ENSAE TD noté, vendredi 16 décembre 2016

Le programme devra être imprimé et rendu au chargé de TD. Toutes les questions valent 2 points. Vous êtes libres d'utiliser numpy ou non à toutes les questions.

1

On suppose qu'on dispose d'un ensemble d'observations (X_i, Y_i) avec $X_i, Y_i \in \mathbb{R}$. La régression linéaire consiste une relation linéaire $Y_i = aX_i + b + \epsilon_i$ qui minimise la variance du bruit. On pose :

$$E(a,b) = \sum_i (Y_i - (aX_i + b))^2$$

On cherche a, b tels que :

$$a^*, b^* = \arg\min E(a,b) = \arg\min \sum_i (Y_i - (aX_i + b))^2$$

La fonction est dérivable et on trouve :

$$\frac{\partial E(a,b)}{\partial a} = -2\sum_i X_i(Y_i - (aX_i + b)) \text{ et } \frac{\partial E(a,b)}{\partial b} = -2\sum_i (Y_i - (aX_i + b))$$

Il suffit alors d'annuler les dérivées. On résoud un système d'équations linéaires. On note :

$$\begin{array}{l} \mathbb{E}(X) = \frac{1}{n}\sum_{i=1}^n X_i \text{ et } \mathbb{E}(Y) = \frac{1}{n}\sum_{i=1}^n Y_i \\ \mathbb{E}\left(X^2\right) = \frac{1}{n}\sum_{i=1}^n X_i^2 \text{ et } \mathbb{E}(XY) = \frac{1}{n}\sum_{i=1}^n X_i Y_i \end{array}$$

Finalement :

$$a^* = \frac{\mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)}{\mathbb{E}(X^2) - (\mathbb{E}(X))^2} \text{ et } b^* = \mathbb{E}(Y) - a^*\mathbb{E}(X)$$

On génère un nuage de points avec le code suivant :

```
import random
def generate_xy(n=100, a=0.5, b=1):
    res = []
    for i in range(0, n):
        x = random.uniform(0, 10)
        res.append((x, x*a + b + random.gauss(0,1)))
    return res
```

1) Recopier la fonction précèdente et génèrer 10 points (ou 10 couples de points).

2) Ecrire une fonction qui calcule $\mathbb{E}(X), \mathbb{E}(Y), \mathbb{E}(XY), \mathbb{E}\left(X^2\right)$.

```
def calcule_exyxyx2(obs):
    ....
```

3) Calculer les grandeurs a^*, b^*.

```
def calcule_ab(obs):
    ....
```

Tout se passe bien quand X_i est une variable continue. Quand c'est une variable catégorielle, cela n'est plus possible.

4) Compléter le programme ci-dessous (....) pour qu'il génère des couples tel que :

```
[('rouge', 1.4281790866123962), ('vert', 3.1438708048684716),
 ('rouge', 0.7193245827013746), ('vert', 0.5293831925619408),
 ('bleu', 0.27344460504234447), ... ]
```

```
def generate_caty(n=100, a=0.5, b=1, cats=["rouge", "vert", "bleu"]):
    res = []
    for i in range(0, n):
        x = ....
        cat = cats[x]
        res.append((cat, 10*x**2*a + b + random.gauss(0,1)))
    return res
```

5) On voudrait quand même faire une régression de la variable Y sur la variable catégorielle. On construit une fonction qui assigne un numéro quelconque mais distinct à chaque catégorie. La fonction retourne un dictionnaire avec les catégories comme clé et les numéros comme valeurs.

```
def numero_cat(obs):
    ....
```

6) On construit la matrice M_{ic} tel que : M_{ic} vaut 1 si c est le numéro de la catégorie X_i, 0 sinon.

```
def matrice_cat(obs, numero):
    ....
```

7) Il est conseillé de convertir la matrice M et les Y au format *numpy*. On ajoute un vecteur constant à la matrice M.

```
def convert_numpy(obs, M):
    ....
```

8) On résoud la régression multidimensionnelle en appliquant la formule $C^* = (M'M)^{-1}M'Y$.

9) La régression détermine les coefficients α dans la régression $Y_i = \alpha_{rouge}\mathbb{1}_{\{X_i=rouge\}} + \alpha_{vert}\mathbb{1}_{\{X_i=vert\}} + \alpha_{bleu}\mathbb{1}_{\{X_i=bleu\}} + \epsilon_i$. Construire le vecteur $\hat{Y_i} = \alpha_{rouge}\mathbb{1}_{\{X_i=rouge\}} + \alpha_{vert}\mathbb{1}_{\{X_i=vert\}} + \alpha_{bleu}\mathbb{1}_{\{X_i=bleu\}}$.

10) Utiliser le résultat de la question 3 pour calculer les coefficients de la régression $Y_i = a^*\hat{Y_i} + b^*$.

ENSAE TD noté, vendredi 16 décembre 2016

Le programme devra être imprimé et rendu au chargé de TD. Toutes les questions valent 2 points. Vous êtes libres d'utiliser numpy ou non à toutes les questions.

