Classification supervisée et SVM

Travaux Pratiques

1 Introduction

Les SVM ont été introduites par Vapnik [3], et sont abordées au chapitre 12 du livre [1]. La popularité des méthodes SVM, pour la classification binaire en particulier, provient du fait qu'elles reposent sur l'application d'algorithmes de recherche de règles de décision linéaires ("hyperplan séparateur"), la recherche s'effectuant toutefois dans un espace ("feature space") de très grande dimension, lequel est l'image de l'espace d'entrée original par une transformation Φ non linéaire. Le but de ce TP est de mettre en pratique ce type de techniques de classification sur données réelles et simulées au moyen du package R e1071 (lequel met en oeuvre la librairie en C libsvm) et d'apprendre à contrôler les paramètres garantissant leur flexibilité (hyperparamètres, noyau).

2 SVM et noyaux

Les techniques SVM (non linéaires) font appel à une fonction implicite Φ transformant l'espace d'entrée $\mathcal{X} \subset \mathbb{R}^d$ en un espace hilbertien $(\mathcal{H},\ \langle .,\ . \rangle$ de plus grande dimension. L'apprentissage s'effectue alors à partir du modèle $(\Phi(X), Y)$ dans l'espace $\mathcal{H}$, de dimension plus grande certes, mais dans lequel on espère que les données soient "davantage linéairement séparables". Du point de vue pratique, il convient de noter que le calcul des projections $\Phi(X)$ n'est pas utilisé dans la méthode, seuls les produits scalaires $\langle \Phi(x),\ \Phi(x') \rangle$, $(x, x') \in \mathcal{X}^2$, sont requis. Or, ceux-ci sont donnés par un noyau K, via la relation ("kernel trick")

$$K(x, x') = \langle \Phi(x),\ \Phi(x') \rangle.$$

La méthode requiert donc de sélectionner un noyau (ainsi que d'autre paramètres). Parmi les choix possibles, on compte en particulier :

- Le noyau linéaire : $K(x, x') = \langle x,\ x' \rangle$ (correspondant aux SVM linéaires)
- Le noyau Gaussien radial (Gaussian RBF) $K(x, x') = \exp\left(-\sigma ||x - x'||^2\right)$
- Les noyaux polynomiaux $K(x, x') = (\alpha + \beta \langle x,\ x' \rangle)^{\delta}$
- Le noyau radial de Laplace (Laplace RBF) $K(x, x') = \exp\left(-\sigma ||x - x'||\right)$
- Le noyau tangente hyperbolique (sygmoïde) $K(x, x') = tanh\left(\alpha + \beta \langle x,\ x' \rangle\right)$

Un classifieur SVM est de la forme

$$C(X) = sign\left(\langle \omega,\ \Phi(X)\rangle + b\right),$$

où $\omega \in \mathcal{H}$ et $b \in \mathbb{R}$ sont des paramètres ajustés lors de la phase d'apprentissage à partir d'un échantillon d'exemples i.i.d. $\{(x_i, y_i) :\ 1 \leq i \leq n\}$. La frontière associée à cette règle de décision a pour équation : $\langle \omega,\ \Phi(x)\rangle + b = 0$. Elle correspond à un hyperplan dans l'espace $\mathcal{H}$, mais est beaucoup plus complexe dans $\mathcal{X}$ (selon la forme du noyau choisi). Dans $\mathcal{H}$, l'hyperplan est obtenu en maximisant la marge séparant les deux classes, ce qui revient à résoudre un problème d'optimisation sous contraintes linéaires :

$$\text{minimiser } \frac{1}{2}||\omega||^2 + \frac{C}{m}\sum_{i=1}^{n}\xi_i$$

sous les contraintes : $\forall i \in \{1,\ \ldots,\ n\}$,

$$\xi_i \geq 0 \text{ et } y_i\left(\langle \omega,\ \Phi(x_i)\rangle + b\right) \geq 1 - \xi_i.$$

On peut montrer que la solution ω peut s'exprimer de la façon suivante

$$\omega = \sum_{i=1}^{n}\alpha_i y_i \Phi(x_i),$$

les indices i pour lesquels $\alpha_i \neq 0$ sont ceux pour lesquels l'égalité est réalisée dans la contrainte, les points x_i correspondants sont appelés *vecteurs supports* (de la décision). Les coefficients α_i désignent les solutions du problème quadratique dual :

$$\text{maximiser } \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{1\leq i,\ j\leq n}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

sous les contraintes : $\forall i \in \{1,\ \ldots,\ n\}$,

$$0 \leq \alpha_i \leq \frac{C}{n} \text{ et } \sum_{i=1}^{n}\alpha_i y_i = 0.$$

Le paramètre C contrôle la complexité du classifieur dans la mesure où il détermine le coût d'une mauvaise classification : plus C est grand, plus la règle obtenue est complexe (le nombre de points pour lesquels on veut minimiser l'erreur de classification croît). Cette approche est appelée C-classification.

Une autre façon de contrôler la complexité (*i.e.* le nombre de vecteurs supports), appelée ν-classification, revient à considérer, à la place du problème dual décrit ci-dessus, le problème suivant :

$$\text{minimiser } \frac{1}{2}\sum_{1\leq i,\ j\leq n}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

sous les contraintes : $\forall i \in \{1,\ \ldots,\ n\}$,

$$0 \leq \alpha_i \leq \frac{1}{n} \text{ et } \sum_{i=1}^{n}\alpha_i \geq \nu \text{ et } \sum_{i=1}^{n}\alpha_i = 0,$$

où $\nu \in [0, 1]$ est un paramètre approchant le pourcentage de vecteurs supports parmi les données d'apprentissage.

3 Extensions aux cas multi-classe

Dans le cas où la variable de sortie Y compte plus de deux modalités, il existe plusieurs façon d'étendre directement les méthodes du cas binaire.

"Un contre un". Dans le cas où l'on cherche à prédire un label pouvant $K \geq 3$ modalités, on peut considérer toutes les paires de labels (k, l) possibles, $1 \leq k < l \leq K$ (il y en a C_K^2) et ajuster un classifieur $C_{k,l}(X)$ pour chacune d'entre elles. La prédiction correspond alors au label qui a gagné le plus de "duels".

"Un contre tous". Pour chaque modalité k, on apprend un classifieur permettant de discriminer entre les populations $Y = k$ et $Y \neq k$. A partir des estimations des probabilités a posteriori, on affecte le label estimé le plus probable.

4 Applications

Au moyen du package E1071, on s'attachera à mettre en oeuvre les techniques de classification SVM sur les jeux de données de l'UCI data repository, dans des cadres binaire et multi-classe. Au moyen de plans d'expérience adéquats (validation croisée), on sélectionnera le noyau et l'hyperparamètre (C ou ν) permettant de minimiser l'erreur de généralisation. On comparera leur efficacité à celles des méthodes de partitionnement récursif (package rpart ou tree).

On trouvera sur le package E1071 de multiples références en ligne, comme par exemple `http://planatscher.net/svmtut/svmtut.html`.

Références

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning : data mining, inference and prediction.* Springer, 2 edition, 2009.

[2] A. Karatzoglou and D. Meyer. Support vector machines in r. *Journal of Statistical Software*, 15(9), April 2006.

[3] Vladimir Vapnik. *Statistical learning theory.* Wiley, 1998.