ECP, SDM 2015

Agrégation de modèles analyse spectral et clustering

M. Mougeot, N. Vayatis,

Rappel: `help(fonc)` pour obtenir de l'aide sur la fonction nommée "fonc".

Contrôle des connaissances du TD: code R à poster sur Claroline avant le 11 Janvier 2016.

Agrégation de modèles

Le but de cette application est de construire un modèle prédictif permettant de classer automatiquement si un courrier électronique est un spam (indicateur 1) ou non (0). Les données sont constituées de plus de 4500 observations. Chaque observation (spam ou non) est décrite par 57 attributs.

Analyse préliminaire

- Charger les données spam et les indices des observations, indtrain, dans l'environnement de travail R

  tab=read.table('spam.txt',header=T,sep=';"); Indiquer le nombre d'observations, le nombre de variables et la nature des variables. Caractériser à l'aide du fichier "names" les variables analysées dans les courriers éléctroniques?

- Les données du fichier "indtrain.txt" correspondent aux indices à retenir pour les données d'apprentissage. La base de test est définie par les données complémentaires.

  Fonctions R: nrow(), ncol(), dim(), names()

Arbres de Classification

- Construire un arbre de décision sur l'échantillon de données d'apprentissage. Visualiser l'arbre de classification. Quelle est la variable la plus influente? Combien de variables interviennent dans l'arbre de classification affiché.

- Calculer les erreurs de classification et la matrice de confusion sur la base d'apprentissage et sur la base de test. Quels sont les taux de fausse alarme et de non détection? Conclusion.

  Fonctions R et de la librairie (rpart): library(rpart), rpart(), predict(); plot(); text(), table(), sum(), print(). Pour information, le package adabag permet également de réaliser du bagging sur des arbres de décision (librairie adabag).

Agrégation de modèles. Bagging

- Utiliser la fonction bagging() pour générer plusieurs arbres de classification à l'aide d'échantillons bootstrap des données d'apprentissage. Combien d'arbres sont par défaut générés?

- Calculer les erreurs de classification et la matrice de confusion sur la base d'apprentissage et sur la base de test. Conclusion.

  Fonctions R de la librarie (ipred): library(ipred), bagging(), predict(); plot(); text(), table(), sum(), print()

Agrégation de modèles. Random Forest

- Utiliser la fonction randomForest() pour génerer plusieurs arbres de classification à l'aide d'échantillons bootstrap des données d'apprentissage en choisissant aléatoirement pour chaque noeud les variables d'étude
- Calculer les performances de ces modèles agrégés sur la base d'apprentissage puis sur la base de test ainsi que la matrice de confusion. Conclusion.
- Etudier et afficher l'importance des variables à l'aide de la fonction `varImpPlot` **Fonctions R et de la librarie (randomForest):** library(randomForest), bagging(), predict(); plot(); text(), table(), sum(), print()

Scoring, Comparaison des modèles de classification

- Estimer les paramètres à l'aide d'une régression logitique, d'une analyse discriminante ou d'un SVM sur l'échantillon de données d'apprentissage
- Calculer les erreurs de classification et la matrice de confusion sur la base d'apprentissage et sur la base de test. Conclusion.

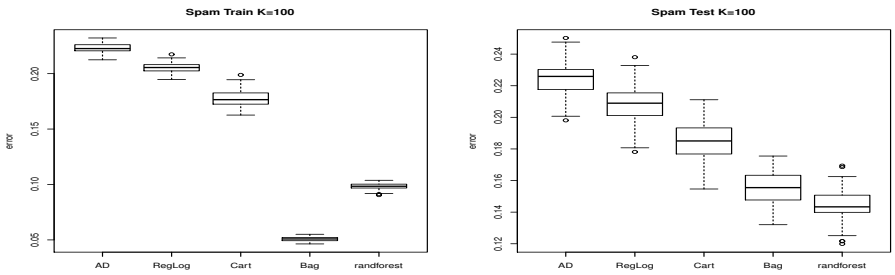
  Fonctions R: glm(), predict.glm(), table(), sum(), print(). library(MASS), lda(), predict(), table(), sum(), print()

