Machines à vecteurs supports

Résumé

Recherche d'un hyperplan, dit de marge optimale (vaste), pour la séparation de deux classes dans un espace hilbertien défini par un noyau reproduisant associé au produit scalaire de cet espace. Estimation de l'hyperplan dans le cas linéaire et séparable ; les contraintes actives du problème d'optimisation déterminent les vecteurs supports. Extension au cas non linéaire par plongement dans un espace hilbertien à noyau reproduisant. Extension au cas non séparable par pénalisation.

Retour au plan du cours

1 Introduction

Les *Support Vector Machines* souvent traduit par l'appellation de Séparateur à Vaste Marge (SVM) sont une classe d'algorithmes d'apprentissage initialement définis pour la discrimination c'est-à-dire la prévision d'une variable qualitative initialement binaire. Ils ont été ensuite généralisés à la prévision d'une variable quantitative. Dans le cas de la discrimination d'une variable dichotomique, ils sont basés sur la recherche de l'*hyperplan de marge optimale* qui, lorsque c'est possible, classe ou sépare correctement les données tout en étant le plus éloigné possible de toutes les observations. Le principe est donc de trouver un classifieur, ou une fonction de discrimination, dont la capacité de généralisation (qualité de prévision) est la plus grande possible.

Cette approche découle directement des travaux de Vapnik en théorie de l'apprentissage à partir de 1995. Elle s'est focalisée sur les propriétés de généralisation (ou prévision) d'un modèle en contrôlant sa complexité. Voir à ce sujet la vignette sur l'estimation d'un risque et la section introduisant la dimension de Vapnik-Chernovenkis comme indicateur du pouvoir séparateur d'une famille de fonctions associé à un modèle et qui en contrôle la complexité. Le principe fondateur des SVM est justement d'intégrer à l'estimation le contrôle de la complexité c'est-à-dire le nombre de paramètres qui est associé dans ce cas au nombre de vecteurs supports. L'autre idée directrice de Vapnik dans ce développement, est d'éviter de substituer à l'objectif initial : la discrimination, un ou des problèmes qui s'avèrent finalement plus complexes à résoudre comme par exemple l'estimation non-paramétrique de la densité d'une loi multidimensionnelle en analyse discriminante.

Le principe de base des SVM consiste de ramener le problème de la discrimination à celui, linéaire, de la recherche d'un hyperplan optimal. Deux idées ou astuces permettent d'atteindre cet objectif :

- La première consiste à définir l'hyperplan comme solution d'un problème d'optimisation sous contraintes dont la fonction objectif ne s'exprime qu'à l'aide de produits scalaires entre vecteurs et dans lequel le nombre de contraintes "actives" ou vecteurs supports contrôle la complexité du modèle.
- Le passage à la recherche de surfaces séparatrices non linéaires est obtenu par l'introduction d'une fonction noyau (*kernel*) dans le produit scalaire induisant implicitement une transformation non linéaire des données vers un espace intermédiaire (*feature space*) de plus grande dimension. D'où l'appellation couramment rencontrée de machine à noyau ou *kernel machine*. Sur le plan théorique, la fonction noyau définit un espace hilbertien, dit auto-reproduisant et isométrique par la transformation non linéaire de l'espace initial et dans lequel est résolu le problème linéaire.

Cet outil devient largement utilisé dans de nombreux types d'application et s'avère un concurrent sérieux des algorithmes les plus performants (agrégation de modèles). L'introduction de noyaux, spécifiquement adaptés à une problématique donnée, lui confère une grande flexibilité pour s'adapter à des situations très diverses (reconnaissance de formes, de séquences génomiques, de caractères, détection de spams, diagnostics...). À noter que, sur le plan algorithmique, ces algorithmes sont plus pénalisés par le nombre d'observations, c'est-à-dire le nombre de vecteurs supports potentiels, que par le nombre de variables. Néanmoins, des versions performantes des algorithmes permettent de prendre en compte des bases de données volumineuses dans des temps de calcul acceptables.

Le livre de référence sur ce sujet est celui de Schölkopf et Smola (2002)[2]. De nombreuses introduction et présentations des SVM sont accessibles sur des sites comme par exemple : `www.kernel-machines.org`. Guermeur et Paugam-Moisy (1999)[1] en proposent une en français.

