IFT 603 : Devoir 4
Travail individuel

Remise : 10 avril 2015, 17h00 (**au plus tard**).

Remettez votre solution au numéro 1 en format papier et aux numéros 2 et 3 via *turnin*, comme suit :

```
turnin -c ift603 -p devoir_4 solution_processus_gaussien.py solution_acp_noyau.py
```

1. **[2 points]** Démontrez que, en supposant que les modèles $y_m(\mathbf{x})$ d'un ensemble ont des erreurs signées $\epsilon_m(\mathbf{x})$ d'espérance nulle et non-corrélées, i.e. :

$$\mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})\right] = 0, \quad \forall m \neq l \tag{1}$$

   alors on a que

$$E_{\text{COM}} = \mathbb{E}_{\mathbf{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M}\epsilon_m(\mathbf{x})\right\}^2\right] = \frac{1}{M^2}\sum_{m=1}^{M}\mathbb{E}_{\mathbf{x}}\left[\epsilon_m(\mathbf{x})^2\right] = \frac{1}{M}E_{\text{AV}} \ . \tag{2}$$

2. **[4 points]** Programmez l'algorithme de l'analyse en composantes principales à noyau, basée sur un noyau gaussien :

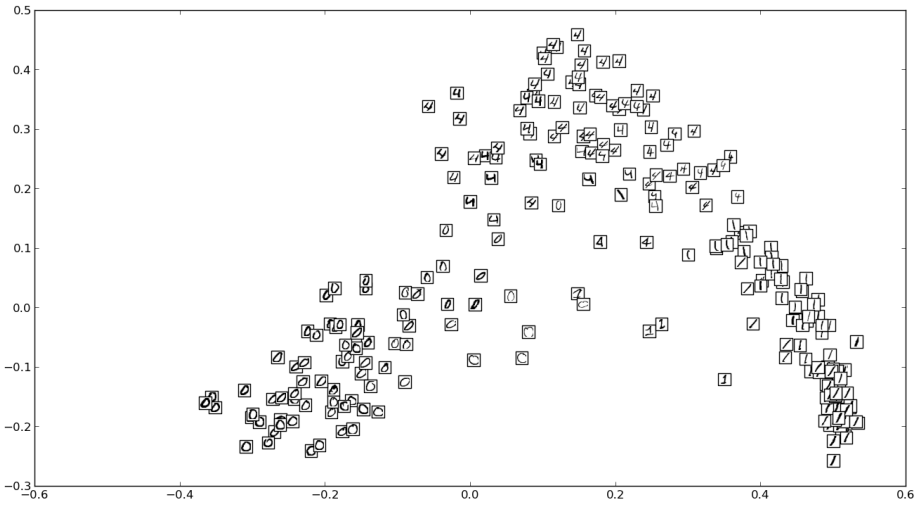
$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right) \ . \tag{3}$$

   L'algorithme doit être implémenté sous la forme d'une classe `ACPNoyau`. Votre implémentation de cette classe doit être placée dans le fichier `solution_acp_noyau.py`, qui contient déjà une ébauche de la classe (fichier disponible dans l'archive `devoir_4.zip` du site web du cours). Veuillez vous référer aux "docstrings" (la chaîne de caractères sous la signature de chaque méthode) des méthodes de la classe `ACPNoyau` afin de savoir comment les implémenter.

   Le fichier `solution_acp_noyau.py` sera importé par le script `acp_noyau.py`, qui exécute votre code sur les données d'entraînement et affiche le résultat de la réduction de dimensionnalité. Ce script nécessite également que les fichiers suivants soient présents dans le même répertoire :

   - Données d'entraînement et de test :
     - `ensemble_entrainement_acp.pkl`
     - `ensemble_test_acp.pkl`
   - Fichier de comparaison avec une implémentation correcte :
     - `solution_predictions_test_acp.pkl`

Les données correspondent à des images de caractères manuscrits 28×28. Chaque caractère correspond à un "0", "1" ou "4". Une implémentation correcte donnera la visualisation de la réduction de dimensionnalité suivante :



Dans votre implémentation, vous aurez besoin d'extraire des valeurs et vecteurs propres. Veuillez vous référer à la démonstration de l'ACP du cours, accessible en format *iPython notebook*.

3. **[4 points]** Programmez l'algorithme de la régression basée sur un processus gaussien, utilisant un noyau gaussien.

   L'algorithme doit être implémenté sous la forme d'une classe `ProcessusGaussien`. Votre implémentation de cette classe doit être placée dans le fichier `solution_processus_gaussien.py`, qui contient déjà une ébauche de la classe. Veuillez vous référer aux "docstrings" (la chaîne de caractères sous la signature de chaque méthode) des méthodes de la classe `ProcessusGaussien` afin de savoir comment les implémenter.

   Le fichier `solution_processus_gaussien.py` sera importé par le script `processus_gaussien.py`, qui exécute votre code sur les données d'entraînement et mesure la performance du modèle de régression sur les ensembles d'entraînement et de test. Ce script nécessite également que les fichiers suivants soient présents dans le même répertoire :

   - Données d'entraînement et de test :
     - `ensemble_entrainement.pkl`
     - `ensemble_test.pkl`
   - Fichiers de comparaison avec une implémentation correcte :
     - `solution_predictions_test.pkl`
     - `solution_variances_test.pkl`
     - `solution_erreurs_test.pkl`

   Le script gère donc déjà le chargement des données. Les données sont les mêmes que pour le Devoir 1. Elles ont été extraites du jeu de données *Housing Data Set*[1]. La tâche associée à ces données est celle d'apprendre à prédire le prix médian de maisons dans différents quartiers de la région de Boston, à

1. Pour en savoir plus, voir `http://archive.ics.uci.edu/ml/datasets/Housing`.

partir d'information telle le taux de criminalité, la proportion d'enfants par enseignant à l'école du quartier, etc.

Une implémentation correcte obtiendra une erreur d'entraînement de 3.04 et une erreur de test de 31.02. Remarquez que les résutlats sont meilleurs que ceux du Devoir 1, utilisant la régression linéaire pour laquelle des erreurs d'entraînement de 17.59 et de test de 42.96 avaient été obtenues.