

TP Apprentissage

Reconstruction d'image par Regression à base de noyaux

1 Objectif

Le but de ce TP est d'utiliser les SVM en regression (aussi appelés SVR pour *Support Vector Regression*) pour :

- reconstituer les pixels manquants dans une image ;
- synthétiser un chiffre à partir d'une base de données de ces chiffres.

Les SVM utilisés sont ceux de la boîte à outils libSVM. Comme dans le TP précédent, les données sont issues de la base de données USPS. Cette base de données contient des images représentant des chiffres manuscrits isolés sur des codes postaux.

2 Préparation des données

La base complète contient 4 matrices : un couple données/étiquettes pour l'apprentissage, un couple données/étiquettes pour le test. Les données (train_patterns et test_patterns) se retrouvent ligne par ligne (une ligne contient une image). Il est donc nécessaire pour visualiser une image de réarranger le vecteur ligne en une matrice (16×16 pixels).

1. Ouvrir la base de données usps.mat disponible dans le répertoire Data et :
 - en extraire une image représentant un chiffre. Cette image servira de source de données pour la reconstruction de pixels manquants
 - en extraire toutes les images d'un chiffre donné, qui serviront pour la synthèse d'une image de ce chiffre.
 - Pour la reconstruction de pixels
 - sélectionner les pixels à supprimer (tirer au hasard $p\%$ des pixels de l'image). Afficher l'image d'origine et l'image tronquée.
 - Créer votre base d'apprentissage : celle-ci est constituée des coordonnées de tous les pixels non supprimés. Les étiquettes correspondantes sont les niveaux de gris de l'image d'origine à ces coordonnées.
 - Créer votre base de test : celle-ci est constituée des coordonnées de tous les pixels supprimés. Les étiquettes correspondantes sont les niveaux de gris de l'image d'origine à ces coordonnées.
 - Pour la synthèse d'image
 - sélectionner $p\%$ des images de ce chiffre dans la base d'apprentissage
 - Créer votre base d'apprentissage : celle-ci est constituée des vecteurs des niveaux de gris de chaque position des pixels (pour un pixel i donné, le label est donc un vecteur de valeurs, correspondant aux niveaux de gris de i dans les images de la base d'apprentissage).

Pour vous aider, quelques commandes utiles.

```
load('...');  
reshape(X,nb_lignes,nb_colonnes);  
imagesc(X);  
colormap('gray');  
[x1,x2] = meshgrid(1:size(X,1),1:size(X,2));  
rn = randperm(n);
```

3 Ce qui est demandé

1. Lire la documentation (en ligne) de libSVM pour connaître les options disponibles et les sorties du SVM/SVR
2. Pour la reconstruction de pixels manquants : lancer l'apprentissage en régression pour une image avec $10\% \leq p \leq 90\%$ de pixels manquants, un noyau de type 'rbf' (gaussien) avec largeur de bande γ , et des paramètres du SVR C , ϵ à déterminer par validation croisée. Attention ici à adapter la procédure qui vous a été fournie dans le TP précédent (pensez à regarder ce que signifient les composantes de accuracy, sortie de la fonction svm-predict). Tester le modèle appris sur vos données de test et afficher l'image complétée. Tracer une courbe évaluant la qualité de la reconstruction en fonction de p . Proposez une méthode d'évaluation de la qualité de l'image reconstruite.
3. Pour la synthèse d'image de chiffre : lancer le même type d'apprentissage avec $10\% \leq p \leq 90\%$ images de ce chiffre, et prédisez une image de ce chiffre à l'aide du SVR appris. Proposez une méthode d'évaluation de la qualité des images synthétisées.

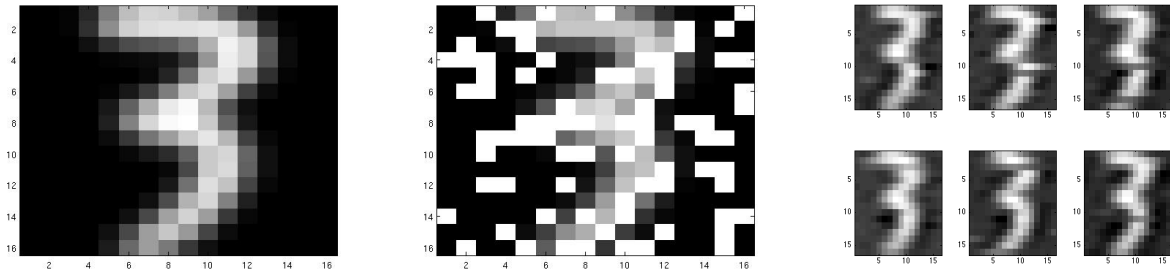


FIGURE 1 – A gauche : l'image complète. Au milieu : l'image avec pixels manquants (ici 30%). A droite : synthèse de quelques 3