Université de Technologie de Compiègne
SY09:Analyse des données et Data-Mining , P17

Rendu TP3

Zineb Slam, Oumaima Talouka

1er juin 2017

1 Classifieur euclidien, K plus proches voisins

1.1 Programmation

1.1.1 Classifieur Euclidien

ceuc.app est une fonction qui retourne les centres d'inertie de chaque classifieur. La fonction *rowsum* sur R nous a été très utile pour sommer les lignes. La fonction *ceuc.val* prédit classe chaque individu des données tests. Nous utilisons ici la fonction *distXY* pour calculer qui sépare chaque individu a chacun des centres d'inertie.

1.1.2 K Plus proches Voisins

Dans cette partie nous avons essaye de programmer l'algorithme des k plus proches voisins en évitant les boucles.

1.1.3 Test de fonctions

1.2 Évaluation des Performances

1.2.1 Estimation des paramètres

Les résultats sont affichés dans Le tableau ci-dessous :

Estimation des Parametres				
Jeu de données	k	μ_k	$\sum_k$	π_k
Synth1-40	1			
	2			
Synth1-100	1			
	2			
Synth1-500	1			
	2			
Synth1-1000	1			
	2			

1.2.2 Calcul du taux d'erreur du classifieur Euclidien

Dans cette partie nous allons tester les performances de chacun des algorithmes programmes precedement en utilisan le critère de l'erreur qu'on exprime comme :

$$Erreur = \frac{1}{N} \sum_{i=1}^{n} \mathbb{1}\hat{z}_i \neq z_i$$

Avec z_i la vrai valeur de z et $\hat{z}_i$ la valeur calculée a partir de l'algorithme. N est le nombre d'individus. $\mathbb{1}\hat{z}_i \neq z_i = 1$ si $\hat{z}_i = z_i$, 0 sinon. Après avoir calculé le taux d'erreur on peut utiliser la formule ci-dessous pour calculer l'intervalle de confiance :

$$Ic = [\mu_\epsilon - t\frac{\sigma_\epsilon^2}{\sqrt{N}}, \mu_\epsilon + t\frac{\sigma_\epsilon^2}{\sqrt{N}}]$$

On choisit un niveau de confiance t de 95%. Notons ici que μ et σ sont ceux calculés pour les différentes valeurs de ϵ et non ceux du jeu de données. Autre point a remarquer est le N qui est ici le nombre d'individus , or comme on a sépare notre jeu de données entre un ensemble d'application et un ensemble de test, on veillera a mettre le nombre d'individus correspondant en fonction si c'est l'intervalle de confiance d'application ou test. D'après l'énoncé en utilisant

la fonction *separ1* napp $=\frac{2n}{3}$ et ntst$=\frac{n}{3}$ n étant le nombre total d'individus du jeu de donnée.

Les résultats obtenus pour les jeux de données sont affichés dans Le tableau qui suit.

Performance du Classifier Euclidien		
Jeu de données	ϵ_{test}	Ic_{test}
Synth1-40		
Synth1-100		
Synth1-500		
Synth1-1000		

1.2.3 Nombre optimal des k voisins

1.2.4 Calcul du taux d'erreur du KPP

De la même manière aussi on traduit les résultats dans le tableau qui suit.

Performance du KPP avec nombre de voisins =		
Jeu de données	ϵ_{test}	Ic_{test}
Synth1-40		
Synth1-100		
Synth1-500		
Synth1-1000		

1.2.5 Jeux de données Synth2-1000

Estimation des Paramètres				
Jeu de données	k	μ_k	$\sum_k$	π_k
Synth2-1000	1			
	2			

Performance du KPP avec nombre de voisins =		
Jeu de données	ϵ_{test}	Ic_{test}
Synth2-1000		

1.2.6 Jeux de données réelles

Estimation des Paramètres				
Jeu de données	k	μ_k	$\sum_k$	π_k
Pima	1			
	2			

Performance du KPP avec nombre de voisins =		
Jeu de données	ϵ_{test}	Ic_{test}
Pima		

Estimation des Paramètres				
Jeu de données	k	μ_k	$\sum_k$	π_k
Breast Cancer	1			
	2			

Performance du KPP avec nombre de voisins =		
Jeu de données	ϵ_{test}	Ic_{test}
Breast Cancer		

2 Règle de Bayes

3 Conclusion