Université de Lille
M2 FCA - Data Sciences celine.duval@univ-lille.fr.fr

Examen - Partie I

Durée 2h (Aucun document autorisé)

Exercice 1. L'objectif est de prédire un label $Y \in \{\circ, \times\}$ à partir d'observations $(X_1, X_2) \in \mathbb{R}^2$. On a la possibilité d'utiliser les modèles suivants :

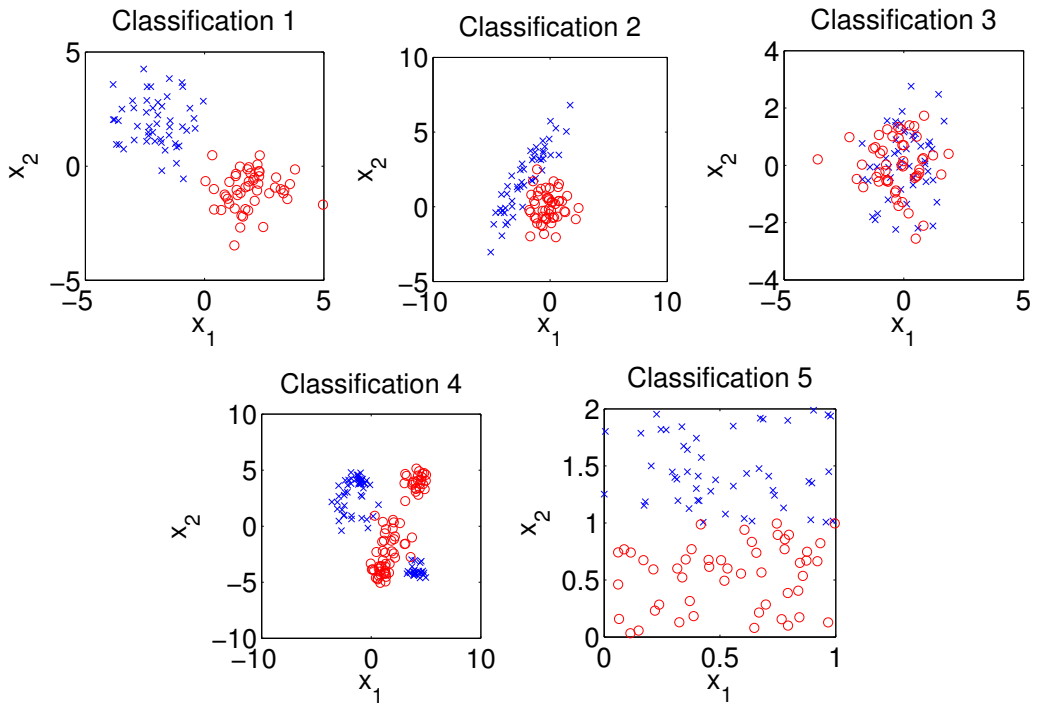
a). Régression logistique sur (X_1, X_2),

b). Analyse discriminante linéaire/quadratique,

c). Régression logistique sur (X_1, X_1^2, X_2, X_2^2),

d). k-plus proches voisins.

Sur les 5 problèmes de classification suivants, quel(s) modèle(s) pensez-vous être adaptés ? Justifier brièvement mais précisément vos réponses.



Exercice 2. Deux statisticiens ont à leur disposition un ensemble de données recueillies sur $n = 100$ patients tous atteints d'une même pathologie. Pour chacun des patients, 20 variables explicatives ont été mesurées. Chacun des patients a également reçu un traitement coûteux, une variable binaire Y indique si le patient a guéri suite au traitement. Le traitement étant coûteux, ils veulent déterminer à partir des données une procédure de classification permettant, pour un nouveau patient atteint de la même pathologie et dont on ne dispose

que de la mesure des 20 variables explicatives, de prédire Y, si le patient guérira suite au traitement ou non.

Le statisticien I veut utiliser la règle des 3 plus proches voisins tandis que le statisticien II propose d'utiliser un arbre de classification binaire. N'arrivant pas à se mettre d'accord, ils mettent chacun au point une procédure permettant de déterminer, à partir des données, quelle méthode de classification est la plus adaptée.

