UNIVERSITÉ PARIS-EST

IGN
INSTITUT NATIONAL DE L'INFORMATION GÉOGRAPHIQUE ET FORESTIÈRE

Mastère DeSIGeo

Yann MÉNEROUX

Cours d'apprentissage statistique

Table des matières

Chapitre 1

Introduction

1.0.1 L'apprentissage supervisé

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

C'est en particulier l'approche suivie par les algorithmes de type Séparateurs à Vaste Marge (SVM), qui construisent un hyper-plan séparateur affine (paramétré par $\boldsymbol{\theta}$) dans l'espace des descripteurs. Une nouvelle donnée (*i.e.* un nouveau point de l'espace) est alors projetée sur l'axe porté par le vecteur normal à l'hyper-plan, et le signe de la valeur résultante indique la prédiction du classifieur pour cette donnée. Les réseaux de neurones artificiels utilisent une approche similaire.

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

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

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

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

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

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

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

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

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

1.0.2 L'apprentissage de données fonctionnelles

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

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

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

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

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

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

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

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

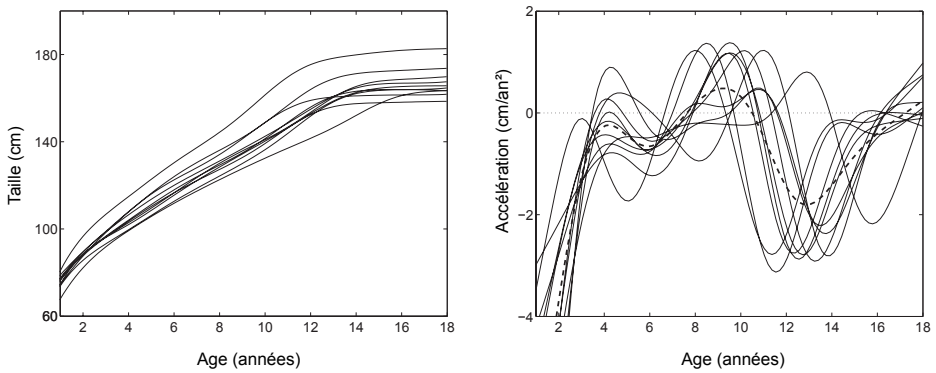

FIGURE 1.1 – Exemple classique d'analyse de données fonctionnelles tiré de Ramsay et Silverman (2007), avec l'étude de la croissance de 10 sujets féminins jusqu'à l'âge de 18 ans, en cm (à gauche) et en cm/an^2 (à droite). En ADF, la clé réside souvent dans une représentation adéquate du jeu de données.

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

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

Chapitre 2

Éléments d'apprentissage statistique

2.1 Méthodes d'apprentissage supervisé

Dans cette section, nous exposons brièvement six algorithmes d'apprentissage supervisé, utilisés dans le cadre de la classification binaire pour des descripteurs continus. Pour la plupart de ces algorithmes, la généralisation au cas de la classification catégorielle est immédiate, en remarquant qu'un problème univarié à k classes peut toujours être transformé en un problème de classification binaire à k variables (Dietterich et Bakiri, 1991).

On désigne par $X = (X_1, X_2, ...X_p) \in \mathbb{R}^p$ l'ensemble des descripteurs d'une instance d'étiquette $Y \in \{0, 1\}$. Le jeu d'entraînement $\mathcal{D}_n = \{(X^1, Y^1)...(X^n, Y^n)\}$ est un n-échantillon de réalisations de la loi jointe inconnue $p(X, Y)$. On note $X^{(n+1)}$ le vecteur de descripteurs d'une nouvelle instance d'étiquette $Y^{(n+1)}$ inconnue et à déterminer.

2.1.1 Classifieur bayésien naïf

Le classifieur bayésien naïf (ou *Naive Bayes* dans la littérature anglo-saxonne) pose l'hypothèse d'indépendance des descripteurs conditionnellement à l'étiquette :

$$P(X_i|Y, X_{j\neq i}) = P(X_i|Y). \tag{2.1}$$

À l'aide de la règle de chaînage, on tire facilement de la relation 2.1 la simplification suivante (pour $i \neq j$) : $P(X_i, X_j|Y) = P(X_i|Y)P(X_j|Y, X_i) = P(X_i|Y)P(X_j|Y)$, d'où l'expression factorisée de la loi jointe :

$$P(X, Y) = P(Y)P(X|Y) = P(Y)\prod_{i=1}^{p} P(X_i|Y). \tag{2.2}$$

La figure 2.1 donne une illustration de modèle graphique probabiliste associé au classifieur bayésien naïf, schématisant l'ensemble des distributions qui se factorisent sous la forme 2.2.

La factorisation 2.2 couplée à la formule de Bayes permet d'exprimer la loi de probabilité sur l'étiquette conditionnellement aux descripteurs :

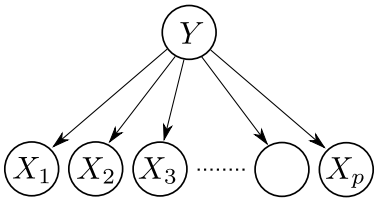

FIGURE 2.1 – Modèle graphique du *Naive Bayes* sur p descripteurs. Les flèches indiquent toutes les dépendances conditionnelles de la distribution de probabilité $p(X, Y)$.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} = \frac{P(Y)\prod_{i=1}^{p} P(X_i|Y)}{\sum_y P(y)\prod_{i=1}^{p} P(X_i|y)}, \tag{2.3}$$

où le dénominateur correspond à la loi marginale de X, *i.e.* au numérateur sommé sur toutes les affectations possibles pour y (en classification binaire la somme opère sur $\{0, 1\}$).

La phase d'entraînement du modèle consiste alors à :

- Estimer les probabilités a priori $P(Y = 0)$ et $P(Y = 1)$ à partir des ratios d'instance positives et négatives. Lorsque le jeu d'entraînement est équilibré, $P(Y = 0) = P(Y = 1)$ et les termes $P(Y)$ s'annulent [1] dans 2.3.

- Estimer les $2p$ lois conditionnelles $P(X_i|Y)$. La littérature est vaste sur le sujet, mais la méthode la plus classiquement utilisée consiste à modéliser ces lois par des gaussiennes (Rogers et Girolami, 2016), de moyennes et variances à déterminer, soit un total de $4p$ paramètres à estimer, à savoir μ_{ij} et σ_{ij} pour $(i, j) \in [\![1; p]\!] \times \{0, 1\}$:

  $$P(X_i = x|Y = j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\Big(-\frac{1}{2}\frac{(x - \mu_{ij})^2}{\sigma_{ij}^2}\Big).$$

  On montre aisément que les estimateurs par maximum de vraisemblance de μ_{ij} et σ_{ij} sont respectivement la moyenne et l'écart-type (biaisé) empiriques des valeurs prises par le i^{eme} descripteur pour les données de la classe j (Zivot, 2009).

La figure 2.2 donne un exemple graphique d'apprentissage du modèle sur un problème simple de classification à 3 classes et 2 descripteurs. L'inférence porte sur les paramètres de 6 distributions (figure de droite) : $\boldsymbol{\mu}_1 = (3.90, 3.02, 4.13)$, $\boldsymbol{\mu}_2 = (7.06, 2.88, 3.88)$, $\boldsymbol{\sigma}_1 = (1.31, 0.45, 0.47)$, $\boldsymbol{\sigma}_2 = (0.56, 0.45, 0.90)$, ainsi que sur les 3 probabilités a priori : $\boldsymbol{\theta} = (0.48, 0.24, 0.28)$. L'hypothèse naïve implique l'alignement des axes principaux des gaussiennes sur les axes du repère (figure de gauche).

L'indépendance conditionnelle des descripteurs est une hypothèse très forte et bien souvent irréaliste en pratique, en particulier lorsque les descripteurs sont significativement corrélés.

1. Sauf à décider expressement d'ajouter un poids sur les classes d'étiquettes pour compenser un éventuel défaut de représentativité statistique de l'échantillon d'entraînement.

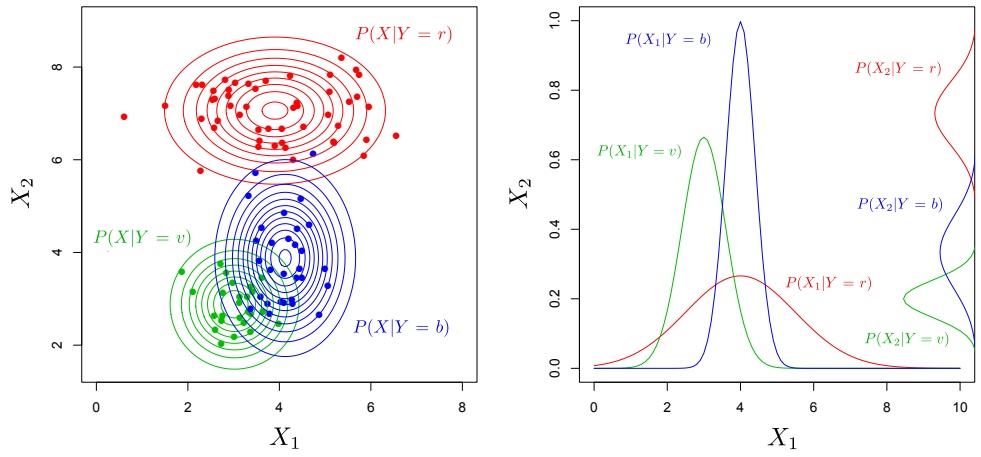

FIGURE 2.2 – Exemple d'apprentissage d'un modèle bayésien naïf pour un problème de classification à 3 classes (rouge, verte et bleue) et 2 descripteurs (X_1, X_2).

En pratique, le classifieur bayésien naïf retourne des décisions robustes, y compris sur des problèmes modérément difficiles, mais avec des valeurs de probabilité associées bien souvent trop catégoriques (Rish et al., 2001).

2.1.2 Les k plus proches voisins

La méthode des k-ppv (ou k-NN pour k-Nearest-Neighbors) est un algorithme simple, requierant uniquement la définition d'un entier k (typiquement 5 à 10) ainsi que d'une métrique sur l'espace des descripteurs. Il s'agit d'une généralisation de la méthode d'interpolation par ppv (méthode de 1-ppv) décrite dans la partie pré-traitements (section ??).

Pour chaque nouvelle donnée $X^{(n+1)}$, la loi conditionnelle $P(Y^{(n+1)}|X^{(n+1)}, \mathcal{D}_n)$ est estimée empiriquement sur un jeu de données réduit composé des k données d'entraînement les plus proches de $X^{(n+1)}$ dans l'espace des descripteurs.

On remarque que dans le cas des k-ppv, la procédure d'entraînement à proprement parler est exécutée avant chaque inférence (ce qui n'exclut pas une phase de pré-calculs, par exemple avec la construction d'un index spatial destiné à optimiser les requêtes de recherche des voisins).

Le nombre k de voisins à sélectionner est un paramètre à régler pour obtenir un compromis entre généralisation et sur-apprentissage (*cf* 2.2).

2.1.3 Les arbres de décision

Introduits par Breiman et al. (1984) sous l'acronyme CART, pour *Classification And Regression Trees*, les arbres de décision peuvent être vus comme une version élaborée des k-ppv.

Le concept de la méthode repose sur un découpage de l'espace des descripteurs à l'aide d'hyperplans séparateurs, de sorte à minimiser une certaine fonction d'*impureté* traduisant l'hétérogénéité des étiquettes des données situées de part et d'autre de la séparation. À chaque nœud j de l'arbre de décision, une donnée $\mathbf{x}$ quelconque est affectée d'un côté ou de l'autre par la règle (Louppe, 2014) :

$$h(\mathbf{x}, \boldsymbol{\theta}_j) = [\phi_j \mathbf{x} > \tau_j] \quad \in \{0, 1\}, \tag{2.4}$$

où ϕ_j est un vecteur unitaire de dimension p et $\boldsymbol{\theta}_j = (\phi_j, \tau_j)$ est le vecteur des paramètres du nœud j, contenant l'indicateur de direction ϕ_j et le seuil de coupure $\tau_j \in \mathbb{R}$. Le vecteur θ_j est déterminé par :

$$\theta_j^* \in \underset{\theta_j}{\operatorname{argmax}} \Big\{ H(\mathcal{S}_j) - \sum_{i \in \{0,1\}} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(\mathcal{S}_j^i) \Big\}, \tag{2.5}$$

avec i qui indice les deux ensembles de la partition créée par l'hyperplan 2.4, où H désigne l'entropie des étiquettes des données d'entraînement et $\mathcal{S}_j^i$ est l'ensemble des données d'entraînement $\mathbf{x}$ telles que $h(\mathbf{x}, \boldsymbol{\theta}_j) = i$. En général, on impose aux hyperplans séparateurs d'être alignés avec les axes du repère, ce qui implique que ϕ est un vecteur indicateur de $\mathbb{R}^p$ ne contenant que des 0 excepté sur la dimension à tester. Dans le cas plus général, on parle d'arbres obliques (Do et al., 2009).

Notons que l'entropie n'est pas le seul choix possible de fonctions d'impureté (Scornet et al., 2015). La profondeur de l'arbre est paramétrée de sorte à obtenir un compromis entre généralisation et sur-apprentissage (*cf* 2.2).

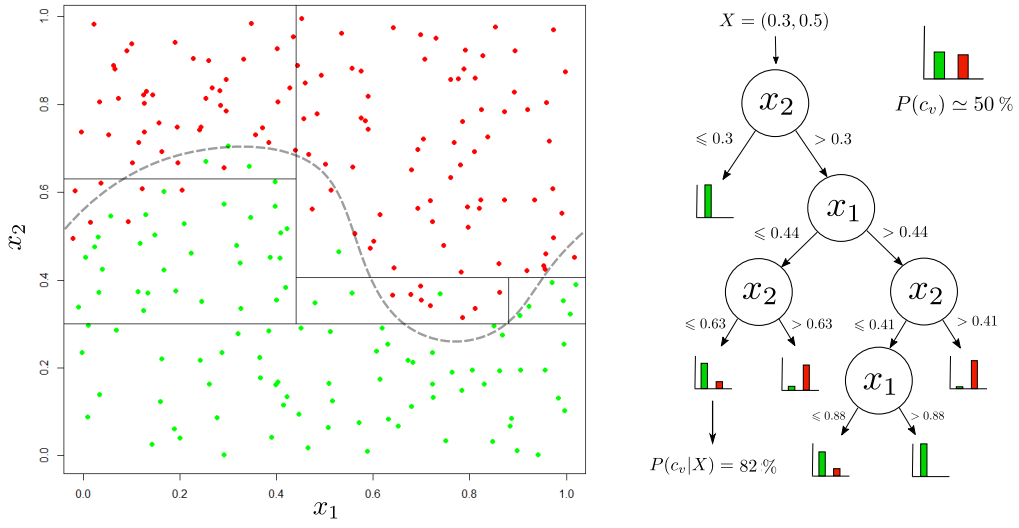

FIGURE 2.3 – Données d'entraînement et arbre de décision (à droite)

Lorsqu'une nouvelle donnée $X^{(n+1)}$ est passée en entrée de l'algorithme, on teste successivement ses descripteurs (en fonction du schéma de l'arbre de décision), jusqu'à être capable de l'affecter à une feuille de l'arbre. La donnée est alors classée selon l'étiquette majoritaire des données d'entraînement situées dans la même feuille.

