TP Apprentissage : Apprentissage d'une catégorie d'images par classification à base de noyaux

1 Objectif

Le but de ce TP est d'utiliser les SVM en classification pour faire la discrimination, d'abord entre deux séries de chiffres (classification binaire), puis entre un chiffre et tous les autres (version multiclasse).

1.1 Outils

- Le code sera en Matlab.
- Les SVM utilisés sont ceux de la boîte à outils libSVM (disponible gratuitement en ligne).
- Les données sont issues de la base de données USPS (fournie avec le TP). Cette base de données contient des images représentant des chiffres manuscrits isolés sur des codes postaux. Cette base est utilisée classiquement pour l'évaluation des méthodes d'apprentissage.

1.2 Etapes

- Préparer la base d'apprentissage et de test.
- Mettre en œuvre le SVM.
- Tester différents réglages du SVM (noyau, relâchement des contraintes C) afin d'obtenir la meilleure discrimination.
- Faire varier les couples de classes à discriminer (on peut chercher à classer les chiffres de 0 à 4 contre ceux de 5 à 9 par exemple).
- Etudier le passage au multiclasse (cf. documentation de libSVM) et tester la performance obtenue (à titre indicatif, l'erreur humaine sur cette base de données est de 2.9%)

2 Préparation des données

1. Ouvrir la base de données usps.mat disponible dans le répertoire Data et en extraire deux séries de chiffres.
2. Créer votre base d'apprentissage : celle-ci est constituée des toutes les données de la base d'apprentissage d'origine ayant pour étiquettes les deux chiffres choisis.
3. Créer votre base de test : idem à partir de la base de test d'origine.

Commande utile :

```
A = find(train_labels==c1);
```

3 SVM binaire et réglage

1. Lire la documentation (en ligne) de libSVM pour connaître les options disponibles et les sorties du SVM.
2. Lancer l'apprentissage en classification type C-SVM avec validation croisée.
3. Tester le modèle appris sur vos données de test et afficher les images mal reconnues.

4 SVM et multiclasse

Il y a deux manières classiques de transformer un algorithme d'apprentissage binaire en une méthode multiclasse : le 1-vs-1 et le 1-vs-all.

Dans le cas du 1-vs-1, on apprend un modèle binaire par couple possible (1-2, 1-3, 1-4, 2-3, 2-4, 3-4 si on a 4 classes par exemple). Pour faire le test d'une nouvelle image, chaque classifieur est interrogé et on attribue la classe la plus fréquement obtenue (système de vote).

Dans le cas du 1-vs-all, on apprend un modèle binaire par classe. Pour chaque classifieur, la classe positive est la classe que l'on cherche à reconnaître et la classe négative est constituée de tous les exemples de toutes les autres classes. Pour faire le test d'une nouvelle image, chaque classifieur est interrogé et on attribue la classe du classifieur qui renvoie la plus grande valeur (dans l'idéal, un seul des classifieurs renvoie une valeur positive).

1. Lire la documentation (en ligne) de libSVM pour appliquer le multiclasse.
2. Lancer l'apprentissage en classification type C-SVM avec validation croisée (sur la base complète, avec toutes les classes).
3. Tester le modèle appris sur vos données de test et afficher les images mal reconnues.

Commandes utiles :

```
addpath('../Toolbox/libsvm-mat-2.88-1/');
model = svmtrain(trainlab, trainvec, '-s 0 -c 1 -g 0.1');
[predict_label, accuracy, dec_values] = svmpredict(testlab, testvec, model);
```