2

On suppose qu'on dispose d'un ensemble d'observations (X_i, Y_i) avec $X_i, Y_i \in \mathbb{R}$. La régression linéaire consiste une relation linéaire $Y_i = aX_i + b + \epsilon_i$ qui minimise la variance du bruit. On pose :

$$E(a,b) = \sum_i (Y_i - (aX_i + b))^2$$

On cherche a, b tels que :

$$a^*, b^* = \arg\min E(a,b) = \arg\min \sum_i (Y_i - (aX_i + b))^2$$

La fonction est dérivable et on trouve :

$$\frac{\partial E(a,b)}{\partial a} = -2\sum_i X_i(Y_i - (aX_i + b)) \text{ et } \frac{\partial E(a,b)}{\partial b} = -2\sum_i (Y_i - (aX_i + b))$$

Il suffit alors d'annuler les dérivées. On résoud un système d'équations linéaires. On note :

$$\begin{array}{l} \mathbb{E}(X) = \frac{1}{n}\sum_{i=1}^n X_i \text{ et } \mathbb{E}(Y) = \frac{1}{n}\sum_{i=1}^n Y_i \\ \mathbb{E}\left(X^2\right) = \frac{1}{n}\sum_{i=1}^n X_i^2 \text{ et } \mathbb{E}(XY) = \frac{1}{n}\sum_{i=1}^n X_i Y_i \end{array}$$

Finalement :

$$a^* = \frac{\mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)}{\mathbb{E}(X^2) - (\mathbb{E}(X))^2} \text{ et } b^* = \mathbb{E}(Y) - a^*\mathbb{E}(X)$$

On génère un nuage de points avec le code suivant :

```
import random
def generate_xy(n=100, a=0.5, b=1):
    res = []
    for i in range(0, n):
        x = random.uniform(0, 10)
        res.append((x, x*a + b + random.gauss(0,1)))
    return res
```

1) Recopier la fonction précédente et génèrer 10 points (ou 10 couples de points).

2) Ecrire une fonction qui calcule $\mathbb{E}(X), \mathbb{E}(Y), \mathbb{E}(XY), \mathbb{E}\left(X^2\right)$.

```
def calcule_exyxyx2(obs):
    ....
```

3) Calculer les grandeurs a^*, b^*.

```
def calcule_ab(obs):
    ....
```

Tout se passe bien quand X_i est une variable continue. Quand c'est une variable catégorielle, cela n'est plus possible.

4) Compléter le programme ci-dessous (....) pour qu'il génère des couples tel que :

```
[('rouge', 1.4281790866123962), ('vert', 3.1438708048684716),
 ('rouge', 0.7193245827013746), ('vert', 0.5293831925619408),
 ('bleu', 0.27344460504234447), ... ]
```

```
def generate_caty(n=100, a=0.5, b=1, cats=["rouge", "vert", "bleu"]):
    res = []
    for i in range(0, n):
        x = ....     # on veut 50% de rouge, 30% de vert, 20% de bleu
        cat = cats[x]
        res.append((cat, 10*x**2*a + b + random.gauss(0,1)))
    return res
```

5) On voudrait quand même faire une régression de la variable Y sur la variable catégorielle. On commence par les compter. Construire une fonction qui compte le nombre de fois qu'une catégorie est présente dans les données (un histogramme).

```
def histogram_cat(obs):
    ....
```

6) Construire une fonction qui calcule la moyenne des Y_i pour chaque catégorie : $\mathbb{E}(Y|rouge)$, $\mathbb{E}(Y|vert)$, $\mathbb{E}(Y|bleu)$. La fonction retourne un dictionnaire `{ couleur : moyenne }`.

```
def moyenne_cat(hist, Y):
    ....
```

7) Construire le vecteur $Z_i = \mathbb{E}(Y|rouge)\,\mathbb{1}_{\{X_i=rouge\}} + \mathbb{E}(Y|vert)\,\mathbb{1}_{\{X_i=vert\}} + \mathbb{E}(Y|bleu)\,\mathbb{1}_{\{X_i=bleu\}}$.

8) Utiliser le résultat de la question 3 pour calculer les coefficients de la régression $Y_i = a^*Z_i + b^*$.

9) Calculer la matrice de variance / covariance pour les variables (Y_i), (Z_i), $(Y_i - Z_i)$, $\mathbb{1}_{\{X_i=rouge\}}$, $\mathbb{1}_{\{X_i=vert\}}$, $\mathbb{1}_{\{X_i=bleu\}}$.

10) On permute rouge et vert. Construire le vecteur $W_i = \mathbb{E}(Y|rouge)\,\mathbb{1}_{\{X_i=vert\}} + \mathbb{E}(Y|vert)\,\mathbb{1}_{\{X_i=rouge\}} + \mathbb{E}(Y|bleu)\,\mathbb{1}_{\{X_i=bleu\}}$. Utiliser le résultat de la question 3 pour calculer les coefficients de la régression $Y_i = a^*W_i + b^*$. Vérifiez que l'erreur est supérieure.