Prenons un exemple simple en figure 2.3 avec un problème de classification binaire : rouge/vert. Supposons que l'on souhaite attribuer une couleur à un nouveau point : $(0.3, 0.5)$.

La probabilité a priori que la couleur de ce point soit verte sachant les données d'exemple $P(y = \text{vert})$ est proche de 50% (les proportions des deux classes sont à peu près égales). La structure d'arbre représentant une partition de l'espace au niveau de ses feuilles, elle nous permet de prendre en compte la position du point dans le processus de décision. Il suffit de suivre le cheminement de l'arbre en répondant aux questions 2.4.

Par exemple, la première intersection de l'arbre teste si $x_2 \geqslant 0.3$, la réponse est positive ($x_2 = 0.5$) et on se déplace donc vers le fils droit du nœud. En suivant le cheminement jusqu'à une racine de l'arbre, l'histogramme des couleurs est beaucoup plus discriminé et on obtient une probabilité a posteriori $P(y = \text{vert} \mid x) = 82\%$ que le point soit vert. On l'affecte donc à la classe verte. En effet, dans la cellule correspondante (figure de gauche), 82% des points appartiennent à la classe verte.

Parmi les nombreux avantages des arbres de décision, on citera notamment leur interprétabilité et leur faible coût computationnel.

2.1.4 Les Random Ferns

La méthode des *Random Ferns* a été introduite par Ozuysal et al. (2007) dans un papier traitant d'une problématique de reconnaissance d'images. Depuis lors, elle a été utilisée dans de nombreux travaux (Villamizar et al., 2012; Aniruddha et Babu, 2014). On peut la considérer comme une généralisation du classifieur bayésien naïf.

Le principe général consiste à relaxer l'hypothèse d'indépendance conditionnelle des descripteurs, au profit de l'hypothèse plus lâche, et plus réaliste en pratique, d'indépendance entre groupes de variables. Plus formellement, étant donnée une partition F de l'ensemble $[\![1; p]\!]$, avec $\text{card}(F) = d$, le nombre de groupes indépendants, l'hypothèse d'indépendance de groupes permet de réécrire 2.3 sous la forme modifiée :

$$P(Y|X) \propto P(Y) \prod_{k=1}^{d} P(\{X_i \ ; \ i \in F_k\}|Y), \tag{2.6}$$

où à nouveau, le facteur de proportionnalité ne dépend pas de Y et s'obtient par calcul de la loi marginale de X. Si $d = 1$ on retrouve l'expression générique de la loi jointe (intractable d'un point de vue numérique). À l'inverse, si $d = p$, chaque groupe ne contient qu'une variable, et l'expression 2.6 dégénère en un classifieur bayésien naïf du type 2.3. Pour cette raison, la méthode des Random Ferns est parfois qaulifée de *semi-naive bayes*, avec un compromis entre expressivité statistique des distributions modélisées et rapidité de calcul. En règle générale, le cardinal d de la partition est choisi de sorte à limiter les

effectifs des groupes à un nombre de descripteurs de l'ordre de 5 à 10.

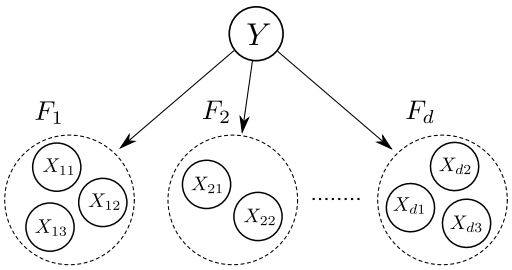

FIGURE 2.4 – Modèle d'indépendance conditionnelle entre groupes. On ne pose aucune hypothèse supplémentaire sur les distributions conditionnelles de F_k sachant Y.

Ayant posé le modèle 2.6 et l'entier d, il reste à définir la partition F. Ozuysal et al. (2007) proposent une stratégie de type fusion de prédicteurs *faibles*, en multipliant les probabilités postérieures fournies par un ensemble de modèles dont les partitions ont été tirées aléatoirement. Par exemple, pour L partitions générées $F^{(1)}, F^{(2)}, ...F^{(L)}$:

$$P(Y|X) = \frac{1}{Z} \prod_{l=1}^{L} P(Y) \prod_{k=1}^{d} P(\{X_i \ ; \ i \in F_k^{(l)}\}|Y), \tag{2.7}$$

où Z est une constante de normalisation permettant à la distribution de se sommer à 1.

Cette méthode de combinaison de prédicteurs faibles partiellement générés aléatoirement rapproche la méthode des Random Ferns des forêts d'arbres aléatoires.

2.1.5 Forêts d'arbres aléatoires

Le concept de stabilité, introduit empiriquement par Breiman et al. (1996), désigne la capacité d'un estimateur à produire des résultats similaires après de petites variations dans le jeu de données d'entraînement. Dans ce même papier, il est mis en évidence (analytiquement ou par simulations) que certains algorithmes, tels que les k-ppv, sont stables par nature, au contraire d'autres estimateurs, comme celui des arbres CART, introduit dans le paragraphe 2.1.3.

Pour les estimateurs instables, l'agrégation des résultats d'un grand nombre de prédicteurs individuels peut permettre d'améliorer les performances de prédiction (Breiman, 1996). Le concept est illustré de manière simplifiée par Friedman et al. (2001) : étant donnée une variable $\bar{x}$ calculée empiriquement par la moyenne d'un échantillon de n réalisations x_i identiquement distribuées (selon une loi quelconque de variance σ mais non-indépendantes (supposons un facteur de corrélation ρ). La variance de $\bar{x}$ s'exprime alors classiquement par :

$$\begin{aligned}\mathrm{Var}(\bar{x}) &= \frac{1}{n^2}\Big[\sum_{i\neq j}\mathrm{Cov}(x_i, x_j) + \sum_{i}\mathrm{Var}(x_i)\Big] \\ &= \Big(\rho + \frac{1-\rho}{n}\Big)\sigma^2.\end{aligned} \tag{2.8}$$

L'équation 2.8 montre que lorsque les prédicteurs individuels sont complètement corrélés, la variance de prédiction est limitée (en borne inférieure) par la valeur de σ^2. Pour un nombre n de prédicteurs fixés, la variance est rendue minimale par une valeur spécifique $\rho^* \in [0, 1]$. L'objectif des méthodes d'ensemble, ou méthodes de *stabilisation* selon la terminologie de Breiman (1996), est de décorréler légèrement les arbres de sorte à abaisser la valeur de ρ, idéalement de 1 à ρ^*.

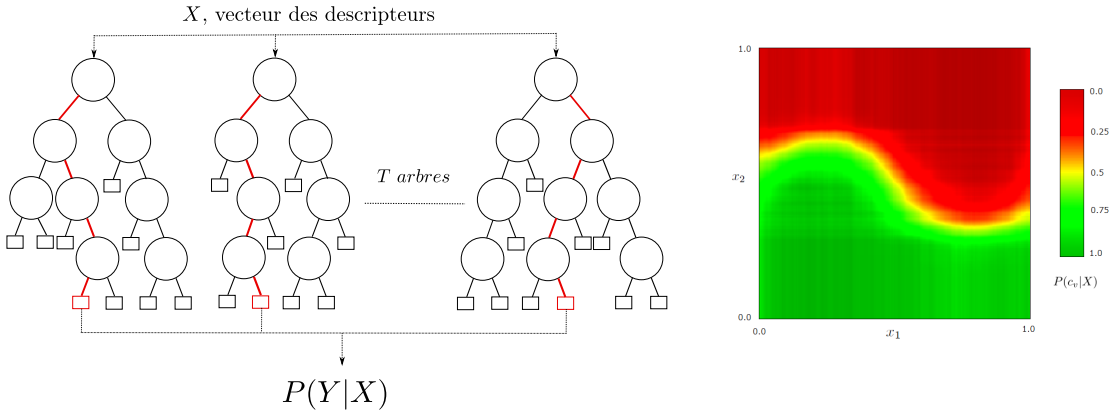

FIGURE 2.5 – À gauche : phase de prédiction du modèle de forêt aléatoire. À droite : résultat de l'inférence sur le problème modèle de classification illustré sur la figure 2.3.

Egalement introduites par Breiman (2001), les forêts aléatoires (*Random Forests*) constituent une version statistiquement robuste des arbres de décision, en s'appuyant au même titre que les Random Ferns sur les concepts de bootstrap statistique (Efron, 1992) et de méthode d'ensemble, pour réduire la variance de prédiction des arbres individuels. En revanche, avec cette *randomisation*, l'interprétabilité des arbres de décision est perdue.

L'idée centrale consiste à générer un nombre T (typiquement plusieurs centaines) d'arbres de décision dans lesquels on introduit une composante aléatoire à deux niveaux :

- À chaque construction d'un nouveau nœud dans l'arbre (equation 2.5), la coupe est réalisée dans le sous-espace vectoriel de $\mathbb{R}^p$ généré par un sous-ensemble de cardinal $m \leqslant p$ déterminé par tirage aléatoire sans remise dans les vecteurs de la base canonique. La littérature de référence (voir Breiman, 2001, par exemple) recommandent de choisir $m = \lceil\sqrt{p}\rceil$ pour les problèmes de classification et $m = \lceil p/3\rceil$ pour les problèmes de régression.

- Chaque arbre t est construit avec une version *bootstrap* $\mathcal{D}_n^t$ du jeu de données : plus formellement $\mathcal{D}_n^t$ contient n données $(X^{(i)}, Y^{(i)})$, échantillonnées aléatoirement et avec remise dans $\mathcal{D}_n$. Notons que rien n'interdit en pratique la présence de données en doublon dans chaque échantillon bootstrap.

Une fois que la collection de T arbres aléatoires a été construite, l'inférence sur une nouvelle donnée $X^{(n+1)}$ est réalisée en calculant les probabilité conditionnelles sur chaque arbre, puis en moyennant les probabilités obtenues, comme illustré sur la figure 2.5 :

$$P(Y^{(n+1)}|X^{(n+1)}) = \frac{1}{T}\sum_{t=1}^{T} P_t(Y^{(n+1)}|X^{(n+1)}). \tag{2.9}$$

On parle alors de *bootstrap aggregating* (ou en abrégé de *bagging*). Notons que cette méthode d'agrégation par la moyenne des probabilités (Bostrom, 2007) est préférable à celle du vote majoritaire lorsque l'on souhaite une estimation robuste de la probabilité a posteriori, bien que les résultats de classification soit sensiblements identiques in fine, quelle que soit la méthode employée (Breiman, 1996). Une solution alternative pourrait consister à calculer le produit (normalisé) des probabilités individuelles :

$$P(Y^{(n+1)}|X^{(n+1)}) = \frac{1}{Z}\prod_{t=1}^{T} P_t(Y^{(n+1)}|X^{(n+1)}), \tag{2.10}$$

$$\text{avec : } Z = \sum_{y\in\mathcal{Y}}\prod_{t=1}^{T} P_t(Y^{(n+1)} = y|X^{(n+1)}). \tag{2.11}$$

L'expression 2.10 serait optimale si les arbres étaient indépendants. Cependant, de par leur processus de génération, ce n'est en pratique jamais le cas, et l'expression 2.9 donne bien souvent de meilleurs résultats.

L'algorithme des forêts aléatoires présente deux intérêts notables :

- **Erreur OOB :** l'erreur *out-of-bag* (OOB) désigne une procédure de mesure de la performance du modèle sans nécessiter de jeu de données de test. L'étape de bootstrap implique que chaque instance du jeu d'entraînement a une probabilité $(1 - 1/n)^n$ de ne pas être sélectionné dans la base d'apprentissage d'un prédicteur individuel. Lorsque le nombre d'échantillons est suffisament grand, la proportion d'exemples non utilisés (échantillons out-of-bag) vaut :

$$\lim_{n\to\infty}\left(1 - \frac{1}{n}\right)^n = e^{-1} \approx 0.367. \tag{2.12}$$

  Autrement dit, chaque arbre individuel n'est construit qu'avec 63 % des données d'entraînement, ce qui permet d'utiliser les données restantes pour valider le modèle. On appelle *erreur OOB* le taux d'erreur empirique mesuré sur l'échantillon OOB. Il est généralement admis que cet estimateur est biaisé (Breiman, 2001; Mitchell, 2011), mais suffisant en pratique lorsque l'on souhaite une évaluation approximative de la

performance de classification, ou pour comparer plusieurs modèles opérant sur des sous-ensembles différents de descripteurs (Genuer et al., 2010).

- **Mesure d'importance des variables :** les forêts aléatoires offrent des méthodes simples pour estimer l'importance relative des descripteurs dans le processus de classification. On relève deux approches principales dans la littérature (Gregorutti, 2015). Dans une première méthode, l'importance d'une variable X_i donnée est calculée en fonction du gain en homogénéité des étiquettes des sous-arbres à chaque coupe dans laquelle elle intervient. En général, on prend le même critère d'homogénéité que celui qui a été utilisé pour la construction de l'arbre. Dans une deuxième approche plus empirique, on créé un jeu de données artificiel à partir du jeu d'origine, en permutant aléatoirement et pour i fixé, toutes les valeurs des vecteurs X en position i. On mesure alors l'importance de $\mathcal{I}(X_i) \in \mathbb{R}^+$ en évaluant la dégradation des performances (pour la fonction de perte $0-1$) :

$$\begin{aligned} \mathcal{I}(X_i) &= \mathbb{E}\big[|Y - f(\mathbf{x}_{(i)})|\big] - \mathbb{E}\big[|Y - f(\mathbf{x})|\big] \\ &= \mathbb{E}\big[|f(\mathbf{x}_{(i)}) - f(\mathbf{x})|\big], \end{aligned} \tag{2.13}$$

  où l'espérance est en pratique estimée empiriquement sur l'échantillon OOB et où $\mathbf{x}_{(i)} = (X_1, ..., \widetilde{X}_i, ...X_p)$ désigne un vecteur de descripteurs dans lequel $\widetilde{X}_i$ est une réplication indépendante et de même loi que X_i. La seconde égalité de 2.13 résulte du fait que la permutation aléatoire de X_i ne peut apporter d'information, impliquant $\mathcal{I}(X_i) \geqslant 0$ (avec égalité uniquement si X_i est complètement non-informative).

  Notons que Gregorutti (2015) a mis en évidence le lien existant entre la mesure d'importance par permutation aléatoire, et les indices de Sobol (Saltelli et al., 2000), démontrant ainsi que les coefficients calculé par 2.13 ont un fondement statistique commun à celui de l'analyse de sensibilité.

  Cette mesure d'importance est particulièrement utile dans les problèmes en grande dimension, pour lesquels on ne sait pas a priori quels descripteurs vont être informatifs dans le processus de décision.

Un intérêt subsidiaire des forêts aléatoires, est leur nombre d'hyper-paramètres relativement réduit, avec des règles de paramétrage empiriques établies par Breiman (2001) dans son papier fondateur.

La complexité du processus de constructions des arbres rend difficile l'établissement de résultats théoriques. On pourra trouver quelques garanties de convergence, moyennant quelques hypothèses simplificatrices, dans les travaux de Breiman (2004), Biau et al. (2008a) ou encore Scornet et al. (2015). En pratique, malgré ces limitations, les forêts aléatoires donnent en général de très bons résultats et ont été utilisées dans de nombreux problèmes concrets, par exemple dans le contexte du véhicule autonome (Zaklouta et al., 2011; Marin et al., 2013), en traitement automatique de la langue (Palomino-Garibay et al., 2015; DeBarr et Wechsler, 2009), en détection de fraudes bancaires (Liu et al., 2015) ou encore en gestion des risques naturels (Wang et al., 2015; Tesfamariam et Liu, 2010).

