Méthodes à noyaux

TP : SVR et SVM

Compte-rendu à rendre via l'ENT

1 Préparation

Récupérez la base usps sur l'ENT et chargez la:

```
uspsXapp = np.load('uspsXapp.npy')
uspsYapp = np.load('uspsYapp.npy')
```

Cette base contient plus de 7000 images de chiffres manuscrits. Pour visualiser une image, utilisez le code suivant

```
nimg = 456
im = uspsXapp[:,nimg]
lab = uspsYapp[nimg]-1

plt.imshow(np.reshape(im,(16,16)))
plt.title('Training: %i' %lab)
plt.show()
```

Notez que le choix d'utiliser la première image est complètement arbitraire est vous pouvez donc utiliser n'importe laquelle pour la suite du TP, voire en essayer plusieurs.

2 SVR, Support Vector Regression

Dans cette première partie, nous allons utiliser les SVM en regression (voir annexes).

2.1 Objectif

L'objectif va être de reconstruire une image dont des pixels sont perdus (voir figure 1).

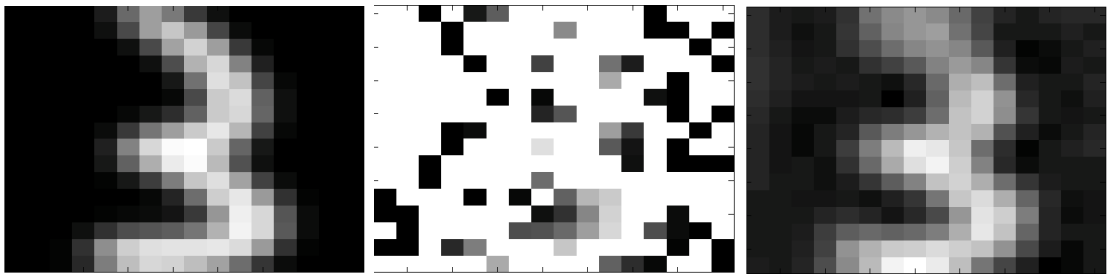

Figure 1: A gauche, l'image originale, au centre, l'image avec seulement 10% de pixels connus et à droite l'image reconstruite par regression

2.2 Méthode

1. construire la base de données: ici, les données sont les coordonnées des pixels, et les étiquettes sont les valeurs des pixels. Il faut donc construire une matrice X de dimension 2*nbPixels contenant tous les couples de coordonnées. Y sera un vecteur contenant toutes les valeurs des pixels correspondant aux coordonnées.
   ```
   [x1,x2] = np.meshgrid(range(16),range(16))
   trainvecT = np.concatenate((np.reshape(x1,(1,16*16)),np.reshape(x2,(1,16*16))),axis=0)
   trainlabT = im
   ```
2. découper la base créée en une base d'apprentissage et une base de tests. Ici il n'y a pas de répartition particulière à respecter comme dans la classification. Les points placés dans la base de test correspondent aux pixels manquants.
3. faire un premier essai de regression
4. faire varier le taux de pixels manquants (donc augmenter ou diminuer la taille de l'ensemble d'apprentissage) afin d'observer la robustesse de la méthode.

2.3 Mettre en place une validation croisée

Les SVR ont plusieurs paramètres:

- la largeur de bande si on utilise un noyau gaussien
- C le paramètre de régularisation
- et en plus ϵ qui règle la largeur d'un tube dans lequel doivent passer les données (notion de marge). Voir annexe.

L'objectif est donc de trouver le meilleur triplet de paramètres.

3 SVM

La base USPS contient 10 classes, les chiffres de 0 à 9. Pour cette partie, le SVM va être utiliser en mode classique, à savoir la classification binaire.

1. Constituer une base comprenant 300 exemples de 2 classes (au choix, mais évitez le 4 contre 9 qui est particulièrement difficile!), soit 600 exemples. Définissez également le vecteur y. Les exemples restants des deux classes formeront la base de test.
2. Mettre en place une validation croisée afin de de déterminer le meilleur couple de paramètres (σ,C) pour le noyau Gaussien. Donner la performance en test.
3. Affichez les exemples mal classés (il sera utile de connaitre la commande `subplot`).
4. Observation de l'influence des paramètres (sans validation croisée)
   (a) Fixez C selon la valeur optimale de la question précédente. Faites varier σ et affichez l'évolution de la performance en apprentissage et en test ainsi que le temps d'apprentissage. Commentez.
   (b) Fixez σ selon la valeur optimale de la question précédente. Faites varier C et afficher l'évolution de la performance en apprentissage et en test ainsi que le temps d'apprentissage. Commentez.

4 Bonus

L'état de l'art sur cette base est

- erreur humaine : entre 1.5% et 2.5% selon les études
- méthode utilisant des techniques avancées telle que la déformation des exemples (pour avoir plus de variété en apprentissage) : de 2.6% à 3%

- SVM : 4%
- kNN : 5.9%

Mettez en oeuvre un SVM multiclasses et essayez d'arriver à l'état de l'art sur la vraie base de test (sans tricher!). Vous pouvez dans cette phase tester d'autres noyaux. Attention les temps de calculs seront plus longs.

A SVR

Le SVR (Support Vector Regression) est l'adaptation du SVM au cas de la regression. L'idée est de trouver un tube de rayon ϵ qui explique au mieux les données (voir figure 2).

$$\begin{cases} \min\limits_{f\in\mathcal{H}} & \frac{1}{2}\|f\|^2_{\mathcal{H}} + C\sum|\xi_i| \\ \text{tel que} & |f(x_i) - y_i| \leq \epsilon + \xi_i \quad 0 \leq \xi_i \quad i = 1, n \end{cases}$$

L'erreur se mesure comme étant la sommes des carrés des différences entre les prévisions et

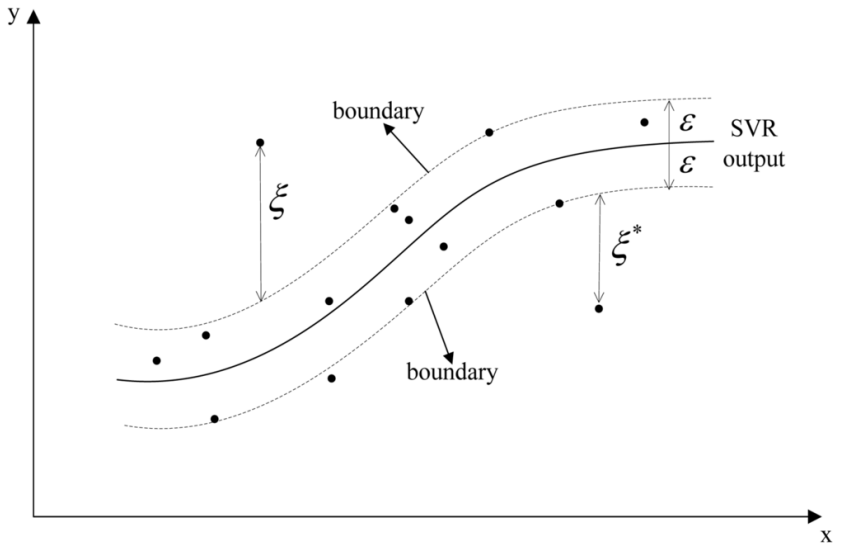

Figure 2: Support Vector Regression

les valeurs attendues. On peut également prendre comme critère la correlation entre l'ensemble des prévision et les valeurs attendues.