\mathrm{X})$, $\lambda_{\mathrm{X}}(\mathrm{S})$, est strictement positive, et ce pour tout T strictement positif. Nous allons établir la propriété qui donne une minoration de la probabilité de passer de $(\mathbb{P}_0, \mathrm{X}_0)$ à $(d\mathbb{P}, \mathrm{Y})$, en fonction de la longueur du chemin minimal permettant d'effectuer la transition $\mathrm{X}_0 \longmapsto \mathrm{Y}$.

Propriété 6.2.2
Supposons donnés $(\mathbb{P}_0, \mathrm{X}_0)$ un état initial du processus et $(d\mathbb{P}, \mathrm{Y})$ de mesure strictement positive tels que le chemin minimal pour effectuer la transition $\mathrm{X} \longmapsto \mathrm{Y}$ est de longueur N :

$$\mathrm{X}_0 \longmapsto \mathrm{X}_1 \longmapsto \ldots \longmapsto \mathrm{X}_{\mathrm{N}} = \mathrm{Y}$$

on a alors la minoration :

$$\inf_{\mathbb{P}_0\in\mathcal{S}(\mathrm{X}_0)} \mathrm{P}_{(\mathbb{P}_0,\mathrm{X}_0)}\left[\tau_{\mathrm{d}\mathbb{P},\mathrm{Y}} \leqslant \mathrm{T}\right] \geqslant (\eta\lambda)^{\mathrm{N}}\, m_{\mathrm{N}} \tag{6.60}$$

avec

$$\begin{cases} m_{\mathrm{N}} > 0 \\ \eta = \underset{\substack{(\mathrm{P}_1,\mathrm{Z}_1)(\mathrm{P}_2,\mathrm{Z}_2)\in\left(\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^f\right)^2 \\ \mathrm{Z}_1 \underbrace{\longmapsto}_{\text{transition possible}} \mathrm{Z}_2}}{\mathrm{Min}} \mathrm{Q}\left(\begin{pmatrix}\mathrm{P}_1\\ \mathrm{Z}_1\end{pmatrix};\begin{pmatrix}\mathrm{P}_2\\ \mathrm{Z}_2\end{pmatrix}\right) \end{cases}$$

Preuve : Pour minorer la probabilité d'atteinte de $(\mathrm{d}\mathbb{P}, \mathrm{Y})$, on découpe l'intervalle de temps T en 2N parties égales telles que :

$$\underbrace{\left[0\,;\frac{\mathrm{T}}{2\mathrm{N}}\right]}_{\text{Diffusion sur}\mathrm{X}_1} \longmapsto \underbrace{\left[\frac{\mathrm{T}}{2\mathrm{N}}\,;\frac{2\mathrm{T}}{2\mathrm{N}}\right]}_{\text{Saut 1}} \longmapsto \underbrace{\left[\frac{2\mathrm{T}}{2\mathrm{N}}\,;\frac{3\mathrm{T}}{2\mathrm{N}}\right]}_{\text{Diffusion sur}\mathrm{X}_2} \longmapsto \underbrace{\left[\frac{3\mathrm{T}}{2\mathrm{N}}\,;\frac{4\mathrm{T}}{2\mathrm{N}}\right]}_{\text{Saut 2}} \longmapsto \ldots$$

On sait alors que si $\mathbb{P}_0$ désigne un point de départ du processus, la probabilité d'atteindre $(\mathrm{d}\mathbb{P}, \mathrm{Y})$ est supérieure à la probabilité d'atteindre ce même point dans la configuration où les sauts sont disposés comme précédemment. En fin de compte, on a

$$\begin{aligned}\mathrm{P}_{(\mathbb{P}_0,\mathrm{X}_0)}\left[\tau_{\mathrm{d}\mathbb{P},\mathrm{Y}} \leqslant \mathrm{T}\right] \geqslant & \int_{\mathrm{T/2N}}^{\mathrm{2T/2N}} \int_{\mathrm{P}_1 \in \mathcal{S}(\mathrm{X}_1)} \mathrm{P}_{\mathbb{P}_0,\mathrm{X}_0}\left[\tau_{\mathrm{dP1},\mathrm{X}_1} = t_1\right] \lambda e^{-\lambda t_1} \mathrm{d}t_1 \mathrm{Q}\left(\begin{pmatrix}\mathbb{P}_1 \\ \mathrm{X}_1\end{pmatrix}; \begin{pmatrix}\pi_{\mathrm{X}_1}(\mathbb{P}_1) \\ \mathrm{X}_2\end{pmatrix}\right) \\ & \int_{\mathrm{3T/N}}^{\mathrm{4T/N}} \int_{\mathrm{P}_2 \in \mathcal{S}(\mathrm{X}_2)} \mathrm{P}_{\pi(\mathbb{P}_1),\mathrm{X}_2}\left[\tau_{\mathrm{dP2},\mathrm{X}_2} = t_2 - t_1\right] \lambda e^{-\lambda(t_2-t_1)} \mathrm{d}t_2 \\ & \ldots\end{aligned}$$

En utilisant l'expression de (6.59), on peut alors établir que

$$\mathrm{P}_{(\mathbb{P}_0,\mathrm{X}_0)}\left[\tau_{\mathrm{d}\mathbb{P},\mathrm{Y}} \leqslant \mathrm{T}\right] \geqslant \left(\lambda e^{-\lambda \mathrm{T}/2\mathrm{N}}\right)^{\mathrm{N}} \eta^{\mathrm{N}} \mathrm{I}$$

où

$$\mathrm{I} = \int_{[\,\mathrm{T/2N}\,;\mathrm{2T/2N}\,]\times\mathcal{S}_{\mathrm{X}_1}\times\ldots[\,(\mathrm{2N-3})\mathrm{T/2N}\,;(\mathrm{N-1})\mathrm{T/N}\,]\times\mathcal{S}_{\mathrm{X}_\mathrm{N}}} p_{\mathrm{X}_1,t_1}(\mathrm{d}\mathbb{P}_1) p_{\mathrm{X}_1,t_2-t_1}(\mathrm{d}\mathbb{P}_2)\ldots$$

Chaque fonction $p_{\mathrm{X},.}(.)$ étant strictement positive, il en est de même pour son intégrale sur chaque espace $\mathcal{S}(\mathrm{X})$ puis lorsqu'on l'intègre par rapport au temps et I est donc strictement positif pour le choix de $\mathrm{X}_1, \ldots \mathrm{X}_\mathrm{N}$. Par ailleurs, le nombre de N-uplets de vecteurs X_i étant fini, on peut donc trouver un minorant m_N de la quantité I, indépendant du chemin permettant de passer de X_0 à X_N. On obtient donc l'inégalité (6.60) en remarquant que η minore également la probabilité de transition d'une forêt à l'autre, et que la quantité minorante est indépendante de $\mathbb{P}_0$. □.

La propriété précédente nous permet d'en déduire une minoration de la probabilité (6.58). En effet, pour tout $\mathbb{P}$, X et S compact de la forme $\mathcal{P} \times \mathrm{Y}$ où Y est un vecteur quantifiant une forêt, et si N est la longueur minimale pour passer de X à Y *via* les règles de transition du chapitre 5, on a alors en prenant $\mathrm{T} = 1$ dans la propriété précédente :

$$\mathrm{P}_{(\mathbb{P},\mathrm{X})}\left[\tau_{\mathcal{P}\times\mathrm{Y}} = 1\right] > (\eta\lambda)^{\mathrm{N}} m_\mathrm{N}$$

En prenant alors le minimum sur N des termes précédents, N variant dans un ensemble fini (il n'y a qu'un nombre fini de couples de forêts possibles), on obtient un minorant absolu de la probabilité d'atteindre le compact $\mathcal{P} \times \mathrm{Y}$, puis un minorant absolu pour atteindre tout compact quelconque de $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$. On a ainsi établi la propriété :

Propriété 6.2.3 (Minoration de la probabilité d'atteinte (6.58))
Pour tout compact S *de* $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$, *on a*

$$\inf_{(\mathbb{P},\mathrm{X})\in\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^f} \mathrm{P}_{(\mathbb{P},\mathrm{X})}\left[\tau_\mathrm{S} \leqslant 1\right] > 0$$

On peut dès lors reprendre point par point la démonstration du théorème 6.2.2 qui n'utilise que des résultats classiques de processus Markovien pour déduire de (6.58) :

Théorème 6.2.3 (Récurrence de $(\mathbb{P}_t, \mathrm{X}_t)$)
Le processus défini par $(\mathbf{E}-\mathbf{8})$ *est un processus récurrent.*

6.2.2 Mesure invariante du processus

L'objet de ce paragraphe est d'utiliser le principal résultat de la section 6.2.1 (théorème 6.2.3) pour en déduire des propriétés sur la mesure invariante du processus général $(\mathbb{P}_t, \mathrm{X}_t)$.

La définition des mesures invariantes ainsi que ses propriétés fondamentales sont données dans [EK86] :

Définition 6.2.3 (Mesure invariante d'un processus Markovien)
Une mesure μ sur $\left(\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f\right)$ est invariante pour le processus $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$ si son semi-groupe P_t vérifie pour tout temps t positif

$$\mu \mathrm{P}_t = \mu$$

Si μ est une mesure invariante du processus X, cela signifie finalement que si X_t est de loi μ, alors à tous les instants ultérieurs à t , X_{t+s} est de loi μ.

On a donc les propriétés :

Propriété 6.2.4 (Mesure invariante)
μ est une mesure invariante pour le processus X*, si l'on a l'une des deux conditions :*

1. X_t *de loi* $\mu \quad \Longrightarrow \forall (s_1, s_2) \quad (\mathrm{X}(t+s_1), \mathrm{X}(t+s_2))$ *sont indépendants.*
2. X_t *de loi* $\mu \quad \Longrightarrow \forall s \geqslant t \quad \mathrm{X}_s$ *de loi* μ.

Preuve : cf [EK86] □

Nous allons essayer de caractériser le comportement asymptotique de $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$ en exhibant, d'une part une mesure invariante pour ce processus, et d'autre part en démontrant qu'en réalité $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$ ne possède qu'une unique mesure invariante.

6.2.2.1 Existence et Unicité de la mesure invariante de $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$

Un raisonnement strictement identique à ce qui est fait dans [ABD01] permet de montrer que la famille de mesures $(\mu_t)_{t\geqslant 0}$ donnée par

$$\forall \mathrm{B} \in \mathcal{B}\left(\mathcal{S}_{\mathcal{A}^\star \times \{0;1\}^f}\right) \qquad \mu_t(\mathrm{B}) = \frac{1}{t}\int_0^t \mathrm{P}_{\mathbb{P},\mathrm{X}}\left((\mathbb{P}_s, \mathrm{X}_s) \in \mathrm{B}\right) \mathrm{d}s$$

est une famille tendue de mesures, ce qui assure l'existence d'une mesure invariante. L'unicité provient alors de la non-dégénérescence de la diffusion, et par la suite du caractère récurrent positif du processus ([HW92] paragraphe 7). On en déduit ainsi le théorème :

Théorème 6.2.4 (Existence et unicité de la mesure invariante de $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$)
Il existe une unique mesure μ invariante pour le processus $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$.

6.2.2.2 Champ de Gibbs de $\mathcal{E}$

Nous allons établir que le champ de Gibbs associé à l'énergie $\mathcal{E}$, donné par

$$\mathrm{d}\mu(\mathbb{P}, \mathrm{X}) = \frac{e^{-\mathcal{E}(\mathbb{P},\mathrm{X})}}{\mathrm{Z}} \mathrm{d}\mathbb{P}\mathrm{d}\mathrm{X} \tag{6.61}$$

est une mesure stationnaire pour le processus $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$; et conclure grâce à l'unicité d'une telle mesure. Le raisonnement est très largement comparable à ce qui est fait dans [SMG97].

Remarquons tout d'abord que μ s'écrit en réalité comme une combinaison de mesures μ_{X}, pour X vecteur de $\{0;1\}^f$ quantifiant une forêt possible. En effet, si l'on numérote les forêts possibles de 1 à N et que l'on désigne par Z_k les quantités

$$\mathrm{Z}_k = \int_{\mathbb{P}\in\mathcal{S}(\mathrm{X}_k)} e^{-\mathcal{E}(\mathbb{P},\mathrm{X}_k)}\mathrm{d}\mathbb{P}$$

on a alors $$\mathrm{Z}_1 + \cdots + \mathrm{Z}_\mathrm{N} = \mathrm{Z}$$

On définit alors la mesure μ_k par

$$\mu_k(\mathcal{S}) = \frac{\displaystyle\int_{\mathrm{S}\cap\{\mathrm{S}_\mathrm{X}=\mathrm{X}_k\}} e^{-\mathcal{E}(\mathbb{P},\mathrm{X}_k)}\mathrm{d}\mathbb{P}}{\mathrm{Z}_k}$$

Par ailleurs, l'énergie $\mathcal{E}$ est « séparable » en $\mathbb{P}$ et X, ainsi :

$$\mathcal{E}(\mathbb{P},\mathrm{X}) = \mathcal{E}_{err}(\mathbb{P}) + \mathcal{E}_{lg}(\mathrm{X}) + \mathcal{E}_\rho(\mathrm{X})$$

Par conséquent $$\boxed{\mathrm{Z}_k = e^{-\mathcal{E}_\rho(\mathrm{X}_k)-\mathcal{E}_{lg}(\mathrm{X}_k)}\int_{\mathcal{S}(\mathrm{X}_k)} e^{-\mathcal{E}_{err}(\mathbb{P})}\mathrm{d}\mathbb{P}}$$

Cette dernière équation associée à (6.61) assure alors que

$$\begin{aligned}
\mu(\mathrm{S}) &= \frac{\displaystyle\int_{\mathrm{S}} e^{-\mathcal{E}(\mathbb{P},\mathrm{X})}\mathrm{d}\mathbb{P}\mathrm{d}\mathrm{X}}{\mathrm{Z}} \\
&= \frac{\displaystyle\sum_{k=1}^{\mathrm{N}}\int_{\mathrm{S}\cap\{\mathrm{S}_\mathrm{X}=\mathrm{X}_k\}} e^{-\mathcal{E}_{lg}(\mathrm{X}_k)-\mathcal{E}_\rho(\mathrm{X}_k)}e^{-\mathcal{E}_{err}(\mathbb{P})}\mathrm{d}\mathbb{P}}{\mathrm{Z}} \\
&= \sum_{k=1}^{\mathrm{N}} \frac{e^{-\mathcal{E}_\rho(\mathrm{X}_k)-\mathcal{E}_\rho(\mathrm{X}_k)}}{\mathrm{Z}}\int_{\mathrm{S}\cap\{\mathrm{S}_\mathrm{X}=\mathrm{X}_k\}} e^{-\mathcal{E}_{err}(\mathbb{P})}\mathrm{d}\mathbb{P} \\
\mu(\mathrm{S}) &= \sum_{k=1}^{\mathrm{N}} \frac{\mathrm{Z}_k}{\mathrm{Z}}\mu_k(\mathrm{S})
\end{aligned}$$

On peut donc en conclure la propriété de combinaison convexe de mesures :

Propriété 6.2.5 (Combinaison convexe de mesures)
Le champ de Gibbs défini par (6.61) est une combinaison convexe de mesures μ_k.

Nous allons alors essayer d'utiliser le critère de la proposition 9.2, chapitre 9, de [EK86] pour en déduire que μ est une mesure invariante du processus. Pour ce faire, nous allons d'abord étudier l'effet du générateur infinitésimal du processus $\mathcal{L}$ sur la mesure μ.

Propriété 6.2.6 (Effet de μ sur $\mathbb{D}$)
Prenons une fonction f de $\mathbb{D}$, cette fonction f vérifie alors

$$\int_{\mathcal{S}^\star\times\{0;1\}^f} \mathcal{L}f(\mathbb{P},\mathrm{X})\mathrm{d}\mu(\mathbb{P},\mathrm{X}) = 0$$

Preuve : Pour démontrer cette proposition, il s'agit d'appliquer le même raisonnement que celui de [SMG97] (théorème 2). Le seul détail à régler est de savoir pourquoi si f s'écrit

$$f = \sum_{k=1}^{n} \chi_{\mathrm{X}_k} f_k$$

alors
$$\int_{\mathcal{S}(\mathrm{X}_k)} \mathcal{L}_1 f_k(\mathbb{P}) \mathrm{d}\mu_k(\mathbb{P}) = 0$$

car le terme de saut utilisant $\mathcal{L}_2$ issu de Q est annulé de la même façon que dans [SMG97]. Pour obtenir le résultat, sur $\mathbb{D}$, il faut donc établir l'égalité

$$\int_{\mathcal{S}(\mathrm{X}_k)} \mathcal{L}_1 f_k(\mathbb{P}) \mathrm{d}\mu_k(\mathbb{P}) = 0$$

pour f_k de classe $\mathcal{C}^2$ vérifiant des conditions de Neumann sur $\partial\mathcal{S}(\mathrm{X}_k)$. En utilisant la définition de μ_k, on a :

$$\mathrm{Z}_k \int_{\mathcal{S}(\mathrm{X}_k)} \mathcal{L}_1 f_k(\mathbb{P}) \mathrm{d}\mu_k(\mathbb{P}) = \int_{\mathcal{S}(\mathrm{X}_k)} \left(-(\nabla^{\mathcal{H}(\mathrm{X}_k)} f_k \mid \nabla^{\mathcal{H}(\mathrm{X}_k)} \mathcal{E}_{err}(\mathbb{P})) + \Delta^{\mathcal{H}(\mathrm{X}_k)} f_k \right) e^{-\mathcal{E}_{err}(\mathbb{P})} \mathrm{d}\mathbb{P}$$

En appliquant la formule d'Ostrogradski [DL92]) sur $\mathrm{S} = \mathcal{S}(\mathrm{X}_k)$ qui est bien un ouvert $\mathcal{C}^1$ par morceaux :

$$\int_{\mathrm{S}} f \Delta g \mathrm{d}s = \int_{\mathrm{S}} (\nabla f \mid \nabla g) \mathrm{d}s + \int_{\partial \mathrm{S}} f (\nabla g \mid \overrightarrow{n(l)}) \mathrm{d}l$$

Pour $g = f_k$ et $f = e^{-\mathcal{E}_{err}}$ sur $\mathrm{S} = \mathcal{S}(\mathrm{X}_k)$, on obtient que

$$\int_{\mathcal{S}(\mathrm{X}_k)} \mathcal{L}_1 f_k(\mathbb{P}) \mathrm{d}\mu_k(\mathbb{P}) = \frac{1}{\mathrm{Z}_k} \int_{\partial\mathcal{S}(\mathrm{X}_k)} e^{-\mathcal{E}_{err}(\mathbb{P})} (\nabla^{\mathcal{H}(\mathrm{X}_k)} f(\mathbb{P}) \mid \overrightarrow{n(\mathbb{P})}) \mathrm{d}\mathbb{P}$$

Mais f_k vérifie des conditions de Neumann sur $\mathcal{S}(\mathrm{X}_k)$, ainsi, l'intégrale précédente est nulle et finalement

$$\int_{\mathcal{S}(\mathrm{X}_k)} \mathcal{L}_1 f_k(\mathbb{P}) \mathrm{d}\mu_k(\mathbb{P}) = 0$$

On en déduit donc la démonstration de la propriété précédente en additionnant les relations précédentes pour obtenir :

$$\forall f \in \mathbb{D} \qquad \int_{\mathcal{S}\times\{0;1\}^f} \mathcal{L} f(\mathbb{P}, \mathrm{X}) \mathrm{d}\mu(\mathbb{P}, \mathrm{X}) = 0 \qquad \square$$

Il reste alors à conclure que μ est une mesure stationnaire du processus. Pour ce faire, nous allons utiliser le thérorème :

Théorème 6.2.5 (Ethier et Kurtz)
Si $\mathcal{L}$ est le générateur d'un processus markovien, fortement continu X *à valeurs dans* E*, de semi-groupe $(\mathrm{P}_t)_{t\geqslant 0}$ sur un sous-espace fermé $\mathbb{D}$ de $\mathbb{D}(\mathcal{L})$, et tel que le problème martingale soit bien posé pour tout f de $\mathbb{D}$, alors si μ est une mesure sur* E*, les trois propriétés sont équivalentes :*

1. *μ est une distribution stationnaire pour $\mathcal{L}$.*
2. $$\forall f \in \mathbb{D} \quad \forall t \geqslant 0 \qquad \int_{\mathrm{E}} \mathrm{P}_t f \mathrm{d}\mu = \int_{\mathrm{E}} f \mathrm{d}\mu \tag{6.62}$$

3.

$$\forall f \in \mathbb{D} \qquad \int_{\mathrm{E}} \mathcal{L}f \,\mathrm{d}\mu = 0 \tag{6.63}$$

Preuve : Voir [EK86] (page 239, proposition 9.2). □

On ne peut cependant pas appliquer le théorème 6.2.5 : $\mathbb{D}$ est bien un sous-espace fermé de $\mathbb{D}(\mathcal{L})$ donné par l'ensemble des fonctions verifiant les deux premières conditions de la propriété 6.1.1 (sauf les hypothèses de Neumann sur $\partial\mathcal{S}_{\mathcal{H}_i}$) qui peut être muni de la métrique :

$$\|f\| = \sum_k \left(\|f_k\|_\infty + \|\nabla^{\mathcal{H}_k} f_k\| \right)$$

Cependant, la nature du problème martingale pour $\mathbb{D}$ associé au générateur A n'est pas claire. En fait, on peut démontrer la propriété plus faible

Théorème 6.2.6 (Mesure invariante du processus)
Le champ de Gibbs μ associé à l'énergie $\mathcal{E}$ est l'unique mesure invariante du processus $(\mathbb{P}, \mathrm{X})$ et le problème martingale associé au générateur A et à la mesure initiale μ est bien posé.

Preuve : Nous cherchons à appliquer une modification du théorème précédent dans le cas où l'ensemble des fonctions du problème martingale est $\mathbb{D}$. Il s'agit donc tout d'abord d'établir que le problème est bien posé pour (A, μ) sur l'ensemble des fonctions tests $\mathbb{D}$.

Comme nous l'avons mentionné dans la remarque qui suit la définition des problèmes bien posés, c'est l'unicité en loi de tels processus qu'il faut vérifier puisque l'existence de tels processus est donnée par l'existence trajectorielle (existence forte) des processus de diffusion réfléchie avec sauts (théorème 5.7.1). Il s'agit donc de démontrer que si $(\mathrm{X}_t)_{t\geqslant 0}$ et $(\mathrm{Y}_t)_{t\geqslant 0}$ sont deux processus vérifiant $\mathrm{X}_0 \sim \mu$ et $\mathrm{Y}_0 \sim \mu$, alors la loi de X_t et celle de Y_t coïncident, en tout temps t. Il faut donc montrer que pour toute fonction mesurable sur $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$, on a

$$\mathbb{E}[f(\mathrm{X}_t)|\mathrm{X}_0 \sim \mu] = \mathbb{E}[f(\mathrm{Y}_t)|\mathrm{Y}_0 \sim \mu]$$

Il suffit par ailleurs de démontrer ce résultat pour les fonctions f mesurables qui sont des indicatrices d'ouverts relatifs de $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$, à savoir

$$\mathbb{E}[\chi_{\mathrm{U}}(\mathrm{X}_t)|\mathrm{X}_0 \sim \mu] = \mathbb{E}[\chi_{\mathrm{U}}(\mathrm{Y}_t)|\mathrm{Y}_0 \sim \mu] \tag{e}$$

où U désigne un ouvert quelconque de $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$. Nous allons approcher une telle indicatrice par une suite de fonctions $(f_\varepsilon)_{\varepsilon\geqslant 0}$ de $\mathbb{D}$. Si $(f_\varepsilon)_{\varepsilon\geqslant 0}$ désigne une suite de fonctions de $\mathbb{D}$, d'après la propriété 6.2.6 et l'équivalence entre (6.62) et (6.63), nous savons que

$$\mathbb{E}[f_\varepsilon(\mathrm{X}_t)|\mathrm{X}_0 \sim \mu] = \mathbb{E}[f_\varepsilon(\mathrm{X}_0)] = \int f_\varepsilon \mathrm{d}\mu = \mathbb{E}[f_\varepsilon(\mathrm{Y}_0)] = \mathbb{E}[f_\varepsilon(\mathrm{Y}_t)|\mathrm{Y}_0 \sim \mu]$$

Il suffit donc de construire une suite de $\mathbb{D}$ approchant χ_{U} pour établir l'égalité (**e**). Si U_k désigne un ouvert d'un simplexe $\mathcal{S}(\mathrm{X}_k)$, le lemme d'Uryshon appliqué au voisinage fermé $\mathcal{V}(\mathrm{U}_k, \varepsilon)$ permet de construire f_ε^k nulle en dehors de $\mathcal{V}(\mathrm{U}_k, \varepsilon)$ et valant 1 sur $\mathcal{V}(\mathrm{U}_k, \varepsilon/2)$ et de classe $\mathscr{C}^\infty(\mathcal{S}(\mathrm{X}_k), \mathbb{R})$. La fonction f_ε donnée par

$$f_\varepsilon = \sum_k f_\varepsilon^k \chi_{\mathcal{S}(\mathrm{X}_k)}$$

appartient bien à $\mathbb{D}$ (le gradient de f_ε^k est nul puisque la fonction est constante sur un voisinage de U de taille ε). Par ailleurs, f_ε converge simplement vers χ_{U} et est majorée par la fonction constante égale à 1 intégrable sous $\mathrm{d}\mu$.