On adoptera les notations suivantes :

— On regroupe les observations dans un tableau $D = (X, Y)$ à n lignes et dont les colonnes sont les 20 variables explicatives, et le label Y.

— Pour un individu i, les valeurs prises par les 20 variables explicatives sont regroupées dans un vecteur $X_i = (X_i^1, \ldots, X_i^{20})$. Ainsi $D = (X_i, Y_i, 1 \leq i \leq n)$.

Procédure du statisticien I :

```
knn = KNeighborsClassifier(n_neighbors = 3).fit(X,Y)
conf_knn = confusion_matrix(Y, knn.predict(X))
err_knn = (conf_knn[1,0] +  conf_knn[0,1])/Y.shape[0]

tree = DecisionTreeClassifier(max_depth=15).fit(X,Y)
conf_tree = confusion_matrix(Y, tree.predict(X))
err_tree = (conf_tree[1,0] +  conf_tree[0,1])/Y.shape[0]
```

Procédure du statisticien II :

```
err_knn = []
err_tree = []
for i in range(10):
    ind = np.binom(1,0.9).rvd(X.shape[0])
    X1, Y1 = X[ind==False].copy(), Y[ind==False].copy()
    X2, Y2 = X[ind].copy(), Y[ind].copy()
    knn = KNeighborsClassifier(n_neighbors = 3).fit(X1,Y1)
    conf_knn = confusion_matrix(Y2, knn.predict(X2))
    err_knn.append((conf_knn[1,0] +  conf_knn[0,1])/Y2.shape[0])

    tree = DecisionTreeClassifier(max_depth=15).fit(X1,Y1)
    conf_tree = confusion_matrix(Y2, tree.predict(X2))
    err_tree.append((conf_tree[1,0] +  conf_tree[0,1])/Y2.shape[0])

err_knn = np.matrix(err_knn).mean()
err_tree = np.matrix(err_tree).mean()
```

1. Décrire rapidement le principe des 3 plus proches voisins.
2. Sur les figures en dernière page tracer (grossièrement) les frontières de séparation entre les classes "$\circ$" et "$\triangle$"' des procédure CART et des 3 plus proches voisins.
3. Détailler ce que renvoie chacune des procédures I et II des codes ci-dessus.
4. Quelle procédure est la plus judicieuse pour choisir la méthode de classification ? Justifier soigneusement votre réponse.

Les résultats de la procédure I sont : `err_tree` = 0.11 et `err_knn` = 0.10.
Les résultats de la procédure II sont : `err_tree` = 0.15 et `err_knn` = 0.17.

5. Quelle méthode est la plus adaptée pour répondre au problème ?
6. Rappeler le concept du bagging.
7. Pour appliquer l'algorithme bagging aux k-plus proches voisins comment faut-il choisir k (justifier) ? Même question avec CART.

Exercice 3. Le but de cet exercice est de montrer que lorsque l'on effectue l'algorithme des K-means (ou centres mobiles), d'une étape m à une étape $m+1$ de l'algorithme, l'inertie intra classe diminue. On note :

— $I = \{1, \ldots, n\}$ l'ensemble des individus décrits par un ensemble de caractéristiques respectivement $X_1, \ldots, X_n$.
— K le nombre de classes considéré.
— $P^{(m)} = [C_1^{(m)}, \ldots, C_K^{(m)}]$ la partition de I obtenue à l'étape m de l'algorithme.
— $g^{(m)} = (g_1^{(m)}, \ldots, g_K^{(m)})$ les centres de gravités associés à la partition $P^{(m)}$.
— Soit $P = [C_1, \ldots, C_K]$ une partition de I et $g = (g_1, \ldots, g_K)$ un ensemble de K points, on note

$$V_a(P, g) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in C_k} \|X_i - g_k\|^2.$$

1. Rappeler rapidement le principe de l'algorithme des centres mobiles, en utilisant les notations ci-dessus.
2. Justifier que $\forall m$, $V_a(P^{(m)}, g^{(m)})$ est l'inertie intra classe à l'étape m de l'algorithme des K means.
3. Montrer que

$$V_a(P^{(m+1)}, g^{(m+1)}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in C_k^{(m)}} \|X_i - g_k^{(m)}\|^2 \leq V_a(P^{(m+1)}, g^{(m)}).$$

4. En déduire que $V_a(P^{(m+1)}, g^{(m+1)}) \leq V_a(P^{(m)}, g^{(m)})$.
5. En déduire que l'inertie intra converge. Quel résultat peut-on en déduire concernant l'algorithme des K means ?