Pour plus de détails sur les forêts aléatoires, on pourra se référer au travail complet et détaillé de Louppe (2014) pour les aspects théoriques, ou encore à Criminisi et al. (2011) pour une large gamme d'applications pratiques.

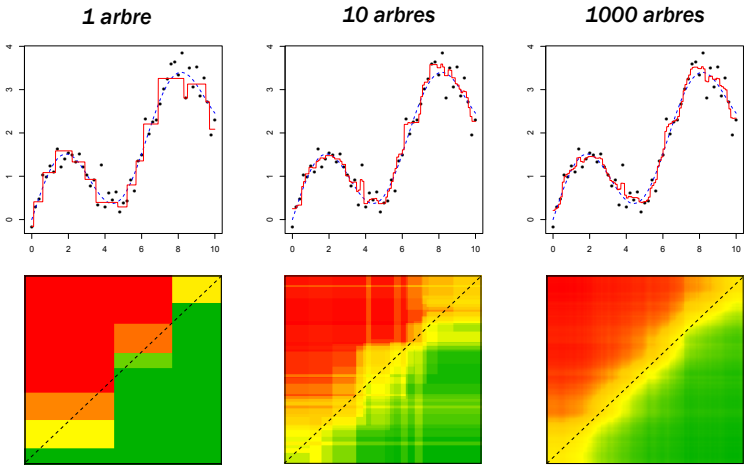

FIGURE 2.6 – Exemples d'application des forêts aléatoires pour différents effectifs d'arbres, sur le problème modèle de régression ?? (en haut) et sur le problème de classification $Y(X) = \mathbb{1}\{X_1 + \varepsilon \geqslant X_2\}$ où ε est une v.a. de ratio signal sur bruit égal à 7 (en bas). On remarque la robustesse de l'algorithme au sur-apprentissage (*cf* 2.2).

2.1.6 Les réseaux de neurones artificiels

Historiquement, les réseaux de neurones artificiels soint issus de la nécessité de modéliser de manière formelle le fonctionnement du cerveau pour tester des hypothèses biologiques simples. Le neurone formel, unité de base du modèle, prend en entrée un certains nombre de signaux, et calcule de manière déterministe une valeur de consigne, qui lui permet de s'activer au delà d'une valeur de seuil prédéterminée. Les poids numériques entrant en jeu dans le calcul sont paramétrables, de manière à simuler la plasticité synaptique du cerveau, et donc sa capacité d'apprentissage en fonction de l'environnement.

En parallèle, les réseaux de neurones peuvent être considérés comme un modèle d'apprentissage statistique qui permet une extension non-linéaire des modèles de régression classiques (Günther et Fritsch, 2010). Étant donné un vecteur de réels en entrée : $\mathbf{x}^i = (x_1^i, x_2^i, ...)$, dans une couche de niveau i du réseau, la sortie $\mathbf{x}^{i+1}$ s'évalue par :

$$x_k^{i+1} = f\Big(\omega_{0k}^i + \sum_j \omega_{jk}^i x_j^i\Big), \tag{2.14}$$

où ω_{jk}^i désigne le j-eme poids du k-eme neurone de la couche i, et f une fonction d'activation, en général non-linéaire. La sortie du réseau est désignée par $\mathbf{x}^n$, où n représente le nombre total de couches. Le modèle 2.15 est appelé perceptron multicouche.

Étant donné un jeu d'entraînement $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1..n}$ d'instances $\mathbf{x}_i$ étiquetées y_i (variable catégorielle ou continue), l'apprentissage statistique du modèle consiste à calculer les poids ω_{jk}^i du réseau, de sorte à minimiser un critère de coût du type 1.4. L'optimisation est en général effectuée par un algorithme itératif de descente de gradient, à l'aide des expressions de rétropropagation (Kelley, 1960), *i.e.* chaque instance $\mathbf{x}_i$ est passée dans le réseau (avec son paramétrage courant), et on compare le résultat obtenu $f(\mathbf{x}_i; \boldsymbol{\omega})$ avec l'étiquette associée y_i. L'erreur de classification (ou de régression) permet de propager des corrections à effectuer (avec un pas $\eta \in \mathbb{R}^+$, éventuellement variable au cours du temps), en *sens inverse*, sur les poids du réseau (voir voir Friedman et al., 2001 pour les détails) :

$$\forall\ i, j, k \qquad \omega_{jk}^{i(t+1)} = \omega_{jk}^{i(t)} - \eta(t)\ \frac{\partial L(y, f(\mathbf{x}; \boldsymbol{\omega}))}{\partial \omega_{jk}^i}. \tag{2.15}$$

Le théorème d'approximation universelle (voir par exemple voir Hornik et al., 1989 ou voir Cybenko, 1989, garantit que toute fonction continue peut être arbitrairement approchée (au sens de la limite uniforme) par un perceptron monocouche (*i.e.* constitué d'une unique couche de neurones cachés). Cette propriété théorique permet aux réseaux de neurones de partager les avantages des bases fonctionnelles passées en revue dans la section 2.3 (Conan-Guez, 2002).

Dans le domaine de la reconnaissance d'images et de signaux audio (entre autres), l'apprentissage profond (ou *deep learning* en anglais) consiste à multiplier le nombres de couches cachées, de manière à améliorer les performances de détection des algorithmes. En particulier, les réseaux de neurones convolutionnels (ou CNN, pour Convolutional Neural Network) consistent à disposer les poids synaptiques de sorte que chaque couche procéde à une convolution de la couche de niveau précédent afin de rechercher dans le signal des motifs de plus en plus complexes. Cette technique permet de partager les poids entre les neurones, et donc de réduire la dimension de la fonction à minimiser, permettant ainsi un gain considérable en temps de calcul, et une réduction du risque de sur-apprentissage des données, tout en conservant un fort potentiel expressif pour l'analyse de données spatialement structurées (Friedman et al., 2001; Guo et al., 2017). En ce sens, les CNN se comportent comme un modèle de régularisation statistique vis-à-vis de l'apprentissage profond classique. En régle général, un modèle classique de CNN, procède à une réduction spatiale de la taille du signal à analyser, en contre-partie d'une augmentation de sa profondeur sémantique.

Les CNN existent depuis les années 1980, mais leurs capacités d'apprentissage n'ont été que récemment mises en évidence avec l'avènement des processeurs graphiques à usage générique (GPGPU), permettant de paralléliser massivement les opérations de matricielles (Li et al., 2016). Depuis lors, les CNN sont utilisés dans de nombreux domaines, par exemple en diagnostique médical (Albarqouni et al., 2016), en cartographie à base d'images satellitaires (Postadjian et al., 2017), en recherche de victimes en montage (Bejiga et al., 2017), en traitement automatique de la langue (Jacovi et al., 2018) ou encore en détection de fraudes bancaires (Lv et al., 2019).

2.2 Le compromis biais-variance

Considérons un cas de problème de régression [2] avec une fonction de perte L quadratique. Le modèle inconnu à estimer est noté f, et on suppose que les descripteurs sont liés aux étiquettes par la relation non-déterministe $Y = f(X)+\varepsilon$ où ε est une variable aléatoire de loi inconnue, de moyenne nulle et de variance σ^2. Un choix naturel pour mesurer les performances d'un modèle $\hat{f}$ retourné par l'algorithme d'apprentissage consiste à évaluer l'espérance de la fonction de perte en un point $\mathbf{x}$ quelconque :

$$\begin{aligned}
\mathbb{E}[L(Y, \hat{f}(\mathbf{x}))] &= \mathbb{E}[(Y - \hat{f}(\mathbf{x}))^2] = \mathbb{E}[(Y - f(\mathbf{x}) + f(\mathbf{x}) - \hat{f}(\mathbf{x}))^2] \\
&= \mathbb{E}[\varepsilon^2] + \mathbb{E}\Big[\Big(f(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{x})] + \mathbb{E}[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})\Big)^2\Big] \\
&= \sigma^2 + \big[f(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{x})]\big]^2 + \mathbb{E}[(\hat{f}(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{x})])^2] \\
&= \text{Erreur de Bayes } + \text{ Biais}^2(\hat{f}(\mathbf{x})) + \text{Var}(\hat{f}(\mathbf{x})),
\end{aligned} \tag{2.16}$$

où les 3 termes de l'expression finale représentent respectivement :

- L'erreur incompressible (due à l'incertitude ε du phénomène à modéliser).
- Le biais du modèle, *i.e.* son erreur moyenne sur un grand nombre de prédictions
- La variance du modèle, *i.e.* sa tendance à reproduire des résultats différents à chaque nouvelle génération d'un jeu de données d'entraînement.

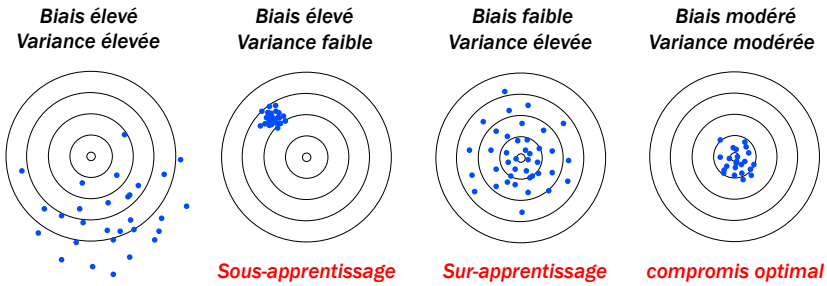

FIGURE 2.7 – Illustration schématique du compromis biais-variance dans le cadre de l'apprentissage statistique.

Dans la situation idéale où le phénomène inconnu est purement déterministe, $\sigma = 0$ et l'équation 2.16 devient :

$$\text{Erreur}(\hat{f}) = \text{Biais}^2(\hat{f}) + \text{Var}(\hat{f}). \tag{2.17}$$

2. Pour un problème de classification binaire dans l'espace des étiquettes $\mathcal{Y} = \{0, 1\}$, la perte quadratique et la perte 0-1 donnent des résultats identiques.

En pratique, plus un modèle est complexe, moins il est biaisé, mais plus sa variance est forte. On parle du *compromis biais-variance*. On pourra trouver des mises en évidences analytiques de ce compromis pour les régressions linéaires, pour les k plus proche voisins, ou encore sous certaines hypothèses réductrices pour les forêts aléatoires dans l'ouvrage de Friedman et al. (2001). On donne en figure 2.8 une illustration graphique du phénomène sur un cas de régression d'un échantillon de points par moindres carrés avec un polynôme de degré variable.

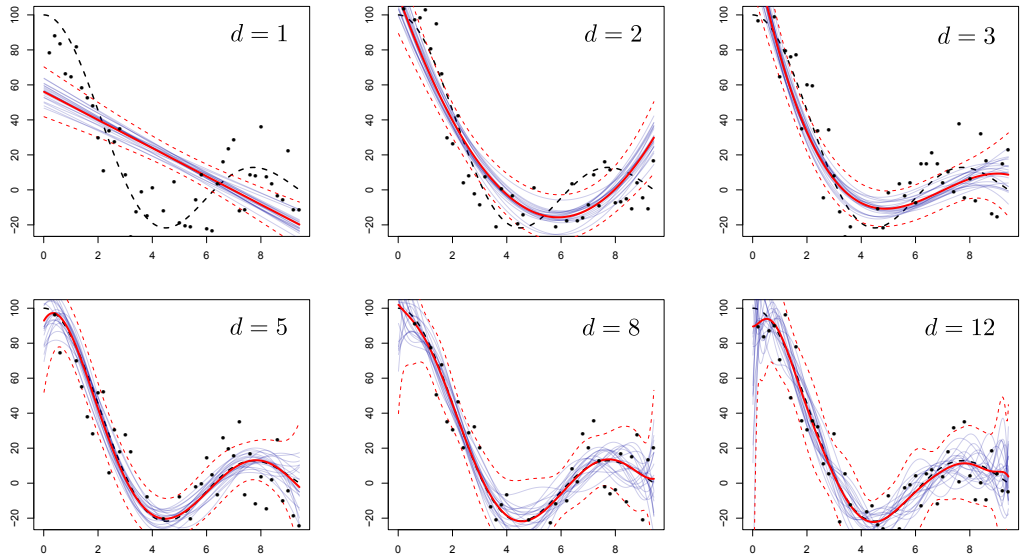

FIGURE 2.8 – Régression d'une fonction f inconnue (en pointillés noirs) par moindres carrés à l'aide de polynômes de degrés croissants d. À mesure que d augmente, l'écart entre l'espérance du modèle et la fonction f (le biais) tend à s'annuler tandis que la dispersion des courbes (la variance) augmente.

Dans la plupart des algorithmes d'apprentissage, un paramètre permet de régler ce compromis, par exemple la profondeur de l'arbre (pour les arbres de décision et les forêts aléatoires), ou encore le nombre k de voisins séléctionnés dans la méthode des k-ppv. La figure 2.9 illustre la nécessité de contrôler la qualité de l'apprentissage sur un jeu de données séparé (qu'on appelle jeu de données de test). En pratique, lorsqu'il faut régler des hyper-paramètres, on a besoin d'un troisième jeu de données pour se prémunir contre le risque de sur-apprentissage de ces hyper-paramètres : le jeu de validation. Certains algorithmes proposent parfois une régularisation a posteriori. C'est le cas par exemple de l'analyse en composantes principales (choix de la cascade de valeur propres), de la régression LASSO (sélection de variables), des arbres de décision (élagage des arbres) ou encore des réseaux de neurones artificiels (régularisation par suppression aléatoire de perceptrons).

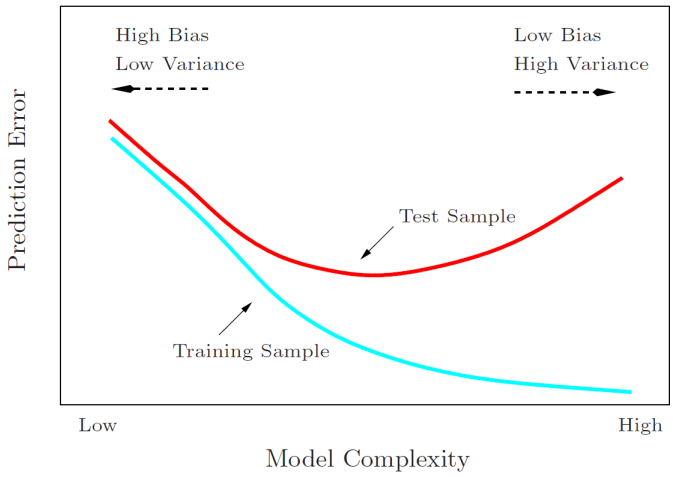

FIGURE 2.9 – Compromis biais-variance sur un modèle d'apprentissage, illustrant le principe d'overfitting. Source : Friedman et al. (2001)

2.3 Apprentissage de données fonctionnelles

Dans cette section nous passons en revue l'extension des algorithmes d'apprentissage au cas des données fonctionnelles. Dans tout ce qui suit, les données sont supposées être des fonctions de l'espace de Hilbert $L^2(\Omega)$, muni du produit scalaire usuel :

$$\forall\ f, g \in L^2(\Omega) \qquad \langle f, g \rangle_{L^2} = \int_\Omega f(x)g(x)dx, \tag{2.18}$$

où $\Omega \subseteq \mathbb{R}$ est un intervalle quelconque (pour la cas des fenêtres glissantes de profils de vitesse, on posera $\Omega = [0, 100]$).