Ainsi, puisque

$$\mathbb{E}\left[f_\varepsilon(\mathrm{X}_t)|\mathrm{X}_0 \sim \mu\right] = \mathbb{E}\left[f_\varepsilon(\mathrm{Y}_t)|\mathrm{Y}_0 \sim \mu\right]$$

en passant à la limite pour $\varepsilon \mapsto 0$, on obtient bien (**e**). Le problème martingale (A, μ) est donc bien posé dans l'ensemble $\mathbb{D}$ quand on impose comme distribution initiale le champ de Gibbs μ.

On ne peut par contre pas appliquer directement le résultat du théorème 6.5.2 pour conclure que le champ de Gibbs μ est bien la mesure invariante associée au processus de diffusion réfléchie avec sauts. En effet, il serait nécessaire pour cela de savoir que le problème martingale est bien posé pour A, quelle que soit la distribution initiale du processus.

La propriété de stationnarité de la mesure μ peut tout de même être établie sans savoir si le problème est bien posé. En effet, (**e**) assure tout de même la stationnarité de μ par densité des fonctions indicatrices dans l'ensemble des fonctions mesurables. Ainsi, si $(\mathrm{P}_t)_{t\geqslant 0}$ désigne le semigroupe associé au processus de diffusion réfléchie avec sauts, on a

$$\forall t \geqslant 0 \qquad \mu \mathrm{P}_t = \mu$$

Utilisant alors l'unicité de la mesure invariante de notre processus évoquée à la fin du paragraphe 6.2.2.1, on en conclut que le champ de Gibbs μ est l'unique mesure stationnaire du processus réfléchi avec sauts. □

> Dans le théorème précédent, il faut noter que l'on n'a pas établi le fait que le problème martingale était bien posé pour le processus réfléchi avec sauts. Notre résultat ne concerne en effet que l'unicité des solutions (en loi) lorsque l'on impose que la loi initiale du processus est le champ de Gibbs μ.

Nous avons donc établi deux résultats importants caractérisant le processus de diffusion avec sauts. Le premier résultat assure que le processus possède une dynamique ergodique, permettant au processus d'explorer tous les états en un temps donné avec une probabilité strictement positive. Ce résultat se traduit en particulier par le fait que le processus $(\mathbb{P}, \mathrm{X})$ ne peut pas atteindre une frontière de $\partial\mathcal{S}_{\mathcal{A}^\star}$ avec une probabilité strictement positive.

Le second résultat utilise alors ce résultat important d'ergodicité et établi en fin de compte que le comportement stationnaire du processus $(\mathbb{P}, \mathrm{X})$ ne dépend pas de ce qui se passe sur la frontière des simplexes, puisqu'en probabilité, le processus ne « vit » pas sur ces frontières. Nous pouvons alors en conclure que la mesure stationnaire du processus est classiquement donnée par le champ de Gibbs définit dans le chapitre 5. Ainsi, nous pouvons affirmer que l'espace des forêts qui seront privilégiées par notre algorithme sont celles qui correspondent à un état énergétique $\mathcal{E}_{\mathrm{L}g} + \mathcal{E}_\rho$ faible.

Chapitre 7 - Approximation stochastique du processus de diffusion sous contraintes avec saut

Dans ce chapitre, nous présentons un algorithme d'approximation stochastique qui va nous permettre d'approcher, dans un sens que nous définirons, le comportement limite de l'équation différentielle stochastique $(\mathbf{E}-\mathbf{8})$. Nous appliquerons alors notre algorithme pour la détection de visages et dans le cadre d'expériences synthétiques.

7.1 Algorithme d'approximation

Nous allons nous inspirer des idées de [KY03] pour simuler le processus donné par $(\mathbf{E}-\mathbf{8})$ (chapitre 5). Pour cela, nous allons utiliser l'équation différentielle stochastique :

$$\forall t \in \left]\, t_{s_i}\,;t_{s_{i+1}}\,\right] \qquad \mathrm{d}\mathbb{P}_t = -\Pi_{\mathcal{H}_t}\left(\nabla \mathcal{E}_{err}(\mathbb{P}_t)\right) + \Sigma_i \mathrm{dW}_t + \mathrm{dZ}_t \tag{7.64}$$

où les instants de sauts (t_{s_i}) suivent une distribution Poissonienne de paramètre λ fixé. Le gradient de $\mathcal{E}$ n'est pas toujours accessible à chaque étape de l'algorithme, il s'agit donc de procéder, comme au chapitre 3, à une approximation stochastique de ce gradient. Nous utiliserons à nouveau dans notre cadre théorique un algorithme d'approximation à pas variable $(\varepsilon_i)_{i\in\mathbb{N}}$ tel que

$$\begin{cases} \sum \varepsilon_i = +\infty \\ \\ \sum \varepsilon_i^2 < +\infty \end{cases}$$

Dans ce cas, la transformation $\tau - m$, qui, à une itération (dans $\mathbb{N}$) de l'algorithme associe un temps (dans $]\,0\,;+\infty\,[$), est à nouveau valide.

7.1.1 Distribution des sauts

Commençons par détailler l'implémentation de la simulation des instants de sauts. Nous générons les instants de sauts (t_{s_i}), et donc les itérations de sauts $(m(t_{s_i}))$ de manière récursive grâce à la simulation :

$$\begin{cases} t_{s_0} \sim \mathcal{P}(\lambda) \\ \\ \forall i \in \mathbb{N} \qquad t_{s_{i+1}} - t_{s_i} \sim \mathcal{P}(\lambda) \end{cases}$$

Nous commençons donc par fixer un instant initial de premier saut t_{s_0}, ainsi qu'une itération de saut $m(t_{s_0})$. Lorsque l'algorithme arrive à cette itération $m(t_{s_0})$, nous générons alors un

deuxième instant de saut *via* la simulation $t_{s_0} + \mathcal{P}(\lambda)$, ce qui détermine une deuxième itération de saut $m(t_{s_1})$ ...

Lors des sauts, on modifie alors le vecteur $\mathrm{X}_{m(t_{s_i})}$, c'est-à-dire la forêt de features *via* les règles données au chapitre 5 et récapitulées dans les tableaux de l'annexe C. Cette modification entraîne alors une modification de $\mathbb{P}_{m(t_{s_i})}$.

7.1.2 Approximation entre les sauts

Entre deux instants de saut, et donc entre deux itérations de saut $m(t_{s_i})$ et $m(t_{s_{i+1}})$, nous appliquons l'algorithme itératif de la forme

$$\mathbb{P}_{i+1} = \Pi_{\mathcal{S}_{t_{s_i}}} \left(\mathbb{P}_i + \varepsilon_i y_i + \sqrt{\varepsilon_i} \xi_i \right) \tag{7.65}$$

où i est un entier quelconque entre $m(t_{s_i})$ et $m(t_{s_{i+1}})$.

La variable aléatoire y_i est alors une approximation du gradient de $\mathcal{E}_{err}$ en $\mathbb{P}_i$ tandis que ξ_i est un bruit gaussien standard sur $\mathbb{R}^{|\mathcal{F}(\mathrm{X}_i)|}$. Nous pouvons expliquer ces choix en associant à chaque terme de (7.64) un terme d'approximation donné dans (7.64).

Ainsi, nous remarquons dans (7.64) que :

- Il est à nouveau nécessaire d'estimer $\nabla \mathcal{E}_{err}$ ainsi que le terme :

$$\int_0^t \nabla \mathcal{E}_{err}(\mathbb{P}_t) \mathrm{d}t$$

  C'est précisément le rôle de y_i et de la somme :

$$\sum_{i=1}^{m(t)} \varepsilon_i y_i$$

  Par la suite, nous choisirons donc

$$\forall \delta \in \mathcal{A}^\star \qquad y_i(\delta) = -\frac{\mathrm{C}(\omega_i, \delta) g(\omega_i)}{\mathbb{P}_i(\delta)}$$

  où ω_i est un p-uplet de loi $\mathbb{P}_i$ avec remise.

- Il faut simuler un mouvement brownien standard W_t (entre deux sauts, la matrice de covariance vaut $\Pi_{\mathcal{S}_{t_i}}(\mathrm{Id})$, sur l'espace « réel » dans lequel vit le mouvement brownien), et cela est fait grâce à

$$\mathrm{W}_t = \int_0^t \mathrm{dW}_t$$

  et l'approximation de dW_t par les termes ξ_i. Par conséquent, le terme

$$\sum \sqrt{\varepsilon_i} \xi_i$$

  est destiné à approcher W_t.

- le terme de rappel dans l'espace des contraintes $\mathcal{S}_{\mathcal{H}_t}$ est en fait « caché », puisqu'il est issu de la projection $\Pi_{\mathcal{S}_{t_{s_i}}}$ dans (7.65) :

$$z_i = \Pi_{\mathcal{S}_{t_{s_i}}} \left(\mathbb{P}_i + \varepsilon_i y_i + \sqrt{\varepsilon_i} \xi_i \right) - \left(\mathbb{P}_i + \varepsilon_i y_i + \sqrt{\varepsilon_i} \xi_i \right)$$

  de sorte que

$$\forall i \in [\![m(t_{s_i}) \,;\, m(t_{s_{i+1}})]\!] \qquad \mathbb{P}_{i+1} = \mathbb{P}_i + \varepsilon_i y_i + \sqrt{\varepsilon_i} \xi_i + z_i$$

7.1.3 Processus interpolés

Grâce aux variables aléatoires précédentes, nous construisons les processus d'interpolation constants par morceaux, grâce aux applications m et τ (dont on rappelle la définition) :

$$\tau_n = \sum_{k \leqslant n} \varepsilon_k$$

Définition 7.1.1 (Processus interpolés)
On définit les processus $(\mathbb{P}^n(t), \mathrm{Y}^n(t), \mathrm{W}^n(t), \mathrm{Z}^n(t))$ *par :*

$$\forall n \in \mathbb{N} \quad \forall t \in \mathbb{R}^+ \qquad \mathbb{P}^n(t) = \mathrm{Y}^n(t) + \mathrm{W}^n(t) + \mathrm{Z}^n(t) \tag{7.66}$$

avec

$$\forall n \in \mathbb{N} \quad \forall t \in \mathbb{R}^+ \qquad \mathrm{Y}^n(t) = \sum_{i=n}^{m(\tau_n+t)} \varepsilon_i y_i \tag{7.67}$$

$$\forall n \in \mathbb{N} \quad \forall t \in \mathbb{R}^+ \qquad \mathrm{W}^n(t) = \sum_{i=n}^{m(\tau_n+t)} \sqrt{\varepsilon_i}\xi_i \tag{7.68}$$

et

$$\mathrm{Z}^n(t) = \sum_{i=n}^{m(\tau_n+t)} z_i \tag{7.69}$$

> La définition des processus interpolés est légèrement différente de celle donnée au chapitre 4, puisqu'ici ces processus ne sont plus continus alors qu'ils l'étaient dans le chapitre 4. Cela n'est pas fondamentalement gênant et se justifie par le fait que le processus que l'on cherche à approcher n'est pas lui non plus continu.

Le but de la suite du chapitre va donc être de montrer en quel sens les processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ approchent la solution de (7.64).

7.2 Ensemble $\mathcal{D}$

7.2.1 Définitions

L'espace naturel dans lequel vit notre processus de diffusion sous contraintes avec sauts va guider notre raisonnement en ce qui concerne l'approximation de (7.64) par les processus interpolés précédents. On note E l'espace $\mathcal{S}_{\mathcal{A}^\star} \times \{0;1\}^f$, on constate alors que le processus $(\mathbb{P}, \mathrm{X})$ appartient à la classe des trajectoires continues à droite sur E ayant une limite à gauche (càdlàg), où E est muni de la norme définie au paragraphe 5.7.2.2 par (5.31). Remarquons enfin que X varie dans un espace discret. On définit alors $\mathcal{D}$ (*Cf* par exemple le chapitre VI de [JS87], le chapitre 3 de [Bil99] ou le chapitre 8 de [KY03] chapitre 8) selon la définition 7.2.1 suivante.

Définition 7.2.1 (Espace $\mathcal{D}$ des trajectoires càdlàg)
$\mathcal{D}$ *est l'espace des fonctions de* $\mathbb{R}$ *dans* E *qui sont continues à gauche, ayant une limite à droite, c'est-à-dire si* x *est une trajectoire dans* $\mathcal{D}$*, alors :*

$$\begin{cases} \forall t \geqslant 0 \qquad \lim\limits_{s \mapsto t \quad s \geqslant t} x(s) = x(t^+) = x(t) \\ \forall t \geqslant 0 \qquad \lim\limits_{s \mapsto t \quad s \leqslant t} x(s) = x(t^-) \end{cases}$$

On définit alors les quantités $w_x(\mathrm{I})$ *pour* $\mathrm{I} \subset \mathbb{R}^+$ *par*

$$w_x(\mathrm{I}) = w(x, \mathrm{I}) = \underset{s,t\in \mathrm{I}}{\mathrm{Sup}} |x(s) - x(t)|$$

Cet espace $\mathcal{D}$ contient donc $\mathcal{C}$, trajectoires continues dans E. Les quantités w_x correspondent alors à un module de continuité sur $\mathcal{C}$ et on a la propriété :

Propriété 7.2.1 (Module de continuité sur $\mathcal{C}$)
Si x est une trajectoire de $\mathcal{C}$, alors

$$\forall \mathrm{T} > 0 \quad \forall x \in \mathcal{C} \qquad \lim_{\delta \mapsto 0} \underset{t\in[\,0\,;\mathrm{T}\,]}{\mathrm{Sup}} w_x\left([\,t\,;t+\delta\,]\right) = 0$$

Malheureusement, une telle propriété n'est pas vraie pour des trajectoires de $\mathcal{D}$. Il est nécessaire pour étudier les éléments de $\mathcal{D}$ d'avoir une entité quantifiant l'amplitude des sauts des trajectoires. On définit alors les distributions de temps d'amplitudes δ par

Définition 7.2.2 (temps (t_i) δ-sparse)
Les temps (t_i) sont δ-sparse s'ils vérifient les conditions :

$$\underset{i}{\mathrm{Min}}\ t_i - t_{i-1} \geqslant \delta$$

avec $$\forall i \qquad t_i \geqslant t_{i-1}$$

On notera alors de tels temps des temps $\delta - sparse$.

De tels intervalles de temps permettent alors de définir le module de saut pour les trajectoires de $\mathcal{D}$:

Définition 7.2.3 (Module de saut $w'_x(\delta)$)
Si x est une trajectoire de $\mathcal{D}$ et si δ est un réel strictement positif, on définit

$$w'_x(\delta) = w'(x,\delta) = \underset{\{t_i\}\in\delta-sparse}{\mathrm{Inf}} \underset{i}{\mathrm{Max}}\ w_x([\,t_i\,;t_{i+1}\,[)$$

On montre alors la propriété ([Bil99] chapitre 3 Lemme 1) 7.2.2 :

Propriété 7.2.2

$$\forall x \in \mathcal{D} \qquad \lim_{\delta\mapsto 0} w'_x(\delta) = 0$$

7.2.2 Topologie sur $\mathcal{D}$

L'objet de ce paragraphe est de rappeler la définition de la métrique sur $\mathcal{D}$ qui rend cet espace séparable et complet.

Si I est un intervalle fermé de $\mathbb{R}^+$ et qu'on note Λ_I l'ensemble des bijections continues croissantes strictement de I dans I, on pose :

Définition 7.2.4 (Distance sur $\mathcal{D}(\mathrm{I})$)
On définit la métrique d_I pour $(f,g) \in \mathcal{D}(\mathrm{I})^2$ par

$$d_\mathrm{I}(f,g) = \mathrm{Inf}\left\{\mu \quad | \quad \underset{s\in\mathrm{I}}{\mathrm{Sup}} |s - \lambda(s)| \leqslant \mu \quad et \quad \underset{s\in\mathrm{I}}{\mathrm{Sup}} |f(s) - g(\lambda(s))| \leqslant \mu \quad et \quad \lambda \in \Lambda_\mathrm{I}\right\}$$

La topologie issue de d_{I} coïncide alors sur $\mathcal{C}(\mathrm{I})$ avec celle de la convergence uniforme. Pour une telle métrique toujours, l'espace $\mathcal{D}(\mathrm{I})$ est séparable mais pas complet ([Bil99] théorème 12.2). Pour rendre $\mathcal{D}(\mathrm{I})$ complet, il est alors nécessaire de contrôler la « dérivée » du paramétrage de temps $\lambda \in \Lambda_{\mathrm{I}}$.

De telles remarques nous amènent donc à la définition :

Définition 7.2.5 (Métrique $\widetilde{d}_{\mathrm{I}}$ sur $\mathcal{D}(\mathrm{I})$ - Distance de Skorokhod)
Pour λ un paramétrage de I, *on pose :*

$$|\lambda| = \underset{\mathrm{Inf\ I} \leqslant s < t \leqslant \mathrm{Max\ I}}{\mathrm{Sup}} \left| \log \frac{\lambda(t) - \lambda(s)}{t - s} \right|$$

La métrique $(\widetilde{d}_{\mathrm{I}})$ est alors définie pour $(f, g) \in \mathcal{D}(\mathrm{I})^2$ par :

$$\widetilde{d}_{\mathrm{I}}(f, g) = \mathrm{Inf} \left\{ \mu \quad | \quad |\lambda| \leqslant \mu \quad et \quad \underset{s \in \mathrm{I}}{\mathrm{Sup}} \, |f(s) - g(\lambda(s))| \leqslant \mu \quad et \quad \lambda \in \Lambda_{\mathrm{I}} \right\}$$

Avec une telle métrique, on a alors :

Théorème 7.2.1 (Complétude de $\mathcal{D}(\mathrm{I})$ et $\mathcal{D}([\,0\,;+\infty\,[)$)
L'espace $\mathcal{D}(\mathrm{I})$ muni de $(\widetilde{d}_{\mathrm{I}})$ est séparable et complet.

Par ailleurs, si on pose :

$$\widetilde{d}(f, g) = \sum_i \widetilde{d}_{[\![\, i\,;\, i+1\,]\!]}(f, g)$$

alors $\mathcal{D}([\,0\,;+\infty\,[)$ est également séparable et complet.

7.2.3 Convergence dans $\mathcal{D}$. Compacité faible et critère de tension sur $\mathcal{D}$

7.2.3.1 Convergence dans $\mathcal{D}$

Nous adopterons le point de vue de la convergence faible (convergence au sens des distributions) ([Bil99], [BMP]) pour les processus à valeurs dans E. C'est-à-dire, X^n tend vers X au sens des distributions si, et seulement si, pour toute fonction continue bornée f sur E, on a

$$\mathbb{E}f(\mathrm{X}^n) \longmapsto \mathbb{E}f(\mathrm{X})$$

Dans notre situation, l'ensemble E étant borné, il suffira de choisir f continue pour caractériser complètement la convergence au sens des distributions d'une suite de processus.

Par ailleurs, une suite de processus $(\mathrm{X}^n)_{n \in \mathbb{N}}$ converge faiblement vers un processus X si, et seulement si,

- $(\mathrm{X}^n)_{n \in \mathbb{N}}$ est faiblement compacte.
- toute sous-suite convergente converge faiblement vers X.

Notre démarche consistera donc par la suite à démontrer que la famille formée par $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ forme une famille faiblement compacte, et telle que la seule limite possible est le processus $(\mathbf{E} - \mathbf{8})$. Nous commençons donc par énumérer quelques critères de compacité faible dans $\mathcal{D}$.

7.2.3.2 Compacité faible

Notre approche va consister à considérer les processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ comme étant une famille de trajectoires de $\mathcal{D}$ et d'établir des critères de compacité faible (pour la topologie précédente sur $\mathcal{D}$) afin d'avoir l'existence d'une sous-suite convergente vers un processus $(\mathbb{P}, \mathrm{Y}, \mathrm{W}, \mathrm{Z})$. Nous avons le résultat :

Théorème 7.2.2 (Parties séquentiellement compacte de $\mathcal{D}$- Parties tendues)
Une partie A *de* $\mathcal{D}$ *est séquentiellement compacte pour la topologie de Skorokhod si, et seulement si les deux conditions suivantes sont vraies :*

1. $$\operatorname*{Sup}_{x\in \mathrm{A}} \|x\| \leqslant +\infty$$

2. $$\lim_{\delta \mapsto 0} \operatorname*{Sup}_{x\in \mathrm{A}} w'_x(\delta) = 0$$

Une telle partie A *est dite tendue dans* $\mathcal{D}$.

Il existe d'autres critères de tension, équivalents aux deux critères précédents, pour des parties A de la forme $(\mathrm{X}^n)_{n\in\mathbb{N}} \in \mathcal{D}^{\mathbb{N}}$, ce qui donne le théorème suivant.

Théorème 7.2.3 (Critère de tension)
Une suite $(\mathrm{X}^n)_{n\in\mathbb{N}}$ *de processus de* $\mathcal{D}$ *est tendue si, et seulement si*

1. $$\lim_{a\mapsto+\infty} \lim_{n\mapsto+\infty} \operatorname{Sup} \mathrm{P}\left[\|\mathrm{X}^n\| \geqslant a\right] = 0$$

2. $$\lim_{\delta\mapsto 0} \lim_{n\mapsto+\infty} \operatorname{Sup} \mathrm{P}\left[w'_{\mathrm{X}^n}(\delta) \geqslant \varepsilon\right] = 0$$

Ces deux dernières conditions sont alors également équivalentes à :

Théorème 7.2.4 (Critère de tension - Kushner et Yin)
Une suite $(\mathrm{X}^n)_{n\in\mathbb{N}}$ *de processus de* $\mathcal{D}(0,+\infty)$ *est tendue si, et seulement si*

1. *Pour tout réel* T *et* ε *strictement positif, il existe* n_0 *et* K *tels que*
$$n \geqslant n_0 \Longrightarrow \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} |\mathrm{X}^n(t)| > \mathrm{K}\right] \leqslant \varepsilon$$

2. $$\forall \mathrm{T} \geqslant 0 \qquad \lim_{\delta\mapsto 0} \lim_{n} \operatorname{Sup} \operatorname*{Sup}_{t\leqslant \mathrm{T}} \operatorname*{Sup}_{s\leqslant\delta} \mathbb{E}[\operatorname{Min}\{\|\mathrm{X}^n(t+s) - \mathrm{X}^n(t)\|; 1\}] = 0$$

C'est précisément cette dernière caractérisation des suites de processus tendus que nous utiliserons dans le cadre de notre approximation stochastique. L'objectif sera donc de montrer qu'il existe une sous-suite des processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ qui converge faiblement avant d'exhiber la limite faible de ce processus.

