4A : INGENIERIE DES DONNEES

TP3 (durée : 3h00)
Analyse en composantes principales

Le but de ce TP est d'étudier l'influence de la réduction de dimension par analyse en composantes principales sur les performances reconnaissance de caractères manuscrits et en compression d'image. Les *fonctions en italique* sont des fonctions de la bibliothèque sklearn. Vous pouvez facilement trouver des informations détaillées sur ces fonctions.

Attention : le texte du TP comporte DEUX pages ☺

Analyse de la base de données

On dispose d'une base de données contenant 1797 images des 10 chiffres manusrits.

1. Charger la base de données *digits* disponible dans sklearn.

   >> from sklearn.datasets import load_digits
   >> digits = load_digits()

   Déterminer la dimension *Dim* des données et le nombre d'exemple par classe. Observer quelques images :

   >> import matplotlib.pyplot as plt
   >> plt.gray()
   >> plt.matshow(digits.images[index]) #index est le numéro de l'image
   >>plt.show()

   Pour récupérer les données et les labels :
   >> X = digits.data
   >> y = digits.target

2. Séparer la base initiale en deux : apprentissage (70%) et test (30%) (*model_selection.train_test_split*).
   Utiliser l'algorithme du plus-proche-voisin pour classer les exemples de la base de test.

   >> NN1 = KNeighborsClassifier(n_neighbors=1)
   >> NN1.fit(X_train, y_train)

   Répeter plusieurs fois la procédure de partition-classification. Conclure.

Analyse en composantes principales

1. Utiliser le solveur PCA de sklearn pour réaliser la PCA.

   >> from sklearn.decomposition import PCA
   >> pca = PCA()
   >> pca.fit(X_train)

   Pourquoi n'utiliser que la base d'apprentissage (et pas toutes les données) pour faire ce traitement ?

2. Tracer le graphe de l'inertie expliquée en fonction du rang de la valeur propre.
   Peut-on appliquer le critère de Catell pour déterminer le nombre optimal de composantes à conserver ?

3. Calculer l'inertie cumulée en considérant M axes ($M \leq Dim$). Tracer le graphe de l'inertie cumulée en fonction du rang de la valeur propre.
   Appliquer le critère de Joliffe (à 90%, par exemple) pour déterminer le nombre optimal de composantes à conserver.

Classification

1. Evaluer, pour les valeurs de M trouvées précédemment, le taux de reconnaissance obtenu par un classifieur 1-ppv lorsque les données (bases de référence et de test) sont codées par les M premières composantes principales.

   Pour ce faire, on projettera les données de l'espace initial dans l'espace de dimension à l'aide de la matrice de passage réduite P_M avant d'appliquer l'algorithme de classification

2. Etendre l'analyse précédente en faisant varier M de 1 à Dim. Tracer le graphe montrant la variation du taux de reconnaissance en fonction de M. Conclure.

Compression

1. Compresser une image sur M composantes principales à l'aide de la matrice de passage réduite P_M puis, reconstruire une image (de dimension Dim) à l'aide de ces M composantes principales. Conclure.

2. Calculer l'erreur de reconstruction : erreur normalisée entre l'image originale et l'image reconstruite.

3. Etendre l'analyse précédente en faisant varier M de 1 à Dim. Tracer le graphe montrant, pour une image, la variation de l'erreur de reconstruction en fonction de M.