Toute fonction de $L\check{s}(\Omega)$ peut être approchée à ε près par une suite d'éléments de l'ensemble $\mathcal{F}$ des fonctions étagées à valeurs rationnelles. L'ensemble $\mathcal{F}$ est donc dense dans $L\check{s}(\Omega)$. D'autre part, $\mathbb{Q}$ étant dénombrable, $\mathcal{F}$ l'est aussi, et par suite $L\check{s}(\Omega)$ est un espace de Hilbert séparable. Toute fonction f admet donc une représentation sous la forme suivante :

$$\forall\ x \in \Omega\ :\quad f(x) = \sum_{k=1}^{+\infty} \langle f, \varphi_k \rangle \varphi_i(x), \tag{2.19}$$

où $\{\varphi_k\ ; k \in \mathbb{N}\}$ désigne une base de $L^2(\Omega)$ orthogonale.

Dans le cadre de l'apprentissage statistique de données fonctionnelles, on utilise souvent une approche similaire à Gregorutti (2015) en projetant les données sur un sous-espace vectoriel de dimension p finie, ce qui revient à tronquer la série de fonctions 2.19 au rang p (moyennant un réordonnancement *ad hoc* des vecteurs de la base) :

$$\forall\ x \in \Omega\ :\quad f(x) = \sum_{i=1}^{p} \langle f, \varphi_i \rangle \varphi_i(x) + \varepsilon_p(x), \tag{2.20}$$

avec $(\varepsilon_p)_{p\in\mathbb{N}}$ une suite de fonctions d'erreur résiduelle dépendant du degré p de l'approximation et convergeant vers la fonction nulle.

On note alors (pour une subdivision suffisamment fine $\{\omega_k\ ;\ k = 1, .., N\}$ de Ω) :

$$X_i = \langle f, \varphi_i \rangle = \int_\Omega f(x)\varphi_i(x)dx \approx \sum_{k=1}^{N} f(\omega_k)\varphi_i(\omega_k), \tag{2.21}$$

et $X = (X_1, ..., X_p)$ devient un vecteur de descripteurs classique de $\mathbb{R}^p$ qui peut être passé en entrée de l'un des algorithmes de classification passés en revue au paragraphe 2.1.

Le théorème suivant montre l'intérêt de travailler dans une base orthogonale :

Théorème 1. Soient $\varphi_1, \varphi_2, ...\varphi_p$ des fonctions deux à deux orthogonales et de norme 1 d'un espace de Hilbert H. Soit F le sous-espace de H engendré les fonctions (φ_i) et $f \in H$ une fonction quelconque. Alors quels que soient $\lambda_1, \lambda_2, ...\lambda_p \in \mathbb{R}$ on a l'inégalité suivante :

$$\Big|\Big|f - \sum_{i=1}^{p} \langle f, \varphi_i \rangle \varphi_i\Big|\Big| \leqslant \Big|\Big|f - \sum_{i=1}^{p} \lambda_i \varphi_i\Big|\Big|, \tag{2.22}$$

avec égalité si et seulement si $\lambda_i = \langle f, \varphi_i \rangle$.

Le théorème 1 nous dit que la projection orthogonale de f sur le sous-espace F est le meilleur représentant (au sens de la norme L^2) de f parmi toutes les fonctions de F. Le vecteur des X_i défini par 2.21 est donc un descripteur naturel des fonctions à analyser.

Dans les sections suivantes, nous considérons plusieurs exemples classiques de bases fonctionnelles qui peuvent être employées à des fins d'apprentissage statistique.

2.3.1 Base de B-splines

Les fonctions splines, passées en revue dans la section **??**, désignent l'ensemble des fonctions polynomiales par morceaux sur une subdivision de Ω. Nous avons vu qu'elles correspondent exactement à l'espace des solutions d'un problème de minimisation de l'energie de flexion, permettant ainsi de donner une interprétation physique à cet ensemble.

La base des B-splines est un ensemble de fonctions à support compact permettant une décomposition des fonctions splines plus aisée (d'un point de vue numérique) qu'avec la base des puissances tronquées. On donne ci-dessous en figure 2.10 une illustration des vecteurs de cette base pour différents ordres m.

Excepté pour le cas des splines d'ordre 0, une base de B-splines n'est pas orthogonale, et la décomposition 2.19 n'est plus valide. L'estimation des coefficients n'est pas directe, et

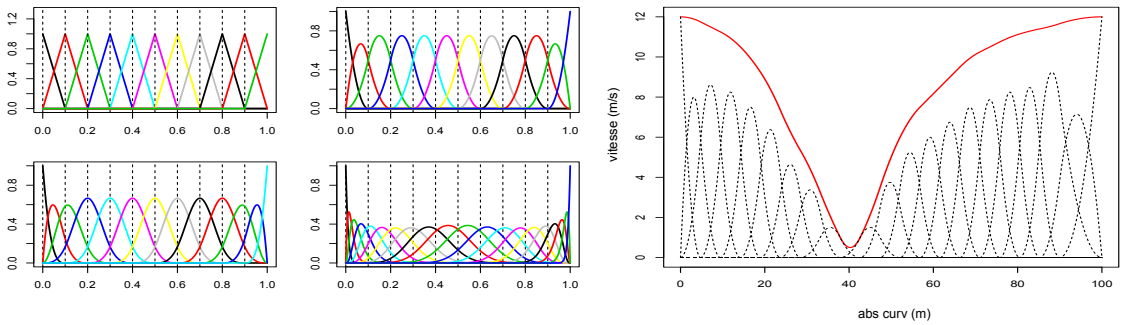

FIGURE 2.10 – À gauche : 4 bases de B-splines (d'ordre 1, 2, 3 et 10, de gauche à droite et de haut en bas). À droite : décomposition d'un profil spatial de vitesse GPS sur une base de B-splines cubiques.

doit se faire par inversion d'une système linéaire. Cependant, les fonctions de base sont à support compact, ce qui permet à la base B-splines de partager d'une certaine manière les avantages numériques de bases orthogonales (Ramsay et Silverman, 2005).

D'autre part, notons que la base des B-splines n'est pas une base de $L^2(\Omega)$. En revanche, le théorème de Stone-Weierstrass nous garantit que toute fonction continue sur un segment est limite uniforme d'une suite de fonctions polynomiales. En conséquence, les B-splines forment une base de l'espace préhilbertien $\mathcal{C}(\Omega)$. Pour un échantillonnage fin de Ω, la base des B-splines peut être utilisée comme une base des données d'entrée (sous réserve de définir a priori le nombre de coefficients nécessaire dans la troncature 2.20).

Une limitation principale des B-splines vient du fait que les fonctions de bases sont spatialement localisées, impliquant ainsi que l'information totale est a priori équitablement répartie dans les coefficients de base, rendant ainsi peu opérante la compression de données par troncature de la série de fonctions, ce qui dans le cadre de l'apprentissage peut poser des problèmes similaires aux données de grandes dimensions (section 1.0.2).

Notons qu'il existe de nombreux autres exemples de bases polynomiales : Legendre, Tchebychev, Laguerre, Hermite... qui constituent des bases de Hilbert des espaces préhilbertiens des fonctions continues et de carré intégrable pour une certaine fonction de poids. Pour plus d'informations sur le sujet, on pourra consulter l'ouvrage de Gilsinger et Jaï (2010).

2.3.2 Base de Fourier

Les théorèmes de convergence de Dirichlet nous enseignent que toute fonction f périodique $\mathcal{C}^1$ par morceaux peut s'écrire comme une somme infinie de polynômes trigonométriques (avec une convergence normale de la série de fonctions si f est de plus continue). On appelle spectre du signal f, la suite des coefficients de la décomposition de f en séries de Fourier. La version trigonométrique du théorème de Stone-Weierstrass garantit que toute fonction continue est limite uniforme d'une série de fonctions sinusoïdales.

La transformation de Fourier (TF) est une extension de ce résultat au cas des fonctions

non-périodiques, en considérant une valeur de pulsation ω asymptotiquement nulle (Cottet, 1997). Le spectre de f devient alors une fonction continue et il existe une expression intégrale analogue à 2.19. La transformée de Fourier est une fonction complexe dont le module $|\mathcal{F}[f](\omega)|$ indique la contribution des signaux de pulsation ω dans le signal original f. On peut montrer qu'il existe une application réciproque $\mathcal{F}^{-1}$ de forme très similaire à la TF directe, à un signe et une constante près, permettant de calculer la synthèse d'un ensemble infini de signaux de pulsations différentes. En ce sens, la TF opère une décomposition réversible d'une fonction donnée pour la représenter dans un espace fréquentiel.

Les données représentables en machine étant nécessairement de taille finie, il existe une version discrète de la TF, adaptée aux signaux échantillonnées : la transformation de Fourier discrète (TFD). Étant donné un N-échantillon, la TFD évalue N coefficients du spectre uniformément distribués entre la fréquence nulle et la moitié de la fréquence d'échantillonnage (en vertue du théorème de Shannon). Par exemple, la TFD d'un profil spatial de vitesse de résolution 1 m retourne 50 coefficients correspondants à la composante continue ainsi qu'à des fonctions de périodes étalées (en décroissance harmonique) entre 100 et 2 m.

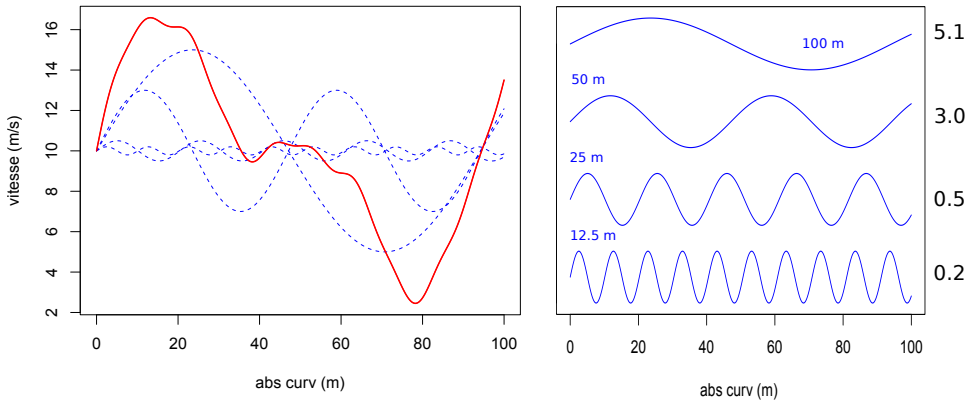

FIGURE 2.11 – Décomposition d'un profil de vitesse comme une somme de 4 fonctions sinusoïdales (de périodicités spatiales indiquées en bleu). Les coefficients à gauche représentent les coordonnées du profil dans la base fonctionnelle.

Toute fonction f de L^2 peut être décrite par la TFD avec une précision arbitrairement choisie en fonction du pas d'échantillonnage. De manière pratique, tout comme pour les splines, le choix de la base fonctionnelle doit être effectué en lien avec de l'erreur acceptable sur la troncature 2.20. En revanche, les decripteurs de Fourier se différencient des coefficients de B-splines dans le sens où ils sont complètement délocalisés dans le domaine spatial (l'information portée par chaque coefficient caractérise uniquement une bande spectrale de la fonction f). Cette limitation rend la TFD peu adaptée à notre cas d'étude.

La TFD possède les avantages notables d'offrir un fort taux de réduction de dimension (pour des signaux suffisamment réguliers) ainsi qu'une extension naturelle au plan complexe, permettant de traiter des courbes non-fonctionnelles. Cette propriété est utilisée en cartographie automatique pour l'analyse des formes baties (voir Bel Hadj Ali, 2001 par exemple).

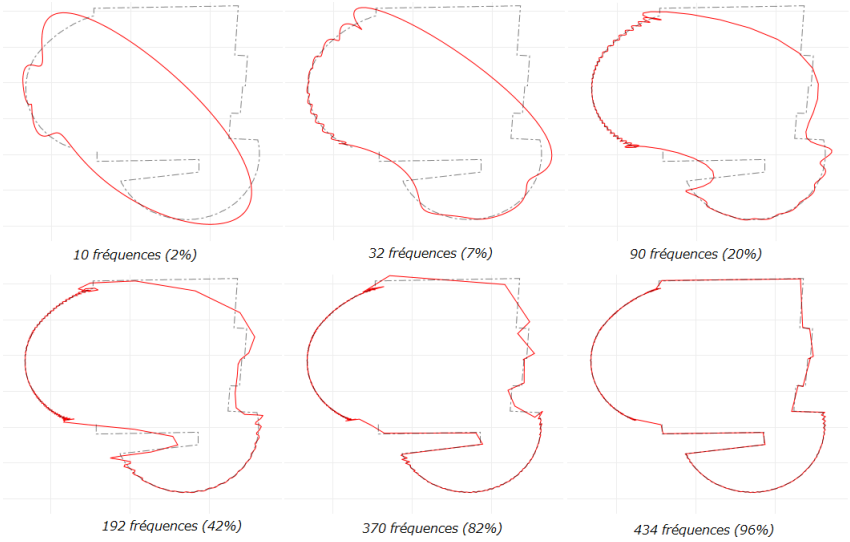

FIGURE 2.12 – Transformation de Fourier Discrète (TFD) d'un bâtiment.

2.3.3 Base d'ondelettes

Historiquement, la notion d'ondelettes est apparue du fait des limitations de la transformée de Fourier pour l'étude temporelle et spectrale de signaux (Chun-Lin, 2010). On sait que multiplication et produit de convolution sont analogues dans les deux espaces duaux, et toute analyse localisée du signal induit invariablement une dégradation du spectre calculé. En 1940, Gabor propose de moduler le signal à étudier par une gaussienne, réalisant ainsi le meilleur compromis possible en termes de résolutions dans le domaine joint temps-fréquence (Wei-lun, 2011). Le concept d'ondelettes sera introduit et théorisé dans les années 80 par Jean Morlet et Alex Grossman, en s'appuyant sur les travaux d'Alfred Haar (1909), dont le système d'ondelettes, restera jusqu'en 1985 la seule base connue d'ondelettes orthogonales. Quelques années plus tard, Ingrid Daubechies propose une méthode systématique de construction d'une base d'ondelettes orthogonales à support compact (Daubechies, 1988).

On considère un ensemble de sous-espaces imbriqués : $V_0 \subset V_1 \subset ... \subset L^2(\Omega)$ tels que pour tout j, il existe un espace W_j en somme directe orthogonale avec V_j dans V_{j+1} et engendré par une base de 2^j vecteurs orthonormaux : $\{\psi_{jk}\ ;\ k = 0, 1, ...2^j - 1\}$. L'espace W_j représente le niveau de détails manquant pour passer d'un niveau de résolution fonctionnelle au niveau plus fin suivant. Ces fonctions de base sont définies par des changements d'échelle et des translations d'une fonction ψ appelée *ondelette mère* : $\psi_{jk} = 2^{j/2}\psi(2^j x - k)$.