2 Principes

2.1 Problème

Comme dans toute situation d'apprentissage, on considère une variable Y à prédire mais qui, pour simplifier cette introduction élémentaire, est supposée dichotomique à valeurs dans $\{-1, 1\}$. Soit $\boldsymbol{X} = X^1, \ldots, X^p$ les variables explicatives ou prédictives et $\phi(\boldsymbol{x})$ un modèle pour Y, fonction de $\boldsymbol{x} = \{x^1, \ldots, x^p\} \in \mathbb{R}^p$. Plus généralement on peut simplement considérer la variable $\boldsymbol{X}$ à valeurs dans un ensemble $\mathcal{F}$.

On note

$$\boldsymbol{z} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$$

un échantillon statistique de taille n et de loi F inconnue. L'objectif est donc de construire une estimation $\widehat{\phi}$ de ϕ, fonction de $\mathcal{F}$ dans $\{-1, 1\}$, de sorte que la probabilité :

$$P(\phi(\boldsymbol{X}) \neq Y)$$

soit minimale.

Dans ce cas (Y dichotomique), le problème se pose comme la recherche d'une frontière de décision dans l'espace $\boldsymbol{\mathcal{F}}$ des valeurs de $\boldsymbol{X}$. De façon classique, un compromis doit être trouvé entre la *complexité* de cette frontière, qui peut s'exprimer aussi comme sa capacité à *pulvériser* un nuage de points par la VC dimension, donc la capacité d'*ajustement* du modèle, et les qualités de *généralisation* ou prévision de ce modèle. Ce principe est illustré par la figure 1.

2.2 Marge

La démarche consiste à rechercher, plutôt qu'une fonction $\widehat{\phi}$ à valeurs dans $\{-1, 1\}$, une fonction réelle f dont le signe fournira la prévision :

$$\widehat{\phi} = \text{signe}(f).$$

L'erreur s'exprime alors comme la quantité :

$$P(\phi(\boldsymbol{X}) \neq Y) = P(Y f(\boldsymbol{X}) \leq 0).$$

De plus, la valeur absolue de cette quantité $|Y f(\boldsymbol{X})|$ fournit une indication sur la confiance à accorder au résultat du classement.

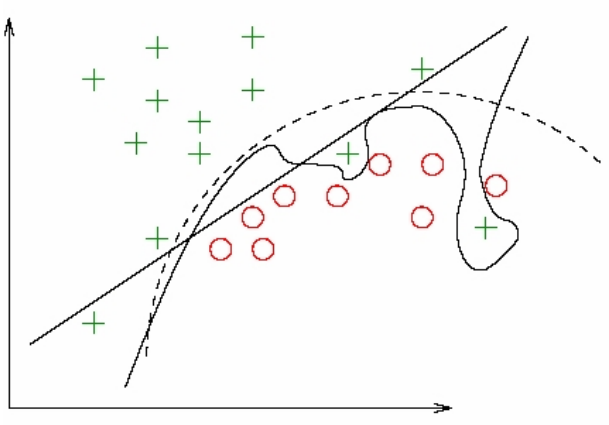

FIGURE 1 – Sous-ajustement linéaire et sur-ajustement local (proches voisins) d'un modèle quadratique.

On dit que $Y f(\boldsymbol{X})$ est la *marge* de f en $(\boldsymbol{X}, Y)$.

2.3 Espace intermédiaire

Une première étape consiste à transformer les valeurs de $\boldsymbol{X}$, c'est-à-dire les objets de $\mathcal{F}$ par une fonction Φ à valeurs dans un espace $\mathcal{H}$ intermédiaire (*feature space*) muni d'un *produit scalaire*. Cette transformation est fondamentale dans le principe des SVM, elle prend en compte l'éventuelle non linéarité du problème posé et le ramène à la résolution d'une séparation linéaire. Ce point est détaillé dans une section ultérieure. Traitons tout d'abord le cas linéaire c'est-à-dire le cas où Φ est la fonction identité.

3 Séparateur linéaire

3.1 Hyperplan séparateur

La résolution d'un problème de séparation linéaire est illustré par la figure 2. Dans le cas où la séparation est possible, parmi tous les hyperplans solutions pour la séparation des observations, on choisit celui qui se trouve le plus "loin"

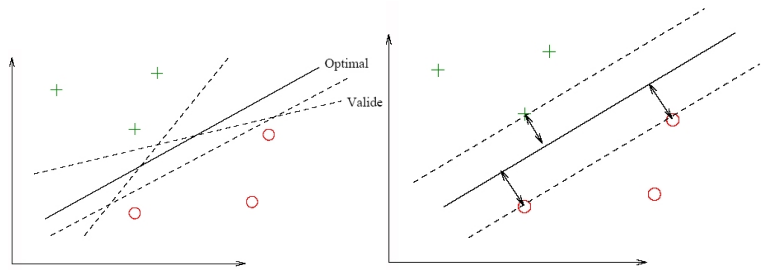

FIGURE 2 – Recherche d'un hyperplan de séparation optimal au sens de la marge maximale.

possible de tous les exemples, on dit encore, de *marge* maximale.

