Classement de caractère manuscrits

La reconnaissance de caractères manuscrits par un système automatisé est un problème aux multiples applications: reconnaissance des codes postales, création de petit système mobile de saisie de texte (par exemple, pour les ordinateurs de poche), numérisation de documents manuscrits, etc. Les méthodes les plus performantes pour reconnaître 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. Par exemple, 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'.

Nous allons évaluer l'algorithme kNN et Naïve Bayes sur une base de données d'images de chiffres manuscrits au format 28 pixels par 28 pixels où chaque pixel est représenté par un niveau de gris allant de 0 à 255 (i.e. un chiffre manuscrit est un vecteur de $\{0, \ldots, 255\}^{28\times 28}$).

1 Visualiser les Données

- Téléchargez le fichier de données "mnist_all.mat"
- Suivez les instructions suivantes pour visualiser le premier chiffre manuscrit.

```
loadmatfile('mnist_all.mat');
y=matrix(train0(1,:),28,28)';
Matplot(y)
```

- Comment peut-on modifier le code pour visualiser les autres chiffres manuscrits? 1%

Attention: Mettez la taille de pile au max, i.e. stacksize('max').

2 L'algorithme de plus proche voisin ou kNN

Le but de cette partie est de réaliser l'algorithme kNN comme méthode de classification. Nous allons pour cela utiliser la distance Euclidienne. La distance Euclidienne $d(\mathbf{x}, \mathbf{y})$ entre

deux vecteurs $\mathbf{x} := (x_1, x_2, \ldots, x_n)$ et $\mathbf{y} := (y_1, \ldots, y_n)$ est donnée par

$$d(\mathbf{x}, \mathbf{y}) := \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

1. Est-ce que les résultats changent si $d^2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} (x_i - y_i)^2$ est utilisée comme distance? 2%

2. Complétez le code de kNN.sci et tester le pour la classification des chiffres manuscrits. Soit $\mathbf{y} := (y_1, \ldots, y_m)$ le vecteur contenant les vraies étiquettes et $\hat{\mathbf{y}} := (\hat{y}_1, \ldots, \hat{y}_m)$ vecteur contenant les classes estimées par l'algorithme. On va calculer l'erreur de classement par la formule suivante

$$e := \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\{\hat{y}_i \neq y_i\} \tag{1}$$

   où pour une expression logique P on a $\mathbb{I}\{P\} := 1$ si P est vrai, et $\mathbb{I}\{P\} := 0$ sinon. Quelles sont les erreurs moyennes de classification pour $k = 1, 3, 5, 10$? 6%

3. Vrai ou faux? "Cette approche est supervisée." Expliquez pourquoi. 2%

3 L'algorithme Naïve Bayes

Nous allons évaluer la performance de l'algorithme Naïve Bayes pour la classification des chiffres manuscrits.

1. Quelle est l'hypothèse principale dans cette approche? 2%

2. Complétez le code de NB.sci et testez le avec $n = 200$ et $n = 1000$ exemplaires d'entrainements. (Calculez les erreurs de classification avec la formule donnée par l'équation (1)). 8%

3. Vrai ou faux? "Cette approche est supervisée." Expliquez pourquoi. 2%

4. Vrai ou faux? "Cette approche est probabiliste." Expliquez pourquoi. 2%

Remarque

Ce problème comporte 10 classes $C_1, \ldots, C_{10}$ qui correspondent aux chiffres $0, \ldots, 9$ respectivement. Dans le code NB.sci, chaque point d'entrainement, est une ligne de la matrice x_train, composée de 728 éléments. Rappelez-vous que dans notre code la matrice x_train est fabriquée de manière à ce que les lignes $1..n$ correspondent aux images de la première classe, les lignes $n + 1..2n$ correspondent aux images de la deuxième classe, les lignes $2n + 1..3n$ correspondent aux images de la troisième classe, et ainsi de suite. *(Le paramètre n est fixé à* 200 *dans le code).*

1. Comment utiliser 'x_train' pour calculer les probabilités conditionnelles? Dans chaque classe calculons la fréquence de chaque valeur $0, 1, 2, ..., 255$ quelque soit le point d'entrainement et quel que soit l'emplacement de cette valeur dans le point. Ceci va donner une *matrice de fréquences* de taille 10×256, où chaque ligne correspond aux fréquences des valeurs différentes des pixels, i.e. 0, 1, ...,255 dans une des classes.

2. Soit $\mathbf{x} :=$ x_test$[j] = x_1, x_2, \ldots, x_m$ la j'ième ligne de la matrice de x_test.

   - Comment calculer $P(\mathbf{x}|C_i)$ pour chaque $i = 1..10$?
     En utilisant la matrice des fréquences et l'hypothèse suivante

     $$P(\mathbf{x} := x_1, x_2, ..., x_m|C_i) = \prod_{l=1}^{m}(x_l|C_i)$$

     sur laquelle est basé Naive Bayes. *(Remarquez qu'ici $m := 728$.)* Chaque élément x_l, $l = 1..m$ peut prendre une valeur entre 0..255. Donc $P(x_l|C_i)$ est la fréquence de x_l donnée par la ligne i de la matrice des fréquences. Par exemple si $x_l = 0$ utilisons la fréquence de '0' dans la classe C_i comme $P(0|C_i)$, si $x_l = 1$ utilisons la fréquence de '1' dans la classe C_i comme $P(1|C_i)$ et ainsi de suite.
   - Comment classifier $\mathbf{x}$? Calculons

     $$\text{argmax}_{i=1..10} \log P(\mathbf{x}|C_i) = \sum_{l=1}^{m} \log P(x_l|C_i)$$

À rendre

Un fichier [votre_Prénom_Nom].zip contenant:

- Vos codes, i.e. kNN.sci et NB.sci que vous avez complétés
- Un fichier .pdf ou .txt contenant de vos réponses aux toute les questions

(Envoyer par e-mail à azadeh.khaleghi@inria.fr)
La date limite de rendu est le mardi 2 février 2013 à 23:59