On peut montrer que l'ensemble des fonctions $\{\psi_{jk}\ ;\ j \in \mathbb{N},\ k = 0, 1, ...2^j - 1\}$ complété par une fonction φ appellée *ondelette père* (ou fonction d'échelle), forme une base orthonormale de $L^2(\Omega)$, avec comme implication directe que toute fonction f_l de $L^2(\Omega)$ peut s'écrire sous la forme :

$$\forall\ x \in \Omega \qquad f_l(x) = \omega_0^l \varphi(x) + \sum_{j=0}^{\infty} \sum_{k=0}^{2^j - 1} \omega_l^{jk} \psi_{jk}(x), \tag{2.23}$$

où les coefficients ω sont donnés par :

$$\omega_0^l = \int_\Omega f_l(x)\varphi(x)dx \qquad \text{et} \qquad \omega_{jk}^l = \int_\Omega f_l(x)\psi_{jk}(x)dx. \tag{2.24}$$

On peut réindicer les vecteurs de base en fonction du niveau de détails souhaité (Berlinet et al., 2008) : $\{\varphi_1, \varphi_2, ...\varphi_p\}$ et le vecteur de descripteurs $X^{(l)}$ d'une donnée fonctionnelle f_l s'écrit $(X_1^{(l)}, X_2^{(l)}, ...X_p^{(l)})$ avec :

$$\forall\ x \in \Omega \qquad f_l(x) = \sum_{i=1}^{p} X_p^{(l)}\varphi_i(x) + \varepsilon_l(x,p). \tag{2.25}$$

Le choix d'ondelettes le plus simple consiste à prendre le système de Haar (Haar, 1910), défini par une fonction d'échelle $\varphi = \mathbb{1}_{[0,1]}$ et une ondelette mère $\psi = \mathbb{1}_{[0,1/2]} - \mathbb{1}_{[1/2,1]}$. Notons qu'il s'agit d'un cas particulier des ondelettes de Daubechies (*cf* figure 2.13).

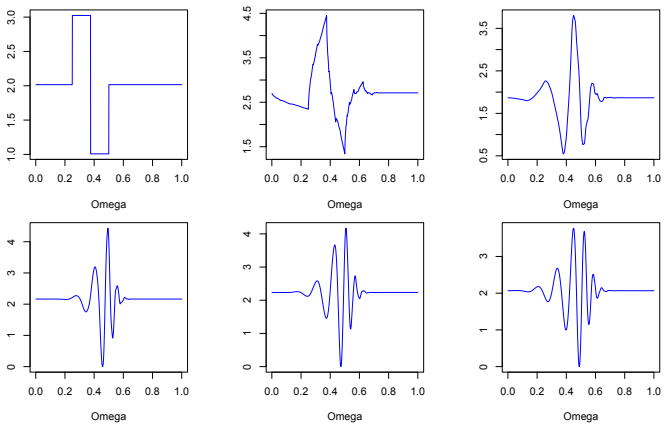

FIGURE 2.13 – Illustration de la forme des ondelettes mères (à différentes échelles) de Daubechies pour un nombre $m = 1, 2, 4, 6, 8$ et 10 de moments dissipants. La cas $m = 1$ (en haut à gauche) correspond au système de Haar.

Une pratique classique pour la réduction de dimension, consiste à ordonner les fonctions de base par niveau de résolution spatial, puis à tronquer la série après les vecteurs de l'espace V_j correspondant au niveau de détail souhaité. Notons qu'il existe d'autres méthodes plus fines, consistant à annuler les coefficients non-significatifs en fonction de différents seuils (déterminés par la théorie ou par validation croisée), donnant ainsi aux ondelettes un fort potentiel dans le domaine de l'estimation statistique non-paramétrique. Pour plus d'informations sur le sujet, on pourra se référer à Donoho et Johnstone (1994) ou encore à Nason (1995).

De par leur construction, les ondelettes réalisent le meilleur compromis entre localisations spatiale et fréquentielle (Berlinet et al., 2008). Elles sont donc pleinement adaptées au cas d'application de l'apprentissage, pour lequel on souhaite réduire la dimension des vecteurs de descripteurs à analyser, tout en conservant une information localisée (l'objectif *in fine* étant d'obtenir un géoréférencement précis de la signalisation routière). En ce sens, on peut

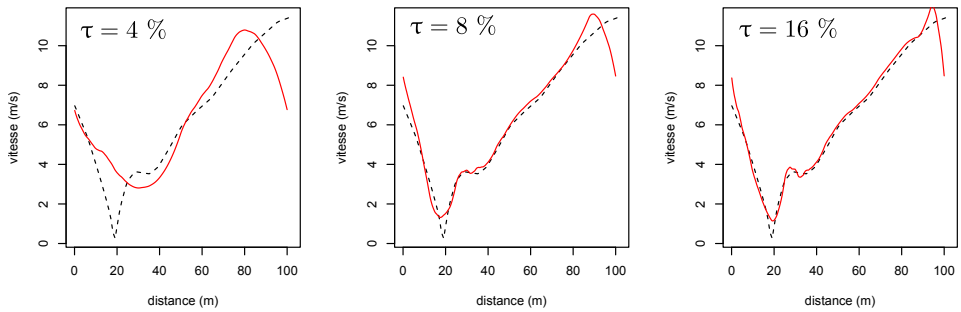

FIGURE 2.14 – Projection d'un profil spatial de vitesse sur l'espace généré par les 4, 8 et 16 premiers vecteurs de la base d'ondelettes de Daubechies d'ordre 4.

les situer entre les bases de splines et la base de Fourier. Elles sont fréquemment utilisées dans la litérature de la classification de signaux, par exemple en acoustique (Daniels, 2010; Morizet et al., 2016), en diagnostic médical (Aydemir et Kayikcioglu, 2011; Sumathi et al., 2014) et en reconnaisance d'images (Lotfi et al., 2009).

2.3.4 Base de Karhunen-Loève

La transformation de Karhunen-Loève (KL) est une généralisation fonctionnelle due initialement au travaux de Deville (1974) de l'analyse en composantes principales multivariée (Hotelling, 1933). Contrairement aux trois cas de bases fonctionnelles vues précédemment, la base de KL est déterminée à partir des données. Formellement, pour un processus stochastique du deuxième ordre X défini sur $[a, b] \subset \mathbb{R}$, de moyenne nulle et de fonction de covariance continue $K(.,.)$, les fonctions de bases φ_i sont les vecteurs propres de l'opérateur intégral de Hilbert-Schmidt de $L^2(\Omega)$:

$$(Af)(x) = \int_a^b K(x, \tau) f(\tau) d\tau. \tag{2.26}$$

On peut démontrer (voir Giambartolomei, 2015 par exemple) que les fonctions φ_i forment une base orthonormale de l'espace permettant d'écrire tout processus f de $L^2([a, b])$ sous la forme :

$$\forall\ x \in [a, b] \qquad f(x) = \sum_{i=1}^{+\infty} Z_i \varphi_i(x), \tag{2.27}$$

avec $Z_i = \langle f, \varphi_i \rangle$, une collection de variables aléatoires deux à deux orthogonales[3], de moyenne nulle et de variance égale à la valeur propre λ_i associée au vecteur de base φ_i. Cette propriété permet de trouver un ordonnancement naturel des fonctions de base, à

3. Pour cette raison, la décomposition KL est dite *doublement orthogonale* : les vecteurs de base (déterministes) sont orthogonaux dans l'espace des fonctions de carré intégrable ; les coefficients de la décomposition le sont dans l'espace des variables aléatoires du deuxième ordre.

partir de la fraction de variance totale expliquée :

$$\mathcal{I}(Z_i) = \frac{\lambda_i}{\sum_{k=1}^{\infty} \lambda_k}, \tag{2.28}$$

où le dénominateur est fini puisque le processus est supposé être du second ordre, et donc de variance intégrable sur l'intervalle $[a, b]$.

Théorème 2. Optimalité de la base de Karhunen-Loève

Étant donnée une base orthonormale $\boldsymbol{\phi} = \{\phi_k\}_{k\in\mathbb{N}}$ de $L^2([a, b])$, on note $\mathcal{E}_p(\boldsymbol{\phi})$ l'intégrale de l'espérance de l'erreur moyenne quadratique entre f et sa projection sur les p premiers vecteurs de la base $\boldsymbol{\phi}$ (on note A_i les coefficients de base) :

$$\mathcal{E}_p(f, \boldsymbol{\phi}) = \int_a^b \mathbb{E}\Big[\big(f(x) - \sum_{i=1}^{p} A_i\phi_i(x)\big)^2\Big]dx. \tag{2.29}$$

Alors, l'erreur $\mathcal{E}_p(f, \boldsymbol{\phi})$ est minimale si et seulement si les fonctions $\{\phi_i\}_{i=1,..,p}$ sont les p premières fonctions de la base de Karhunen-Loève dans l'ordre décroissant des valeurs propres. On pourra trouver la preuve dans Giambartolomei (2015).

Autrement dit, à dimension fixée, la décomposition dans une base KL fournit la meilleure approximation possible. Ce résultat s'explique par le fait que contrairement au cas de la TF ou des ondelettes, la base KL est *data-driven*, et donc naturellement plus optimale. Cette propriété est illustrée sur la figure 2.17 à la fin de cette section.

On donne ci-dessous un exemple d'apprentissage utilisant une projection de profils spatiaux de vitesse sur les 10 premières fonctions de base. On applique un algorithme de classification non-supervisé pour classifier les profils de vitesse dans une fenêtre comprenant un feu tricolore en séparant les trajectoires des véhicules marquant l'arrêt au feu des autres, la vérité terrain ayant été établie par annotation manuelle. Nous avons utilisé la technique des k-means dont l'objectif consiste à répartir les données en k courbes (k étant un paramètre fixé) de sorte à minimiser la variance des données au sein de chaque classe, formellement, on cherche à minimiser un terme de distorsion :

$$W(\mathbf{c}) = \mathbb{E}\Big[\min_{l=1,...|\mathbf{c}|} ||X - \mathbf{c}||_2\Big] \approx \frac{1}{n}\sum_{i=1}^{N} \min_{l=1,...|\mathbf{c}|} ||X_i - \mathbf{c}||_2, \tag{2.30}$$

où l'approximation de droite représente le terme empirique calculable. En réalité, le problème de la minimisation de 2.30 est NP-complet, et on utilise en pratique des méthodes itératives telles que l'algorithme de Lloyd (Lloyd, 1982), issue du domaine de la quantification numérique des signaux.

Des approches similaires ont été utilisées dans de nombreux travaux de classification non-supervisée de courbes : Abraham et al. (2003) par exemple utilisent une projection des données sur la base des B-splines ; Auder et Fischer (2012) comparent différentes stratégies en amont de l'application des k-means, notamment, via les descripteurs de Fourier (2.3.2),

les ondelettes de Haar (2.3.3), Karhunen-Loève (2.3.4), ainsi qu'une stratégie de sélection optimale au sein d'une famille de base d'ondelettes ; Barreyre et al. (2016) utilisent une approche similaire à la projection sur une base de Karhunen-Loève mais en substituant à la fonction de covariance, un noyau gaussien de paramètres *ad hoc* dans 2.26, la classification étant alors effectuée par un one-class-SVM. Enfin, dans une approche plus théorique, Biau et al. (2008b) analysent la performance des algorithmes de clustering dans un espace fonctionnel, par projection aléatoire sur un sous-espace vectoriel, en s'appuyant sur le lemme de Johnson–Lindenstrauss [4].

La méthode employée permet d'obtenir de bons résultats dans la classification des feux rouge / feux vert des profils. Dans cet exemple, la part de variance expliquée (équation 2.28) par la base tronquée s'élève à 99.997 %, ce qui permet une représentation très parcimonieuse des données fonctionnelles. Le taux de classifications correctes (évalué sur un ensemble représentatif de 15 fenêtres glissantes) avoisine les 99.6 %, contre 97.9 % avec la méthode par k-means sur les données représentées dans l'espace original.

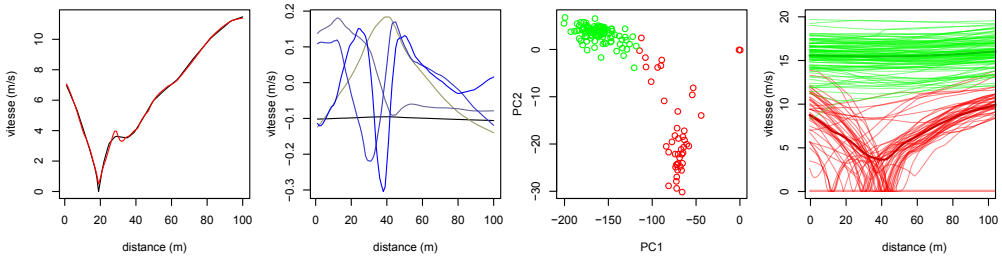

FIGURE 2.15 – De gauche à droite : 1) projection d'un profil sur les 10 premières fonctions de la base KL. 2) 5 premières fonctions de base. 3) Étiquettes inférées par l'algorithme des k-means représentées dans l'espace des 2 premières fonctions de base. 4) Résultat de la classification non-supervisée sur la séparation feu rouge / feu vert des profils.

La figure 2.15 (à droite) illustre également les fonctions centroïdes estimées par k-means [5].

La figure 2.16 fournit un autre intérêt notable de la transformation de KL dans un contexte de simulation. Nous avons remarqué que les vecteurs de la base KL formaient une rotation de la base canonique, dans laquelle les coefficients des fonctions sont entièrement décorrélés. La simulation d'un nouveau profil de vitesse peut être effectuée selon un processus décrit par Phoon et al. (2002). Cette possibilité de génération de données synthétiques offre de nombreuses opportunités, par exemple en simulation de trafic, en analyse de sensibilité ou encore pour l'équilibrage d'un jeu de données d'entraînement.

4. Pour les problèmes en grande dimension, pour tout ensemble de données, il existe une projection sur un sous-espace de dimension logarithmique en le nombre total de données et préservant approximativement (et avec une certaine probabilité) les distances entre les données.

5. Plus précisément, la projection sur les p premières fonctions de base étant par définition non-réversible, il s'agit de la moyenne d'ensemble des profils par classe dans l'espace original des courbes.

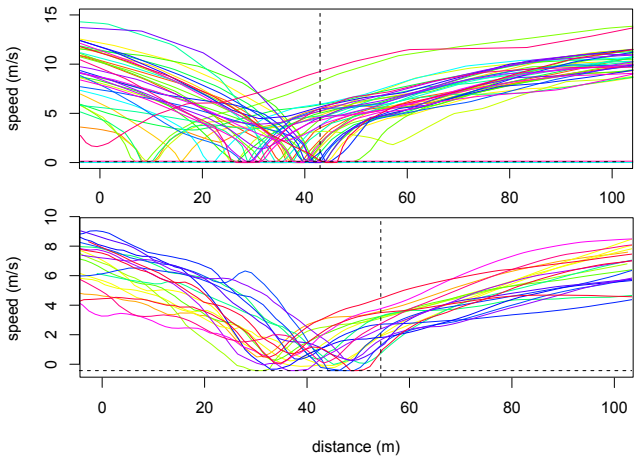

FIGURE 2.16 – En haut : profils de vitesse GPS de véhicules arretés au niveau d'un feu tricolore. En bas : génération synthétique de 20 profils de vitesse. Les positions des feux tricolores sont indiquées par les lignes verticales pointillées.