Dans le cas linéaire, un hyperplan est défini à l'aide du produit scalaire de $\mathcal{H}$ par son équation :

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0$$

où $\boldsymbol{w}$ est un vecteur orthogonal au plan tandis que le signe de la fonction

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$$

indique de quel côté se trouve le point $\boldsymbol{x}$ à prédire. Plus précisément, un point est bien classé si et seulement si :

$$yf(\boldsymbol{x}) > 0$$

mais, comme le couple $(\boldsymbol{w}, b)$ qui caractérise le plan est défini à un coefficient multiplicatif près, on s'impose :

$$yf(\boldsymbol{x}) \geq 1.$$

Un plan $(\boldsymbol{w}, b)$ est un séparateur si :

$$y_i f(\boldsymbol{x}_i) \geq 1 \quad \forall i \in \{1, \ldots, n\}.$$

La distance d'un point $\boldsymbol{x}$ au plan $(\boldsymbol{w}, b)$ est donnée par :

$$d(\mathbf{x}) = \frac{|\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b|}{\|\boldsymbol{w}\|} = \frac{|f(\boldsymbol{x})|}{\|\boldsymbol{w}\|}$$

et, dans ces conditions, la marge du plan a pour valeur $\frac{2}{\|\boldsymbol{w}\|^2}$. Chercher le plan séparateur de marge maximale revient à résoudre le problème ci-dessous d'optimisation sous contraintes (problème primal) :

$$\begin{cases} \min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \\ \text{avec } \forall i, y_i(< \boldsymbol{w}, \boldsymbol{x}_i > +b) \geq 1. \end{cases}$$

Le problème dual est obtenu en introduisant des multiplicateurs de Lagrange. La solution est fournie par un *point-selle* $(\boldsymbol{w}^*, b^*, \boldsymbol{\lambda}^*)$ du lagrangien :

$$L(\boldsymbol{w}, b, \boldsymbol{\lambda}) = 1/2\|w\|_2^2 - \sum_{i=1}^{n} \lambda_i \left[y_i \left(< \boldsymbol{w}, \boldsymbol{x}_i > +b \right) - 1 \right].$$

Ce point-selle vérifie en particulier les conditions :

$$\lambda_i^* \left[y_i \left(< \boldsymbol{w}^*, \boldsymbol{x}_i > +b^* \right) - 1 \right] = 0 \quad \forall i \in \{1, \ldots, n\}.$$

Les *vecteurs support* sont les vecteurs $\boldsymbol{x}_i$ pour lesquels la contrainte est active, c'est-à-dire les plus proches du plan, et vérifiant donc :

$$y_i \left(< \boldsymbol{w}^*, \boldsymbol{x}_i > +b^* \right) = 1.$$

Les conditions d'annulation des dérivées partielles du lagrangien permettent d'écrire les relations que vérifient le plan optimal, avec les λ_i^* non nuls seulement pour les points supports :

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \lambda_i^* y_i \boldsymbol{x}_i \quad \text{et} \quad \sum_{i=1}^{n} \lambda_i^* y_i = 0.$$

Ces contraintes d'égalité permettent d'exprimer la formule duale du lagrangien :

$$W(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j < \boldsymbol{x}_i, \boldsymbol{x}_j > .$$

Pour trouver le point-selle, il suffit alors de maximiser $W(\boldsymbol{\lambda})$ avec $\lambda_i \geq 0$ pour tout $i \in \{1, ...n\}$. La résolution de ce problème d'optimisation quadratique de

taille n, le nombre d'observations, fournit l'équation de l'hyperplan optimal :

$$\sum_{i=1}^{n} \lambda_i^* y_i < \boldsymbol{x}, \boldsymbol{x}_i > + b^* = 0 \quad \text{avec} \quad b^0 = -\frac{1}{2}\left[< \boldsymbol{w}^*, sv_{class+1} > + < \boldsymbol{w}^*, sv_{class-1} >\right].$$

Pour une nouvelle observation $\boldsymbol{x}$ non apprise présentée au modèle, il suffit de regarder le signe de l'expression :

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i^* y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b^*$$

pour savoir dans quel demi-espace cette forme se trouve, et donc quelle classe il faut lui attribuer.

