TP 2 - Régression linéaire

M2 IWOCS, Apprentissage Automatique

Exercice 1 - Régression linéaire univarié

Les données utilisées pour cet exercice décrivent les tailles d'enfants d'âge compris entre 2 et 8 ans : le fichier `ex1x.dat` correspond à leur âge et `ex1y.dat` correspond à leur taille (en mètres). On a dans ces 2 fichiers, les données de 50 enfants rangées dans le même ordre.

Ces données constituent des exemples d'apprentissage qui vont être utilisées afin de construire un modèle de régression linéaire qui a pour objectif de prédire la taille d'un enfant à partir de son âge.

1. Tout d'abord, nous allons utiliser la fonction `loadtxt` du package `numpy` en Python (ici désigné par `np`). Cette fonction permet de lire les fichiers et de les concertir en vecteurs/matrices :

   ```
   x=np.loadtxt('ex1x.dat');
   y=np.loadtxt('ex1y.dat');
   ```

2. Afficher le nuage de points (x_i, y_i).

3. Définir en Python, la fonction hypothèse correspondant à un modèle de régression linéaire. Soit le vecteur $\theta = (\theta_0, \theta_1)$, cette fonction hypothèse s'écrit
   $$h(\theta_0, \theta_1, x) = \theta_0 + \theta_1 x$$

4. Définir en Python, la fonction de coût $J(\theta_0, \theta_1)$ telle que vue en cours.

5. Définir en Python, une fonction qui réalise une itération et qui va renvoyer θ_0^* et θ_1^*, en fonction de θ_0 et θ_1, selon les formules de descente de gradient vue en cours. Pour faire fonctionner cette méthode de gradient, il faut définir la valeur du coefficient d'apprentissage noté α dans le cours ; il régle la profondeur de descente. On propose ici que ce coefficient prenne une valeur constante égale à 0,07. On partira aussi des valeurs initiales $\theta_0 = \theta_1 = 0$.

6. Faire tourner la méthode de descente de gradient sur quelques itérations puis représenter la droite
   $$y = \theta_0 + \theta_1 x$$
   sur le nuage de points.

7. Faire tourner la méthode de descente de gradient pendant toutes les itérations nécessaires à faire converger la solution θ recherchée. Pour cela on définit le critère

d'arrêt du processus itératif par

$$\left|\frac{J(\theta^*) - J(\theta)}{J(\theta)}\right| < 10^{-3}$$

Afficher la droite obtenue suite à cette convergence sur le nuage de points.

8. On peut désormais utiliser le modèle pour faire des prédictions : quelle est la taille de 3 enfants d'âges respectifs 3, 5 et 7 ans ?

9. Nous allons maintenant visualiser la fonction de coût $J(\theta)$ en 3D sur une grille de base 100 x 100 et des valeurs pour θ dans les intervalles suivants (il faudra donc prendre 100 valeurs réparties dans ces intervalles) : $\theta_0 \in [-30; 30]$ et $\theta_1 \in [-3; 3]$.

Exercice 2 - Régression linéaire multivariée et écriture matricielle de la méthode de descente du gradient

Dans ce problème, on utilise des données correspondant à 47 exemples d'apprentissage sur des données immobilières à Portland, Oregon (USA). Les données d'entrée x (stockées dans le ficheir `ex2x.dat`) correspondent aux surfaces et au nombre de pièces de ces 47 appartements et la donnée cible y (stockée dans le fichier `ex2y.dat`) correspond au prix de ces mêmes appartements.

1. Utiliser la fonction `loadtxt` du package `numpy` en Python (ici désigné par `np`). Cette fonction permet de lire les fichiers et de les concertir en vecteurs/matrices :

```
x=np.loadtxt('ex2x.dat');
y=np.loadtxt('ex2y.dat');
```

2. Prétraitement des données : utiliser la fonction `sklearn.preprocessing.StandardScaler` pour normaliser les données.

3. Définir les fonctions en Python sous forme matricielle pour représenter la fonction hypothèse $h_\theta(X)$, le vecteur tel que défini dans le cours :$E = h_\theta(X) - Y$, la fonction décrivant une itération et la fonction de coût $J(\theta)$.

4. Pour la valeur du taux d'apprentissage $\alpha = 0,07$, effectuer le calcul de régression permettant d'obtenir le vecteur $\theta = (\theta_0, \theta_1, \theta_2)$ optimal permettant de calculer la meilleure régression linéaire multivariée sur le jeu de données d'apprentissage.

5. Nous allons maintenant automatiser la recherche du meilleur taux d'apprentissage $\alpha \in [0,001; 10]$. Pour ce faire, on devra calculer pour chaque itération la valeur de la fonction de coût $J(\theta)$ et on stockera toutes ces valeurs dans un vecteur. Comme on veut sélectionner un taux d'apprentissage efficace, on va comparer les résultats de calcul de $J(\theta)$ sur 50 itérations en changenat de taux d'apprentissage à chaque

série d'itérations. Les valeurs de ce taux doivent rester dans l'intervalle initialement donné $\alpha \in [0,001; 10]$. On tracera alors les courbes représentant en abscisse, le nombre d'itérations et en ordonnées les valeurs de la fonction de coût ; chaque courbe correspond à une valeur du taux d'apprentissage. A partir de ces courbes, sélectionner ce qui parît être le meilleur taux d'apprentissage et recalculer le vecteur θ jusqu'à la convergence. Utiliser ce vecteur θ pour prédire le prix d'un logement de 1650 m2 et de 3 pièces.