Apprentissage supervisé et non supervisé
TP3 : Support Vector Machines (Régression)
M2 SISE - Université Lyon 2 - 2017/2018

Responsable : Julien Ah-Pine

1 Présentation de l'étude de cas

L'étude de cas est celui des biscuits vu lors du TP1. Ce sont des données de grande dimension $p > n$ et il s'agit d'un problème de régression. Nous avons à notre disposition 40 observations pour lesquelles nous savons la teneur en sucre et également les données du signal. Il faut prédire la teneur en sucre à partir du signal. L'objectif est de tester le modèle svm sur ces données.

2 Lecture et decription des données

1. Charger les données contenues dans le fichier `cookie.RData` à l'aide de la commande `load`.

3 Régression par SVM

Nous allons utiliser la librairie `e1071` qui donne une interface R à libsvm, une performante librairie pour les svm développée en C++. Rappelons que le problème d'optimisation du svm pour la régression se formalise de la façon suivante :

$$\begin{array}{ll} & \min_{a_0, \mathbf{a} \in \mathbb{R}^p, \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \in \mathbb{R}^n} \frac{1}{2}\|\mathbf{a}\|^2 + c\sum_{i=1}^n (\xi_i^+ + \xi_i^-) \\ slc & \forall i, y_i - (a_0 + \mathbf{a}^\top \mathbf{x}_i) \leq \epsilon + \xi_i^+ \\ & \forall i, (a_0 + \mathbf{a}^\top \mathbf{x}_i) - y_i \leq \epsilon + \xi_i^- \\ & \forall i, \xi_i^+, \xi_i^- \geq 0 \end{array}$$

où c est une constante positive tel un coefficient de pénalité, permettant de contrôler l'équilibre entre la maximisation de la marge et les erreurs.

2. Installez puis chargez la librairie `e1071`.
3. Entrez, exécutez et commentez les commandes suivantes :

```
svm_fit=svm(sucres~.,data=D,type="eps-regression",kernel="linear
   ")
svm_fit$residuals
D$sucres-predict(svm_fit,D)
sum(svm_fit$residuals^2)/length(svm_fit$residuals)
```

4. En étudiant `svm_fit`, identifiez les vecteurs supports et leur coefficient.
5. Quelles sont les valeurs par défaut des hyper-paramètres ϵ et c ?
6. Stockez dans la variable `c_seq` la séquence de valeurs $1, 2, 10, 50, 100$. Ecrire un programme permettant de stocker dans la variable `mse_c` le critère mse pour chaque valeur dans `c_seq`. Etudiez et commentez les mse obtenus pour chaque valeur de c.

4 Validation croisée

Les résultats précédents ne permettent pas d'apprécier l'erreur en généralisation. Nous allons pour cela utiliser la validation croisée déjà implémentée dans la commande `svm`.

7. Entrez, exécutez et commentez les commandes suivantes :

```
svm_fit=svm(sucres~.,data=D,type="eps-regression",kernel="linear
   ",cross=5)
svm_fit$MSE
mean(svm_fit$MSE)
```

8. Ecrire un programme permettant de stocker dans la variable `mse_cv` l'estimation de l'erreur en généralisation selon le critère mse pour chaque valeur dans `c_seq`. Comparez `mse_c` et `mse_cv` et commentez.

5 Utilisation de plusieurs noyaux

L'objectif est d'étudier les résultats du svm avec l'utilisation de noyaux non linéaires. Le noyau rbf ou gaussien est obtenu en utilisant `radial` à la place de `linear` et le noyau polynomial en utilisant `poly`.

9. Pour chaque noyau pré-cité quels sont les paramètres par défaut utilisés par la commande `svm` ?
10. Utilisez les commandes des section précédentes afin d'écrire un programme permettant de déterminer les variables `mse_c` et `mse_cv` avec un noyau rbf. Comparez `mse_c` et `mse_cv` et commentez.
11. Même question mais avec un noyau polynomial.

6 Optimisation des hyper-paramètres

Pour contrôler les performances du svm, nous avons vu précédemment que nous pouvions "jouer" avec les valeurs de l'hyper-paramètre c et avec le type de noyau utilisé. Il existe un autre hyper-paramètre dans le cas de la régression qui est la tolérance ϵ utilisée dans la fonction objectif. Le but ici est de déterminer les meilleures couples (ϵ, c) parmi une grille de recherche (le noyau étant cette fois-ci fixé).

12. Entrez, exécutez et commentez les commandes suivantes :

```
c_seq=c(1,10,50)
eps_seq=c(0.05,0.1,0.5)
svm_grid_search=tune(method = svm, kernel = "linear", sucres~.,
   data=D, ranges = list(epsilon=eps_seq,cost=c_seq))
print(svm_grid_search)
plot(svm_grid_search)
```

13. Différentes valeurs d'un noyau paramétrique peuvent également être données en entrée de la commande `tune`. Ecrivez un programme permettant d'étudier un noyau polynomial avec les valeurs du paramètre d (le degré) appartenant à la séquence $1, 2, 3, 4$.