> On pourra également utiliser une variante du critère 2 issue du théorème 13.2, chapitre 3 de [Bil99], à savoir :
>
> $$\forall \varepsilon > 0 \qquad \lim_{\delta\mapsto 0} \lim_{n} \operatorname{Sup} \mathrm{P}\left[w'_{\mathrm{X}^n}(\delta) \geqslant \varepsilon\right] = 0 \tag{7.70}$$

7.3 Compacité des trajectoires de $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$

Nous allons démontrer que les trajectoires $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ forment une suite tendue de processus de $\mathcal{D}$. Pour ce faire, il s'agira donc de vérifier les deux conditions du théorème 7.2.4 pour chacune des familles des processus $(\mathbb{P}^n)$, $(\mathbb{Y}^n)$, $(\mathbb{W}^n)$ et $(\mathbb{Z}^n)$. Nous allons commencer par démontrer un lemme qui sera essentiel pour obtenir ces deux résultats et qui utilise la notion d'uniforme intégrabilité.

Définition 7.3.1 (Uniforme intégrabilité de (x_i))
Une suite de variables aléatoires (x_i) est uniformément intégrable si, et seulement si,

$$\lim_{\mu(\mathrm{B})\mapsto 0} \operatorname*{Sup}_n \mathbb{E}[|x_n|\chi_{x_n\in\mathrm{B}}] = 0$$

Cette propriété est alors équivalente à

$$\lim_{\mathrm{K}\mapsto+\infty} \operatorname*{Sup}_n \mathbb{E}\left[|x_n|\chi_{|x_n|>\mathrm{K}}\right] = 0 \tag{7.71}$$

Nous pouvons dès lors énoncer le lemme :

Lemme 7.3.1 (Famille uniformément intégrable $\Rightarrow$ Critère 1 de tension)
Supposons que (x_i) sont uniformément intégrables, alors la suite de processus (X^n) définis par

$$\mathrm{X}^n(t) = \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i x_i$$

vérifie immédiatement le critère 1 du théorème 7.2.4.

Preuve : Nous allons démontrer que si T est un réel strictement positif,

$$\lim_{\mathrm{K}\mapsto+\infty} \operatorname*{Sup}_n \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant\mathrm{T}} |\mathrm{X}^n(t)| > \mathrm{K}\right] = 0$$

Nous en déduirons alors la propriété 1 du théorème 7.2.4. Nous allons utiliser une méthode de troncature pour démontrer ce résultat.

Prenons K un réel strictement positif quelconque et T un réel strictement positif, on pose :

$$x_{n,\mathrm{K}} = x_n\left(1 - \chi_{|x_n|\geqslant\mathrm{K}}\right)$$

On a alors

$$x_n = x_{n,\mathrm{K}} + \underbrace{x_n\chi_{|x_n|\geqslant\mathrm{K}}}_{=\zeta_{n,\mathrm{K}}}$$

En revenant aux probabilités, on obtient :

$$\mathrm{P}\left[\operatorname*{Sup}_{t\leqslant\mathrm{T}} |\mathrm{X}^n(t)| \geqslant \mathrm{K}\right] = \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant\mathrm{T}} \left|\sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i x_{i,\mathrm{K}} + \varepsilon_i\zeta_{i,\mathrm{K}}\right| \geqslant \mathrm{K}\right]$$

$$\leqslant \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant\mathrm{T}} \left|\sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i x_{i,\mathrm{K}}\right| \geqslant \mathrm{K}/2\right] + \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant\mathrm{T}} \left|\sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i\zeta_{i,\mathrm{K}}\right| \geqslant \mathrm{K}/2\right]$$

car $(a+b\geqslant\mathrm{K}) \subset \{(a\geqslant\mathrm{K}/2)\cup(b\geqslant\mathrm{K}/2)\}$.

Par construction de $(x_{n,\mathrm{K}})$, on sait que

$$\operatorname*{Sup}_{t\leqslant \mathrm{T}} \left| \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i x_{i,\mathrm{K}} \right| \leqslant \mathrm{K} \sum_{i=n}^{m(\tau_n+\mathrm{T})} \varepsilon_i \leqslant \mathrm{K} \log\left(1+\mathrm{T}/\tau_n\right)$$

Comme $\tau_n \longmapsto +\infty$, il existe n_0 tel que :

$$\forall n \geqslant n_0 \qquad \log\left(1+\mathrm{T}/\tau_n\right) \leqslant \frac{1}{3}$$

Et pour un tel n_0, on en déduit que

$$\operatorname*{Sup}_{t\leqslant \mathrm{T}} \left| \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i x_{i,\mathrm{K}} \right| \leqslant \frac{\mathrm{K}}{3}$$

Soit finalement

$$\forall n \geqslant n_0 \qquad \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} \left| \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i x_{i,\mathrm{K}} \right| \geqslant \mathrm{K}/2 \right] = 0 \tag{7.72}$$

Il reste donc à étudier le terme en « $\zeta_{.,.}$ ». L'inégalité de Markov implique que

$$\mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} |\mathrm{Y}(t)| \geqslant \mathrm{K} \right] \leqslant \frac{\mathbb{E} \operatorname*{Sup}_{t\leqslant \mathrm{T}} |\mathrm{Y}(t)|}{\mathrm{K}}$$

En appliquant cette inégalité à $\mathrm{Y} = \sum \varepsilon_i \zeta_{i,\mathrm{K}}$, on a alors :

$$\mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} \left| \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i \zeta_{i,\mathrm{K}} \right| \geqslant \mathrm{K}/2 \right] \leqslant 2 \frac{\mathbb{E}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i |\zeta_{i,\mathrm{K}}| \right]}{\mathrm{K}}$$

En majorant brutalement, on voit que :

$$\operatorname*{Sup}_{t\leqslant \mathrm{T}} \mathbb{E} \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i |\zeta_{i,\mathrm{K}}| \leqslant \operatorname*{Sup}_{n\geqslant n_0} \mathbb{E}|\zeta_{n,\mathrm{K}}| \sum_{i=n}^{m(\tau_n+\mathrm{T})-1} \varepsilon_i \leqslant \operatorname*{Sup}_{n\geqslant n_0} \mathbb{E}|\zeta_{n,\mathrm{K}}| \log\left(1+\frac{\mathrm{T}}{\tau_n}\right)$$

Mais l'uniforme intégrabilité de (x_n) appliquée à de $\zeta_{n,\mathrm{K}}$ assure alors que

$$\lim_{\mathrm{K}\mapsto+\infty} \operatorname*{Sup}_n \mathbb{E}|\zeta_{n,\mathrm{K}}| = 0$$

Finalement :

$$\lim_{\mathrm{K}\mapsto+\infty} \operatorname*{Sup}_n \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} \left| \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i \zeta_{i,\mathrm{K}} \right| \geqslant \mathrm{K}/2 \right] = 0 \tag{7.73}$$

En revenant au processus X^n, (7.72) et (7.73) assurent alors que :

$$\lim_{\mathrm{K}\mapsto+\infty} \operatorname*{Sup}_n \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} |\mathrm{X}^n(t)| > \mathrm{K} \right] = 0 \qquad \square$$

Nous allons utiliser le lemme précédent pour démontrer que les processus étudiés sont tendus.

Lemme 7.3.2 (Vérification du critère 1 de tension)
Les processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ *vérifient la condition 1 du théorème 7.2.4.*

Preuve : L'ensemble des variables $(\mathbb{P}^n)$ sont bornées par construction (trajectoire dans $\mathcal{S}_{\mathcal{A}^\star}$.). Ainsi, dès que K est supérieur à $\underset{\mathbb{P}\in\mathcal{S}_{\mathcal{A}^\star}}{\mathrm{Sup}}\{\|\mathbb{P}\|_2\}$, on a bien

$$\underset{n}{\mathrm{Sup}}\ \mathrm{P}\left[\underset{t\leqslant \mathrm{T}}{\mathrm{Sup}}\ |\mathbb{P}^n(t)| > \mathrm{K}\right] = 0$$

et donc la famille de processus $(\mathbb{P}^n)$ vérifie le critère 1 du théorème 7.2.4.

On utilise le lemme précédent pour démontrer que (y_i) vérifie le premier critère de tension. On étudie l'uniforme intégrabilité de la famille (y_i). En réalité, si on prend $|y| = \|y\|_\infty$, on a :

$$\mathbb{E}\left[|y_i|\chi_{|y_i|\geqslant \mathrm{K}}\right] = \sum_\omega \underset{\delta}{\mathrm{Sup}}\ \frac{\mathrm{C}(\omega,\delta)g(\omega)}{\mathbb{P}_i(\omega)}\mathbb{P}_i(\omega)\chi_{|y_i|\geqslant \mathrm{K}}$$

Comme on sait que :

$$\forall\omega \quad \forall\delta \qquad \frac{\mathrm{C}(\omega,\delta)g(\omega)}{\mathbb{P}_i(\omega)}\mathbb{P}_i(\omega) \leqslant \mathrm{G}|\omega|$$

où G est l'erreur la plus importante $g(\omega)$, on peut alors écrire que :

$$\mathbb{E}\left[|y_i|\chi_{|y_i|\geqslant \mathrm{K}}\right] \leqslant \mathrm{G}|\omega|\mathbb{E}\chi_{|y_i|\geqslant \mathrm{K}}$$

Mais on sait également que si $|y_i| > \mathrm{K}$, alors il existe δ_i tel que

$$\frac{\mathrm{C}(\omega,\delta_i)g(\omega)}{\mathbb{P}_i(\delta_i)} > \mathrm{K}$$

et finalement

$$\mathbb{P}_i(\delta_i) < \frac{\mathrm{G}|\omega|}{\mathrm{K}}$$

Mais l'événement $|y_i(\omega)| > \mathrm{K}$ est inclus dans l'évènement :

$$\exists\,\delta\in\omega \qquad \mathbb{P}_i(\delta_i) < \frac{\mathrm{G}|\omega|}{\mathrm{K}}$$

et cet évènement a une mesure majorée par $\mathrm{G}|\omega|/\mathrm{K}$. En fin de compte :

$$\mathbb{E}\left[|y_i|\chi_{|y_i|\geqslant \mathrm{K}}\right] \leqslant \frac{\mathrm{G}^2|\omega|^2}{\mathrm{K}} \longmapsto 0 \qquad \text{pour} \qquad \mathrm{K}\mapsto+\infty$$

Finalement, (y_i) est uniformément intégrable (condition (7.71)) et le lemme 7.3.1 assure alors que (Y^n) vérifie le premier critère de tension.

En étudiant W^n, on voit que si μ est un réel positif :

$$\mathrm{P}\left[\underset{t\leqslant \mathrm{T}}{\mathrm{Sup}}\ |\mathrm{W}^n(t)| \geqslant \mu\right] \leqslant \frac{\mathbb{E}\left[\sum_{i=n}^{m(\tau_n+\mathrm{T})-1}\sqrt{\varepsilon_i}\xi_i\right]^2}{\mu^2}$$

Comme les variables ξ_i sont indépendantes, centrées, de variances bornées, on a alors

$$\mathrm{P}\left[\underset{t\leqslant \mathrm{T}}{\mathrm{Sup}}\ |\mathrm{W}^n(t)| \geqslant \mu\right] \leqslant \frac{\mathrm{K}\sum_{i=n}^{m(\tau_n+\mathrm{T})-1}\varepsilon_i}{\mu^2} \leqslant \frac{\mathrm{K}}{\mu^2}\log\left(1+\mathrm{T}/\tau_n\right) \underbrace{\longmapsto}_{n\mapsto+\infty} 0$$

Ainsi, on a également

$$\lim_{\mathrm{K}\mapsto+\infty} \operatorname*{Sup}_n \mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}} |\mathrm{W}^n(t)| > \mathrm{K}\right] = 0$$

La famille (W^n) vérifie donc également le critère 1. Enfin, on déduit de l'égalité $\mathrm{Z}^n = \mathbb{P}^n - \mathrm{Y}^n - \mathrm{W}^n$ que (Z^n) vérifie le critère 1 du théorème 7.2.4. $\square$.

Enfin, nous allons utilisé à nouveau le lemme 7.3.1 sur les variables uniformément intégrables pour montrer que les familles de processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ vérifient bien le second critère de tension du théorème 7.2.4.

Lemme 7.3.3 (Vérification du critère 2 de tension)
Les processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ *vérifient la condition 2 du théorème 7.2.4 :*

$$\forall \mathrm{T} \geqslant 0 \qquad \lim_{\delta\mapsto 0} \lim_n \operatorname{Sup} \operatorname*{Sup}_{t\leqslant \mathrm{T}} \operatorname*{Sup}_{s\leqslant\delta} \mathbb{E}[\operatorname{Min}\{\|\mathrm{X}^n(t+s) - \mathrm{X}^n(t)\|; 1\}] = 0$$

pour $\mathrm{X}^n \in \{\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n\}$.

Preuve : Prenons $\mathrm{T} \geqslant 0$ et deux instants t et $t+s$ tels que $s \leqslant \delta$.

On étudie dans un premier temps l'expression de $\mathbb{E}[\operatorname{Min}\{\|\mathrm{Y}^n(t+s) - \mathrm{Y}^n(t)\|; 1\}]$ en effectuant la disjonction des cas : il y a un saut entre $\tau_n + t$ et $\tau_n + t + s$ ou il n'y en a pas. On a donc :

$$\mathbb{E}[\operatorname{Min}\{\|\mathrm{Y}^n(t+s) - \mathrm{Y}^n(t)\|; 1\}] = \underbrace{\mathbb{E}\left[\chi_{\{\exists\, t_{s_i} \quad t_{s_i}\in[\tau_n+t\,;\tau_n+t+s\,]\}} \operatorname{Min}\{\|\mathrm{Y}^n(t+s) - \mathrm{Y}^n(t)\|; 1\}\right]}_{=\mathrm{A}}$$
$$+ \underbrace{\mathbb{E}\left[\chi_{\{\forall t_{s_i} \quad t_{s_i}\notin[\,\tau_n+t\,;\tau_n+t+s\,]\}} \operatorname{Min}\{\|\mathrm{Y}^n(t+s) - \mathrm{Y}^n(t)\|; 1\}\right]}_{=\mathrm{B}}$$

A se majore par

$$\mathrm{A} \leqslant \mathbb{E}\left[\chi_{\{\exists\, t_{s_i} \quad t_{s_i}\in[\,\tau_n+t\,;\tau_n+t+s\,]\}} \times 1\right] = \mathrm{P}\left[\exists\, t_{s_i} \quad t_{s_i} \in [\,\tau_n + t\,; \tau_n + t + s\,]\right]$$

Comme les sauts sont répartis selon une loi exponentielle, on en déduit que

$$\mathrm{A} \leqslant \frac{e^{-\lambda(\tau_n+t)}\left(1 - e^{-\lambda s}\right)}{\lambda}$$

Enfin, on étudie B en remarquant que s'il n'y a pas de sauts entre $\tau_n + t$ et $\tau_n + t + s$, alors

$$\mathrm{Y}^n(t+s) - \mathrm{Y}^n(t) = \sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} \varepsilon_i y_i$$

Prenons ε positif quelconque. Comme y_i est uniformémént intégrable, on décompose y_i comme dans le lemme 7.3.1 en

$$y_i = y_{i,\mathrm{K}} + \zeta_{i,\mathrm{K}}$$

et on peut trouver K tel que

$$\operatorname*{Sup}_n \mathbb{E}\left[\sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} \varepsilon_i \left|\zeta_{i,\mathrm{K}}\right|\right] \leqslant \varepsilon/2$$

et *a fortiori* :

$$\operatorname*{Sup}_n \operatorname*{Sup}_{t\leqslant \mathrm{T}, s\leqslant\delta} \mathbb{E}\left[\sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} \varepsilon_i \left|\zeta_{i,\mathrm{K}}\right|\right] \leqslant \varepsilon/2$$

Pour un tel réel K, le terme en $y_{i,\mathrm{K}}$ se majore en

$$\mathbb{E}\left[\sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} \varepsilon_i \left|y_{i,\mathrm{K}}\right|\right] \leqslant \mathrm{K}\int_{\tau_n+t}^{\tau_n+t+s} \frac{\mathrm{d}\alpha}{\alpha} \leqslant \mathrm{K}\log\left(1+\frac{\delta}{\tau_n}\right)$$

et ce terme-là a bien une limite nulle lorsque n tend vers $+\infty$ et δ vers 0. Ainsi, le processus (Y^n) vérifie bien le second critère du théorème 7.2.4.

On utilise la formule (7.70) pour démontrer que (W^n) vérifie la condition 2 du théorème 7.2.4. Si ε est strictement positif, on a alors :

$$\mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}, s\leqslant\delta} \left|\mathrm{W}^n(t+s)-\mathrm{W}^n(t)\right| \geqslant \varepsilon\right] \leqslant \frac{\mathbb{E}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}, s\leqslant\delta} \left|\mathrm{W}^n(t+s)-\mathrm{W}^n(t)\right|^2\right]}{\varepsilon^2}$$

La différence $\mathrm{W}^n(t+s)-\mathrm{W}^n(t)$ se réécrit en $\sum \varepsilon_i\xi_i$ et donc

$$\mathrm{P}\left[\operatorname*{Sup}_{t\leqslant \mathrm{T}, s\leqslant\delta} \left|\mathrm{W}^n(t+s)-\mathrm{W}^n(t)\right| \geqslant \varepsilon\right] \leqslant \frac{\mathrm{C}\sum_{i=n}^{+\infty}\varepsilon_i}{\varepsilon^2} \leqslant \mathrm{C}\frac{\log\left(1+\frac{\delta}{\tau_n}\right)}{\varepsilon^2}$$

où C est la constante majorant la norme quadratique des ξ_i. Comme τ_n tend vers $+\infty$, on obtient :

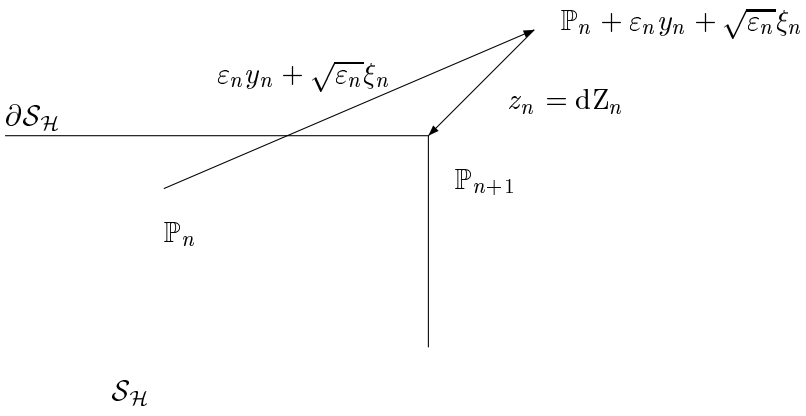
$$\lim_{\delta\mapsto 0}\lim_{n} \mathrm{P}\left[w'_{\mathrm{W}^n}(\delta) \geqslant \varepsilon\right] = 0$$

et (W^n) vérifie le second critère de tension.

Pour étudier le terme de réflexion, (Z^n), il suffit de remarquer que l'inégalité suivante est vraie

$$\left|\mathrm{Z}^n(t+s)-\mathrm{Z}^n(t)\right| \leqslant \sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} \left|\varepsilon_i y_i + \sqrt{\varepsilon_i}\xi_i\right|$$

En effet, la projection z_n se traduit par :



On constate donc ici que

$$\forall i \in \mathbb{N} \qquad |z_i| \leqslant \left|\varepsilon_i y_i + \sqrt{\varepsilon_i}\xi_i\right|$$

et $$\left|\mathrm{Z}^n(t+s)-\mathrm{Z}^n(t)\right| \leqslant \left|\sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} z_i\right| \leqslant \sum_{i=m(\tau_n+t)}^{m(\tau_n+t+s)} \left|\varepsilon_i y_i + \sqrt{\varepsilon_i}\xi_i\right|$$

En utilisant alors ce qui a été démontré pour les processus (Y^n) (variables y_i) et (W^n) (variables ξ_i), on en déduit également que la famille de processus (Z^n) vérifie le critère 2 de tension.

Enfin, on obtient le même résultat pour le processus $(\mathbb{P}^n)$ en remarquant que

$$\mathbb{P}^n = \mathrm{Y}^n + \mathrm{W}^n + \mathrm{Z}^n \qquad \square$$

Avec les deux lemmes précédents, on peut donc conclure que la famille de processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ est une famille tendue de processus, et donc séquentiellement faiblement compacte. On peut donc établir le théorème 7.3.1.

Théorème 7.3.1 (Compacité des trajectoires $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$)
La famille de processus $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ est tendue, et donc faiblement compacte.

7.4 Limite faible de $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$

Dans la section précédente, nous avons démontré que la famille de processus ainsi définie était tendue. Nous allons enfin voir que toute sous-suite faiblement convergente de $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ tend vers une diffusion sous-contraintes avec sauts du type $(\mathbf{E}-\mathbf{8})$.

Supposons tout d'abord que la suite de processus s'effectue sans saut, et reste confinée dans $\mathcal{S}_{\mathcal{H}}$, on a alors les deux propriétés ([KY03], chapitre 8) :

Propriété 7.4.1 (Approximation de W, processus de Wiener)
Le processus (W^n) tend faiblement vers W, processus de Wiener.

Propriété 7.4.2 (Approximation du gradient $\nabla^{\mathcal{H}}\mathcal{E}$)
Si $(\mathbb{P}^n)$ converge faiblement vers $\mathbb{P}$, alors les processus (G^n) définis par

$$\mathrm{G}^n(t) = -\sum_{i=n}^{m(\tau_n+t)-1} \nabla^{\mathcal{H}}\mathcal{E}_{err}(\mathbb{P}^n(s))\mathrm{d}s$$

convergent faiblement vers un processus G vérifiant l'équation différentielle stochastique :

$$\mathrm{dG}_t = -\nabla^{\mathcal{H}}\mathcal{E}_{err}(\mathbb{P}(t))\mathrm{d}t$$

La première propriété assure que le processus W^n va approcher faiblement le mouvement brownien standard dans $\mathcal{H}$ tandis que la propriété précédente nous assure que la convergence de $(\mathbb{P}^n)$ vers $\mathbb{P}$ implique l'approximation de la descente de gradient sur $\mathcal{E}_{err}$. Ces deux propriétés permettent alors de démontrer le théorème qui assure l'approximation stochastique faible du processus de diffusion réfléchi sans saut :

Théorème 7.4.1
Si $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ converge faiblement vers $(\mathbb{P}, \mathrm{Y}, \mathrm{W}, \mathrm{Z})$, alors $(\mathbb{P}, \mathrm{Z})$ vérifient le système d'équation différentielle $(\mathbf{E}-\mathbf{8})$ sans terme de sauts. Le processus tend alors faiblement vers une diffusion réfléchie sans sauts avec pour initialisation la limite faible des suites $(\mathbb{P}^n(0), \mathrm{Y}^n(0), \mathrm{W}^n(0), \mathrm{Z}^n(0))$.