Comparaison des modèles de classification

- Etape Préliminaire
  - Sélectionner aléatoitement 75% des observations pour estimer les paramètres du modèle et 25% pour la base de test en utilisant la fonction sample().
  - Calculer l'erreur de classification la matrice de confusion sur la base d'apprentissage et sur la base de test
- Variations des échantillons d'apprentissage et de test
  - Répéter l'étape prélimaire pour $K = 10$ réplications
  - Sauvegarder les erreurs de classification pour chacune des méthodes étudiées dans deux tableaux ETrain et ETest.
  - Calculer les erreurs de classifications sur la moyenne des K réplications, ainsi que l'écart-type observé pour chacune des méthodes pour chacun des échantillons d'apprentissage et de test.
  - Synthétiser vos résultats à l'aide d'un graphique adapté.

Fonctions R: rep(), matrix(), as.data.frame(), boxplot()

Synthèse des résultats



Décomposition en valeurs singulières & Analyse en composantes principales

Préliminaire:

Liste de commandes R indicatives: rnorm(), matrix(), cov(), svd(), barplot(),diag(), sum(), cbind(), nrow(), ncol(), t(). Pour obtenir de l'aide sur une fonction R, taper help(nomfunction). L'opérateur $\%*\%$ effectue un produit matriciel

Exercice 1:

Partie A:

- Simuler n réalisations d'un vecteur aléatoire gaussien multivarié de taille p: $X = (X_1, X_2, \ldots, X_p)$ avec $X_j \sim \mathcal{N}(0,1)$ i.i.d. On prendra dans un premier temps $n = 50$ et $p = 5$. $\mathcal{N}$ est la loi normale.
- Calculer la matrice de covariance empirique, notée S. Analyser S. Que remarquez-vous?
- La décomposition en valeur singulière de S est donnée par $S = U\Sigma V^t$. Donner la définition et caractériser les matrices U, V et Σ. Utiliser le logiciel R pour effectuer une SVD de S à l'aide de la commande svd(). En étudiant l'aide de la fonction, affecter les résultats dans des matrices notées U, V et Σ, définiées précédemment.
- Que contient Σ? Proposer deux méthodes différentes pour comparer U et V? Calculer et comparer les traces des matrices S et Σ. Que peut-on dire? Expliquer théoriquement le résultat observé.
- Calculer $U\Sigma V^t$. Que remarquez-vous?

Exercice 2: Analyse en composantes principales

L'analyse en composantes principales est une méthode d'analyse exploratoire de données multidimentionnelles quantitatives. Cette méthode permet de projeter les données initiales sur des plans, appelés "plans factoriels", tout en conservant un maximum d'information (en maximisant la variance des données projetés). Cette méthode est un outil d'introspection de distributions empiriques de données utilisé pour découvrir des "features" dans les données.

Liste de commandes R indicatives: dim(), ncol(), nrow(), prcomp(), library(ade4), dudi.pca(), s.corcircle().
Application: Les données "cardata.txt" présentent les caractéristiques d'un ensemble de voitures.

Analyse Préliminaire

1. Analyser rapidement la structure du fichier de données à l'aide d'un éditeur classique (blocnote, wordpad)? Que remarquez-vous?

2. Récupérer les variables numériques dans un dataframe noté X à l'aide de la commande `read.table()`. Utiliser les options de la fonction de lecture pour affecter un nom aux variables et aux individus statistiques étudiés (options `row.names, header`)

3. Caractériser l' échantillon de données (effectif n, et nombre de variables p)
   Exécuter la commande plot(X). Que représente ce graphique? Calculer la matrice des corrélations `cor(X)` pour compléter l'analyse du graphique. Indiquer vos premières conclusions.

ACP

La commande `prcomp()` de R est une fonction standard de R utilisée pour l'ACP. Effectuer une ACP sur les données **centrées réduites** en étudiant puis utilisant cette fonction `help(prcomp)`.
Sauvegarder les résultats dans la variable res. Analyser et caractériser les champs de l'objet res en vous aidant de l'aide et de l'instruction `attributes(res)`: `res$sdev res$rotation res$center res$scale res$x`.

