SY19 - TP 6

Automne 2011

I-Problématique

Mettre en œuvre la méthode des séparateurs à vaste marge, en testant l'influence du choix des paramètres.
On désire appliquer la méthode sur les données du fichier breast-cancer-wisconsin.data dans le but discriminer les tumeurs bénignes des tumeurs malignes en utilisant des données morphologiques. L'ensemble d'exemples disponible est constitué de 699 cas correctement classifiés. Un travail supplémentaire sur le problème de reconnaissance de caractères est à faire si le temps le permet.

II-Travail Préliminaire

Manipulation et compréhension des données

Avant de commencer, il vaut mieux ouvrir le fichier *breast-cancer-wisconsin.data* dans un éditeur de texte, de préférence *wordpad*, et regarder les données.
Ci-après quelques opérations et commandes en R, avec les commentaires explicatifs, pour vous aider à manipuler et comprendre les données du fichier *breast-cancer-wisconsin.data*.

```
##################################################

#lire " breast cancer data " :

bcdata <- read.csv('breast-cancer-wisconsin.data',head=TRUE)

#Regarder les noms des colonnes.

names(bcdata)

#regarder une colonne, par exemple:

bcdata$ClumpThickness #ou bcdata$Samplecodenumber

#regarder les classes:

bcdata$Class

#Enlever de bcdata la colonne "Samplecodenumber" et la colonne
#"Class". Le signe moins pour dire unselect les deux colonnes:

databcall <- subset(bcdata,select=c(-Samplecodenumber,-Class))

# vous pouvez faire names(databcall)

# Selectionner les classes seulement:
```

```
classesbcall <- subset(bcdata,select=Class)

#prendre une partie de databcall pour l'apprentissage:

databctrain <- databcall[1:400,]

classesbctrain <- classesbcall[1:400,]

#prendre une partie de databcall pour le test:

databctest <- databcall[401:699,]

classesbctest <- classesbcall[401:699,]


##################################################
```

Application des SVM sous R

La fonction du langage R à utiliser est *svm* du package e1071. Un exemple d'application aux données Breast-Cancer est le suivant :

```
##################################################


# Faire help(svm) pour bien comprendre la fonction "svm" :

model <- svm(databctrain, classesbctrain)

# Faire str(model)

# Faire help(predict.svm) #Attention, ici "model" est un objet de
#la classe "svm". Ainsi, il faut faire help(predict.svm) et pas
#help(predict)

# Validation de "model" :

pred <- predict(model, databctest)


# Comparer la prediction et les vraies classes :

table(pred,t(classesbctest))

#Les hyperparametres affecte les performances du noyau. Le
#package e1071 offre une fonction, tune(), qui fait ce qu'on
#appelle "grid search" et donne ainsi une estimation des
#parametres. Pour cela, faire : help(tune)

#Exemple d'application de la fonction "tune" :

a=tune(svm, train.x=databctrain, train.y=classesbctrain,
validation.x=databctest, validation.y=classesbctest, ranges =
list(gamma = 10^(-1:1), cost = c(1,1.5,2)), control =
tune.control(sampling = "fix"))
```

```
# Faire  str(a)

model <- svm(databctrain,
classesbctrain,gamma=a$best.parameters$gamma,cost=a$best.parameters$cost)


# Comparaison de la prediction et des vraies classes

pred <- predict(model, databctest)

table(pred,t(classesbctest))


##################################################
```

III-Travail demandé

La mise en œuvre des SVM nécessite un certain nombre de choix et de réglages. La fonction *tune* présentée ci-dessus permet de faire ce réglage pour un noyau choisi a priori. Il est cependant important que vous testiez l'influence de ces différents choix sur la qualité de vos résultats. En utilisant l'erreur moyenne comme critère de mesure de performance, mettez en évidence l'intérêt des choix suivants :

1. Le choix du paramètre C de pénalisation ;
2. Le choix du noyau ;
3. Le choix du paramètre associé au noyau, comme par exemple le choix de la *largeur de bande* d'un noyau gaussien, ou le *degré* d'un noyau polynomial...

Vous pouvez par exemple, pour un noyau de type fixé préalablement, tracer l'erreur moyenne en fonction du paramètre C pour mettre en évidence l'influence du choix du paramètre C.
Pour un noyau de type choisi préalablement, gaussien par exemple, vous pouvez tracer l'erreur moyenne en fonction du paramètre du noyau.

IV-Travail Supplémentaire

Pour les plus motivés, vous pouvez utiliser la fonction *ksvm* du package *kernlab* pour répondre au problème de de reconnaissance de caractères. Vous pouvez aussi utiliser d'autres fonctions de ce même package telles que la fonction *kpca*, pour l'analyse en composante principale à noyau.