Preuve : Il suffit de suivre la démonstration du théorème 2.3 de [KY03] ou du théorème 5.1 de [BK02] en définissant

$$\delta \mathrm{M}_n = \mathrm{Y}_n - \nabla^{\mathcal{H}} \mathcal{E}_{err}(\mathbb{P}_n)$$

et

$$\mathrm{M}^n(t) = \sum_{i=n}^{m(\tau_n+t)-1} \varepsilon_i \delta \mathrm{M}_i$$

puis approcher la descente de gradient exacte par le terme G^n défini dans la propriété précédente. $(\mathbb{P}^n)$ vérifie alors

$$\mathbb{P}^n(t) = \mathbb{P}^n(0) + \mathrm{G}^n(t) + \mathrm{W}^n(t) + \mathrm{Z}^n(t) + \mathrm{M}^n(t)$$

Le terme correspondant à $\mathrm{M}^n(t)$ a une espérance nulle sachant les événements $(\mathcal{F}(s), s \leqslant t)$. On en déduit alors que

$$\mathbb{E}\left[\mathbb{P}^n(t) - (\mathbb{P}^n(0) + \mathrm{G}^n(t) + \mathrm{W}^n(t) + \mathrm{Z}^n(t)) \,|\mathcal{F}(s), s \leqslant t\right] = 0$$

Mais le processus X, limite faible de $\mathbb{P}^n - (\mathbb{P}^n(0) + \mathrm{G}^n + \mathrm{W}^n + \mathrm{Z}^n)$, vaut précisément

$$\mathrm{X}^n \underbrace{\Longrightarrow}_{faible} \mathrm{X} = \mathbb{P} - \mathbb{P}_0 - \mathrm{G} - \mathrm{W} - \mathrm{Z}$$

Ce processus étant continu et nul en 0, localement lipschitzien X, on peut alors appliquer le théorème 1.1 du chapitre 4 de [KY03] pour en déduire que X est nul. Finalement $(\mathbb{P}, \mathrm{Z})$ est solution de l'équation $(\mathbf{E}-\mathbf{8})$ puisqu'en plus Z vérifie les propriétés données en 6 pour le problème de Skorokhod. En effet, Z n'augmente que lorsque $\mathbb{P}$ atteint la frontière et sa différentielle appartient bien à l'ensemble des vecteurs admissibles de réflection donnés en 5.2.2.2. □

Comme il existe une unique solution de $(\mathbf{E}-\mathbf{8})$, on en déduit donc que les limites faibles possibles pour $(\mathbb{P}^n, \mathrm{Y}^n, \mathrm{W}^n, \mathrm{Z}^n)$ sont en réalité unique (solution de $(\mathbf{E}-\mathbf{8})$). Ainsi, en appliquant la caractérisation de la limite faible des processus de $\mathcal{D}$ ([Bil99]), on en déduit que sans sauts, les processus $(\mathbb{P}^n, \mathrm{Z}^n)$ convergent faiblement vers la solution unique de $(\mathbf{E}-\mathbf{8})$ sans le terme de saut.

Enfin, les transitions de sauts étant simulées de façon exacte (on peut générer les (t_{s_i}) de façon exacte, on en déduit immédiatement que la suite des processus $(\mathbb{P}^n, \mathrm{Z}^n)$ converge en réalité vers la solution de l'équation différentielle stochastique du problème de diffusion sous contraintes avec sauts (sauts paramétrés par Q) donnée dans le chapitre 6 :

Théorème 7.4.2 (Convergence faible de $(\mathbb{P}^n, \mathrm{Z}^n)$)
Les processus constants par morceaux $(\mathbb{P}^n, \mathrm{Z}^n)$ tendent faiblement vers l'unique solution de l'équation différentielle stochastique $(\mathbf{E}-\mathbf{8})$ de loi l'unique loi stationnaire de la diffusion réfléchie avec sauts qui est le champ de Gibbs associé à l'énergie $\mathcal{E}(\mathcal{F}, \mathbb{P})$.

Preuve :

D'après le théorème 7.4.1 et la simulation exacte des instants de sauts, on sait que la suite des processus $(\mathbb{P}^n, \mathrm{Z}^n)$ est faiblement compacte et converge faiblement vers une diffusion réfléchie de générateur identique à celui de $(\mathbf{E}-\mathbf{8})$. Toute sous-suite $\left(\mathbb{P}^{\mathrm{N}_k}, \mathrm{Z}^{\mathrm{N}_k}\right)$ extraite de $(\mathbb{P}^n, \mathrm{Z}^n)$ converge faiblement vers la diffusion réfléchie avec sauts $(\mathbf{E}-\mathbf{8})$ de distribution initiale la limite faible de $(\mathbb{P}^{\mathrm{N}_k}(0), \mathrm{Z}^{\mathrm{N}_k}(0))$ notée ν_∞. Démontrons que cette limite faible est le champ de Gibbs μ associé à $\mathcal{E}$.

Il suffit pour cela de démontrer que si ν_∞ désigne une limite faible d'une sous-suite $(\mathbb{P}^{\mathrm{N}_k}(0), \mathrm{Z}^{\mathrm{N}_k}(0))$, alors pour toute fonction ϕ :

$$\int \phi(y)\mathrm{d}\nu_\infty(y) = \int \phi(y)\mathrm{d}\mu(y)$$

Si l'on note P_ν^t la loi du processus donné par $(\mathbf{E}-\mathbf{8})$ à l'instant t, initialisé avec la loi ν, comme μ est la loi stationnaire du processus on a

$$\forall \nu \in \mathrm{K} \quad \forall \varepsilon > 0 \quad \exists\, \mathrm{T} > 0 \quad \forall\, t \geqslant \mathrm{T} \qquad \left| \int \phi(y)\mathrm{dP}_\nu^t(y) - \int \phi(y)\mathrm{d}\mu(y) \right| \leqslant \varepsilon \tag{7.74}$$

où K est un ensemble compact de mesures. Fixons $\varepsilon > 0$ et considérons un tel T, quitte à extraire à nouveau une sous-suite on peut noter ν'_∞ la limite faible du processus $\left(\mathbb{P}^{\mathrm{N}_k}(.-\mathrm{T}), \mathrm{Z}^{\mathrm{N}_k}(.-\mathrm{T})\right) = (\mathbb{P}(\tau_{\mathrm{N}_k}-\mathrm{T}), \mathrm{Z}(\tau_{\mathrm{N}_k}-\mathrm{T}))$. La famille des lois de $\left(\mathbb{P}^{\mathrm{N}}, \mathrm{Z}^{\mathrm{N}}\right)$ est tendue, et donc séquentiellement compacte, on peut appliquer l'inégalité (7.74) et on a alors :

$$\begin{aligned}
\left| \int \phi(y)\mathrm{d}\nu_\infty(y) - \int \phi(y)\mathrm{d}\mu(y) \right| \leqslant & \left| \int \phi(y)\mathrm{d}\nu_\infty(y) - \mathbb{E}\left[\phi\left(\mathbb{P}(\tau_{\mathrm{N}_k}), \mathrm{Z}(\tau_{\mathrm{N}_k})\right)\right] \right| \\
& + \left| \mathbb{E}\left[\phi\left(\mathbb{P}(\tau_{\mathrm{N}_k}), \mathrm{Z}(\tau_{\mathrm{N}_k})\right)\right] - \int \phi(y)\mathrm{dP}^{\mathrm{T}}_{\nu'_\infty}(y) \right| \\
& + \left| \int \phi(y)\mathrm{dP}^{\mathrm{T}}_{\nu'_\infty}(y) - \int \phi(y)\mathrm{d}\mu(y) \right|
\end{aligned}$$

En faisant tendre N_k vers l'infini, τ_{N_k} tend également vers l'infini et par construction de T et définition des limites faibles ν_∞ et ν'_∞ :

$$\left| \int \phi(y)\mathrm{d}\nu_\infty(y) - \int \phi(y)\mathrm{d}\mu(y) \right| \leqslant \varepsilon$$

Finalement $$\nu_\infty = \mu \qquad p.s.$$

Ainsi, toute sous-suite convergente tend faiblement vers la distribution de Gibbs stationnaire du processus $(\mathbf{E}-\mathbf{8})$. Cela assure finalement que la suite de processus $(\mathbb{P}^n, \mathrm{Z}^n)$ ainsi que $(\mathbb{P}^n(0), \mathrm{Z}^n(0))$ tendent faiblement vers la loi stationnaire μ. □

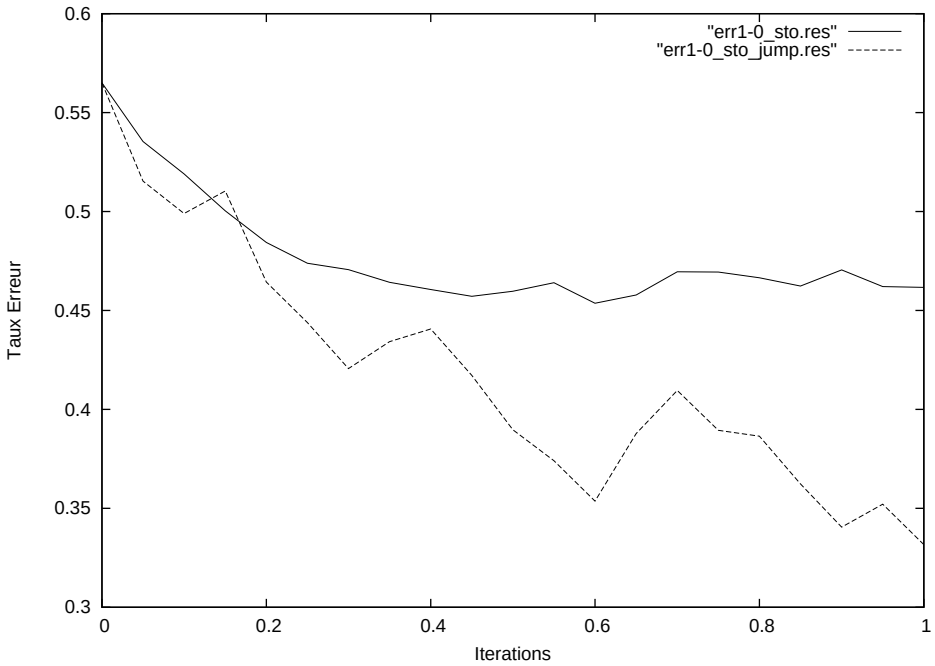
7.5 Expériences

Parmi les listes des données étudiées dans le chapitre 3, on ne peut pas appliquer notre algorithme à la classification de messages électroniques. Par contre, notre modélisation s'applique naturellement au cas des données synthétiques ainsi qu'à la reconnaissance d'objets dans une image. Nous allons présenter les résultats des expériences obtenus sur les données synthétiques et sur la base de données [MIT].

7.5.1 Données synthétiques

Pour une description précise de la façons dont sont générées les données, nous renvoyons au paragraphe 3.7.1. Nous rappelons que les features élémentaires sont à valeurs ternaires et que les 3 classes $\mathcal{C}_i$ sont issues des sources $\mathcal{F}_i^j$. Par ailleurs, nous avons simulé différents types de données synthétiques, et toutes les expériences que nous avons réalisées *via* notre algorithme de « Jump-Diffusion » ont révélé des résultats similaires, quels que soient le nombre de classes étudiées ou le nombre de sources $\mathcal{F}_i^j$ ayant permis de générer chacune des $\mathcal{C}_i$.

En considérant toutefois le même exemple qu'au chapitre 3, les résultats obtenus sont probants : Voici l'évolution du taux d'erreur lorsque l'on exécute notre algorithme de diffusion avec sauts. La courbe obtenue dans cette situation peut alors être comparée à la courbe d'erreur obtenue sans l'ajoût du processus de saut au chapitre 3.



Évolution des taux d'erreur par Descente de gradient et Jump-Diffusion en fonction du temps

Les performances obtenues lors des expériences avec sauts sont nettement meilleures que lorsqu'on choisit un processus sans modification des variables (pas de terme de saut). Cet exemple synthétique nous permet donc de constater l'efficacité de notre approximation et de notre modèle. La quantité d'arbres initiaux était de 20 et ne diminue pas franchement, mais en revanche, la nature de ces arbres évolue complètement. On constate en effet que la profondeur moyenne de ces arbres est d'environ 3, ce qui constitue déjà une bonne complexité à la vue de la génération des données synthétiques.

Par ailleurs, l'étude de cet exemple simple nous permet de stocker numériquement l'occupation moyenne des arbres $\mathcal{A}$ à l'intérieur des forêts $\mathcal{F}_t$ données par la définition suivante.

Définition 7.5.1 (Mesure d'occupation moyenne)
Si $\mathcal{A}$ est un arbre de $\mathcal{A}^\star$, on définit alors la quantité :

$$\mu_t(\mathcal{A}) = \frac{1}{t}\int_0^t \chi_{\mathcal{A}\in\mathcal{F}_{t_s}}\,\mathrm{d}s$$

Cette quantité permet de mesurer, en moyenne, la présence de l'arbre $\mathcal{A}$ dans $\mathcal{F}_t$, au cours du temps.

Grâce aux quantités suivantes, nous pouvons alors exhiber les arbres « les plus souvent présents » dans la forêt aléatoire $\mathcal{F}_t$. Il se trouve que les arbres les plus présents sont alors ceux

qui sont définis pour constituer les sources $\mathcal{F}_i^j$. Rappelons tout d'abord que les sources étaient exactement :

$$\mathcal{F}_1{}^1 = \{\delta_1; \delta_3; \delta_5; \delta_7\} \quad \text{et} \quad \mathcal{F}_1{}^2 = \{\delta_1; \delta_5\} \quad \text{et} \quad \mathcal{F}_1{}^3 = \{\delta_3; \delta_7\}$$

puis $$\mathcal{F}_2{}^1 = \{\delta_2; \delta_4; \delta_6; \delta_8\} \quad \text{et} \quad \mathcal{F}_2{}^2 = \{\delta_2; \delta_4\} \quad \text{et} \quad \mathcal{F}_2{}^3 = \{\delta_6; \delta_8\}$$

et enfin $$\mathcal{F}_3{}^1 = \{\delta_1; \delta_4; \delta_8; \delta_9\} \quad \text{et} \quad \mathcal{F}_3{}^2 = \{\delta_1; \delta_8\} \quad \text{et} \quad \mathcal{F}_3{}^3 = \{\delta_4; \delta_9\}$$

On trouve alors dans l'ordre décroissant de plus forte présence dans la forêt des arbres constitués de noeuds principaux valant :

$$\{\delta_2; \delta_4\} \quad \{\delta_1; \delta_5\} \quad \{\delta_4; \delta_9\} \quad \{\delta_1; \delta_8\} \quad \{\delta_6; \delta_8\} \quad \{\delta_3; \delta_7\} \quad \delta_1 \quad \delta_4 \quad \delta_8$$

Puis viennent ensuite des arbres dont les noeuds principaux sont formés des tests :

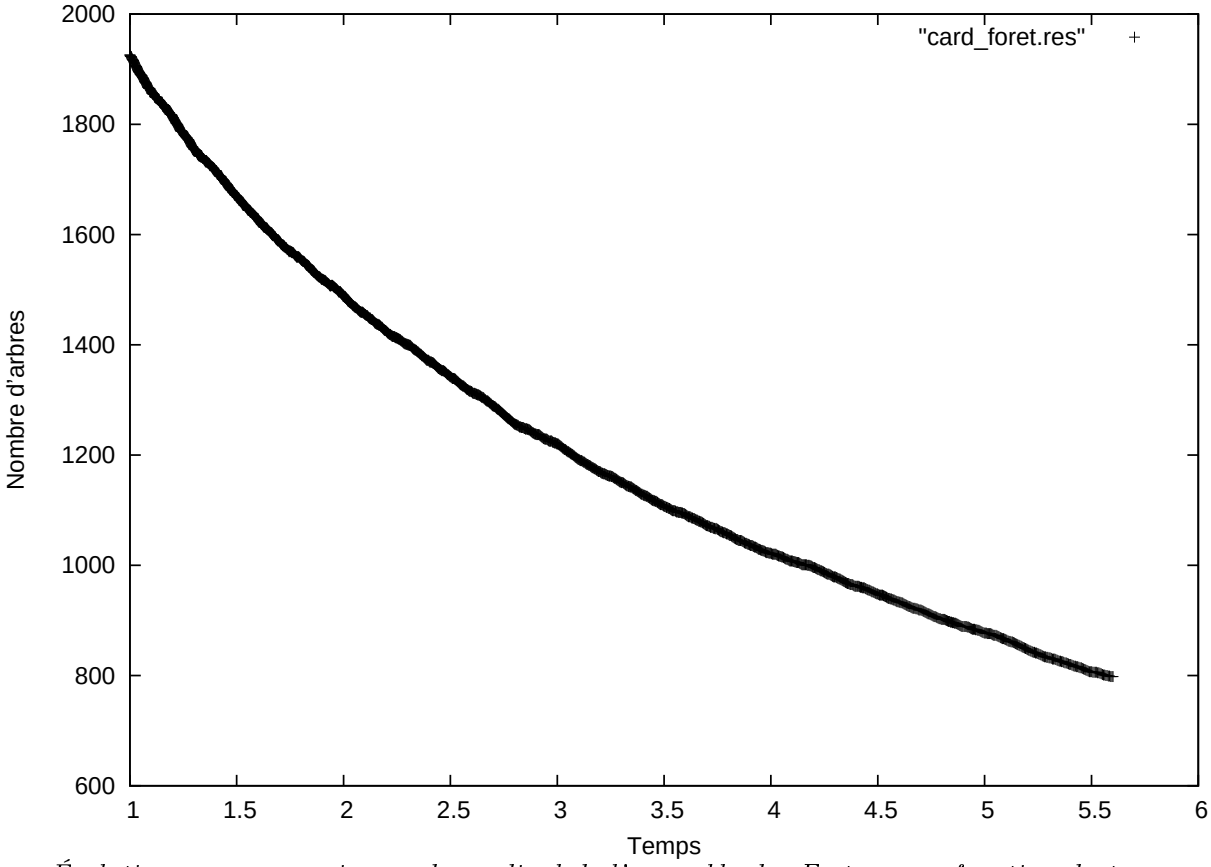
$$\{\delta_1; \delta_5; \delta_3\} \quad \{\delta_2; \delta_4; \delta_8\} \quad \{\delta_4; \delta_9; \delta_8\}$$

Il est important de constater que vis-à-vis des sources initiales, les arbres qui sont les plus présents dans $\mathcal{F}_t$ tendent à reconstituer précisément ces $\mathcal{F}_i^j$.

Enfin, les arbres ne comprenant pas de tests élémentaires constituant les $\mathcal{F}_i^j$ sont purement et simplement effacés des ensembles de features $\mathcal{F}_t$, ce qui n'était bien évidemment pas possible dans le cadre du modèle développé au chapitre 3.

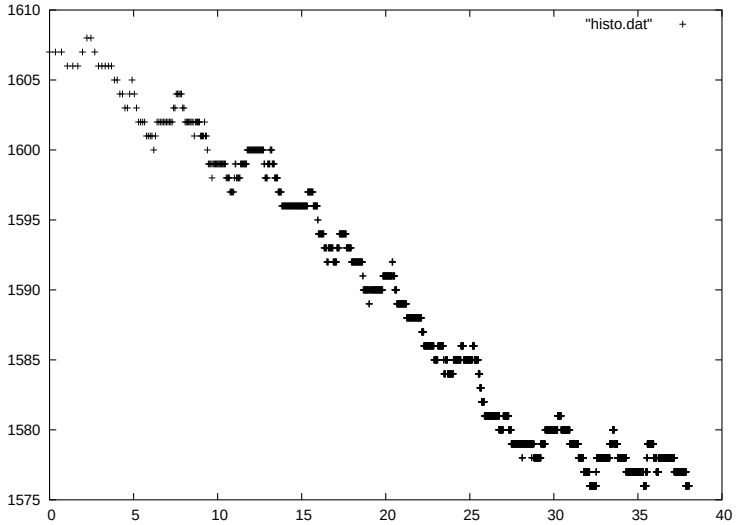
7.5.2 Détection de Visages

Nous pouvons représenter l'évolution du cardinal de la forêt $\mathcal{F}_{\tau(n)}$, c'est-à-dire la quantité d'éléments présents dans la forêt au cours du temps.



Évolution « macroscopique » du cardinal de l'ensemble des Features en fonction du temps

La décroissance du nombre d'arbres dans $\mathcal{F}_t$ est donc non-négligeable. Même s'il paraît que le nombre d'arbres tend à se stabiliser lorsque t est grand, ce n'est absolument pas le cas. En « zoomant », nous obtenons en effet une courbe de la forme :



Évolution « microscopique » du cardinal de l'ensemble des Features en fonction du temps

Dans la courbe précédente, nous n'avons pas tout à fait utilisé la même échelle de temps, ce qui explique la différence d'unités et d'« allure » entre la première et la seconde courbe.

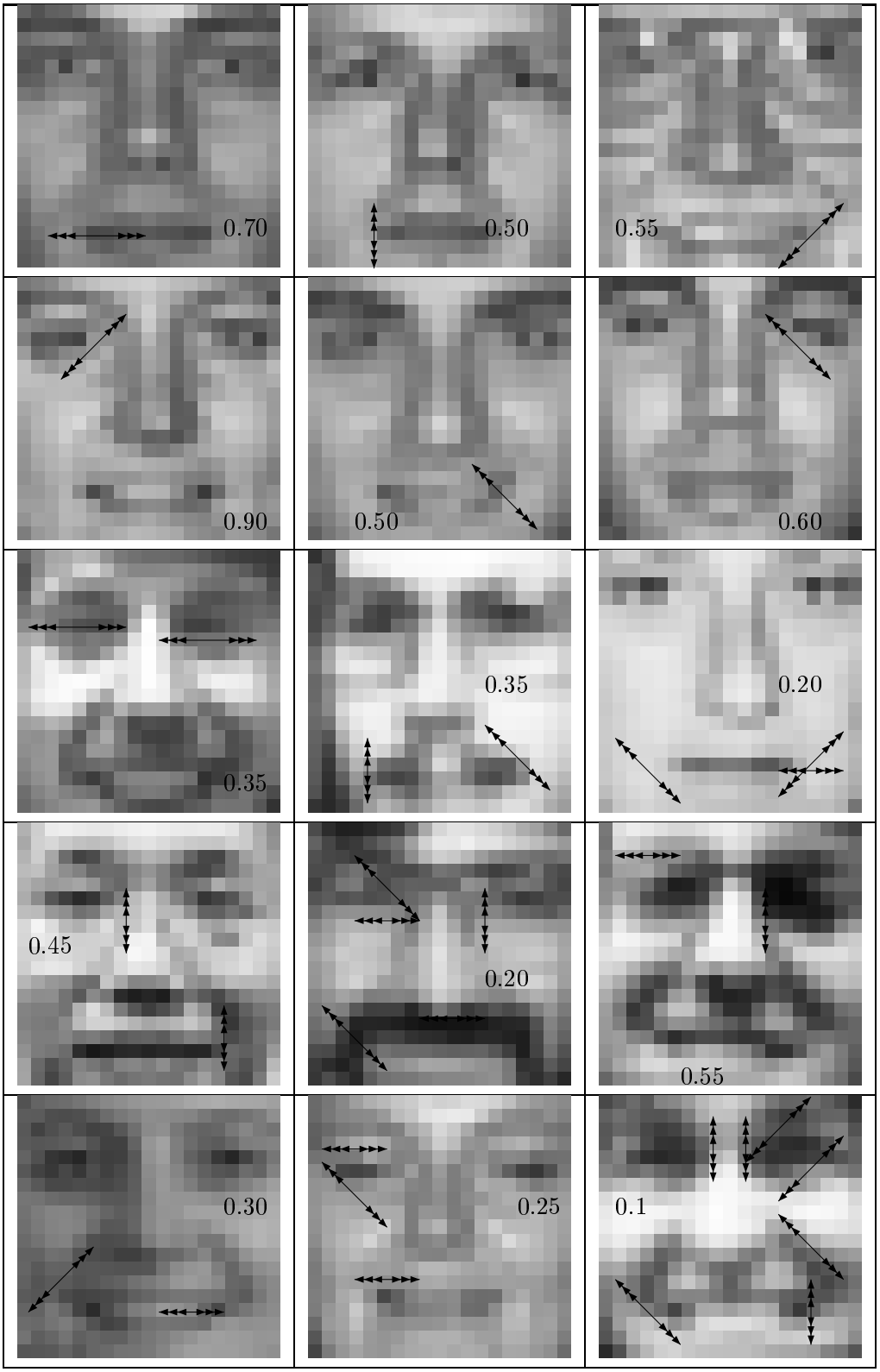
Pour obtenir le taux d'erreur général de l'algorithme sur le Test-Set, nous effectuons là encore un tirage selon $\mathbb{P}_t$ et organisons un vote de détecteurs, comme ce qui est effectué dans le paragraphe 3.8.

Nous tirons donc 10 p-uplets pour $p = 50$ ou $p = 100$, et nous obtenons des taux d'erreurs très convaincants vis-à-vis de notre problème de détection :

- Le taux de faux-positifs est nul
- Le taux de faux négatif vaut 1.5% dans le cas où $p = 50$ et 1.2% lorsque $p = 100$.

Nous rappelons que lors de l'optimisation de l'algorithme à dictionnaire fixe, le meilleur taux global que nous avions obtenu était de 3.5% d'erreur. Il y a donc une très nette amélioration des performances de la détection lorsque nous utilisons notre algorithme de « Jump-Diffusion » pour agréger les détecteurs binaires de bords.