3.2 Cas non séparable

Lorsque les observations ne sont pas séparables par un plan, il est nécessaire d"'assouplir" les contraintes par l'introduction de termes d'erreur ξ_i qui en contrôlent le dépassement :

$$y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq +1 - \xi_i \quad \forall i \in \{1, \ldots, n\}.$$

Le modèle attribue ainsi une réponse fausse à un vecteur $\boldsymbol{x}_i$ si le ξ_i correspondant est supérieur à 1. La somme de tous les ξ_i représente donc une borne du nombre d'erreurs.

Le problème de minimisation est réécrit en introduisant une pénalisation par le dépassement de la contrainte :

$$\begin{cases} \min \frac{1}{2} \|\boldsymbol{w}\|^2 + \delta \sum_{i=1}^{n} \xi_i \\ \forall i, y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \geq +1 - \xi_i \end{cases}$$

Remarques

- Le paramètre δ contrôlant la pénalisation est à régler. Plus il est grand et plus cela revient à attribuer une forte importance à l'ajustement. Il est le paramètre qui ajuste le compromis entre bon ajustement et bonne généralisation.
- Le problème dans le cas non séparable se met sous la même forme duale que dans la cas séparable à une différence près : les coefficients λ_i sont tous bornés par la constante δ de contrôle de la pénalisation.
- De nombreux algorithmes sont proposés pour résoudre ces problèmes d'optimisation quadratique. Certains, proposant une décomposition de l'ensemble d'apprentissage, sont plus particulièrement adaptés à prendre en compte un nombre important de contraintes lorsque n, le nombre d'observation, est grand.
- On montre par ailleurs que la recherche des hyperplans optimaux répond bien au problème de la "bonne" généralisation. On montre aussi que, si l'hyperplan optimal peut être construit à partir d'un petit nombre de vecteurs supports, par rapport à la taille de la base d'apprentissage, alors la capacité en généralisation du modèle sera grande, indépendamment de la taille de l'espace.
- Plus précisément, on montre que, si les $\boldsymbol{X}$ sont dans une boule de rayon R, l'ensemble des hyperplans de marge fixée δ a une VC-dimension bornée par

$$\frac{R^2}{\delta^2} \quad \text{avec } \|\boldsymbol{x}\| \leq R.$$

- L'erreur par validation croisée (*leave-one-out*) et bornée en moyenne par le nombre de vecteurs supports. Ces bornes d'erreur sont bien relativement prédictives mais néanmoins trop pessimistes pour être utiles en pratique.

4 Séparateur non linéaire

4.1 Noyau

Revenons à la présentation initiale du problème. Les observations faites dans l'ensemble $\mathcal{F}$ (en général $\mathbb{R}^p$) sont considérées comme étant transformées par une application non linéaire Φ de $\mathcal{F}$ dans $\mathcal{H}$ muni d'un produit scalaire et de plus grande dimension.

Le point important à remarquer, c'est que la formulation du problème de minimisation ainsi que celle de sa solution :

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i^* y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b^*$$

ne fait intervenir les éléments $\boldsymbol{x}$ et $\boldsymbol{x}'$ que par l'intermédiaire de *produits scalaires* : $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle$. En conséquence, il n'est pas nécessaire d'expliciter la transfor-

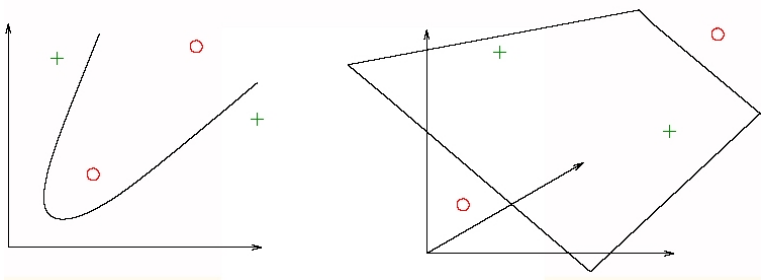

FIGURE 3 – Rôle de l'espace intermédiaire dans la séparation des données.

mation Φ, ce qui serait souvent impossible, à condition de savoir exprimer les produits scalaires dans $\mathcal{H}$ à l'aide d'une fonction $k : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ symétrique appelée *noyau* de sorte que :

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x'}) \rangle .$$

Bien choisi, le noyau permet de matérialiser une notion de "proximité" adaptée au problème de discrimination et à sa structure de données.