Correction

1. On considère la partition $P^{(m)}$ dont les centres de gravité de chaque classe sont donnés par $g^{(m)}$, d'après le cours l'inertie intraclasse est :

$$V_a = \sum_{k=1}^{K} p(\mathbf{C_k^{(m)}}) \sum_{i \in \mathbf{C_k^{(m)}}} \frac{p_i}{p(\mathbf{C_k^{(m)}})} \|X_i - g_k^{(m)}\|^2 = \sum_{k=1}^{K} \sum_{i \in \mathbf{C_k^{(m)}}} \frac{1}{n} \|X_i - g_k^{(m)}\|^2 = V_a(P^{(m)}, g^{(m)}).$$

2. Initialisation : on choisit K individus aléatoirement dans I, ce sont les centres initiaux.
   Etape $m+1$: on suppose l'étape m de l'algorithme réalisée :
   (a) On forme la partition $P^{(m+1)}$ en regroupant les individus de I : autour de chaque centre de gravité $g_k^{(m)}$ sont associés les individus qui sont plus proches de $g_k^{(m)}$ que des autres centres.

(b) On calcul les centres de gravité $g^{(m+1)}$ associés à la partition $P^{(m+1)}$.

On arrête l'algorithme quand la partition obtenue n'évolue plus d'une étape à l'autre.

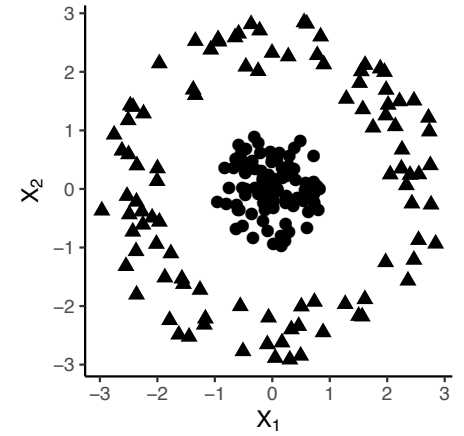
3. On a

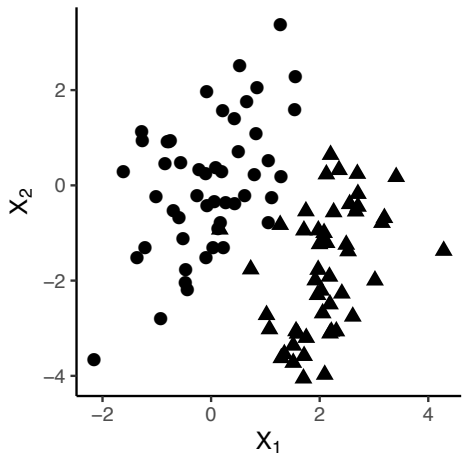
$$\|X_i - g_k^{(m)}\|^2 = \|X_i - g_k^{(m+1)}\|^2 + \|g_k^{(m+1)} - g_k^{(m)}\|^2 + 2\langle X_i - g_k^{(m+1)}, g_k^{(m+1)} - g_k^{(m)}\rangle$$

On somme sur tous les individus de la classe $C_k^{(m+1)}$ l'égalité ci-dessus. Le dernier terme de la partie droite de l'égalité s'annule et on déduit le résultat voulu.

4. C'est une conséquence de l'étape (a) de l'algorithme, l'individu $i \in C_k^{(m+1)}$ est par construction plus proche du point $g_k^{(m)}$ que du point $g_k^{(m+1)}$.
5. Qu 3+4.
6. L'inertie intraclasse décroît à chaque étape et est minorée par 0, elle converge donc. Au bout d'un grand nombre d'étapes l'algorithme de Forgy converge vers une partition correspondant à un minimum local (donc à un maximum local) de l'inertie intraclasse (resp. interclasse).

Frontière des 3 plus proches voisins





Frontière de CART