IFT 603 : Devoir 1
Travail individuel

Remise : 7 février 2014, 17h00 (**au plus tard**).

Remettez votre solution aux numéros 1, 2 et 3 en format papier et au numéro 4 via *turnin*.

1. **[1 points]** Démontrez la propriété de l'entropie suivante :

$$\mathrm{H}[\mathbf{x}, \mathbf{y}] = \mathrm{H}[\mathbf{y}|\mathbf{x}] + \mathrm{H}[\mathbf{x}]$$

   Vous pouvez faire la démontration pour le cas discret ou continu.

2. **[1 points]** Démontrez la propriété de l'information mutuelle suivante :

$$\mathrm{I}[\mathbf{x}, \mathbf{y}] = \mathrm{H}[\mathbf{x}] - \mathrm{H}[\mathbf{x}|\mathbf{y}]$$

   Vous pouvez faire la démontration pour le cas discret ou continu.

3. **[3 points]** Soit une variable aléatoire X pouvant prendre 3 valeurs possibles $\{1, 2, 3\}$ avec probabilités $p(X = 1) = p_1$, $p(X = 2) = p_2$ et $p(X = 3) = p_3$.
   Démontrez que la loi de probabilité ayant l'entropie la plus élevée **et** satisfaisant la contrainte $p_1 = 2p_2$ a les probabilités suivantes :

$$\begin{aligned} p_1 &= \frac{2}{(2^{2/3} + 3)} \\ p_2 &= \frac{1}{(2^{2/3} + 3)} \\ p_3 &= \frac{2^{2/3}}{(2^{2/3} + 3)} \end{aligned}$$

   Pour ce faire, utilisez des multiplicateurs de Lagrange afin de tenir compte de la contrainte de sommation à 1 **et** de la contrainte $p_1 = 2p_2$.

4. **[5 points]** Programmez l'algorithme de la régression linéaire. Pour ce faire, vous devez télécharger et décompresser le fichier `devoir_1.zip` du site web du cours.

   L'algorithme doit être implémenté sous la forme d'une classe `RegressionLineaire`. Votre implémentation de cette classe doit être placée dans le fichier `solution_regression_lineaire.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 `RegressionLineaire` afin de savoir comment les implémenter. Votre implémentation doit être efficace et utiliser les fonctionalités de la librairie Numpy (e.g. vous devez éviter les boucles `for`).

   Le fichier `solution_regression_lineaire.py` sera importé par le script `regression_lineaire.py`, qui exécute votre code sur les données d'entraînement et mesure la performance du modèle de régression linéaire 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_entrainement.pkl`
  - `solution_predictions_test.pkl`
  - `solution_erreurs_entrainement.pkl`
  - `solution_erreurs_test.pkl`

Le script gère donc déjà le chargement des données. Les données 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 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 17.59 et une erreur de test de 42.96.

Vous devez remettre votre solution via l'outil *turnin*, comme suit :

```
turnin -c ift603 -p devoir_1 solution_regression_lineaire.py
```

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