Exemple

Prenons le cas trivial où $\boldsymbol{x} = (x_1, x_2)$ dans $\mathbb{R}^2$ et $\Phi(\boldsymbol{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ est explicite. Dans ce cas, $\mathcal{H}$ est de dimension 3 et le produit scalaire s'écrit :

$$\begin{aligned} \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x'}) \rangle &= x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2 x_2'^2 \\ &= (x_1x_1' + x_2x_2')^2 \\ &= \langle \boldsymbol{x}, \boldsymbol{x'} \rangle^2 \\ &= k(\boldsymbol{x}, \boldsymbol{x'}). \end{aligned}$$

Le calcul du produit scalaire dans $\mathcal{H}$ ne nécessite pas l'évaluation explicite de Φ. D'autre part, le plongement dans $\mathcal{H} = \mathbb{R}^\ni$ peut rendre possible la séparation linéaire de certaines structures de données (cf. figure 3).

4.2 Condition de Mercer

Une fonction $k(.,.)$ symétrique est un noyau si, pour tous les $\boldsymbol{x}_i$ possibles, la matrice de terme général $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ est une matrice définie positive c'est-à-dire quelle définit une matrice de produit scalaire.

Dans ce cas, on montre qu'il existe un espace $\mathcal{H}$ et une fonction Φ tels que :

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x'}) \rangle .$$

Malheureusement, cette condition théorique d'existence est difficile à vérifier et, de plus, elle ne donne aucune indication sur la construction de la fonction noyau ni sur la transformation Φ. La pratique consiste à combiner des noyaux simples pour en obtenir des plus complexes (multidimensionnels) associés à la situation rencontrée.

4.3 Exemples de noyaux

- Linéaire
$$k(\boldsymbol{x}, \boldsymbol{x'}) = \langle \boldsymbol{x}, \boldsymbol{x'} \rangle$$
- Polynômial
$$k(\boldsymbol{x}, \boldsymbol{x'}) = (c + \langle \boldsymbol{x}, \boldsymbol{x'} \rangle)^d$$
- Gaussien
$$k(\boldsymbol{x}, \boldsymbol{x'}) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x'}\|^2}{2\sigma^2}}$$

Beaucoup d'articles sont consacrés à la construction d'un noyau plus ou moins exotique et adapté à une problématique posée : reconnaissance de séquences, de caractères, l'analyse de textes... La grande flexibilité dans la définition des noyaux, permettant de définir une notion adaptée de similitude, confère beaucoup d'efficacité à cette approche à condition bien sur de construire et tester le bon noyau. D'où apparaît encore l'importance de correctement évaluer des erreurs de prévision par exemple par validation croisée.

Attention, les SVM à noyaux RBF gaussiens, pour lesquels, soit on est dans le cas séparable, soit la pénalité attribuée aux erreurs est autorisée à prendre n'importe quelle valeur, ont une VC-dimension infinie.

4.4 SVM pour la régression

Les SVM peuvent également être mis en œuvre en situation de régression, c'est-à-dire pour l'approximation de fonctions quand Y est quantitative. Dans le cas non linéaire, le principe consiste à rechercher une estimation de la fonction par sa décomposition sur une base fonctionnelle. la forme générale des

fonctions calculées par les SVM se met sous la forme :

$$\phi(\boldsymbol{x}, \boldsymbol{w}) = \sum_{i=1}^{\infty} w_i v_i(\boldsymbol{x}).$$

Le problème se pose toujours comme la minimisation d'une fonction coût mais, plutôt que d'être basée sur un critère d'erreur quadratique (moindre carrés), celle-ci s'inspire des travaux de Huber sur la recherche de modèle robustes et utilise des écarts absolus.

On note $|.|_\epsilon$ la fonction qui est paire, continue, identiquement nulle sur l'intervalle $[0, \epsilon]$ et qui croit linéairement sur $[\epsilon, +\infty]$. La fonction coût est alors définie par :

$$E(\boldsymbol{w}, \gamma) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \phi(\boldsymbol{x}_i, \boldsymbol{w})|_\epsilon + \gamma \|\boldsymbol{w}\|^2$$

où γ est, comme en régression *ridge*, un paramètre de régularisation assurant le compromis entre généralisation et ajustement. De même que précédemment, on peut écrire les solutions du problèmes d'optimisation. Pour plus de détails, se reporter à Schölkopf et Smola (2002)[2]. Les points de la base d'apprentissage associés à un coefficient non nul sont là encore nommés vecteurs support.

