Classification des chiffres manuscrits

Présentation des données

La reconnaissance de caractères manuscrits par un système automatisé est un problème aux multiples applications : reconnaissance des codes postales, création de petits systèmes mobiles de saisie de texte (par exemple, pour les ordinateurs de poche), numérisation de documents manuscrits, ...

Les méthodes les plus performantes pour reconnaitre un caractère manuscrit sont basées sur des méthodes d'apprentissage statistique : leur principe commun est de fonder leur prédiction sur la comparaison de l'image du caractère manuscrit à classer à d'autres images de caractères manuscrits pour lesquels la nature du caractère est connue. Typiquement, si une image arrive et qu'elle est très similaire à l'image de nombreux '1' de notre base d'apprentissage, l'algorithme classera l'image dans la catégorie '1'.

Les données considérées ici proviennent de la base MNIST (http://yann.lecun.com/exdb/mnist/) sur laquelle plusieurs chercheurs ont travaillé. Elle est constituée de 70 000 chiffres manuscrits au format 28 pixels par 28 pixels où chaque pixel est représenté par un niveau de gris allant de 1 à 256 (i.e. un chiffre manuscrit est donc un vecteur de $\{1, \ldots, 256\}^{28\times 28}$).

Dans ce TD, pour limiter le temps de calcul et la mémoire nécessaire, nous ne considérons que 800 chiffres manuscrits (que des '3' et des '5') : 400 chiffres manuscrits seront utilisés pour apprendre, i.e. calibrer l'algorithme. Ces 400 chiffres manuscrits et la classe ('3' ou '5') qui leur est associée constituent l'ensemble d'apprentissage. Les 400 autres chiffres manuscrits seront uniquement utilisés pour évaluer la qualité des algorithmes. Ces 400 caractères manuscrits constituent l'ensemble de test.

On dispose d'une matrice de taille (800, 1 + (28 × 28)) correspondant aux 800 caractères manuscrits où

- chaque ligne correspond à un caractère manuscrit
- les premières colonnes contiennent les valeurs des 28 × 28 pixels en commençant par le coin supérieur gauche et parcourant l'image ligne par ligne.
- la dernière colonne contient la classe du caractère (c'est un texte qui vaut soit trois soit cinq).

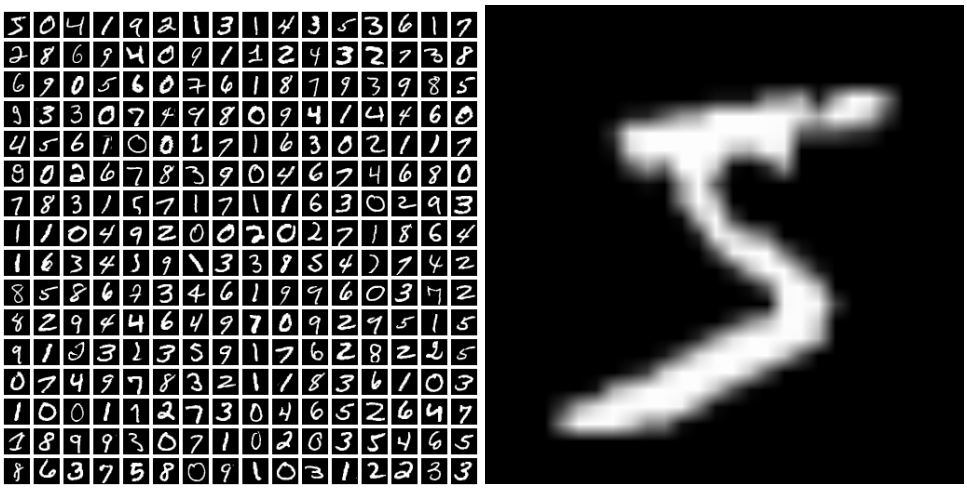

Figure 1 – A gauche : exemple de caractères manuscrits 28 × 28. A droite : zoom sur le premier caractère manuscrit 28 × 28. Ce caractère manuscrit appartient à la classe 'cinq'.

Classification avec Weka

Dans un premier temps, nous allons tester quelques uns des algorithmes d'apprentissage supervisé présents sur Weka. Pour cela :

1. Lancer le logiciel Weka (java ….) puis, depuis le bouton Explorer, charger le fichier chiffres.arff. Vous pouvez commencer par Editer le fichier et visualiser son contenu (bouton Edit).
2. Via l'onglet Classify, tester les différents algorithmes de classification. En particulier, la méthodes Naive Bayes et la méthode SVM.
3. Pour chacune des méthodes, évaluer la qualité des classifieurs utilisés pour ce problème en particulier. Observer les résultats obtenus en utilisant la cross-validation puis une partition jeux d'entrainement/jeux de test.

Entrainement et prédiction

A présent, nous allons entrainer nos modèles puis les utiliser pour prédire les classes de nouvelles instances. Sous Weka, entrainer votre machine à reconnaître les 3 et les 5 puis enregistrer le modèle obtenu.
Ensuite, utiliser ce modèle pour classifier les données dans le fichier unclassified.arff :

```
java –cp /opt/weka.jar weka.classifiers.bayes.NaiveBayes -T unclassified.arff -l modelBayes.model -p 0
```