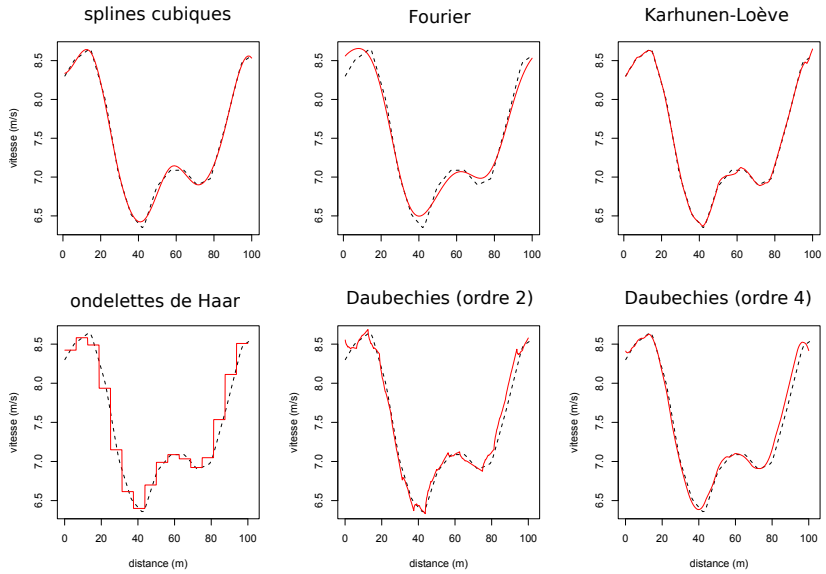

FIGURE 2.17 – Comparaison de plusieures bases fonctionnelles. Toutes les séries sont tronquées après le 16eme terme.

2.4 Évaluation d'un classifieur

Dans le cas d'un classifieur binaire (à 2 modalités de sortie), on peut utiliser les métriques suivantes pour évaluer la qualité d'apprentissage :

- La *sensibilité* (parfois appelé rappel), notée **STV** ci-après, définie par la probabilité de détecter une instance positive :

$$\mathbb{P}(\hat{Y} = 1|Y = 1) = \frac{TP}{TP + FN}, \tag{2.31}$$

où TP et FN représentent repectivement les nombre de vrais positifs et faux négatifs résultant du processus de classification. Cet indicateur mesure l'exhaustivité de la méthode de détection.

- La *spécificité*, notée **SPC**, mesure à l'inverse la probabilité du classifieur de ne pas détecter une instance négative :

$$\mathbb{P}(\hat{Y} = 0|Y = 0) = \frac{TN}{TN + FP}, \tag{2.32}$$

où TN et FP représentent repectivement les nombre de vrais négatifs et faux positifs.

- La *précision*, notée **PPV** (pour *positive predictive value*), quantifie le nombre de prédictions justes parmi les objects détectés positifs. On parle de mesure de précision *utilisateur* (a posteriori), par opposition aux deux indicateurs ci-dessus, qui sont qualifiés de précision *producteur* (a priori) :

$$\mathbb{P}(Y = 1|\hat{Y} = 1) = \frac{TP}{TP + FP}. \tag{2.33}$$

En particulier, à l'aide de la loi de Bayes, on peut réécrire la précision sous la forme :

$$PPV = \frac{\mathbb{P}(\hat{Y} = 1|Y = 1)\mathbb{P}(Y = 1)}{\mathbb{P}(\hat{Y} = 1)} \propto STV \times \frac{P}{TP + FP}, \tag{2.34}$$

mettant ainsi en évidence le fait que l'indicateur PPV dépend des proportions des classes, et ne fournit donc pas nécessairement une image représentative des performances du classifieur.

- La mesure F_1, notée **F1M**, est la moyenne harmonique des mesures de rappel et de précision :

$$\left(\frac{STV^{-1} + PPV^{-1}}{2}\right)^{-1} = \frac{2TP}{2TP + FP + FN}. \tag{2.35}$$

Le choix de la moyenne harmonique, plutôt que de la moyenne arithmétique, se justifie par une motivation naturelle à pénaliser plus fortement les classifieurs déséquilibrés. On pourra trouver des justifications plus théoriques et générales dans les travaux de Van Rijsbergen (1974), qui mettent en évidence le fait que la famille des mesures F_β (pour un paramètre de pondération relatif entre rappel et précision) est l'unique solution d'un ensemble de contraintes naturelles dans la recherche d'une métrique scalaire de l'évaluation des performances d'un classifieur binaire. La mesure

F_1 n'a toutefois pas d'interprétation intrinsèque en termes de valeur de probabilité, contraitrement à tous les autres indicateurs présentés dans cette section.

- La mesure d'*Accuracy*, notée *ACC*, désigne la probabilité pour le classifieur de retourner une prédiction correcte :

$$\mathbb{P}[\hat{Y} = Y] = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.36}$$

En remarquant qu'on peut réécrire le terme de gauche de 2.36 sous la forme :

$$\begin{aligned} \mathbb{P}[\hat{Y} = Y] &= \mathbb{P}[\hat{Y} = 1|Y = 1]\mathbb{P}[Y = 1] + \mathbb{P}[\hat{Y} = 0|Y = 0]\mathbb{P}[Y = 0] \\ &= STV \times \mathbb{P}[Y = 1] + SPC \times \mathbb{P}[Y = 0], \end{aligned} \tag{2.37}$$

on observe qu'à sensibilité et spécificité fixées, cet indicateur évolue avec les proportions d'instances de chaque classe dans la base de validation. De ce fait, la mesure d'accuracy n'est pas stable.

- Les mesures globales F_1 et accuracy, posent le problème d'être dépendantes de la proportion des étiquettes dans le jeu de données de validation. L'aire sous la courbe ROC, ou **AUC** (pour *area under curve*) permet de combler cette lacune.

Considérons un classifieur qui à chaque donnée $\mathbf{x} \in \mathcal{X}$ associe un score $f(\mathbf{x})$, d'autant plus élevé (resp. faible) que la donnée porte vraisemblablement l'étiquette 1 (resp. 0). À partir de cet unique classifieur, on peut construire une famille infinie de classifieurs dérivés $(f_t)_{t\in\mathbb{R}}$ en faisant varier le seuil de décision t :

$$\hat{Y} = f_t(\mathbf{x}) = \begin{cases} 1 & \text{si } f(\mathbf{x}) \geqslant t \\ 0 & \text{sinon.} \end{cases} \tag{2.38}$$

On appelle courbe ROC (pour *Receiver Operating Caracteristics*) le lieu des points $(x, y) = (1 - SPC(t), STV(t)) \in [0, 1]^2$ pour l'ensemble des seuils $t \in \mathbb{R}$. En considérant t comme une variable muette, l'aire AUC désigne alors l'intégrale sur $[0, 1]$ de la fonction qui à la variable $x = 1 - SPC$ associe $y = STV$.

On peut montrer que l'AUC est égale à $\mathbb{P}[f(X^{(1)}) \leqslant f(X^{(2)})]$, où $X^{(1)}$ et $X^{(2)}$ sont 2 instances échantillonnées aléatoirement et telles que $Y^{(1)} < Y^{(2)}$ (Hanley et McNeil, 1982). Notons qu'une AUC égale à 50 %, correspond à un classifieur purement aléatoire. Pour plus d'informations sur l'estimation et l'interprétation des courbes ROC, voir Gonçalves et al. (2014).

Chapitre 3

Apprentissage structuré

3.1 Introduction

Lorsque les instances d'apprentissage sont corrélées (en particulier par exemple, lorsqu'elles se chevauchent), nous sortons donc du cadre classique de l'apprentissage, tel que formulé dans la section 1.0.1. Par exemple, sachant que l'instance X_i est positive, la probabilité que l'instance voisine X_{i+1} le soit également peut être considérée comme plus faible ou à l'inverse plus élevée. En conséquence, on a vraisemblablement $P(Y_i|X_i, Y_{i+1}) \neq P(Y_i|X_i)$, d'où la nécessité d'inférer les variables Y_i et Y_{i+1} simultanément, ou du moins, de manière globale et cohérente.

Les techniques d'apprentissage collectif, ou *relational learning* (Lu et Getoor, 2003; Dhurandhar et Dobra, 2010) offrent plusieurs stratégies pour répondre à cette problématique. La méthode la plus simple consiste à diviser l'entraînement en deux blocs distincts :

- *L'apprentissage statique* (ou intrinsèque), qui cherche à inférer les variables cibles à partir des données connues, *i.e.* à partir des variables explicatives propres à l'instance concernée, mais aussi celles des instances de son voisinage ainsi que les variables cibles *connues* dans ce même voisinage.

- *L'apprentissage dynamique*, qui cherche à inférer les variables cibles à partir des variables cibles *inconnues* des instances de son voisnage.

La méthode dite de classification itérative (Neville et Jensen, 2000) consiste à itérer entre ces deux types de classification (à l'aide de deux modèles de classifieurs distincts) jusqu'à convergence des étiquettes de toutes les instances de la zone d'étude. D'autres stratégies existent, notamment à base d'échantillonneur de Gibbs (Geman et Geman, 1987), dont le paradigme fondé sur la génération itérative de variables à partir de données incomplètes (à chaque itération), le rend particulièrement adapté à l'apprentissage dynamique. De nombreuses variantes en découlent, dont celle de Chakrabarti et al. (1998), qui à chaque itération affecte une distribution de probabilité à chaque étiquette du modèle, plutôt qu'une affectation en dur.

En réalité, les premiers travaux d'apprentissage structuré remontent aux années 50 (Macskassy et Provost, 2007), et sont issues de la physique statistique, avec notamment la modélisation des phénomènes globaux composés d'une multitude d'intéractions locales. On citera

notamment les travaux d'Ising (1925) ainsi que l'extension proposée par Potts (1952), qui sont principalement utilisés pour dérire et trouver les configurations physiques d'énergie minimale de réseaux d'éléments à états discrets (comme par exemple les spins des électrons). Ces modèles appartiennent à la classe plus générale des champs de Markov (ou MRF pour *Markov Random Field*) et sont aujourd'hui intensivement utilisés dans tous les domaines où les prédictions doivent être effectuées sur un ensemble d'objets dont les positions spatiales (ou temporelles) entraînent des corrélations, comme par exemple dans le cas des pixels d'une image à segmenter (Kato, 1994), pour la classification de séries temporelles (Jebreen, 2017) ou encore pour la cartographie du risque en épidémiologie (Azizi, 2011). Les MRF sont eux-mêmes un cas particulier des modèles graphiques probabilistes (Koller et Friedman, 2009).

3.2 Les modèles graphiques probabilistes

Cette section constitue une brève présentation de la théorie générale des modèles graphiques. On pourra trouver plus de détails dans les ouvrages très complets de Wainwright et al. (2008), Koller et Friedman (2009) ou encore Sutton et al. (2012).

3.2.1 Les modèles dirigés

Définition 5.1. *On appelle réseau bayésien, ou modèle graphique dirigé (DAG pour Directed Acyclic Graph), un ensemble de variables aléatoires $X = \{X_i\}_{i\in\mathcal{I}}$, définies sur un espace $\mathcal{X}$, munit d'une structure de graphe $G(X, E)$ représentant les dépendances conditionnelles $p(X_i|\Pi(X_i))$, avec $\Pi(X_i) \subseteq X$ l'ensemble des nœuds antécédents de X_i dans G.*

Prenons un exemple de modèle simpliste, caractérisant une famille de lois sur 4 variables (à gauche sur la figure 3.1) : la présence de pluie (P), la présence d'un accident sur la chaussée (A), la présence d'un feu tricolore (F) et l'arrêt momentané du véhicule (S).

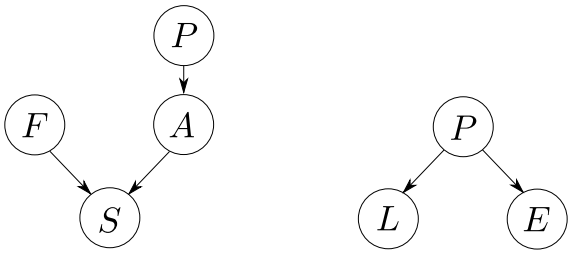

FIGURE 3.1 – Deux modèles graphiques probabilistes. P = Pluie, A = Accident, F = Feu tricolore, S = Arrêt du véhicule (stop), L = Allure lente, E = Essuie-glace.

Le modèle graphique indique les 2 points suivants :

- Les lois $p(A|P)$ (la pluie a une incidence sur la fréquence des accidents) et $p(S|A, F)$ (un arrêt du véhicule peut être dû, et de manière non exclusive, à la présence d'un

accident sur la chaussée, ou à un feu tricolore) modélisent les causalités du phénomène étudié.

- La loi jointe sur les 4 variables s'exprime sous la forme factorisée :

$$p(X) = p(\{A, F, P, S\}) = p(P)p(F)p(A|P)p(S|A, F)$$

D'une manière plus générale, la probabilité jointe s'écrit :

$$p(X) = \prod_{i \in \mathcal{I}} p(X_i|\Pi(X_i)), \tag{3.1}$$

avec la convention $p(X_i|\varnothing) = p(X_i)$ lorsque X_i n'admet pas d'antécédent dans G.

On vérifie aisément que la loi p est normalisée, c'est-à-dire que (en notant n le nombre de nœuds du graphe) :

$$\sum_{X \in \mathcal{X}^n} p(X) = 1. \tag{3.2}$$

Le modèle graphique permet de formaliser les indépendances de la loi de probabilité. Par exemple, sur le modèle de gauche de la figure 3.1, on peut lire que la présence d'un feu tricolore est indépendante de la météo, ou formellement : $p(F|P) = p(F)$. En effet, à l'aide de la factorisation, on écrit :

$$\begin{aligned} p(F|P) = \frac{p(F,P)}{p(P)} &= \frac{\sum_a \sum_s p(P)p(F)p(a|P)p(s|a,F)}{p(P)} \\ &= p(F) \sum_a \sum_s p(a|P)p(s|A,F) = p(F). \end{aligned}$$

De manière similaire, on peut facilement montrer des relations de dépendances conditionnelles. Par exemple, la probabilité de présence d'un feu tricolore sachant que le véhicule s'est arrêté est en général modifiée si on sait en plus qu'un accident est survenu sur la chaussée. Formellement : $p(F|S) \neq p(F|A, S)$. D'une certaine manière, la présence d'un accident diminue la probabilité de présence d'un feu, tandis que son abscence, au contraire, tend à l'augmenter. La connaissance de la conséquence commune S a donc brisé l'indépendance entre les causes potentielles F et A. On parle de mécanisme de type *explaining away*.

L'indépendance entre deux variables est donc une propriété qui dépend de l'observation d'autres variables. Par exemple, sur le second modèle de la figure 3.1, l'occurence de pluie est une cause probable commune à l'allure réduite des véhicules et à l'utilisation des essuie-glace. A l'aide de la forme factorisée 3.1, on montre facilement que nécessairement : $p(L|E, P) = p(L|P)$, tandis qu'en général $p(L|E) \neq p(L)$. Autrement dit, l'observation de la présence (ou abscence) de pluie, rend indépendantes l'utilisation des essuie-glace et l'adoption d'une allure réduite. En effet, n'ayant aucune information sur la météo, l'activation des essuie-glace augmente la probabilité de présence de pluie, qui en retour augmente

celle de l'allure réduite du véhicule. En revanche, lorsque les conditions météorologiques sont connues, l'utilisation des essuie-glace n'apporte aucune information supplémentaie sur l'allure du véhicule. Nous nous trouvons donc ici dans une situation inverse à celle du paragraphe précédent, dans lequel une nouvelle observation tendait à renforcer les dépendences entre les variables du voisinage.

De par leur interprétabilité en termes de relations causales, les DAG sont particulièrement utilisés dans le cadre des diagnositics (par exemple en médecine, ou encore en maintenance préventive). Comme nous l'avons vu dans les expérimentations menées aux chapitres 2 et 3, un second avantage indéniable de ces modèles réside dans leur efficacité d'entraînement, réduite à l'inférence [1] des lois conditionnelles intervenant dans le produit 3.1.