Dans cette situation, les noyaux k utilisés sont ceux naturellement associés à la définition de bases de fonctions. Noyaux de splines ou encore noyau de Déricl et associé à un développement en série de Fourier sont des grands classiques. Ils expriment les produits scalaires des fonctions de la base.

5 Exemples

Comme pour les réseaux de neurones, l'optimisation des SVM qui, en plus du choix de noyau, peut comporter de 1 à 3 paramètres (pénalisation et éventuels paramètres du noyau) est délicate. La figure 4 montre 3 résultats de validation croisée pour le simple noyau linéaire dans le cas des données NIR.

5.1 Cancer du sein

La prévision de l'échantillon test par un Séparateur à Vaste marge conduit à la matrice de confusion :

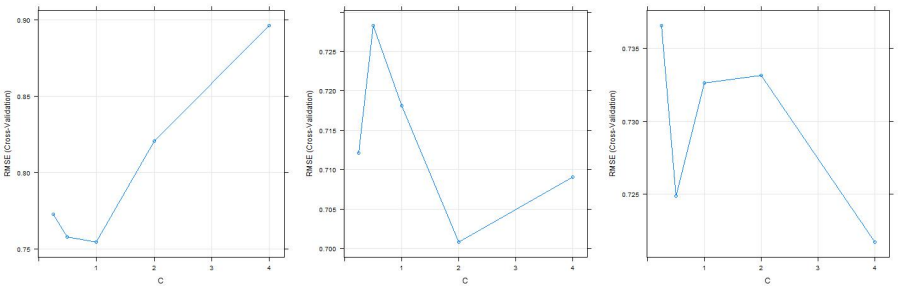

FIGURE 4 – Cookies : trois exécutions de la validation croisée estimant l'erreur en fonction de la pénalisation d'un noyau linéaire.

```
ign malignant
  benign         83          1
  malignant       3         50
```

et donc une erreur estimée de 3%.

5.2 Concentration d'ozone

Un modèle élémentaire avec noyau par défaut (gaussien) et une pénalisation de 2 conduit à une erreur de prévision estimée à 12,0% sur l'échantillon test. La meilleure prévision de dépassement de seuil sur l'échantillon test initial est fournie par des SVM d'ε-régression. Le taux d'erreur est de 9,6% avec la matrice de confusion suivante :

```
          0   1
FALSE   161  13
TRUE      7  27
```

Ce résultat serait à confirmer avec des estimations systématiques de l'erreur. Les graphiques de la figure 5 montre le bon comportement de ce prédicteur. Il souligne notamment l'effet "tunnel" de l'estimation qui accepte des erreurs autour de la diagonale pour se concentrer sur les observations plus éloignées donc plus difficiles à ajuster.

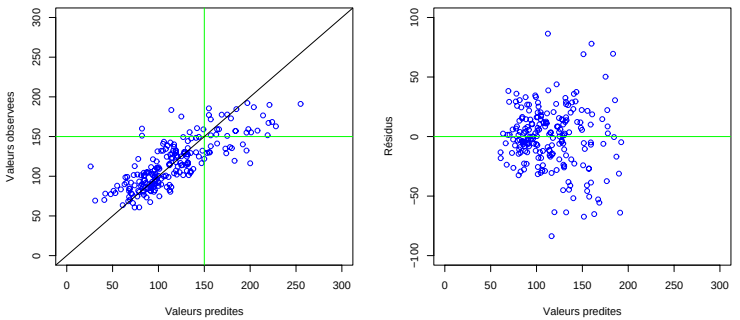

FIGURE 5 – Ozone : Valeurs observées et résidus en fonction des valeurs prédites pour l'échantillon test.

5.3 Données bancaires

Les données bancaires posent un problème car elles mixent variables quantitatives et qualitatives. Celles-ci nécessiteraient la construction de noyaux très spécifiques. Leur traitement par SVM n'est pas détaillé ici.

Références

[1] Y. Guermeur et H. Paugam-Moisy, *Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines*, Apprentissage automatique (M. Sebban et G. Venturini, réds.), Hermes, 1999, p. 109–138.

[2] B Schölkopf et A Smola, *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, 2002.