Enfin, il est intéressant de représenter là encore les exemples de features (composés ou non) qui sont privilégiés par notre algorithme.



Notre modèle s'avère particulièrement adapté au problème de la détection de visages, et ce pour plusieurs raisons :

- La quantité de features diminue fortement en fonction du temps $\tau(n)$.
- Le nombre moyen de détecteurs de bords pour chaque feature vaut autour de 4, ce qui signifie que la forêt aléatoire est composé d'arbres relativement complexes. De plus, nous avons stoppé l'algorithme dès que le nombre de variables dans $\mathcal{F}_t$ nous a semblé suffisament bas, mais la poursuite de l'algorithme nous permet de faire encore augmenter la profondeur moyenne des arbres. L'arbre le plus long que nous obtenons, et qui est réalisé avec une probabilité non négligeable (supérieure à 5%), a pour profondeur 12.
- La quantité d'erreur sur le Test-Set diminue énormément puisque le taux de faux positif est nul alors que le taux de faux négatif vaut approximativement 1.2%. Ceci s'explique par le fait que même si les arbres sont relativement profonds, la probabilité de réalisation de tels arbres sur les images de fonds est maintenue suffisament grande par le terme $\mathcal{E}_\rho$ de l'énergie
- Enfin, on peut constater que les néologismes formés (nouveaux arbres) ne sont pas laissés de côté par l'algorithme mais bien utilisés puisque par exemple un arbre de longueur 7 fait partie des 15 arbres les plus « utiles » (voir schéma précédent).

Chapitre 8 - Conclusion

8.1 Bilan

L'objet de notre travail est de proposer une méthode de sélection de variables, afin d'optimiser le problème de la reconnaissance de formes. Cette problématique est motivée initialement par la possibilité, à partir de la proposition de certaines variables explicatives, d'effectuer un nombre important de tâches visuelles. La modélisation de notre problème consiste alors à construire une loi de probabilité sur ces variables explicatives qui peuvent être des variables à la fois sur des images, mais aussi sur d'autre type de signal.

Nous avons alors présenté successivement deux algorithmes d'apprentissage, l'un à dictionnaire fixe, l'autre à dictionnaire variable, permettant de cerner les variables influançant favorablement la tâche de la reconnaissance de formes dans un signal.

Le premier algorithme permet de traiter des données quelconques. En revanche, l'approche à dictionnaire variable ne s'applique, lui, qu'au traitement de données conférant une signification particulière à l'agrégation de variables, par exemple dans le cas de variables à valeurs binaires ou ternaires.

Les principaux outils utilisés pour construire un tel algorithme peuvent se résumer en :

- l'approximation stochastique d'une descente de gradient et d'une diffusion sous contrainte.
- la dynamique markovienne de l'évolution des dictionnaires, lorsque ceux-ci peuvent varier, une telle dynamique permet alors l'exploration de tous les espaces possibles de variables, lorsque celles-ci peuvent subir des règles de greffe, de coupe et de renaissance.
- l'utilisation systématique d'un algorithme $\mathbb{A}$ de classification rapide sur une quantité de variables restreintes.

L'algorithme de construction et d'évolution de notre population de variables ressemble un peu à ce qui fait dans le cadre des algorithmes génétiques ([Cer94], [Cer96],[Cer98]) . Néanmoins, les méthodes pour étudier théoriquement de tels algorithmes sont nettement plus complexes ([FW98]) que les techniques d'approximation et de calcul stochastique que nous utilisons.

Nous prenons en compte différentes propriétés statistiques d'apprentissage, de la théorie de l'information ou de la théorie du traitement de l'image pour obtenir des bonnes performances sur les exemples que nous avons abordés.

Le cas des variables synthétiques nous permet d'exhiber l'importance d'un critère basé sur l'information mutuelle pour générer une agrégation pertinente de variables tandis que dans le cas particulier des images, c'est précisément la simplicité des features élémentaires (détecteurs de bords, annexe A) et les techniques de hiérarchisation de tests ([FG01]) qui nous permet d'obtenir le taux de 1.5% d'erreur sur la détection de visages. Nous avons par ailleurs vu comment aménager notre approche lorsque l'on souhaite implémenter un algorithme où les détecteurs sont alors invariants par translation.

8.2 Points forts

De notre modélisation, nous pouvons donc retenir les points forts suivants.

- L'algorithme nécessite une faible quantité de stockage de variables lors de la détection, ce qui est d'une importance capitale lorsque'on souhaite rendre portable l'algorithme (système électronique embarqué à taille réduite par exemple).
- Le traitement de la reconnaissance de formes dans les signaux étudiés est exécuté très rapidement (moins de 1 ms pour des emails, 3 ms pour des images de taille restreinte), ce qui peut également permettre d'utiliser plusieurs collaborations des résultats de nos algorithmes (vote de détecteurs par exemple, ou implémentation de patchs utilisant nos détecteurs lors du parcours de l'espace des poses dans une image de grande taille).
- Notre approche est générique à tout problème de sélection de variables (du moins à dictionnaire fixe), indépendamment de la nature des features et des données. Ainsi, nous pourrions également appliquer notre modélisation et sa résolution pour tout type de sélection de features dans des problèmes d'imagerie par ordinateur mais aussi dans la détection de protocole pour le scan des couches TCP/IP.
- Les résultats de détections en terme de pourcentage d'erreur, sont à la hauteur de ce qui a été fait dans le cadre d'autres algorithmes (taux d'erreur sur la détection de SPAM [UCI], de visages [MIT] ou de chiffres manuscrits [USP])
- Notre approche permet de révéler un sens cognitif intéressant pour certaines variables, c'est notamment le cas dans le problème de la détection de SPAM (sélection de mots comme « George », « free » etc.).

8.3 Points faibles

Nous pouvons cependant mentionner quelques points faibles à notre approche.

- Pour obtenir des taux d'erreurs relativement bons, il a été necessaire de manipuler des bases de données contenant un nombre important d'objets (4600 messages électroniques, 7000 images pour la détection de visages et environ 60000 pour la reconnaissance de chiffres manuscrits).
- L'implémentation de l'algorithme d'apprentissage est couteuse puisqu'il est nécessaire de représenter récursivement les arbres binaires des forêts et que l'apprentissage peut être très long (deux jours pour le problème de la reconnaissance de visages, quelques heures pour le SPAM).
- Les performances pour la base de données [USP] à dictionnaire variable, en terme de taux d'erreur se sont révélées assez décevantes. Nous n'avons pas réussi à établir une baisse significative du taux d'erreur. De plus, la sélection de variables n'a pas permis de construire des tests interprétables « facilement » sur une classe de chiffres manuscrits comme dans le cas des visages [MIT].
- L'implémentation de la hiérarchisation des tests pourrait être améliorée, en terme d'efficacité algorithmique notamment. Nous n'avons pas privilégié l'exploration de certaines poses de tests et nous n'avons pas choisi l'évaluation prioritaire de certains tests pour calculer un détecteur constitués de plusieurs tests complexes (absence de stratégie « type jeux des 20 questions »).

8.4 Poursuite des travaux

Il est possible de donner quelques développements possibles pour poursuivre notre approche.

- Du point de vue théorique, il serait intéressant d'étudier un algorithme de diffusion basé sur une diminution du logarithme de l'erreur moyenne. Les critères pour obtenir la convergence d'un tel algorithme d'apprentissage pourraient alors utiliser le taux d'ergodicité de Dobrushin par exemple (*cf* Annexe C, section 6).
- Nous pouvons également penser à tester des patchs formés *via* les features sélectionnés par notre apprentissage et explorer l'ensemble des poses dans une image de grande taille. Le problème serait alors de repérer un visage, ou un chiffre manuscrit dans une image, sans que la pose soit définie (ce qui est le cas dans les différents exemples étudiés dans cette thèse).
- Par ailleurs, le point de vue du traitement informatif des tests statistiques et d'efficacité algorithmique pourrait être largement amélioré, puisque en l'état, aucune stratégie de parcours des tests binaires n'est privilégiée et les agrégations de tests binaires utilisés pour [USP] se sont montrés par exemple trop discriminant dès l'agrégation d'au moins deux features.
- Enfin, on peut imaginer qu'un algorithme de « Jump-Diffusion » pourrait très bien être mis en oeuvre pour l'apprentissage d'un langage naturel et d'une grammaire naturelle à partir d'une grand base de données de textes numérisés. De tels algorithmes s'utiliseraient également pour des problèmes de classification de textes, en fonction de leurs auteurs, époques, etc.

Annexe A -Espaces des Features pour les images

A-1 Détecteurs de bords

Détecteurs primitifs de bords verticaux

Les détecteurs primitifs ε_1 et ε_2 de bords verticaux sont des détecteurs de bords orientés, les transitions foncé-clair et clair-foncé sont donc différenciées. Si I désigne l'image étudiée et (x, y) la position dans l'image où l'on désire se placer, on pose :

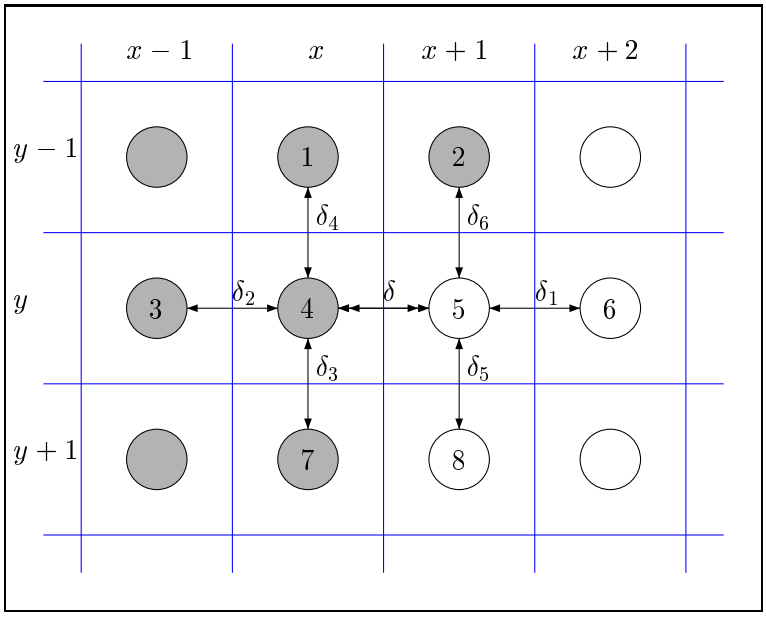
$$\begin{cases} \delta(x,y)(\mathrm{I}) = |\mathrm{I}(x,y) - \mathrm{I}(x+1,y)| \\ \delta_1(x,y)(\mathrm{I}) = |\mathrm{I}(x+1,y) - \mathrm{I}(x+2,y)| \\ \delta_2(x,y)(\mathrm{I}) = |\mathrm{I}(x-1,y) - \mathrm{I}(x,y)| \\ \delta_3(x,y)(\mathrm{I}) = |\mathrm{I}(x,y) - \mathrm{I}(x,y+1)| \\ \delta_4(x,y)(\mathrm{I}) = |\mathrm{I}(x,y-1) - \mathrm{I}(x,y)| \\ \delta_5(x,y)(\mathrm{I}) = |\mathrm{I}(x+1,y) - \mathrm{I}(x+1,y+1)| \\ \delta_6(x,y)(\mathrm{I}) = |\mathrm{I}(x+1,y-1) - \mathrm{I}(x+1,y)| \end{cases}$$

Enfin, on définit le détecteur binaire ε_1 par

$$\varepsilon_1(x,y)(\mathrm{I}) = 1 \Longleftrightarrow \begin{cases} \text{Card } \{i \,|\, \delta_i(x,y)(\mathrm{I}) < \delta(x,y)(\mathrm{I})\} \geqslant 5 \\ \mathrm{I}(x,y) > \mathrm{I}(x+1,y) \end{cases}$$

et le détecteur ε_2 par

$$\varepsilon_2(x,y)(\mathrm{I}) = 1 \Longleftrightarrow \begin{cases} \text{Card } \{i \,|\, \delta_i(x,y)(\mathrm{I}) < \delta(x,y)(\mathrm{I})\} \geqslant 5 \\ \mathrm{I}(x,y) < \mathrm{I}(x+1,y) \end{cases}$$



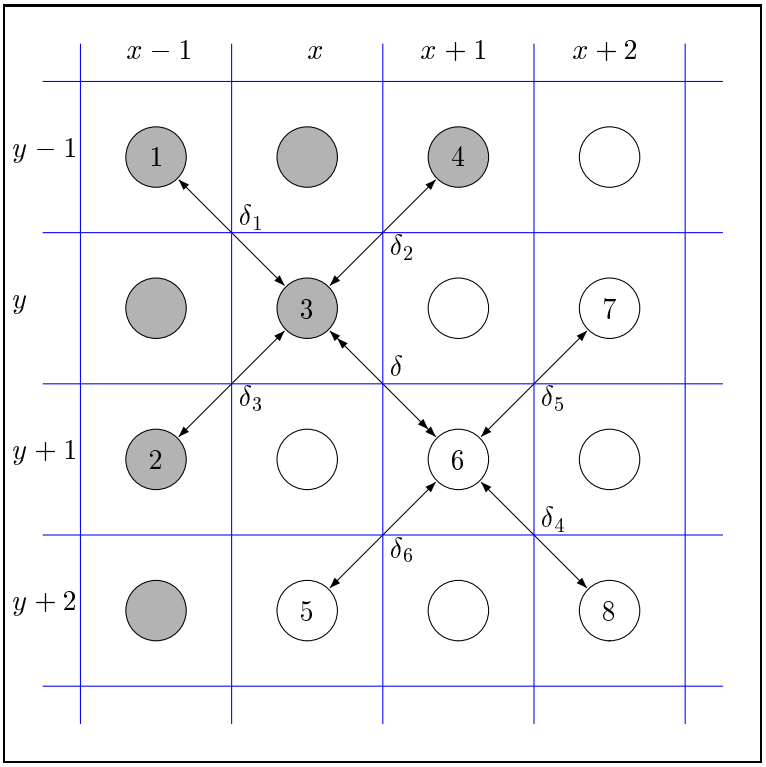
Voisinage pour la détection de bord vertical en (x, y).

Détecteurs primitifs de bords horizontaux

Les détecteurs primitifs ε_3 et ε_4 se construisent aisément de la même manière que dans le paragraphe précédent, en utilisant un voisinage de 8 pixels contenant le pixel (x, y) obtenu par rotation d'angle $\pi/2$ par rapport au précédent voisinage. Le mode de calcul est strictement identique, les détecteurs renvoient 1 dès que la différence δ est supérieure strictement à au moins 5 différences δ_i.

Détecteurs primitifs de bords obliques

Il y a quatre détecteurs primitifs de bords obliques. ε_5 et ε_6 sont des détecteurs de bords dans la direction diagonale orientée selon l'angle $\pi/4$, tandis que ε_7 et ε_8 sont les détecteurs de bords orientés selon la direction $-\pi/4$. Là encore, le mode de calcul est similaire aux précédents calculs des détecteurs primitifs. On décide de détecter un bord dès que δ est en valeur absolue supérieur à 5 des 6 autres différences. Pour une bonne compréhension de la définition de ce détecteur de bord, nous renvoyons au schéma suivant du voisinage et de la détection d'un bord selon la direction $\pi/4$.



Voisinage pour la détection de bord diagonal $\pi/4$ en (x, y).

Voici quelques exemples de bords détectés pour les tests ε_i positifs précédents calculés sur les images de chiffres manuscrits de la base [USP] constituée de chiffres de taille 16×16.

Classe	Image I	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7	ε_8
$\mathcal{C}_0$									
$\mathcal{C}_1$									
$\mathcal{C}_2$									
$\mathcal{C}_3$									
$\mathcal{C}_4$									
$\mathcal{C}_5$									
$\mathcal{C}_6$									
$\mathcal{C}_7$									
$\mathcal{C}_8$									
$\mathcal{C}_9$									

Définitions des détecteurs élémentaires

Tous les features plus complexes qui seront construits sont formés des briques élémentaires constituant l'alphabet $\mathcal{A}$. Cet alphabet $\mathcal{A}$ est construit initialement à partir du learning set d'images [USP].
Une lettre de l'alphabet $l \in \mathcal{A}$ est un test élémentaire orienté (Noir/Blanc ou Blanc/Noir) localisé en un pixel (c_x, c_y) de direction verticale, horizontale, ou suivant l'une des deux diagonales et basé sur un flou f qui est particulier à chacune des lettres de l'alphabet.
Via toutes les définitions précédentes, on peut donc définir :

Définition 1 (Tests élémentaires δ détecteurs de bords)
On appelle δ un test élémentaire détecteur de bords, la variable booléenne dépendant des quatre paramètres $(\varepsilon, f, c_x, c_y)$ où

- *ε paramétrise l'orientation de la discontinuité ($\varepsilon \in \{\varepsilon_1, \dots \varepsilon_{16}\}$)*
- *f paramétrise le flou associé au test*
- *(c_x, c_y) paramétrise la localisation du test dans l'image.*

La définition précédente mérite un ultime éclaircissement au sujet du paramètre de « flou » mentionné. Le paramètre de flou est un paramètre qui détermine la taille du voisinage sur lequel on calculera la disjonction des détecteurs de bords précédemment définis sur une image. Ce flou est, par ailleurs, orthogonal à l'orientation du bord. Plus précisément :

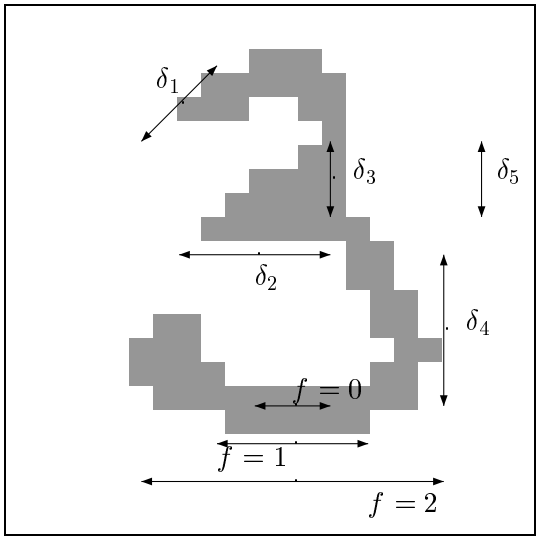
$$\forall i \in \{1,2\} \qquad \delta_{\varepsilon_i,f,c_x,c_y} = \max\{\varepsilon_i(c_x - f, c_y), \dots \varepsilon_i(c_x + f, c_y)\}$$

$$\forall i \in \{3,4\} \qquad \delta_{\varepsilon_i,f,c_x,c_y} = \max\{\varepsilon_i(c_x, c_y - f), \dots \varepsilon_i(c_x, c_y + f)\}$$

$$\forall i \in \{5,6\} \qquad \delta_{\varepsilon_i,f,c_x,c_y} = \max\{\varepsilon_i(c_x - f, c_y - f), \dots \varepsilon_i(c_x + f, c_y + f)\}$$

$$\forall i \in \{7,8\} \qquad \delta_{\varepsilon_i,f,c_x,c_y} = \max\{\varepsilon_i(c_x + f, c_y - f), \dots \varepsilon_i(c_x - f, c_y + f)\}$$

Voici enfin un exemple d'image réelle sur laquelle certains tests floués sont réalisés : une flèche correspond à la réalisation d'un détecteur de bord, la taille de la flèche est proportionnelle au flou associé au détecteur tandis que son orientation est identique à celle du détecteur de bord. Le centre de la flèche est identique bien entendu à la localisation du détecteur.



L'alphabet $\mathcal{A}$ est construit initialement en balayant tous les tests possibles l (positions des pixels, valeurs des flous, orientation du test, direction de la discontinuité).

On perçoit dès lors toute la difficulté de la sélection des bons features pour la tâche de détection, mais aussi sa nécessité puisque nous disposons de $16 \times \mathrm{F} \times \mathrm{N}_x \times \mathrm{N}_y$ détecteurs de bords possibles, soit la même quantité de valeurs binaires si N_x et N_y désignent la taille des images et F le nombre possible de flous.

Sélection des détecteurs élémentaires pour la tâche de classification

Le but de la sélection de features, et de la construction de features plus complexes est, nous l'avons vu, d'obtenir un nombre restreint de features, préservant tout de même une bonne capacité de classification. Nous allons donc dans cette optique opérer, comme ce qui a été fait dans [FG01] une première sélection grossière des features élémentaires pour le problème de classification sur les classes $\mathcal{C}_i$.

On pré-sélectionne dans l'ensemble initial des features que l'on va manipuler les tests de façon à ce qu'il y ait suffisamment d'images sur lesquelles les détecteurs répondent 1. On joue pour cela sur le paramètre f de flou, quitte à ne pas garder le test si aucune des valeurs de flou ne donne des valeurs satisfaisantes de probabilité de réussite. Plus précisément, on définit le détecteur initial ${\mathcal{D}_0}^+$ par :

Définition 2 (Dictionnaire initial ${\mathcal{D}_0}^+$)
L'ensemble ${\mathcal{D}_0}^+$ des détecteurs élémentaires positifs initiaux soit donné par

$$\delta_{\varepsilon,f,c_x,c_y} \in {\mathcal{D}_0}^+ \Longleftrightarrow \begin{cases} \exists\, i \in \{1 \dots \mathrm{C}\} \qquad \mathrm{P}_{\mathcal{C}_i}(\delta_{\varepsilon,f,c_x,c_y} = 1) \geqslant 1/2 \\ \varepsilon \in \{\varepsilon_1, \dots \varepsilon_8\} \\ f = \min\left\{ \widetilde{f} \mid \delta_{\varepsilon,\widetilde{f},c_x,c_y} \in {\mathcal{D}_0}^+ \right\} \\ f \leqslant 5 \end{cases}$$

On notera que dans la définition de ${\mathcal{D}_0}^+$, nous avons imposé que le flou est borné par 5, ceci pour garder un aspect de localisation du test, les images sont en effet dans le cas de la base de donnée [USP] de taille relativement petite (16×16).

Annexe B - Conditions de stabilité de $\mathcal{S}_\mathcal{F}$

Théorème 3.4.1 (Stabilité de $\mathcal{S}_\mathcal{F}$ sous l'équation d'évolution ($\mathbf{E}-\mathbf{5}$))
Si l'on suppose que α et β vérifient

$$\frac{\alpha}{\beta} \leqslant \frac{\mathcal{U}_\mathcal{F}^2}{2 \text{ Sup } g\omega} \tag{3.75}$$

et que le nombre ω de features tirés dans $\mathcal{F}$ vérifie

$$\omega \leqslant \frac{|\mathcal{F}|}{2}$$

alors on peut trouver un pas γ_n, variant en $1/n$ suffisamment petit, tel que l'intervalle $\left[\frac{\mathcal{U}_\mathcal{F}}{2}\,;\frac{2\omega}{|\mathcal{F}|}\right]$ est stable par ($\mathbf{E}-\mathbf{5}$). Plus précisément :

$$\exists \gamma_0 \in \mathbb{R}^+ \quad 0 \leqslant \gamma \leqslant \gamma_0 \Longrightarrow \forall n \in \mathbb{N} \quad \forall \delta \in \mathcal{F} \qquad \mathbb{P}_n(\delta) \in \left[\frac{\mathcal{U}_\mathcal{F}}{2}\,;\frac{2\omega}{|\mathcal{F}|}\right]$$

Preuve : Si n désigne un entier quelconque, on a d'après ($\mathbf{E}-\mathbf{5}$) :

$$\begin{aligned}
\forall \delta \in \mathcal{F} \qquad \mathbb{P}_{n+1}(\delta) &\geqslant \mathbb{P}_n(\delta) + 2\gamma_n\beta(\mathcal{U}_\mathcal{F} - \mathbb{P}_n(\delta)) - \gamma_n \alpha g(\omega_n)\mathrm{C}(\omega_n,\delta)/\mathbb{P}_n(\delta) \\
&= \mathbb{P}_n(\delta) + \gamma_n\left[2\beta(\mathcal{U}_\mathcal{F} - \mathbb{P}_n(\delta)) - \alpha g(\omega_n)\mathrm{C}(\omega_n,\delta)/\mathbb{P}_n(\delta)\right] \\
\forall \delta \in \mathcal{F} \qquad \mathbb{P}_{n+1}(\delta) &\geqslant \underbrace{\mathbb{P}_n(\delta) + \gamma_n\left[2\beta(\mathcal{U}_\mathcal{F} - \mathbb{P}_n(\delta)) - \alpha \text{ Sup } g\,\omega/\mathbb{P}_n(\delta)\right]}_{=f_n(\mathbb{P}_n(\delta))}
\end{aligned}$$

Pour assurer l'existence d'un intervalle stable, on cherche un premier intervalle $\mathrm{I} = [\,m\,;\mathcal{U}_\mathcal{F}\,]$ tel que les fonctions

$$f_n(x) = x + \frac{k_1}{n}(\mathcal{U}_\mathcal{F} - x) - \frac{k_2}{x}$$

laissent stable cet intervalle I pour tout n, avec

$$\begin{cases} k_1 = 2\beta\gamma \\ k_2 = \gamma\alpha \text{ Sup } g\omega \end{cases}$$

en supposant (ce qui sera le cas ultérieurement) que γ_n varie en γ/n.