Etude des Valeurs propres:

Récupérer les valeurs propres dans un tableau que l'on nommera valpr. Représenter l'éboulis des valeurs propres. Calculer le pourcentage de variance expliquée par chaque axe principal, puis afficher le pourcentage de variance expliquée par le premier plan factoriel puis la totalité des axes `, fonction cumsum()`. Combien de composantes principales conviendrait-il de garder pour expliquer 95% de la variance? 98% de la variance?

Etude des Vecteurs propres:

Visualiser les coordonnées des axes principaux dans l'ancienne base. Expliquer l'information retenue par les quatre premières composantes.

Analyses des individus projetés et cercle des corrélations:

Tracer à l'aide de la commande `biplot()` un graphique représentant les projections des individus sur le 1er plan factoriel (axes 1 et 2) et sur le 2ème plan factoriel (axes 2 et 3). Que constatez vous sur la répartition des individus sur les plans factoriels? Appuyer votre analyse sur l'étude du cercle des corrélations. Commenter le graphique obtenu.

Remarque: La libraririe ade4 de R propose un ensemble de fonctions dédiées à l'analyse factorielle. La fonction `dudi.pca()` permet de réaliser une analyse en composante principale. La fonction `s.corcircle()` permet de tracer le cercle des corrélations.

Classification non supervisée (kmeans)

La méthode des kmeans nécessite de faire une hypothèse sur le nombre de classes. La fonction `kmeans()` de R regroupe des exemples en k classes en fonction de l'algorithme des K-means `kmeans(X,k,it)`.

- Regrouper les données cardata en 1,2,3,4 classes à l'aide de la commande kmeans.
- Analyser et décrire statistiquement les champs de sorties de la function kmeans :

  `resk$centers ; resk$cluster ; resk$withinss ; resk$betweenbss`
- Utiliser les données projetées de l'A.C.P. pour visualiser dans le 1er plan factoriel les classes constituées. Que constatez-vous?

Exercice 3: Caractères Manuscrits

Charger le fichier "digits3-8.RData" se trouvant sur le site du cours en double cliquant dessus. Il s'agit d'un environnement R où deux tableaux de données sont chargés. Chacun de ces tableaux, `d3` et `d8`, contient 1100 images de caractères, respectivement 3 et 8, sous formes de vecteurs à valeurs dans $[0; 255]$. Chaque ligne d'un tableau contient 256 valeurs que l'on peut réordonner sous forme de tableau de taille 16×16 en utilisant la commande `matrix(x, 16, 16)`.
Partie A:

- Définir une fonction `mImage` qui réordonne les 256 valeurs contenue dans un vecteur en un tableau de taille 16×16, puis affiche l'image correspondante. Tester cette fonction sur plusieurs lignes de `d3` et `d8`.
  indication : utiliser la fonction `image`() en spécifiant:

  $$\texttt{axes} = \texttt{FALSE}, \texttt{col} = \texttt{gray}(0 : 255/255)$$

  Transposer la matrice s'il faut.

Partie B:

- Scinder les données en quatre matrices `d3train`, `d3test`, `d8train`, `d8test` où les matrices d'apprentissage contiennent 1000 lignes et les matrices de test 100 lignes chacune.
  On travaillera par la suite sur les matrices d'apprentissage.
- Calculer, puis afficher le "3 moyen" et le "8 moyen". Stocker les vecteurs "3 moyen", et "8 moyen", on en aura besoin plus tard. Sauvegarder les images obtenues.
- Centrer, normaliser les données et calculer les matrices de covariances.
- Calculer les composantes principales en utilisant `svd`() sur la matrice des données centrées et normalisées ou bien `eigen` sur la matrice de covariance. Expliquer la relation entre les deux.
- Afficher quelques "modes propres" : si $Cov(X) = \sum_{i=1}^{r} \sigma_i u_i u_i^T$, on appelle $u_i u_i^T$ le $i$ème mode propre. Que remarquez-vous ?
- Calculer la matrice de projection sur le sous-espace vectoriel engendré par les 5 premières composantes principales. Vérifier qu'il s'agit bien d'une matrice de projection.
- En n'oubliant pas que l'on a centré et normalisé les données, suggérer une méthode de *reconstruction* des images test en utilisant les vecteurs principaux. Voyez-vous l'avantage en terme de stockage ?