3.2.2 Les modèles non-dirigés

Si les graphes dirigés sont parfaitement adaptés pour représenter les relations causales, ils sont insuffisants lorsque la direction des dépendances entre variables est plus floue, notamment lorsqu'elle est inconnue, symétrique ou fluctuante dans le temps. Koller et Friedman (2009) cite par exemple le cas de 4 étudiants (nommons les $(X_i)_{i=1..4}$) ayant préparé un examen en groupe, chacun d'eux travaillant avec exactement 2 amis, comme illustré sur la figure 3.2 (à gauche). Dans ce cas, il est difficile de définir un lien de causalité entre les erreurs commises par les candidats à l'examen. D'autre part, notons que X_1 peut avoir une incompréhension d'un point quelconque du cours, qui se répercute sur X_2, qui le transmet à X_3, ce dernier le communiquant à X_4 qui à son tour la renforce chez X_1 et ainsi de suite. L'abscence de boucles dans les DAG (par définition, mais surtout pour que l'équation 3.1 comporte un nombre fini de termes), interdit ce genre de modélisation, d'où l'intérêt des modèles non-dirigés.

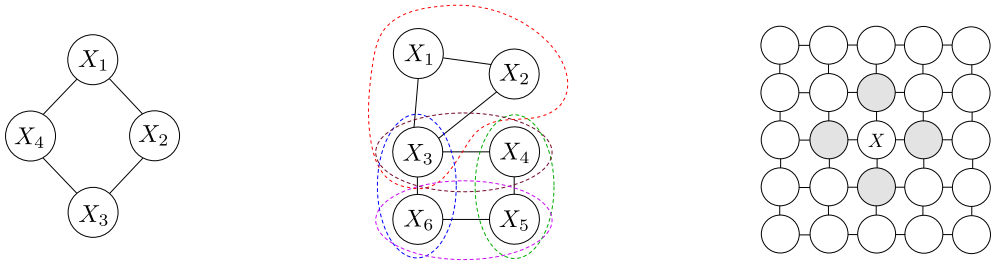

FIGURE 3.2 – Trois exemples de modèles graphiques non-dirigés. À gauche : le modèle des étudiants de Koller et Friedman (2009). Au centre : factorisation de la loi d'un champ de Markov en fonction des potentiels de clique. À droite : *couverture de Markov* d'un champ indiquant que la variable X est indépendante de toutes les autres variables du graphe dès lors que les cellules grisées ont été observées.

Ces modèles semblent également assez pertinents pour notre cadre de modélisation. La présence d'un feu tricolore sur un axe routier d'un carrefour n'implique pas (au sens causal du terme) la présence d'un feu sur les autres axes du carrefour. Mais la présence simultanée

1. Par inférence paramétrique pour les variables continues, et par tableau de contingence pour les variables discrètes.

d'un feu sur tous les axes (ou sur aucun) est plus probable qu'une configuration mixte. À l'inverse, et comme illustré sur la figure 3.3, la corrélation peut être négative, par exemple lorsque deux arcs se suivent, la présence d'un feu sur les deux arcs simultanément paraît moins probable que toutes les autres configurations.

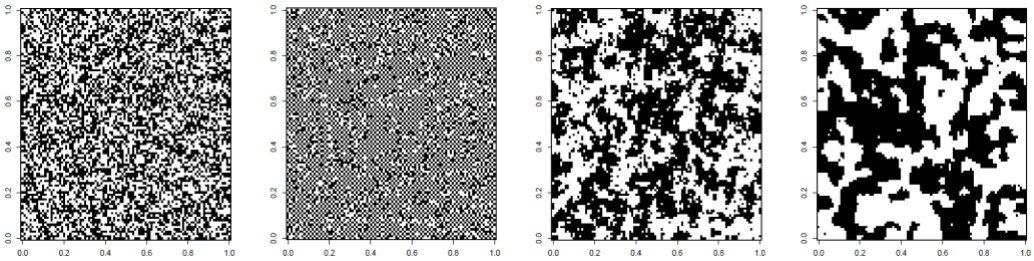

FIGURE 3.3 – Simulations d'un modèle d'Ising sur la grille $\mathbb{Z}^2$. À gauche : bruit blanc. Au centre gauche : champ négativement autocorrélé. Au centre droit : champ positivement et modérément autocorrélé. À droite : champ positivement autocorrélé.

Un autre intérêt fondamental de l'utilisation d'un modèle non-dirigé réside dans la lecture aisée des indépendances du modèle. Précisons un peu cette observation : nous avons vu dans le paragraphe précédent que les indépendances d'un DAG ne se déduisent pas immédiatement des voisinages du graphe. Certaines règles permettent de déterminer (à partir de la réprésentation graphique) si deux variables quelconques sont indépendantes, sachant une liste de variables observées (voir le concept de *Bayes Ball* de Shachter (2013)).

En revanche, dans le cas des modèles non-dirigés, et sous certaines conditions, le théorème suivant, dû à Hammersley et Clifford (1971), établit un lien direct entre les indépendances du modèle graphique, et la forme factorisée de sa loi jointe. Introduisons d'abord une définition formelle de champ de Markov.

Définition 5.2. *On appelle champ de Markov (ou MRF pour Markov Random Field), un modèle graphique G non-dirigé sur un ensemble X de variables aléatoires, tel que tout nœud y est indépendant des autres nœuds du graphe, sachant ses voisins. Formellement :*

$$p(X_i|X_{j\neq i}) = p(X_i|\mathcal{V}(X_i)), \tag{3.3}$$

où $\mathcal{V} : X \to 2^X$ est la fonction multivoque qui à un nœud $X_i \in X$ associe l'ensemble de ses voisins dans G.

Théorème 5.1 (Hammersley-Clifford). *Soit p une distribution de probabilité sur un graphe G non-dirigé, telle que $p(X) > 0$ pour toute affectation $X \in \mathcal{X}^n$. Alors G est un champ de Markov si et seulement si p admet la factorisation :*

$$p(X) = \frac{1}{Z} \prod_{c \in \mathcal{C}_G} \psi_c(X_c) \tag{3.4}$$

avec $Z = \sum_{\mathcal{X}} \prod_c \psi_c(X_c)$ *une constante de normalisation,* $\mathcal{C}_G$ *l'ensemble des cliques du graphe* G*,* $X_c \subseteq X$ *l'ensemble des nœuds de la clique* c *et* ψ_c *un ensemble de fonctions strictement positives, appelées potentiels de cliques.*

Une loi se factorisant sous la forme 3.4 est appelée une *distribution de Gibbs*. Le théorème de Hammersley-Clifford nous dit que, sur le sous-ensemble des distributions strictement positives, les champ de Markov représentent exactement les distributions de Gibbs.

La preuve du sens réciproque découle directement de la substitution de la loi factorisée 3.4 dans le terme de gauche de l'égalité 3.3 à démontrer. La démonstration du sens directe est nettement plus délicate, et on pourra en trouver une version dans Cheung (2008).

Notons que la condition de stricte positivité de p est importante, ce qui exclut les relations déterministes entre les variables. Un contre-exemple célèbre pourra être trouvé dans Moussouris (1974).

La figure 3.2 (au centre) donne un exemple de factorisation sur un ensemble de cliques. D'après le théorème 5.1, si $p(X) > 0$ sur $\mathcal{X}^n$, alors le graphe G définit un champ de Markov si et seulement $p(X) \propto \psi_{123}(x_1, x_2, x_3)\psi_{34}(x_3, x_4)\psi_{45}(x_4, x_5)\psi_{56}(x_5, x_6)\psi_{36}(x_3, x_6)$.

Un avantage pratique des distributions de Gibbs par rapport aux réseaux bayésiens réside dans la facilité de lecture des zones d'indépendance. On peut démontrer que les propriétés de Markov locale, globale et *pairwise* (Gandolfi et Lenarda, 2017) constituent trois notions équivalentes, et y sont donc nécessairement toutes vérifiées. Sur le modèle de droite de la figure 3.2, la propriété locale se traduit par l'indépendance conditionnelle de X à toutes les autres variables sachant les variables grisées (qui constituent la *couverture de Markov*, *i.e.* la zone dont l'observation permet une caractérisation probabiliste complète de la variable à inférer X, les autres variables plus *éloignées* étant alors rendues superflues).

Malgré tous ces avantages pratiques, contrairement aux réseaux bayésiens, les modèles dirigés sont clairement plus difficile à entraîner.

3.2.3 Les problèmes type

S'étant munit d'un modèle graphique (dirigé ou non) pré-entraîné (*i.e.* dont les lois conditionnelles ou les potentiels de cliques ont été inférés à l'aide d'un jeu d'entraînement), on dénombre trois problèmes principaux (Schmidt, 2007). Etant donné un ensemble (éventuellement vide) de nœuds observés dans, on peut chercher à :

- **Simuler** des réalisations de l'ensemble des variables aléatoires non observées.
- **Décoder** : calculer la configuration de variables la plus probable et qui coïncide avec les valeurs prises par les variables observées.

$$\hat{x} = \underset{x \in \mathcal{X}^n}{\operatorname{argmax}} \prod_{c \in \mathcal{C}_G} \psi_c(x_c). \tag{3.5}$$

Notons que la constante Z n'intervient pas explicitement dans ce problème.

- **Inférer** : estimer les probabilités marginales (conditionnelles aux observations) des variables (inconnues) du modèle.

$$p(X_s = x_s) = \frac{1}{Z} \sum_{\{x_t, t \neq s\}} \prod_{c \in \mathcal{C}_G} \psi_c(x_c). \tag{3.6}$$

En pratique, cette tâche se résume souvent au calcul de la constante Z. De plus, pour deux sous-ensembles disjoints de nœuds S et T, la loi marginale conditionnelle $p(X_T|X_S)$ se calcule par la définition classique :

$$p(X_T|X_S) = \frac{p(X_{S\cup T})}{p(X_S)}, \tag{3.7}$$

les deux membres de la fraction étant estimés à partir de l'équation 3.6.

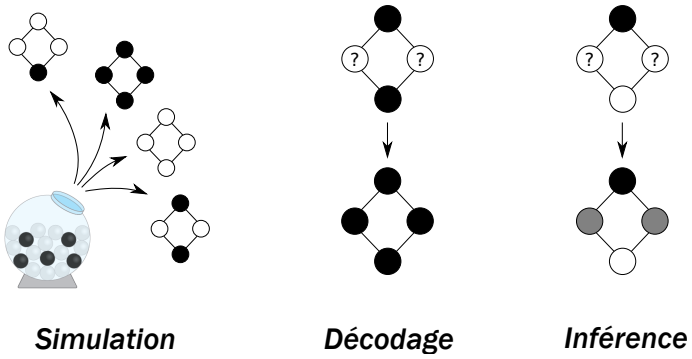

FIGURE 3.4 – Typologie des problèmes à résoudre avec un modèle graphique probabiliste (et étant donné éventuellement un sous-ensemble de variables observées). À gauche : simulation de réalisations distribuées suivant la loi du modèle. Au centre : calcul de la configuration la plus probable (*i.e.* le mode de la distribution). À droite : calcul des probabilités marginales. [Source du schéma de l'urne : Thomas Soell].

Dans notre cadre d'étude, le décodage permet de se placer du point du vue du gestionnaire, qui souhaite obtenir une cartographie détaillée et cohérente sur l'ensemble du réseau routier (par exemple pour mener à bien des travaux d'aménagement). À l'inverse, l'inférence présente un intérêt du point de vue du véhicule individuel. En un point local donné, nous pensons que la probabilité marginale de présence d'un élément de signalisation est plus pertinente que le mode global. En particulier, on cherche à affecter en chaque point le mode de la loi marginale (qui est en général localement différent du mode global).

En toute généralité, les trois problèmes présentés ci-dessus sont NP-difficiles (Koller et Friedman, 2009), indiquant ainsi qu'il n'existe pas d'algorithme permettant, pour toute instance de problème posé, de trouver une solution en un temps acceptable en pratique. Dans certains cas particuliers cependant, des algorithmes efficaces existent. Dans les autres

cas de figure, on a recours à des algorithmes d'approximation. Passons en revue les méthodes de résolution pour le problème de l'inférence.

Algorithmes de résolution

L'algorithme *Belief Propagation* (BP), aussi appelé *max-product* est une généralisation de l'algorithme de Viterbi (Yedidia et al., 2003). Le principe de l'algorithme consiste à transmettre à chaque nœud v un message contenant l'ensemble des informations nécesaires à la résolution du problème, collectées par les voisins de v sur chacun des sous-arbres partant de v, comme illustré sur la figure 3.5.

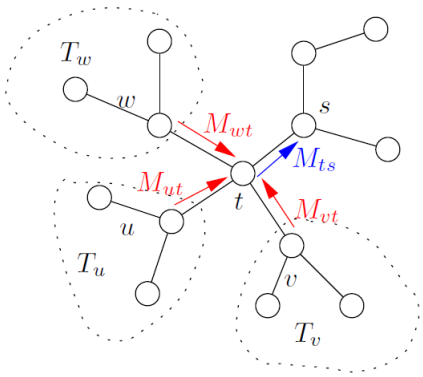

FIGURE 3.5 – Algorithme belief propagation. Source : Wainwright et al. (2008).

Il a été démontré que l'algorithme BP est exact pour des modèles de graphes en arbre (*i.e.* sans boucle). Notons que l'algorithme *max-product* permet de calculer le mode de la distribution, et donc de résoudre le problème de décodage. Il existe une version inférentielle de l'algorithme, appelée *sum-product*, dont la formulation est quasi-identique, pour calculer les lois marginales (Kschischang et al., 2001).

Pour des graphes avec boucles, plusieurs solutions alternatives ont été proposées :

- Lorsque la suppression d'un petit nombre de nœuds (dont les variables aléatoires associées sont à valeurs discrètes) permet d'obtenir une structure d'arbre, on peut adopter la stratégie des *cutsets* (Suermondt et Cooper, 1990) : on crée un graphe conditionné pour chaque affectation possible des variables des nœuds concernés. Le problème peut alors être résolu par l'algorithme *sum-product* indépendamment sur chaque arbre, puis les solutions sont moyennées pour obtenir l'inférence finale. Le nombre d'arbre a créer étant une fonction exponentielle du nombre de nœuds contenus dans le *cutset*, cette solution devient vite impracticable lorsque le nombre de nœuds à éliminer augmente.

- Dans le cas particulier où les variables aléatoires sont binaires et où les potentiels de cliques vérifient des propriétés de sous-modularité, les techniques de flots maximums issus de la théorie des graphe permettent de résoudre le problème de manière exacte

(Kolmogorov et Zabih, 2004).

- Lorsque le graphe possède des boucles, mais reste tout de même proche d'une structure d'arbre, ce qui est objectivement quantifié à l'aide de la notion de *largeur arborescente* (Kloks, 1994), on peut utiliser une technique de *Junction Tree* dans laquelle les variables individuelles sont groupées en *super-nœuds*, de sorte à ce que la structure globale du graphe soit un arbre. La technique d'inférence s'effectue alors par l'algorithme *sum-product*, en consdérant toutes les affectations possibles des variables dans les super-nœuds. La complexité de l'algorithme croît exponentiellement avec le nombre de variables par *super-nœuds* et n'est donc practicable que pour des graphes dont la largeur d'arborescente reste modérée. Voir Vats et Nowak (2014) par exemple, pour plus de détails.

Lorsque toutes ces solutions échouent, une alternative consiste à itérer l'algorithme BP, sans garanties de convergence, sur le graphe du modèle (Ihler et al., 2005). Cette méthode est appelée *Loopy Belief Propagation* (LBP).