Les fonctions f_n se réécrivent en

$$f_n(x) - m = x(1 - k_1/n) - \frac{k_2/n}{x} + (k_1/n\,\mathcal{U}_\mathcal{F} - m)$$

Ces fonctions sont croissantes sur I et il faut donc avoir d'une part

$$\begin{aligned} f_n(m) \geqslant m &\iff f_n(m) - m \geqslant 0 \\ &\iff m(1 - k_1/n) - \frac{k_2/n}{m} + (k_1/n\,\mathcal{U}_\mathcal{F} - m) \geqslant 0 \\ &\iff k_1(\mathcal{U}_\mathcal{F} - m) \geqslant \frac{k_2}{m} \\ f_n(m) \geqslant m &\iff m(\mathcal{U}_\mathcal{F} - m) \geqslant \frac{k_2}{k_1} \end{aligned}$$

D'autre part, il faut bien entendu s'assurer que $f_n(\mathcal{U}_\mathcal{F})$ n'est pas trop grand, ce qui est vrai puisque

$$\forall n \in \mathbb{N} \qquad f_n(\mathcal{U}_\mathcal{F}) = \mathcal{U}_\mathcal{F} - \frac{k_2}{\mathcal{U}_\mathcal{F}} \leqslant \mathcal{U}_\mathcal{F}$$

On souhaite enfin maximiser l'effet de la fonction d'erreur $\mathcal{E}_1$ dans l'énergie, cela signifie donc avoir le α le plus grand possible, soit encore la constante k_2 la plus grande possible. Cela impose donc que le rapport k_2/k_1 soit maximal, et la borne maximale précédente correspond alors aux choix $m = \mathcal{U}_\mathcal{F}/2$ puisque l'application $x \mapsto x(u-x)$ est maximale pour $x = u/2$.

Ainsi
$$\frac{\alpha}{\beta} = \frac{2\beta}{\text{Sup } g\omega}\frac{\mathcal{U}_\mathcal{F}^2}{4} \qquad (\mathbf{C})$$
correspond au choix donnant le plus de poids au terme $\mathcal{E}_1$ dans $\mathcal{E}$ et laissant comme intervalle stable $\mathrm{I} = \left[\frac{\mathcal{U}_\mathcal{F}}{2}\,;\mathcal{U}_\mathcal{F}\right]$.

Il faut ensuite trouver un second intervalle, cette fois de la forme $\mathrm{J} = [\mathcal{U}_\mathcal{F}\,;\mathrm{M}]$, qui n'est pas exactement laissé stable par $(\mathbf{E}-\mathbf{5})$, mais tel que son image soit contenue dans $\mathrm{I} \cup \mathrm{J}$.

De la même façon que précédemment, on peut écrire que

$$\mathbb{P}_{n+1} \geqslant \mathbb{P}_n - \gamma_n\alpha \text{ Sup } g\frac{\omega}{\mathbb{P}_n} + 2\gamma_n\beta(\mathcal{U}_\mathcal{F} - \mathbb{P}_n)$$

Pour que l'image de J soit contenue dans $\mathrm{I} \cup \mathrm{J}$, il suffit donc d'avoir tout d'abord

$$\mathbb{P}_n - \gamma_n\alpha \text{ Sup } g\frac{\omega}{\mathbb{P}_n} + 2\gamma_n\beta(\mathcal{U}_\mathcal{F} - \mathbb{P}_n) \geqslant m = \frac{\mathcal{U}_\mathcal{F}}{2}$$

Cette condition est trivialement vérifiée pour les mêmes raisons que celles évoquées précédemment, puisque c'est une fonction croissante de $\mathbb{P}$ et que l'image de $\mathcal{U}_\mathcal{F}$ est supérieure à l'image de $\mathcal{U}_\mathcal{F}/2$ qui est précisément supérieure à $\mathcal{U}_\mathcal{F}/2$.

Enfin, il faut avoir la condition

$$\mathbb{P}_{n+1} \leqslant \mathrm{M}$$

On rappelle que $\mathbb{P}_{n+1}$ est donnée par

$$\mathbb{P}_{n+1}(.) = \mathbb{P}_n(.) + \gamma_n\alpha g(\omega_n)\underbrace{\left(-\frac{\mathrm{C}(\omega_n,.)}{\mathbb{P}_n(.)} + \sum_{\delta\in\omega_n}\frac{\mathrm{C}(\omega_n,\delta)}{|\mathcal{F}|\mathbb{P}_n(\delta)}\right)}_{=\mathrm{A}} + 2\gamma_n\beta\left(\frac{1}{|\mathcal{F}|} - \mathbb{P}_n(.)\right)$$

et A se majore par
$$\mathrm{A} \leqslant \frac{2\omega}{|\mathcal{F}|\mathcal{U}_\mathcal{F}} - \frac{1}{\mathbb{P}_n}$$

puisque tous les $\mathbb{P}_n$ sont supérieurs à $\mathcal{U}_\mathcal{F}/2$.

Finalement, pour trouver M tel que $\mathbb{P}_n$ est inférieure à M pour tout n, il suffit que

$$\mathbb{P}_n + \gamma_n \alpha g \left(\frac{2\omega}{|\mathcal{F}|\mathcal{U}_\mathcal{F}} - \frac{1}{\mathbb{P}_n} \right) + 2\gamma_n \beta (\mathcal{U}_\mathcal{F} - \mathbb{P}_n) \leqslant \mathrm{M}$$

Or la fonction $x \mapsto -\gamma_n \alpha g 1/x - 2\beta\gamma_n x$ est décroissante sur J, donc il suffit d'avoir

$$\alpha g \left(\frac{2\omega}{|\mathcal{F}|\mathcal{U}_\mathcal{F}} - \frac{1}{\mathcal{U}_\mathcal{F}} \right) \leqslant 0$$

C'est-à-dire

$$\omega \leqslant \frac{\mathcal{F}}{2}$$

C'est bien le cas puisque le nombre de caractéristiques tirées est bien inférieur au nombre d'éléments de $\mathcal{F}$.

Ainsi, l'équation $(\mathbf{E-5})$ assure que l'on peut trouver $\mathrm{M} = 2\omega/|\mathcal{F}| \geqslant \mathcal{U}_\mathcal{F}$ qui a son image inclue dans $\mathrm{I} \cup \mathrm{J}$. Ce qui achève la démonstration. □

Théorème 3.4.2 (Convexité de $\mathcal{E}$)
Si les coefficients α et β vérifient l'inégalité

$$\beta \geqslant \frac{\alpha \ \mathrm{Sup}\ g\omega^2}{2} \tag{3.76}$$

alors la fonctionnelle $\mathcal{E}$ est convexe et admet donc un unique minimum.

Preuve :

On peut évaluer la matrice hessienne H de $\mathcal{E}$:

$$\forall (\delta, \mu) \in \mathcal{F}^2 \qquad \mathrm{H}_{\delta,\mu} = \frac{\partial^2 \mathcal{E}(\mathbb{P})}{\partial \mathbb{P}(\delta) \partial \mathbb{P}(\mu)}$$

On trouve que

$$\forall (\delta, \mu) \in \mathcal{F}^2 \qquad \mathrm{H}_{\delta,\mu} = \alpha \sum_{\omega \in \mathcal{F}^p} g(\omega) \frac{\partial^2 \mathbb{P}(\omega)}{\partial \mathbb{P}(\delta) \partial \mathbb{P}(\mu)} + 2\beta \Delta_{\delta=\mu}$$

En réutilisant les fonctions C définies précédemment :

$$\forall (\delta, \mu) \in \mathcal{F}^2 \qquad \mathrm{H}_{\delta,\mu} = \alpha \sum_{\omega \in \mathcal{F}^p} g(\omega) \mathrm{C}(\omega, \delta) \mathrm{C}(\omega \setminus \delta, \mu) \frac{\mathbb{P}(\omega)}{\mathbb{P}(\delta)\mathbb{P}(\mu)} + 2\beta \Delta_{\delta=\mu}$$

ou encore

$$\mathrm{H} = 2\beta \mathrm{I}d_\mathcal{F} + \alpha \mathrm{G}$$

Pour que $\mathcal{E}$ soit convexe, il faut que H soit définie positive. Il suffit alors que H vérifie :

$$\forall \delta \in \mathcal{F} \qquad 2\beta > \alpha \ \mathrm{Max}_{\mu \in \mathcal{F}} \ |\mathrm{G}_{\delta,\mu}|$$

pour assurer que la matrice H soit définie positive. En effet, si G s'écrit

$$\mathrm{G} = \mathrm{G}_d + \mathrm{G}_a$$

où G_d est la matrice restreinte à la diagonale issue de G, on a :

$$\forall \lambda \in \mathbb{R} \qquad 2\beta \mathrm{I}d + \alpha \mathrm{G} - \lambda \mathrm{I}d = ((2\beta - \lambda)\mathrm{I}d + \alpha \mathrm{G}_d) \left[\mathrm{I}d + \alpha (2\beta - \lambda) \mathrm{I}d + \alpha \mathrm{G}_d)^{-1} \mathrm{G}_a \right]$$

On cherche donc une condition sur les coefficients α et β pour assurer que les valeurs propres de H soit positives. Si $\mathrm{H} - \lambda \mathrm{I}d$ est non inversible, on doit donc avoir :

$$| \ \alpha ((2\beta - \lambda)\mathrm{I}d + \alpha \mathrm{G}_d)^{-1} \mathrm{G}_a \ | \ > 1 \tag{3.77}$$

En prenant pour norme matricielle

$$|\,\mathrm{M}\,| \;=\; \underset{i\in[\![\,1\,;\,|\mathcal{F}|\,]\!]}{\mathrm{Max}} \; \underset{j\in[\![\,1\,;\,|\mathcal{F}|\,]\!]}{\mathrm{Max}} |\mathrm{M}_{i,j}|$$

on voit donc que (3.77) est satisfaite avec des valeurs λ strictement positives dès que

$$\exists i \in [\![\,1\,;\,|\mathcal{F}|\,]\!] \quad \exists j \in [\![\,1\,;\,|\mathcal{F}|\,]\!] \qquad 2\beta + \alpha \mathrm{G}_{d_i} > \beta |\mathrm{G}_{d_j}|$$

Il suffit donc de prendre les coefficients α et β vérifiant l'inégalité :

$$\forall (\delta,\mu) \in \mathcal{F}^2 \qquad 2\beta > \alpha \sum_{\omega\in\mathcal{F}^p} g(\omega)\mathrm{C}(\omega,\delta)\mathrm{C}(\omega,\mu)\mathrm{P}(\omega|\delta,\mu\in\omega)$$

Soit encore $\qquad \forall (\delta,\mu) \in \mathcal{F}^2 \qquad 2\beta > \alpha \mathrm{E}_{\mathrm{P}(\,|\delta,\mu)}\left[\mathrm{C}(\omega,\delta)\mathrm{C}(\omega,\mu)g(\omega)\right]$

Or par hypothèse, l'inégalité précédente est bien vérifiée. □

Annexe C - Calculs des règles de sauts

C-1 Proposition des sauts

Au chapitre 5, nous avons suggéré une méthode de proposition des sauts ($\mathbf{R_{1-2}}$) qui n'est pas « indépendante » de la probabilité $\mathbb{P}$ à l'instant de saut t_{s_i}. Ceci est fait dans le but de ne pas faire un saut aveugle à l'apprentissage par diffusion effectué entre les instants de saut $t_{s_{i-1}}$ et t_{s_i}.

- En effet, il paraît plus naturel d'augmenter la probabilité de proposer la coupe d'un arbre $\mathcal{A}$ (avec ($\mathbf{T_c}$)) lorsque celui-ci possède une probabilité faible $\mathbb{P}_{t_{s_i}^-}(\mathcal{A})$.
- Inversement, il parait également plus judicieux de choisir une greffe ($\mathbf{T_{g,.}}$) d'un arbre $\mathcal{A}_1$ et $\mathcal{A}_2$ avec une plus forte probabilité lorsque ceux-ci possèdent cette fois des probabilités $\mathbb{P}_{t_{s_i}^-}(\mathcal{A}_1)$ et $\mathbb{P}_{t_{s_i}^-}(\mathcal{A})_2$ plutôt élevées.

A chaque instant de sauts, nous prenons donc le parti, une fois l'étape ($\mathbf{R_1}$) franchie et la règle de transition ($\mathbf{T}$) sélectionnée selon $p_g, p_{g;sg}, p_{g;sd}, p_{g;sgd}, p_c$ ou p_i (section 5.x), de sélectionner les arbres sur lesquels vont s'appliquer ces règles *via* une probabilité différente de la loi uniforme sur tous ces arbres et dépendant en réalité de $\mathbb{P}_{t_{s_i}^-}$.

La loi de sélection de ces arbres est alors définie par un seuillage de $\mathbb{P}_{t_{s_i}^-}$.

C-1-1 Sélection pour une greffe

Dans ces cas précis, on souhaite privilégier les arbres qui ont une forte probabilité $\mathbb{P}_{t_{s_i}^-}$ puisque ce sont ceux qui sont le plus utiles à la minimisation de g. On espère alors qu'un choix utilisant favorablement les arbres à probabilité élevée pour une agrégation de features créera un meilleur feature qu'un arbre issu de deux sous-arbres qui possèdent des performances moyennes (pour la fonction g).

On forme donc la nouvelle distribution de probabilité Q_g en définissant plusieurs ensembles :

1. On détermine l'ensemble des arbres $\mathcal{A}$ de $\mathcal{F}_{t_s}^-$ (ensemble $\mathcal{I}$) tels que :

$$\mathcal{A} \in \mathcal{I} \Leftrightarrow \left| \left[\mathcal{B} \in \mathcal{F}_{t_s}^- \quad | \quad \mathbb{P}_{t_s}(\mathcal{B}) \geqslant \mathbb{P}_{t_s}(\mathcal{A}) \right] \right| \leqslant \frac{1}{10} |\mathcal{F}_{t_s}^-|$$

ainsi $$|\mathcal{I}| \simeq \frac{\mathcal{F}_{t_s}}{10}$$

Dans l'égalité précédente, l'ensemble $\mathcal{I}$ n'ayant pas un cardinal divisible par 10, cela explique le signe $\simeq$ plutôt que $=$ dans l'égalité précédente.

2. $$\forall \mathcal{A} \in \mathcal{I} \qquad \mathrm{Q}_g(\mathcal{A}) = \frac{9}{10}\frac{1}{|\mathcal{I}|}$$

3. $$\forall \mathcal{A} \notin \mathcal{I} \qquad \mathrm{Q}_g(\mathcal{A}) = \frac{1}{10}\frac{1}{|\mathcal{F}^-_{t_s} \setminus_{\mathcal{I}}|}$$

On choisit donc de mettre 90% du poids de la probabilité Q sur les 10% des arbres les plus chargés par $\mathbb{P}^-_{t_s}$.

C-1-2 Sélection pour une coupe

Dans ce cas, au contraire, on souhaite couper en priorité les arbres n'ayant que peu d'intêret pour le problème de détection. Dans ce cas, nous choisissons plutôt d'éliminer les features faiblement chargés par $\mathbb{P}^-_{t_s}$. D'où la construction de la probabilité Q_c de proposer un feature pour la coupe :

1. On détermine l'ensemble des arbres $\mathcal{A}$ de $\mathcal{F}^-_{t_s}$ (ensemble $\mathcal{I}$) tels que :

$$\mathcal{A} \in \mathcal{I} \Leftrightarrow \left| \left[\mathcal{B} \in \mathcal{F}^-_{t_s} \quad | \quad \mathbb{P}_{t_s}(\mathcal{B}) \geqslant \mathbb{P}_{t_s}(\mathcal{A}) \right] \right| \geqslant \frac{9}{10}|\mathcal{F}^-_{t_s}|$$

2. $$\forall \mathcal{A} \in \mathcal{I} \qquad \mathrm{Q}_c(\mathcal{A}) = \frac{9}{10}\frac{1}{|\mathcal{I}|}$$

3. $$\forall \mathcal{A} \notin \mathcal{I} \qquad \mathrm{Q}_c(\mathcal{A}) = \frac{1}{10}\frac{1}{|\mathcal{F}^-_{t_s} \setminus_{\mathcal{I}}|}$$

C-2 Calcul des probabilités de transition (R2)

Les calculs suivants sont légèrement conditionnés à la nature exacte de la forêt au temps du saut t_s. Plus précisément, on distingue deux cas :

C-2-1 Cas où $\mathcal{F}_0 \setminus \mathcal{F}_{t_s} \neq \varnothing$

C-2-1-1 Calcul de la probabilité de faire un saut basé sur une greffe

Le nombre de couples possibles d'arbre distincts de $\mathcal{F}_{t_s}$ est noté N_{t_s} exactement $|\mathcal{F}_{t_s}|\,(|\mathcal{F}_{t_s}| - 1)$:

d'où $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_g}(\mathcal{F}_{t_s}; \mathcal{A}_1; \mathcal{A}_2)\right) = p_g \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

On a également $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_{g;sg}}(\mathcal{F}_{t_s}; \mathcal{A}_1; \mathcal{A}_2)\right) = p_{g;sg} \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

puis $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_{g;sd}}(\mathcal{F}_{t_s}; \mathcal{A}_1; \mathcal{A}_2)\right) = p_{g;sd} \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

Enfin $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_{g;sgd}}(\mathcal{F}_{t_s}; \mathcal{A}_1; \mathcal{A}_2)\right) = p_{g;sgd} \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

C-2-1-2 Calcul de la probabilité de faire un saut basé sur $(\mathbf{T_c})$

Le calcul est quasi-identique, seule la probabilité de sélectionner l'arbre dans $\mathcal{F}_{t_s}$ est différente.

On a donc $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_c}(\mathcal{F}_{t_s}; \mathcal{A})\right) = p_c\, \mathrm{Q}_c(\mathcal{A})$$

C-2-1-3 Calcul de la probabilité de faire un saut basé sur une renaissance d'un arbre élémentaire

Ce calcul est conditionné au fait qu'il y ait ou non des arbres élémentaires absents de la forêt $\mathcal{F}_{t_s}$ à l'instant du saut. Mais dans notre cas particulier, on a précisément supposé que certains arbres élémentaires ne sont pas dans $\mathcal{F}_{t_s}$:

soit $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_i}(\mathcal{F}_{t_s};\mathcal{A})\right) = \frac{p_i}{|\mathcal{F}_0 \setminus \mathcal{F}_{t_s}|}$$

C-2-2 Cas où $\mathcal{F}_0 \setminus \mathcal{F}_{t_s} = \varnothing$:

Les calculs sont similaires aux calculs précédents, sauf qu'il faut renormaliser certaines probabilités puisqu'ici :

$$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_i}(\mathcal{F}_{t_s};\mathcal{A})\right) = 0$$

Ainsi $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_g}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2)\right) = \frac{p_g}{(1-p_i)}\mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

et $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_{g;sg}}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2)\right) = \frac{p_{g;sg}}{(1-p_i)}\mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

puis $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_{g;sd}}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2)\right) = \frac{p_{g;sd}}{(1-p_i)}\mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

Enfin $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_{g;sgd}}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2)\right) = \frac{p_{g;sgd}}{(1-p_i)}\mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)$$

et $$\mathrm{P}\left(\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_c}(\mathcal{F}_{t_s};\mathcal{A})\right) = \frac{p_c}{(1-p_i)}\mathrm{Q}_c(\mathcal{A})$$

C-3 Calcul de τ_1

Comme nous l'avons évoqué au chapitre 5, pour calculer τ_1, il s'agit en réalité d'énumérer toutes les transitions possibles lors du saut en étudiant précisément quelles sont les règles qui permettent d'« inverser » ces transitions.

C-3-1 Greffes $(\mathbf{T_g})$, $(\mathbf{T_{g;sg}})$, $(\mathbf{T_{g;sd}})$ et $(\mathbf{T_{g;sgd}})$

Si $\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_g}(\mathcal{F}_{t_s};\mathcal{A}_1;\mathcal{A}_2)$, alors comme le nouvel arbre n'appartient pas à $\mathcal{F}_{t_s}$, le seul moyen de faire le chemin inverse est d'utiliser $(\mathbf{T_c})$. Le rapport vaut alors :

$$\tau_1 = \frac{p_c\mathrm{Q}_c(\mathcal{A}_1 :: \mathcal{A}_2)}{p_g\mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)}$$

Il est à noter dans cette formule que le nombre d'arbres de $\mathcal{F}_{t_s+\mathrm{d}t}$ a augmenté d'une unité puisqu'on y a ajouté $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$. Cela modifie donc légèrement le calcul de $\mathrm{Q}_c(\mathcal{A}_1 :: \mathcal{A}_2)$. Les calculs sont alors similaires pour $(\mathbf{T_{g;sg}})$, $(\mathbf{T_{g;sd}})$ et $(\mathbf{T_{g;sgd}})$.

C-3-2 Coupe ($\mathbf{T_c}$)

Si $\mathcal{F}_{t_s+\mathrm{d}t} = \widetilde{\mathrm{T}_c}(\mathcal{F}_{t_s}; \mathcal{A})$, il est nécessaire de distinguer deux cas.

- L'arbre coupé était un arbre élémentaire, le retour se fait donc en réinjectant dans la forêt l'arbre initial *via* ($\mathbf{T_i}$) :

  donc $$\tau_1 = \frac{p_i}{(|\mathcal{F}_0 \setminus \mathcal{F}_{t_s}| + 1) p_c \mathrm{Q}_c(\mathcal{A})}$$

- L'arbre coupé n'était pas un arbre élémentaire, il s'agit alors de faire une autre distinction :
  - les fils gauche et droit appartenaient déjà à $\mathcal{F}_{t_s}$, alors la règle ($\mathbf{T_g}$) permet de faire le « retour » et
    $$\tau_1 = \frac{p_g \mathrm{Q}_g(\mathcal{A}.g)\mathrm{Q}_g(\mathcal{A}.d)}{\mathrm{Q}_c(\mathcal{A})\, p_c}$$
  - le fils gauche ou le fils droit n'appartenait pas à $\mathcal{F}_{t_s}$, il faut alors faire une greffe avec suppression pour permettre le « retour » donc
    $$\tau_1 = \frac{p_{g;sg} \mathrm{Q}_g(\mathcal{A}.g)\mathrm{Q}_g(\mathcal{A}.d)}{\mathrm{Q}_c(\mathcal{A})\, p_c}$$
    ou $$\tau_1 = \frac{p_{g;sd} \mathrm{Q}_g(\mathcal{A}.g)\mathrm{Q}_g(\mathcal{A}.d)}{\mathrm{Q}_c(\mathcal{A})\, p_c}$$
  - les fils gauche et droit n'appartenaient pas à $\mathcal{F}_{t_s}$, il faut alors faire une greffe avec deux suppressions ($\mathbf{T_{g;sgd}}$) pour effectuer le « retour » :
    $$\tau_1 = \frac{p_{g;sgd} \mathrm{Q}_g(\mathcal{A}.g)\mathrm{Q}_g(\mathcal{A}.d)}{\mathrm{Q}_c(\mathcal{A})\, p_c}$$

  Dans les quatres formules précédentes, il faut prendre garde que le calcul de Q_g se fait de manière différente, selon que les arbres gauches ou droits sont présents ou non dans la forêt avant le saut.

C-3-3 Renaissance ($\mathbf{T_i}$)

Si $\mathcal{F}_{t_s+\mathrm{d}t}$ est issue d'une renaissance d'un arbre élémentaire, il faut donc effectuer une coupe d'un arbre pour effectuer le retour, donc

$$\tau_1 = \frac{p_c \mathrm{Q}_c(\mathcal{A}) |\mathcal{F}_0 \setminus \mathcal{F}_{t_s}|}{p_i}$$

C-4 Calcul des différentiels énergétiques $\Delta\mathcal{E}_{err}$

Propriété 5.6.2 (Calcul de $\Delta\mathcal{E}_{err}$ pour ($\mathbf{T_{g;sg}}$) ou ($\mathbf{T_{g;sd}}$))
Si l'on note t la quantité

$$t = \frac{\mathbb{E}[g(\omega) \,|\, \mathcal{A} \in \omega]}{\mathbb{E}[g(\omega)]}$$

alors le signe de $\Delta\mathcal{E}_{err}$ dépend de ce rapport t. Plus précisément,

- *Si $t > 1$, $\Delta\mathcal{E}_{err}$ est négatif.*
- *Si $t < 1$, il existe p_0 tel que*

$$\begin{cases} \mathbb{P}_{t_s}(\mathcal{A}) < p_0 \Longrightarrow \Delta\mathcal{E}_{err} > 0 \\ \mathbb{P}_{t_s}(\mathcal{A}) > p_0 \Longrightarrow \Delta\mathcal{E}_{err} < 0 \end{cases}$$

Cette proposition nous donne donc l'influence de la suppression d'un arbre $\mathcal{A}$ de $\mathcal{F}_{t_s}$.
Preuve : Puisque $\mathcal{B}$ est le mot formé à l'issue de la greffe, on a :

$$\mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{B}) = 0$$

Donc, en imposant formellement que $\mathbb{P}_{t_s+\mathrm{d}t}(\mathcal{A}) = 0$

> En réalité, $\mathcal{A}$ n'appartient plus à $\mathcal{F}_{t_s+\mathrm{d}t}$ et donc $\mathbb{P}_{t_s+\mathrm{d}t}$ n'y est pas définie. Cependant, imposer une telle valeur ne modifie pas la valeur de l'énergie $\mathcal{E}_{err}$ et cela nous facilitera un peu l'écriture.

$$\begin{aligned}
\Delta\mathcal{E}_{err}\left[\begin{array}{c}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{array} \longrightarrow \begin{array}{c}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{array}\right] &= \sum_{\omega \in \left(\underbrace{\mathcal{F}_{t_s+\mathrm{d}t} \setminus \mathcal{B}}_{=\mathcal{F}_{t_s}\setminus\mathcal{A}}\right)^p} g(\omega)\mathbb{P}_{t_s+\mathrm{d}t}(\omega) + \underbrace{\sum_{\omega\in(\mathcal{F}_{t_s+\mathrm{d}t})^p \,|\, \mathcal{B}\in\omega} g(\omega)\mathbb{P}_{t_s+\mathrm{d}t}(\omega)}_{=0 \text{ car } \mathbb{P}_{t_s+\mathrm{d}t}(\omega)=0} \\
&\quad - \sum_{\omega\in(\mathcal{F}_{t_s})^p} g(\omega)\mathbb{P}_{t_s}(\omega) \\
&= \sum_{\omega\in(\mathcal{F}_{t_s})^p} g(\omega)\left[\mathbb{P}_{t_s+\mathrm{d}t}(\omega) - \mathbb{P}_{t_s}(\omega)\right] \\
&= \sum_{\omega\in(\mathcal{F}_{t_s})^p} g(\omega)\left[\frac{\mathbb{P}_{t_s}(\omega_1)\dots\mathbb{P}_{t_s}(\omega_p)}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p} - \mathbb{P}_{t_s}(\omega_1)\dots\mathbb{P}_{t_s}(\omega_p)\right] \\
&\quad - \sum_{\omega\in(\mathcal{F}_{t_s})^p|\mathcal{A}\in\omega} g(\omega)\frac{\mathbb{P}_{t_s}(\omega_1)\dots\mathbb{P}_{t_s}(\omega_p)}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}
\end{aligned}$$

$$\Delta\mathcal{E}_{err}\left[\begin{array}{c}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{array} \longrightarrow \begin{array}{c}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{array}\right] = \mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\left[\frac{1}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p} - 1\right] - \frac{p\,\mathbb{P}_{t_s}(\mathcal{A})\mathcal{E}_{err,\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}$$

> Dans le calcul précédent, nous définissons la quantité $\mathcal{E}_{err,\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$ par
>
> **Définition 3 (Erreur relative à $\mathcal{A}$: $\mathcal{E}_{err,\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$)**
> *La quantité $\mathcal{E}_{err,\mathcal{A}}(\mathcal{F};\mathbb{P})$ est définie par*
>
> $$\mathcal{E}_{err,\mathcal{A}}(\mathcal{F};\mathbb{P}) = \sum_{\omega\in\mathcal{F}^{p-1}} g(\omega;\mathcal{A})\mathbb{P}(\omega)$$
>
> *Cette quantité désigne donc l'erreur moyenne g commise par le classifieur sachant que les features tirés contiennent $\mathcal{A}$ puisque :*
>
> $$\mathcal{E}_{err,\mathcal{A}}(\mathcal{F};\mathbb{P}) = \mathbb{E}\left[g(\omega)\,|\,\mathcal{A}\in\omega\right]$$

En revenant au calcul précédent, on observe donc que

$$\Delta\mathcal{E}_{err}\left[\begin{array}{c}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{array}\longrightarrow\begin{array}{c}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{array}\right]=\frac{\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\left[1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p\right]-p\mathbb{P}_{t_s}(\mathcal{A})\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}$$

$$=\frac{\left[\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})-\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\right]\left[1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p\right]}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}$$

$$+\frac{\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\left[1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p-p\mathbb{P}_{t_s}(\mathcal{A})\right]}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}$$

Ce qui s'écrit encore

$$\Delta\mathcal{E}_{err}\left[\begin{array}{c}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{array}\longrightarrow\begin{array}{c}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{array}\right]=\left[\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})-\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\right]\mathrm{A}+\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\mathrm{B}$$

où les constantes A et B sont définies par

$$\begin{cases}\mathrm{A}=\dfrac{1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}\\ \mathrm{B}=\dfrac{1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p-p\,\mathbb{P}_{t_s}(\mathcal{A})}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}\end{cases}$$

On constate immédiatement que la constante A est strictement positive et d'autant plus proche de 0 que $\mathbb{P}_{t_s}(\mathcal{A})$ l'est.

$$\mathrm{A}=\frac{p\,\mathbb{P}_{t_s}(\mathcal{A})}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}+\mathrm{o}\left(\mathbb{P}_{t_s}(\mathcal{A})\right)$$

Un développement limité en $\mathbb{P}_{t_s}(\mathcal{A})$ (qui est petit) permet d'écrire que

$$\mathrm{B}=\frac{p(p-1)\mathbb{P}_{t_s}(\mathcal{A})^2}{2(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}+\mathrm{o}\left(\mathbb{P}_{t_s}(\mathcal{A})^2\right)$$

> Le calcul précédent montre donc que si $\mathbb{P}_{t_s}(\mathcal{A})$ est suffisamment petit, la quantité $\Delta\mathcal{E}_{err}$ est du signe de $\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})-\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$. Cette remarque permet d'interpréter facilement l'influence de cette différence sur le seuil τ_2 : il est d'autant plus élevé que la diminution d'énergie est grande et cette diminution d'énergie est d'autant plus grande que $\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})-\mathcal{E}_{err;\mathcal{A}}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})$ est petit. Le saut va donc être accepté avec une probabilité d'autant plus grande que l'arbre coupé $\mathcal{A}$ possède une erreur relative importante. Et cette approximation est en plus d'autant meilleure que la probabilité de tirer cet arbre est faible.

Pour obtenir un résultat plus précis, on peut poser

$$t=\frac{\mathcal{E}_{err;\mathcal{A}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}}{\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}$$

Avec cette notation, on obtient que $\Delta\mathcal{E}$ est du signe de

$$\left[1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p\right]-p\mathbb{P}_{t_s}(\mathcal{A})t$$

C'est une fonction de $\mathbb{P}_{t_s}(\mathcal{A})$ qui s'écrit :

$$h_t : x \in [\,0\,;1\,] \longmapsto [1-(1-x)^p] - pxt$$

La dérivée de cette fonction vaut

$$h'_t(x) = p\left[(1-x)^{p-1} - t\right]$$

- Si la quantité t est supérieure strictement à 1, la fonction est strictement décroissante puisque

  $$\forall x \in [\,0\,;1\,] \qquad h'_t(x) \leqslant 0$$

  et on a donc dans ce cas :

  $$\forall x \in [\,0\,;1\,] \qquad h_t(x) \leqslant h_t(0) = 0$$

  Autrement dit, dans le cas où le rapport

  $$\frac{\mathcal{E}_{err;\mathcal{A}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}}{\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}$$

  est strictement supérieur à 1, le différentiel énergétique d'erreur $\Delta\mathcal{E}_{err}$ est négatif, ce qui signifie encore que dès que l'erreur relative au mot $\mathcal{A}$ est supérieure à l'erreur moyenne, le différentiel énergétique sera positif, et dans ce cas, le rapport τ_2 sera supérieur à 1, on aura donc tendance à accepter le saut.
- Si t est inférieur à 1, la fonction h_t n'est plus monotone, si x_0 désigne le point de $[\,0\,;1\,]$ tel que

  $$(1-x_0)^{p-1} = t$$

  on a alors h_t croissante sur $[\,0\,;x_0\,]$ et décroissante sur $[\,x_0\,;1\,]$.
  Comme de plus $h_t(1) = 1 - pt$ et que pt est tout de même grand devant 1, on en déduit que h_t est positive sur un intervalle $[\,0\,;x_1\,]$ et négative ailleurs avec $x_1 \geqslant x_0$.

x	0	x_0	x_1	1
h_t	0	$h_t(x_0)$ ↗	↘ 0	↘ $h_t(1) < 0$

La précédente disjonction des cas permet alors de conclure. □

Propriété 5.6.3 (Calcul de $\Delta\mathcal{E}_{err}$ pour $(\mathbf{T_{g;sgd}})$:)
$\Delta\mathcal{E}_{err}$ est du signe de

$$\mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\,[1-(1-s)^p] - ps\{\mathcal{E}_{err;\mathcal{A}_1} - \mathcal{E}_{err;\mathcal{A}_2}\} + p(p-1)q\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}$$

où

$$\begin{cases} s = \mathbb{P}_{t_s}(\mathcal{A}_1) + \mathbb{P}_{t_s}(\mathcal{A}_2) \\ q = \mathbb{P}_{t_s}(\mathcal{A}_1)\mathbb{P}_{t_s}(\mathcal{A}_2) \end{cases}$$

et

$$\{\mathcal{E}_{err;\mathcal{A}_1} - \mathcal{E}_{err;\mathcal{A}_2}\} = \frac{\mathbb{P}_{t_s}(\mathcal{A}_1)\mathcal{E}_{err;\mathcal{A}_1} + \mathbb{P}_{t_s}(\mathcal{A}_2)\mathcal{E}_{err;\mathcal{A}_2}}{s}$$

Preuve :

$$\Delta\mathcal{E}_{err}\left[\begin{array}{c}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{array}\longrightarrow\begin{array}{c}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{array}\right] = \sum_{\omega\in\left(\underbrace{\mathcal{F}_{t_s+\mathrm{d}t}\setminus\mathcal{B}}_{=\mathcal{F}_{t_s}\setminus\mathcal{A}_1\cup\mathcal{A}_2}\right)}{}^{p} g(\omega)\mathbb{P}_{t_s+\mathrm{d}t}(\omega) + \underbrace{\sum_{\omega\in(\mathcal{F}_{t_s+\mathrm{d}t})^p\,|\,\mathcal{B}\in\omega} g(\omega)\mathbb{P}_{t_s+\mathrm{d}t}(\omega)}_{=0\ \mathrm{car}\ \mathbb{P}_{t_s+\mathrm{d}t}(\omega)=0}$$

$$- \sum_{\omega\in(\mathcal{F}_{t_s})^p} g(\omega)\mathbb{P}_{t_s}(\omega)$$

$$= \sum_{\omega\in(\mathcal{F}_{t_s})^p} g(\omega)\left[\mathbb{P}_{t_s+\mathrm{d}t}(\omega) - \mathbb{P}_{t_s}(\omega)\right]$$

$$= \sum_{\omega\in(\mathcal{F}_{t_s})^p} g(\omega)\left[\frac{\mathbb{P}_{t_s}(\omega_1)\ldots\mathbb{P}_{t_s}(\omega_p)}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1-\mathbb{P}_{t_s}(\mathcal{A}_2))^p} - \mathbb{P}_{t_s}(\omega_1)\ldots\mathbb{P}_{t_s}(\omega_p)\right]$$

$$- \sum_{\omega\in(\mathcal{F}_{t_s})^p|\mathcal{A}_1\in\omega} g(\omega)\frac{\mathbb{P}_{t_s}(\omega_1)\ldots\mathbb{P}_{t_s}(\omega_p)}{(1-\mathbb{P}_{t_s}(\mathcal{A})_1-\mathbb{P}_{t_s}(\mathcal{A}_2)^p}$$

$$- \sum_{\omega\in(\mathcal{F}_{t_s})^p|\mathcal{A}_2\in\omega} g(\omega)\frac{\mathbb{P}_{t_s}(\omega_1)\ldots\mathbb{P}_{t_s}(\omega_p)}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1)-\mathbb{P}_{t_s}(\mathcal{A}_2)^p}$$

$$+ \sum_{\omega\in(\mathcal{F}_{t_s})^p|\mathcal{A}_1\in\omega\,,\mathcal{A}_2\in\omega} g(\omega)\frac{\mathbb{P}_{t_s}(\omega_1)\ldots\mathbb{P}_{t_s}(\omega_p)}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1)-\mathbb{P}_{t_s}(\mathcal{A}_2)^p}$$

En reprenant les notations $\mathcal{E}_{err;\mathcal{A}_1}$ et $\mathcal{E}_{err;\mathcal{A}_2}$ ainsi que $\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}(\mathcal{F};\mathbb{P})$. On peut alors écrire le différentiel $\Delta\mathcal{E}_{err}$ comme :

$$\Delta\mathcal{E}_{err}\left[\begin{array}{c}\mathcal{F}_{t_s}\\ \mathbb{P}_{t_s}\end{array}\longrightarrow\begin{array}{c}\mathcal{F}_{t_s+\mathrm{d}t}\\ \mathbb{P}_{t_s+\mathrm{d}t}\end{array}\right] = \mathcal{E}_{err}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})\left[\frac{1}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1)-\mathbb{P}_{t_s}(\mathcal{A}_2))^p}-1\right]$$

$$-p\frac{\mathbb{P}_{t_s}(\mathcal{A}_1)\mathcal{E}_{err;\mathcal{A}_1}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})+\mathbb{P}_{t_s}(\mathcal{A}_2)\mathcal{E}_{err;\mathcal{A}_2}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1)-\mathbb{P}_{t_s}(\mathcal{A}_2))^p}$$

$$+p(p-1)\frac{\mathbb{P}_{t_s}(\mathcal{A}_1)\mathbb{P}_{t_s}(\mathcal{A}_2)\mathcal{E}_{err;\mathcal{A}_1;\mathcal{A}_2}(\mathcal{F}_{t_s};\mathbb{P}_{t_s})}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1)-\mathbb{P}_{t_s}(\mathcal{A}_2))^p}$$

En multipliant alors par $(1-s)^p$ et en introduisant les quantités s et p, on obtient donc le résultat attendu. □

C-5 Récapitulatif de la dynamique des sauts

On peut donc résumer la dynamique des probabilités de transition dans le tableau suivant. Il y est mentionné la probabilité de proposer un saut, ainsi que la probabilité de l'accepter, ceci en fonction des grandeurs introduites précédemment. Si $\mathcal{F}_{t_s}$ désigne donc la forêt à l'instant de saut et $\mathbb{P}_{t_s}$ la probabilité à cet instant, alors, la synthétisation des seuils et probabilités de transition pour les greffes sont :

Opération choisie	Opération inverse	Tirage	seuil d'acceptation τ
$(\mathbf{T_g})$ sur $\mathcal{A}_1;\mathcal{A}_2$	$(\mathbf{T_c})$ de $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$	p_g	$\frac{p_c \mathrm{Q}_c(\mathcal{A}_1 :: \mathcal{A}_2)}{p_g \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)} \times e^{\left[-\alpha\left\lvert \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)\right\rvert + \beta \mathrm{I}(\mathcal{A}_1;\mathcal{A}_2)\right]}$
$(\mathbf{T_{g;sg}})$ sur $\mathcal{A}_1;\mathcal{A}_2$	$(\mathbf{T_c})$ de $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$	$p_{g;sg}$	$\frac{p_c \mathrm{Q}_c(\mathcal{A}_1 :: \mathcal{A}_2)}{p_g \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)} e^{\left[-\alpha\left[\underbrace{\left\lvert \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)\right\rvert - \left\lvert \mathcal{A}_1\right\rvert}_{=1}\right]\right]}$ $\times e^{\beta[\mathrm{I}(\mathcal{A}_1;\mathcal{A}_2) - \mathrm{I}(\mathcal{A}_1.g;\mathcal{A}_1.d)]}$ $\times e^{\gamma\left[\frac{1-(1-\mathbb{P}_{t_s}(\mathcal{A}_1))^p - p\mathbb{P}_{t_s}(\mathcal{A}_1)\mathcal{E}_{err;\mathcal{A}_1}/\mathcal{E}_{err}}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1))^p}\right]}$
$(\mathbf{T_{g;sd}})$ sur $\mathcal{A}_1;\mathcal{A}_2$	$(\mathbf{T_c})$ de $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$	$p_{g;sd}$	$\frac{p_c \mathrm{Q}_c(\mathcal{A}_1 :: \mathcal{A}_2)}{p_g \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)} e^{\left[-\alpha\left[\underbrace{\left\lvert \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)\right\rvert - \left\lvert \mathcal{A}_2\right\rvert}_{=1}\right]\right]}$ $\times e^{\beta[\mathrm{I}(\mathcal{A}_1;\mathcal{A}_2) - \mathrm{I}(\mathcal{A}_2.g;\mathcal{A}_2.d)]}$ $\times e^{\gamma\left[\frac{1-(1-\mathbb{P}_{t_s}(\mathcal{A}_2))^p - p\mathbb{P}_{t_s}(\mathcal{A}_2)\mathcal{E}_{err;\mathcal{A}_2}/\mathcal{E}_{err}}{(1-\mathbb{P}_{t_s}(\mathcal{A}_2))^p}\right]}$
$(\mathbf{T_{g;sgd}})$ sur $\mathcal{A}_1;\mathcal{A}_2$	$(\mathbf{T_c})$ de $\mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)$	$p_{g;sgd}$	$\frac{p_c \mathrm{Q}_c(\mathcal{A}_1 :: \mathcal{A}_2)}{p_g \mathrm{Q}_g(\mathcal{A}_1)\mathrm{Q}_g(\mathcal{A}_2)} e^{\left[-\alpha\left[\left\lvert \mathrm{T}_g(\mathcal{A}_1;\mathcal{A}_2)\right\rvert - \left\lvert \mathcal{A}_1\right\rvert - \left\lvert \mathcal{A}_2\right\rvert\right]\right]}$ $\times e^{\beta[\mathrm{I}(\mathcal{A}_1;\mathcal{A}_2) - \mathrm{I}(\mathcal{A}_1.g;\mathcal{A}_1.d) - \mathrm{I}(\mathcal{A}_2.g;\mathcal{A}_2.d)]}$ $\times e^{\gamma \Delta\mathcal{E}_{err}}$

En ce qui concerne la coupe et la renaissance, on a également le tableau :

Opération choisie	Opération inverse	Tirage	seuil d'acceptation τ
$(\mathbf{T_c})$ $(\mathcal{A})$ $\quad \mathcal{A} \in \mathcal{F}_0$	$(\mathbf{T_i})$ de $\mathcal{A}$	p_c	$\frac{p_i}{p_c \mathrm{Q}_c(\mathcal{A})\left(\lvert\mathcal{F}_0 \setminus \mathcal{F}_{t_s}\rvert + 1\right)} e^{\alpha\lvert\mathcal{A}\rvert}$ $\times e^{\gamma\left[\frac{1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p - p\mathbb{P}_{t_s}(\mathcal{A})\mathcal{E}_{err;\mathcal{A}}/\mathcal{E}_{err}}{(1-\mathbb{P}_{t_s}(\mathcal{A}))^p}\right]}$
$(\mathbf{T_c})$ $(\mathcal{A})$ $\mathcal{A}.g \in \mathcal{F}_{t_s}$ $\mathcal{A}.d \notin \mathcal{F}_{t_s}$	$(\mathbf{T_{g;sd}})$ de $\mathrm{T}_g(\mathcal{A}_g; \mathcal{A}.d)$	p_c	$\frac{p_{g;sd}\mathrm{Q}_g(\mathcal{A}.g)\mathrm{Q}_g(\mathcal{A}.d)}{p_c \mathrm{Q}_c(\mathcal{A})} \times e^{-\alpha\left[\underbrace{\lvert\mathcal{A}.d\rvert - \lvert\mathcal{A}\rvert}_{=-1}\right]}$ $\times e^{\beta[\mathrm{I}((\mathcal{A}.d).g;(\mathcal{A}.d).d) - \mathrm{I}(\mathcal{A}.g;\mathcal{A}.d)]}$ $\times e^{\gamma\left[\frac{1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p - p\mathbb{P}_{t_s}(\mathcal{A})\mathcal{E}_{err;\mathcal{A}}/\mathcal{E}_{err}}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1))^p}\right]}$
$(\mathbf{T_c})$ $(\mathcal{A})$ $\mathcal{A}.g \notin \mathcal{F}_{t_s}$ $\mathcal{A}.d \in \mathcal{F}_{t_s}$	$(\mathbf{T_{g;sg}})$ de $\mathrm{T}_g(\mathcal{A}_g; \mathcal{A}.d)$	p_c	$\frac{p_{g;sg}\mathrm{Q}_g(\mathcal{A}.g)\mathrm{Q}_g(\mathcal{A}.d)}{p_c \mathrm{Q}_c(\mathcal{A})} \times e^{-\alpha\left[\underbrace{\lvert\mathcal{A}.g\rvert - \lvert\mathcal{A}\rvert}_{=-1}\right]}$ $\times e^{\beta[\mathrm{I}((\mathcal{A}.g).g;(\mathcal{A}.g).d) - \mathrm{I}(\mathcal{A}.g;\mathcal{A}.d)]}$ $\times e^{\gamma\left[\frac{1-(1-\mathbb{P}_{t_s}(\mathcal{A}))^p - p\mathbb{P}_{t_s}(\mathcal{A})\mathcal{E}_{err;\mathcal{A}}/\mathcal{E}_{err}}{(1-\mathbb{P}_{t_s}(\mathcal{A}_1))^p}\right]}$
$(\mathbf{T_c})$ $(\mathcal{A})$ $\mathcal{A}.g \notin \mathcal{F}_{t_s}$ $\mathcal{A}.d \notin \mathcal{F}_{t_s}$	$(\mathbf{T_{g;sgd}})$ de $\mathrm{T}_g(\mathcal{A}_g; \mathcal{A}.d)$	p_c	$\frac{p_{g;sgd}\mathrm{Q}_g(\mathcal{A}.g)\mathrm{Q}_g(\mathcal{A}.d)}{p_c \mathrm{Q}_c(\mathcal{A})} e^{-\alpha[\lvert\mathcal{A}.g\rvert + \lvert\mathcal{A}.d\rvert - \lvert\mathcal{A}\rvert]}$ $\times e^{\beta[\mathrm{I}((\mathcal{A}.g).g;(\mathcal{A}.g).d) + \mathrm{I}((\mathcal{A}.d).g;(\mathcal{A}.d).d)]}$ $\times e^{-\beta\mathrm{I}(\mathcal{A}.g;\mathcal{A}.d)} \times e^{\gamma\Delta\mathcal{E}_{err}}$
$(\mathbf{T_i})$ $(\mathcal{A})$	$(\mathbf{T_c})$ $(\mathcal{A})$	p_i	$\frac{p_c \mathrm{Q}_c(\mathcal{A})\lvert\mathcal{F}_0 \setminus \mathcal{F}_{t_s}\rvert}{p_i} e^{-\alpha\lvert\mathcal{A}\rvert}$

C-6 Énergie en $\log(\mathbb{E})$

Nous donnons ici quelques pistes qui pourraient permettre de traiter le cas d'une diffusion utilisant un terme d'énergie en log de l'espérance de g, alors :

$$\mathcal{E}_{err}(\mathbb{P}) = \log\left(\sum_{\omega} \mathbb{P}(\omega) g(\omega)\right)$$

Le critère de saut du chapitre 4, détaillé dans les paragraphes précédents, devient par exemple pour la suppression d'un mot μ :

$$\Delta\mathcal{E}_{err} = \log\left(\frac{1-\omega\mathbb{P}_n(\mu)\mathbb{E}_{\mathbb{P}_n}[g|\mu]\,/\mathbb{E}_{\mathbb{P}_n}[g]}{(1-\mathbb{P}_n(\mu))^{\omega}}\right)$$

L'influence de la probabilité $\mathbb{P}_n(\mu)$ est donc nettement plus évidente dans cette situation que dans le cas où l'on choisit une diffusion basée sur une énergie comme l'espérance de l'erreur.