D'autres algorithmes d'approximation peuvent être envisagés : méthodes empiriques itératives (Besag, 1986), méthodes de Monte-Carlo, méthodes variationnelles (Wainwright et al., 2008)...

3.2.4 L'apprentissage

L'apprentissage du modèle ainsi défini s'effectue par minimisation de la log-vraisemblance, en alternant une phase d'inférence et une phase de descente de gradient des paramètres θ du modèle. Un lemme bien pratique (voir Koller et Friedman (2009), p 947 par exemple) nous enseigne que la matrice hessienne de $\log Z(\theta)$ est égale à la matrice de covariance des statistiques exhaustives de la loi p. Par suite, elle est définie-positive, et $-\log Z(\theta)$ est donc une fonction concave, admettant un unique maximum, et garantissant la convergence de l'algorithme de descente de gradient vers une solution optimale.

Bibliographie

Abraham, C., Cornillon, P.-A., Matzner-Løber, E., et Molinari, N. (2003). Unsupervised curve clustering using b-splines. *Scandinavian journal of statistics*, 30(3) :581–595.

Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., et Navab, N. (2016). Aggnet : deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging*, 35(5) :1313–1321.

Aniruddha, A. K. et Babu, R. V. (2014). Visual object tracking via random ferns based classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6533–6537. IEEE.

Attal, F. (2015). *Classification de situations de conduite et détection des événements critiques d'un deux roues motorisé*. PhD thesis, Université Paris-Est.

Auder, B. et Fischer, A. (2012). Projection-based curve clustering. *Journal of Statistical Computation and Simulation*, 82(8) :1145–1168.

Aydemir, O. et Kayikcioglu, T. (2011). Wavelet transform based classification of invasive brain computer interface data. *Radioengineering*, 20(1) :31–38.

Azizi, L. (2011). *Champs aléatoires de Markov cachés pour la cartographie du risque en épidémiologie*. PhD thesis, Grenoble.

Barreyre, C., Laurent, B., Loubes, J.-M., Cabon, B., et Toulouse, I. (2016). Détection d'événements atypiques dans des données fonctionnelles. *Les journées de la Statistique*.

Bejiga, M., Zeggada, A., Nouffidj, A., et Melgani, F. (2017). A convolutional neural network approach for assisting avalanche search and rescue operations with uav imagery. *Remote Sensing*, 9(2) :100.

Bel Hadj Ali, A. (2001). *Qualité géométrique des entités géographiques surfaciques : Application à l'appariement et définition d'une typologie des écarts géométriques*. PhD thesis, Université de Marne-la-Vallée.

Berlinet, A., Biau, G., et Rouviere, L. (2008). Functional supervised classification with wavelets. In *Annales de l'ISUP*, volume 52, page 19.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society : Series B (Methodological)*, 48(3) :259–279.

Besse, P. (1979). *Etude descriptive d'un processus : Approximation et interpolation*. PhD thesis.

Biau, G., Devroye, L., et Lugosi, G. (2008a). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(Sep) :2015–2033.

Biau, G., Devroye, L., et Lugosi, G. (2008b). On the performance of clustering in hilbert spaces. *IEEE Transactions on Information Theory*, 54(2) :781–790.

Bostrom, H. (2007). Estimating class probabilities in random forests. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pages 211–216. IEEE.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2) :123–140.

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

Breiman, L. (2004). Consistency for a simple model of random forests.

Breiman, L. et al. (1996). Heuristics of instability and stabilization in model selection. *The annals of statistics*, 24(6) :2350–2383.

Breiman, L., Friedman, J., Stone, C. J., et Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Chakrabarti, S., Dom, B., et Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD Record*, volume 27, pages 307–318. ACM.

Cheung, S. (2008). Proof of hammersley-clifford theorem. *Unpublished, February*.

Choe, H. C., Karlsen, R. E., Gerhart, G. R., et Meitzler, T. J. (1996). Wavelet-based ground vehicle recognition using acoustic signals. In *Wavelet Applications III*, volume 2762, pages 434–446. International Society for Optics and Photonics.

Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*.

Conan-Guez, B. (2002). *Modélisation supervisée de données fonctionnelles par perceptron multi-couches*. PhD thesis, Université Paris Dauphine-Paris IX.

Cottet, F. (1997). Traitement des signaux et acquisition de données.

Criminisi, A., Shotton, J., et Konukoglu, E. (2011). Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114*, 5(6) :12.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4) :303–314.

Daniels, M. (2010). Classification of percussive sounds using wavelet-based. *CCRMA, Stanford University thesis*.

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7) :909–996.

DeBarr, D. et Wechsler, H. (2009). Spam detection using clustering, random forests, and active learning. In *Sixth Conference on Email and Anti-Spam. Mountain View, California*, pages 1–6. Citeseer.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6) :141–142.

Deville, J.-C. (1974). Méthodes statistiques et numériques de l'analyse harmonique. In *Annales de l'INSEE*, pages 3–101. JSTOR.

Dhurandhar, A. et Dobra, A. (2010). Collective vs independent classification in statistical relational learning. *Submitted for publication*.

Dietterich, T. G. et Bakiri, G. (1991). Error-correcting output codes : A general method for improving multiclass inductive learning programs. In *AAAI*, pages 572–577. Citeseer.

Do, T.-N., Lallich, S., Pham, N.-K., et Lenca, P. (2009). Un nouvel algorithme de forêts aléatoires d'arbres obliques particulièrement adapté à la classification de données en grandes dimensions. In *EGC*, pages 79–90.

Donoho, D. L. et Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3) :425–455.

Efron, B. (1992). Bootstrap methods : another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer.

Ferraty, F. et Vieu, P. (2006). *Nonparametric functional data analysis : theory and practice*. Springer Science & Business Media.

Flamary, R. (2011). *Apprentissage statistique pour le signal : applications aux interfaces cerveau-machine*. PhD thesis, Université de Rouen.

Friedman, J., Hastie, T., et Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA :.

Gandolfi, A. et Lenarda, P. (2017). A note on gibbs and markov random fields with constraints and their moments. *Mathematics and Mechanics of Complex Systems*, 4(3) :407–422.

Geman, S. et Geman, D. (1987). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision*, pages 564–584. Elsevier.

Genuer, R., Poggi, J.-M., et Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14) :2225–2236.

Giambartolomei, G. (2015). *The Karhunen-Loeve Theorem*. PhD thesis.

Gilsinger, J.-M. et Jaï, M. (2010). *Éléments d'analyse fonctionnelle : fondements et applications aux sciences de l'ingénieur*. PPUR Presses polytechniques.

Giraud, C. (2014). *Introduction to high-dimensional statistics*. Chapman and Hall/CRC.

Gonçalves, L., Subtil, A., Oliveira, M. R., et Bermudez, P. (2014). Roc curve estimation : An overview. *REVSTAT–Statistical Journal*, 12(1) :1–20.

Gregorutti, B. (2015). *Forêts aléatoires et sélection de variables : analyse des données des enregistreurs de vol pour la sécurité aérienne*. PhD thesis, Université Pierre et Marie Curie-Paris VI.

Gregorutti, B., Michel, B., et Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing*, 27(3) :659–678.

Günther, F. et Fritsch, S. (2010). neuralnet : Training of neural networks. *The R journal*, 2(1) :30–38.

Guo, Z., Chen, Q., Wu, G., Xu, Y., Shibasaki, R., et Shao, X. (2017). Village building identification based on ensemble convolutional neural networks. *Sensors*, 17(11) :2487.

Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3) :331–371.

Hammersley, J. M. et Clifford, P. (1971). Markov fields on finite graphs and lattices. *Unpublished manuscript*, 46.

Hanley, J. A. et McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1) :29–36.

Hornik, K., Stinchcombe, M., et White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5) :359–366.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6) :417.

Ihler, A. T., John III, W. F., et Willsky, A. S. (2005). Loopy belief propagation : Convergence and effects of message errors. *Journal of Machine Learning Research*, 6(May) :905–936.

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1) :253–258.

Jacovi, A., Shalom, O. S., et Goldberg, Y. (2018). Understanding convolutional neural networks for text classification. *arXiv preprint arXiv :1809.08037*.

Jebreen, K. (2017). *Modèles graphiques pour la classification et les séries temporelles*. PhD thesis, Aix-Marseille.

Kato, J. Z. (1994). *Modelisations markoviennes multiresolutions en vision par ordinateur. Application a la segmentation d'images SPOT*. PhD thesis, Nice.

Kelley, H. J. (1960). Gradient theory of optimal flight paths. *Ars Journal*, 30(10) :947–954.

Kloks, T. (1994). *Treewidth : computations and approximations*, volume 842. Springer Science & Business Media.

Koita, A., Daucher, D., et Fogli, M. (2013). New probabilistic approach to estimate vehicle failure trajectories in curve driving. *Probabilistic Engineering Mechanics*, 34 :73–82.

Koller, D. et Friedman, N. (2009). *Probabilistic graphical models : principles and techniques*. MIT press.

Kolmogorov, V. et Zabih, R. (2004). What energy functions can be minimizedvia graph cuts ? *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2) :147–159.

Kschischang, F. R., Frey, B. J., Loeliger, H.-A., et al. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2) :498–519.

Li, X., Zhang, G., Huang, H. H., Wang, Z., et Zheng, W. (2016). Performance analysis of gpu-based convolutional neural networks. In *2016 45th International Conference on Parallel Processing (ICPP)*, pages 67–76. IEEE.

Liu, C., Chan, Y., Alam Kazmi, S. H., et Fu, H. (2015). Financial fraud detection model : based on random forest. *International journal of economics and finance*, 7(7).

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2) :129–137.

Lotfi, M., Solimani, A., Dargazany, A., Afzal, H., et Bandarabadi, M. (2009). Combining wavelet transforms and neural networks for image classification. In *System Theory, 2009. SSST 2009. 41st Southeastern Symposium on*, pages 44–48. IEEE.

Loubes, J.-M., Maza, É., Lavielle, M., et Rodriguez, L. (2006). Road trafficking description and short term travel time forecasting, with a classification method. *Canadian Journal of Statistics*, 34(3) :475–491.

Louppe, G. (2014). Understanding random forests : From theory to practice. *arXiv preprint arXiv :1407.7502*.

Lu, Q. et Getoor, L. (2003). Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 496–503.

Lv, F., Wang, W., Wei, Y., Sun, Y., Huang, J., et Wang, B. (2019). Detecting fraudulent bank account based on convolutional neural network with heterogeneous data. *Mathematical Problems in Engineering*, 2019.

Macskassy, S. A. et Provost, F. (2007). Classification in networked data : A toolkit and a univariate case study. *Journal of machine learning research*, 8(May) :935–983.

Marin, J., Vázquez, D., López, A. M., Amores, J., et Leibe, B. (2013). Random forests of local experts for pedestrian detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2592–2599.

Mitchell, M. W. (2011). Bias of the random forest out-of-bag (oob) error for certain input parameters. *Open Journal of Statistics*, 1(03) :205.

Morizet, N., Godin, N., Tang, J., Maillet, E., Fregonese, M., et Normand, B. (2016). Classification of acoustic emission signals using wavelets and random forests : Application to localized corrosion. *Mechanical Systems and Signal Processing*, 70 :1026–1037.

Moussouris, J. (1974). Gibbs and markov random systems with constraints. *Journal of statistical physics*, 10(1) :11–33.

Nason, G. (1995). Choice of the threshold parameter in wavelet function estimation. In *Wavelets and statistics*, pages 261–280. Springer.

Neville, J. et Jensen, D. (2000). Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20.

Ozuysal, M., Fua, P., et Lepetit, V. (2007). Fast keypoint recognition in ten lines of code. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. Ieee.

Palomino-Garibay, A., Camacho-Gonzalez, A. T., Fierro-Villaneda, R. A., Hernandez-Farias, I., Buscaldi, D., Meza-Ruiz, I. V., et al. (2015). A random forest approach for authorship profiling. In *Proceedings of CLEF*.

Phoon, K., Huang, S., et Quek, S. (2002). Simulation of second-order processes using karhunen–loeve expansion. *Computers & structures*, 80(12) :1049–1060.

Postadjian, T., Le Bris, A., Sahbi, H., et Mallet, C. (2017). Investigating the potential of deep neural networks for large-scale classification of very high resolution satellite images. *ISPRS Annals*, 4 :183–190.

Potts, R. B. (1952). Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press.

Ramsay, J., Hooker, G., et Graves, S. (2009). *Functional data analysis with R and MATLAB*. Springer Science & Business Media.

Ramsay, J. O. et Dalzell, C. (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 539–572.

Ramsay, J. O. et Silverman (2005). *Functional Data Analysis*. Springer series in statistics.

Ramsay, J. O. et Silverman, B. W. (2007). *Applied functional data analysis : methods and case studies*. Springer.

Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.

Rogers, S. et Girolami, M. (2016). *A first course in machine learning*. CRC Press.

Saltelli, A., Chan, K., et Scott, E. (2000). Wiley series in probability and statistics. In *Sensitivity analysis*. Wiley.

Saporta, G. (1981). Méthodes exploratoires d'analyse de données temporelles. *Cahiers du bureau universitaire de recherche opérationnelle*, (37-38).

Schmidt, M. (2007). Ugm : A matlab toolbox for probabilistic undirected graphical models.

Scornet, E., Biau, G., Vert, J.-P., et al. (2015). Consistency of random forests. *The Annals of Statistics*, 43(4) :1716–1741.

Shachter, R. D. (2013). Bayes-ball : The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *arXiv preprint arXiv :1301.7412*.

Suermondt, H. J. et Cooper, G. F. (1990). Probabilistic inference in multiply connected belief networks using loop cutsets. *International Journal of Approximate Reasoning*, 4(4) :283–306.

Sumathi, S., Beaulah, H. L., et Vanithamani, R. (2014). A wavelet transform based feature extraction and classification of cardiac disorder. *Journal of medical systems*, 38(9) :98.

Suquet, C. (2003). Lois des grands nombres.

Sutton, C., McCallum, A., et al. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4) :267–373.

Tesfamariam, S. et Liu, Z. (2010). Earthquake induced damage classification for reinforced concrete buildings. *Structural safety*, 32(2) :154–164.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Van Rijsbergen, C. J. (1974). Foundation of evaluation. *Journal of documentation*, 30(4) :365–373.

Vats, D. et Nowak, R. D. (2014). A junction tree framework for undirected graphical model selection. *The Journal of Machine Learning Research*, 15(1) :147–191.

Villamizar, M., Garrell, A., Sanfeliu, A., et Moreno-Noguer, F. (2012). Online human-assisted learning using random ferns. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2821–2824. IEEE.

Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2) :1–305.

Wang, Z., Lai, C., Chen, X., Yang, B., Zhao, S., et Bai, X. (2015). Flood hazard risk assessment model based on random forest. *Journal of Hydrology*, 527 :1130–1141.

Wei-lun, C. (2011). Gabor wavelet transform and its application. *R98942073*.

Wohlfarth, T. (2013). *Machine-learning pour la prédiction des prix dans le secteur du tourisme en ligne*. PhD thesis, Télécom ParisTech.

Yedidia, J. S., Freeman, W. T., et Weiss, Y. (2003). Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8 :236–239.

Zaklouta, F., Stanciulescu, B., et Hamdoun, O. (2011). Traffic sign classification using kd trees and random forests. In *The 2011 International Joint Conference on Neural Networks*, pages 2151–2155. IEEE.

Zivot, E. (2009). Maximum likelihood estimation. *Lecture Notes on course" Econometric Theory I : Estimation and Inference (first quarter, second year PhD)", University of Washington, Seattle, Washington, USA*.