En revanche, la simulation d'une diffusion basée sur une telle énergie parait nettement plus délicate. En effet, on a :

$$\nabla\mathcal{E}_{err}(\mathbb{P})(\mu) = \frac{\sum\limits_{\omega} g(\omega)\dfrac{\mathbb{P}(\omega)}{\mathbb{P}(\mu)}\mathrm{C}(\omega,\mu)}{\mathbb{E}_{\mathbb{P}}[g]}$$

Il s'agit donc ici de simuler une loi de probabilité sur les $|\omega|$-uplets donnée par

$$\mu(\omega)\frac{g(\omega)}{\sum\limits_{\omega'} g(\omega')\mathbb{P}(\omega')}$$

Cette loi ne peut être simulée directement, puisqu'elle est alors définie comme un quotient d'un terme calculable avec une somme sur tous les $|\omega|$-uplets, somme incalculable.

On peut par contre simuler le tirage des ω, en utilisant une simulation par chaîne de Markov de probabilité de transition

$$\mathrm{Q}(\omega,\mu) = \mathbb{P}(\mu)\ \mathrm{Min}\left\{1; \frac{g(\mu)}{g(\omega)}\right\}$$

Le coefficient d'ergodicité de cette chaîne de Markov [Dob56] est donné par l'égalité :

$$\tau(\mathrm{Q}) = 1 - \inf_{\omega_1,\omega_2}\sum_{\mu}\mathrm{Min}\left\{\mathrm{Q}(\omega_1,\mu);\mathrm{Q}(\omega_2,\mu)\right\}$$

peut être majoré par

$$\tau(\mathrm{Q}) = 1 - \frac{\mathrm{Min}\ g}{\mathrm{Max}\ g} < 1$$

On obtient alors une convergence de la chaîne de markov utilisant la probabilité de transition Q vers sa loi invariante qui est précisément $\mu(.)$ avec un taux majoré par une puissance de $\tau(\mathrm{Q})$.

Nous pouvons donc conclure que dans ce cas, même si les critères de sauts sont plus « sympathiques », la simulation de la diffusion est nettement plus difficile à effectuer, et la convergence de l'algorithme d'approximation de la descente de gradient semble être plus complexe.

Annexe D - Existence des diffusions réfléchies avec sauts

Rappelons les énoncés des deux théorèmes d'existence et d'unicité de diffusion sous contraintes donnés dans le chapitre 4 et 5 :

Théorème 4.2.2 (Existence de diffusions contraintes dans G)
Pour tout x de G, il existe un unique couple de processus $(\mathrm{X}^x(t), k(t))$ adapté à $\mathcal{T}_t$ ainsi que $\gamma(t)$ tel que

$$\forall t \geqslant 0 \qquad \mathrm{X}^x(t) \in \mathrm{G}\, p.s.$$

$$\forall t \geqslant 0 \qquad \mathrm{X}^x(t) = x + \int_0^t \sigma(\mathrm{X}^x(s))\mathrm{dW}(s) + \int_0^t b(\mathrm{X}^x(s))\mathrm{d}s + k(t)\, p.s.$$

$$\forall \mathrm{T} \geqslant 0 \qquad |k|(\mathrm{T}) < +\infty\, p.s.$$

Théorème 5.2.1 (Existence des diffusions sous contraintes avec sauts)
En supposant que les conditions $(\mathbf{C_4})$, $(\mathbf{C_5})$, $(\mathbf{C_6})$ *et* $(\mathbf{C_7})$ *soient satisfaites, on a l'existence et l'unicité d'un processus couplé* $(\mathbb{P}_t; \mathrm{X}_t)_{t\geqslant 0}$ *vérifiant l'équation intégrale* $(\mathbf{E-6})$.

Le premier théorèmé étant un cas particulier du second (le terme de sauts est nul dans le premier), nous ne donnerons que les pistes pour démontrer le second théorème. Cette preuve reprend les idées de [SV79] (chapitre 5, théorème 5.1).
Preuve : Nous considérons donc une suite de processus $(\mathbb{P}^n, \mathrm{X}^n)$ qui satisfont l'équation de récurrence :

$$\begin{cases} \begin{pmatrix} \mathrm{Y}^n \\ \\ \mathrm{X}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathbb{P}_0 \\ \\ \mathrm{X}_0 \end{pmatrix} + \displaystyle\int_0^t \mathrm{G}\begin{pmatrix} \mathbb{P}^n \\ \\ \mathrm{X}^{n+1} \end{pmatrix}(s)\mathrm{d}s + \int_0^t \sigma\left(\begin{pmatrix} \mathbb{P}^n \\ \\ \mathrm{X}^{n+1} \end{pmatrix}\right)(s)\mathrm{dW}s \\ \qquad\qquad + \displaystyle\int_{\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^{|\mathcal{A}^\star|}} q\left(\begin{pmatrix} \mathbb{P}^n(t) \\ \mathrm{X}^n(t) \end{pmatrix}; \begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix}\right) \mathrm{N}\left(\mathrm{d}\begin{pmatrix} \mathbb{P} \\ \mathrm{X} \end{pmatrix}; \mathrm{d}t\right) \\ \mathbb{P}^{n+1}(t) = \Gamma_{\mathrm{X}^n}(\mathrm{Y}^n(t)) \end{cases}$$

On définit par ailleurs les deux quantités :

$$\begin{cases} \Delta_1^n(t) = \mathrm{E}\left[\underset{s\leqslant t}{\mathrm{Sup}} \left\| \begin{matrix} \mathrm{Y}^{n+1}(s) - \mathrm{Y}^n(s) \\ \mathrm{X}^{n+1}(s) - \mathrm{X}^n(s) \end{matrix} \right\|^2\right] \\ \\ \Delta_2^n(t) = \mathrm{E}\left[\underset{s\leqslant t}{\mathrm{Sup}} \left\| \begin{matrix} \mathbb{P}^{n+1}(s) - \mathbb{P}^n(s) \\ \mathrm{X}^{n+1}(s) - \mathrm{X}^n(s) \end{matrix} \right\|^2\right] \end{cases}$$

L'application $\Gamma_\cdot(.)$ étant Lipchitzienne d'après $(\mathbf{C_7})$, on en déduit immédiatement que :

$$\Delta_2^n(t) \leqslant \theta_4^2 \Delta_1^n(t)$$

Par ailleurs, en réécrivant la définition de Y^n, et en utilisant $(\mathbf{C_1})$, $(\mathbf{C_2})$ et $(\mathbf{C_3})$, on obtient de manière identique à [SV79] :

$$\Delta_1^n(t) \leqslant \mathrm{K}(1+t)\int_0^t \Delta_2^n(s)\mathrm{d}s$$

Finalement $$\exists\, \mathrm{C} > 0 \qquad \Delta_2^n(t) \leqslant \mathrm{C}(1+t)\int_0^t \Delta_2^n(s)\mathrm{d}s$$

On montre alors par récurrence sur n que si t est un réel inférieur strictement à T, on a :

$$\forall \mathrm{T} \geqslant 0 \qquad \underset{t\leqslant \mathrm{T}}{\mathrm{Sup}}\ \Delta_2^n(t) \leqslant \frac{[\mathrm{K}(1+\mathrm{T})]^{n+1}}{n!}$$

La suite des processus $(\mathrm{Y}^n)_{n\in\mathbb{N}}$, $(\mathbb{P}^n)_{n\in\mathbb{N}}$ et $(\mathrm{X}^n)_{n\in\mathbb{N}}$ sont donc de Cauchy dans un espace complet, il existe donc Y, X et $\mathbb{P}$ limites de tels processus :

$$\lim_{m\mapsto+\infty} \int_0^{\mathrm{T}} |\mathrm{Y}_s - \mathrm{Y}_s^n|^2 \mathrm{d}s = 0$$

$$\lim_{m\mapsto+\infty} \int_0^{\mathrm{T}} |\mathbb{P}_s - \mathbb{P}_s^n|^2 \mathrm{d}s = 0$$

et

$$\lim_{m\mapsto+\infty} \int_0^{\mathrm{T}} |\mathrm{X}_s - \mathrm{X}_s^n|^2 \mathrm{d}s = 0$$

En posant alors

$$\begin{cases} \begin{pmatrix} \overline{\mathrm{Y}}_t \\ \\ \overline{\mathrm{X}}_t \end{pmatrix} = \begin{pmatrix} \mathbb{P}_0 \\ \\ \mathrm{X}_0 \end{pmatrix} + \displaystyle\int_0^t \mathrm{G}\begin{pmatrix} \mathbb{P} \\ \\ \mathrm{X} \end{pmatrix}(s)\mathrm{d}s + \int_0^t \sigma\left(\begin{pmatrix} \mathbb{P} \\ \\ \mathrm{X} \end{pmatrix}\right)(s)\mathrm{d}\mathrm{W}s \\ \qquad\qquad + \displaystyle\int_{\mathcal{S}_{\mathcal{A}^\star}\times\{0;1\}^{|\mathcal{A}^\star|}} q\left(\begin{pmatrix} \mathbb{P}(t) \\ \mathrm{X}(t) \end{pmatrix}; \begin{pmatrix} \mathbb{Q} \\ \mathrm{Z} \end{pmatrix}\right) \mathrm{N}\left(\mathrm{d}\begin{pmatrix} \mathbb{Q} \\ \mathrm{Z} \end{pmatrix}; \mathrm{d}t\right) \\ \overline{\mathbb{P}}(t) = \Gamma_{\overline{\mathrm{X}}}(\mathrm{Y}(t)) \end{cases}$$

on montre alors que les processus $(\mathrm{Y}^n)_{n\in\mathbb{N}}$, $(\mathbb{P}^n)_{n\in\mathbb{N}}$ et $(\mathrm{X}^n)_{n\in\mathbb{N}}$ convergent également vers $\overline{\mathrm{Y}}$, $\overline{\mathbb{P}}$ et $\overline{\mathrm{X}}$. Les processus limites $\mathbb{P}, \mathrm{X}$ et Y sont donc solution de l'équation intégrale.

Par la suite, il suffit d'utiliser la caractéristique des solutions de SP pour obtenir que le processus Z défini par

$$\mathrm{Z} = \mathbb{P} - \mathrm{Y}$$

vérifie les conditions mentionnées dans l'énoncé [DI91].

Pour démontrer l'unicité d'un tel processus, il suffit de considérer deux couples de processus solutions $\left(\mathbb{P}^1, \mathrm{X}^1\right)$ et $\left(\mathbb{P}^2, \mathrm{X}^2\right)$ et poser :

$$\Delta(t) = \underset{s\leqslant t}{\mathrm{Sup}} \left\| \begin{pmatrix} \mathbb{P}^1(s) - \mathbb{P}^2(s) \\ \\ \mathrm{X}^1(s) - \mathrm{X}^2(s) \end{pmatrix} \right\|$$

On a alors (en vertu du caractère Lipchitzien) que :

$$\Delta(t) \leqslant \kappa \int_0^t \Delta(s)\mathrm{d}s$$

Le lemme de Gromwall permet alors de conclure que Δ est nul presque sûrement. $\square$

Bibliographie

[AB02] Rami Atar and Amarjit Budhiraja. Stability properties of constrained jump-diffusion processes. *Electron. J. Probab.*, 7 :no. 22, 31 pp. (electronic), 2002.

[ABD01] Rami Atar, Amarjit Budhiraja, and Paul Dupuis. Correction note : "On positive recurrence of constrained diffusion processes". *Ann. Probab.*, 29(3) :1404, 2001.

[Abr00] Romain Abraham. Reflecting Brownian snake and a Neumann-Dirichlet problem. *Stochastic Process. Appl.*, 89(2) :239–260, 2000.

[AG97a] Yali Amit and Donald Geman. A computational model for visual selection. 1997.

[AG97b] Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7) :1545–1588, 1997.

[AGW97] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. PAMI*, 19 :1300–1306, 1997.

[AO76] Robert F. Anderson and Steven Orey. Small random perturbation of dynamical systems with reflecting boundary. *Nagoya Math. J.*, 60 :189–216, 1976.

[Ben96] Michel Benaim. A dynamical system approach to stochastic approximations. *SIAM J. Control Optim.*, 34(2) :437–472, 1996.

[Ben00] Michel Benaïm. Convergence with probability one of stochastic approximation algorithms whose average is cooperative. *Nonlinearity*, 13(3) :601–616, 2000.

[Bil99] Patrick Billingsley. *Convergence of probability measures.* Wiley Series in Probability and Statistics : Probability and Statistics. John Wiley & Sons Inc., New York, second edition, 1999. A Wiley-Interscience Publication.

[BK02] Robert Buche and Harold J. Kushner. Rate of convergence for constrained stochastic approximation algorithms. *SIAM J. Control Optim.*, 40(4) :1011–1041 (electronic), 2001/02.

[BMP] A. Benveniste, M. Métivier, and P. Priouret. *Algorithmes adaptatifs et approximations stochastiques.* Masson. Théorie et applications à l'identification, au traitement du signal et à la reconnaissance des formes.

[Bou00] Nicolas Bouleau. *Processus stochastiques et applications.* Hermann, 2000.

[Bre68] Leo Breiman. *Probability.* Addison-Wesley Publishing Company, Reading, Mass., 1968.

[Bre98] Leo Breiman. Arcing classifiers. *Ann. Statist.*, 26(3) :801–849, 1998. With discussion and a rejoinder by the author.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, 1998.

[Car98] Jean-François Cardoso. Blind signal separation : statistical principles. In R.-W. Liu and L. Tong, editors, *Proceedings of the IEEE, special issue on blind identification and estimation*, 1998.

[Cer94] Raphaël Cerf. Une théorie asymptotique des algorithmes génétiques. *These de l'Université Montpellier II*, 1994.

[Cer96] Raphaël Cerf. The dynamics of mutation-selection algorithms with large population sizes. *Ann. Inst. H. Poincaré Probab. Statist.*, 32(4) :455–508, 1996.

[Cer98] Raphaël Cerf. Asymptotic convergence of genetic algorithms. *Adv. in Appl. Probab.*, 30(2) :521–550, 1998.

[CM98] Vladimir Cherkassky and Filip Mulier. *Learning from data, Concepts, Theory, and Methods.* John Wiley and Sons, Inc. New York, USA, 1998.

[CRR03] Olivier Cappé, Christian P. Robert, and Tobias Rydén. Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 65(3) :679–700, 2003.

[CT91] Thomas Cover and Joy Thomas. *Information Theory.* Wiley, New York, 1991.

[CVBM02] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1–3) :131–159, 2002.

[DI91] Paul Dupuis and Hitoshi Ishii. On Lipschitz continuity of the solution mapping to the Skorokhod problem, with applications. *Stochastics Stochastics Rep.*, 35(1) :31–62, 1991.

[DL92] Robert Dautray and Jacques-Louis Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 2.* Springer-Verlag, Berlin, 1992. Evolution problems. I, With the collaboration of Michel Artola, Michel Cessenat and Hélène Lanchon, Translated from the French by Alan Craig.

[Dob56] R. Dobrushin. Central limit theorems for non-stationary markov chains. *Theory of Probability and its Applications*, 1(1) :65–80, 1956.

[DR99] Paul Dupuis and Kavita Ramanan. Convex duality and the Skorokhod problem. I, II. *Probab. Theory Related Fields*, 115(2) :153–195, 197–236, 1999.

[Duf96] Marie Duflo. *Algorithmes stochastiques*, volume 23 of *Mathématiques & Applications (Berlin) [Mathematics & Applications].* Springer-Verlag, Berlin, 1996.

[EK86] Stewart Ethier and Thomas Kurtz. *Markov Processes.* John Willey and Sons, New York, 1986.

[ET95] Bradley Efron and Robert Tibshirani. Cross-validation and the bootstrap : Estimating the error rate of a prediction rule. 1995.

[FG01] Francois Fleuret and Donald Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41(1/2) :85–107, 2001.

[FHT00] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression : a statistical view of boosting. *Ann. Statist.*, 28(2) :337–407, 2000. With discussion and a rejoinder by the authors.

[Fle00] François Fleuret. *Détection hiérarchique de visages par apprentissage statistique - These university Paris 6.* 2000.

[Fle03] François Fleuret. Binary feature selection with conditional mutual information. 2003.

[FS99] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games Econom. Behav.*, 29(1-2) :79–103, 1999. Learning in games : a symposium in honor of David Blackwell.

[FS03] Patrick Ferrari and Herbert Spohn. Constrained brownian motion : fluctuations away from circular and parabolic barriers. 2003.

[FW98] M. I. Freidlin and A. D. Wentzell. *Random perturbations of dynamical systems*, volume 260 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, New York, second edition, 1998. Translated from the 1979 Russian original by Joseph Szücs.

[Gad04] Sebastien Gadat. Features space construction by jump diffusion algorithms for signal classification. *Preprint CMLA en préparation*, 2004.

[GBD92] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4 :1–58, 1992.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6) :721–741, November 1984.

[GM94] Ulf Grenander and Michael I. Miller. Representations of knowledge in complex systems. *J. Roy. Statist. Soc. Ser. B*, 56(4) :549–603, 1994. With discussion and a reply by the authors.

[Gre95] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82 :711–732, 1995.

[GY04] Sebastien Gadat and Laurent Younes. A stochastic algorithm of features extraction for pattern recognition. *Preprint CMLA*, page 19, 2004.

[Has70] W. K. Hastings. Monte carlo sampling methods using markov chains, and their applications. *Biometrika*, 57 :97–109, 1970.

[HR81] J. Michael Harrison and Martin I. Reiman. Reflected Brownian motion on an orthant. *Ann. Probab.*, 9(2) :302–308, 1981.

[Hsu04] Elton P. Hsu. A brief introduction to brownian motion on a riemannian manifold. *Note de cours- Department of Mathematics, Northwestern University*, 2004.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

[HW92] J. M. Harrison and R. J. Williams. Brownian models of feedforward queueing networks : quasireversibility and product form solutions. *Ann. Appl. Probab.*, 2(2) :263–293, 1992.

[Hyv99] Aapo Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2 :94–128, 1999.

[JH91] Christian Jutten and Jeanny Hérault. Blind separation of sources, part 1 : An adaptative algorithm bases on neuromimetic architecture. *Signal Processing*, 24 :1–10, 1991.

[JK00] T. Joachims and R Klinkenberg. Detecting concept drift with support vector machines. 2000.

[Joa02] T. Joachims. Optimizing search engines using clickthrough data. 2002.

[JS87] J. Jacod and A Shiryaev. *Limit theorems for Stochastic Processes*. Springer-Verlag, 1987.

[Jut87] Christian Jutten. Calcul neuromimétique et traitement du signal. analyse en composantes indépendantes. *Thèse d'état de l'INPG*, 1987.

[KGA02] Samuel Krempp, Donald Geman, and Yali Amit. Sequential learning of reusable parts for object detection. 2002.

[Kni99] K Knight. *Mathematical Statistics*. Chapmand & Hall, 1999.

[KS01] Thomas G. Kurtz and Richard H. Stockbridge. Stationary solutions and forward equations for controlled and singular martingale problems. *Electron. J. Probab.*, 6 :no. 17, 52 pp. (electronic), 2001.

[Kus00] Harold J. Kushner. Jump-diffusions with controlled jumps : existence and numerical methods. *J. Math. Anal. Appl.*, 249(1) :179–198, 2000. Special issue in honor of Richard Bellman.

[Kus02] Harold J. Kushner. Numerical approximations for stochastic differential games. *SIAM J. Control Optim.*, 41(2) :457–486 (electronic), 2002.

[KY03] Harold J. Kushner and G. George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, second edition, 2003. Stochastic Modelling and Applied Probability.

[LBBH98] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[MIT] MIT. Cbcl face database - mit center for biological and computation learning , http ://www.ai.mit.edu/projects/cbcl.

[MPB98] Laurent Mazliak, Pierre Priouret, and Paolo Baldi. *Martingales et chaînes de Markov*. Hermann, 1998.

[MRR+53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chem. Phys.*, 21 :1087–1092, 1953.

[Rei84] Martin I. Reiman. Open queueing networks in heavy traffic. *Math. Oper. Res.*, 9(3) :441–458, 1984.

[Ris83] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11 :2 :416–431, 1983.

[Ris89] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* Series in Computer Science vol 15. World Scientific, 1989.

[Ros58] Frank Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 :386–408, 1958.

[RW00] L. C. G. Rogers and David Williams. *Diffusions, Markov processes, and martingales. Vol. 1.* Cambridge Mathematical Library. Cambridge University Press, Cambridge, 2000. Foundations, Reprint of the second (1994) edition.

[RY94] Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*, volume 293 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, second edition, 1994.

[Sap90] Gilbert Saporta. *Probabilités, analyse des données et statistique.* Technip, 1990.

[SC98] Patrice Simard and Yann Le Cun. Memory based character recognition using a transformation invariant metric. *AT&T Bell Laboratories*, 1998.

[Sko61] A.V. Skorohod. Stochastic equations for diffusions in a bounded region. *Theory Probab. Appl*, 6 :264–274, 1961.

[SMG97] A. Srivastava, M. I. Miller, and U. Grenander. Ergodic algorithms on special Euclidean groups for ATR. In *Systems and control in the twenty-first century (St. Louis, MO, 1996)*, volume 22 of *Progr. Systems Control Theory*, pages 327–350. Birkhäuser Boston, Boston, MA, 1997.

[SV79] Daniel W. Stroock and S. R. Srinivasa Varadhan. *Multidimensional diffusion processes*, volume 233 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1979.

[UCI] UCI. Uci machine learning repository : http ://www.ics.uci.edu/ mlearn/mlrepository.html.

[USP] USPostal. The national institute of standards and technology- http ://www.nist.gov/srd/intro.htm.

[Vap00] Vladimir N. Vapnik. *The nature of statistical learning theory.* Statistics for Engineering and Information Science. Springer-Verlag, New York, second edition, 2000.

[Wil91] David Williams. *Probability with martingales.* Cambridge University Press, 1991.

[WMC+00] Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. Feature selection for SVMs. In *NIPS*, pages 668–674, 2000.

[You89] Laurent Younes. Parametric inference for imperfectly observed gibbsian fields. *Probab. Theory Related Fields*, 82(4) :625–645, 1989.

[You91] Laurent Younes. Maximum likelihood estimation for Gibbsian fields. In *Spatial statistics and imaging (Brunswick, ME, 1988)*, volume 20 of *IMS Lecture Notes Monogr. Ser.*, pages 403–426. Inst. Math. Statist., Hayward, CA, 1991.