TP Module - Probabilité et Statistique
Master 1 - Informatique - Université Paris-Sud

Jeudi 28 Mars 2019
Durée : 2h00
Documents autorisés : supports et notes de cours

Pour ce TP, il vous est demandé de rendre un fichier notebook jupyter clairement commenté et exécutable. Vous enverrez le fichier à l'adresse aurelien.decelle@u-psud.fr à la fin de la séance. Le sujet est composé de deux parties indépendantes, vous pouvez traiter les questions dans l'ordre qui vous va.

1 Variables aléatoires continues :

Dans cette partie, on va générer un ensemble de variables aléatoires et voir ce que l'on peut dire sur la statistique de certaines observables. Nous allons utiliser par la suite la distribution exponentielle

$$p(x) = \frac{1}{\lambda} e^{-x/\lambda}$$

La distribution est caractérisée entièrement par le paramètre λ. La valeur moyenne est donnée par $m = \lambda$ et la variance est donnée par $\sigma^2 = \lambda^2$.

1. Tracer la distribution exponentielle pour $\lambda = 2$ et $x \in [0; 10]$, d'abord en échelle linéaire puis en échelle logarithmique pour l'axe des ordonnées.
2. Générer un ensemble de 10000 variables exponentielles avec $\lambda = 2$. Enuiste, estimer la valeur moyenne, la variance du jeu de données et l'erreur sur la moyenne.

Vous allez maintenant réaliser l'expérience suivante. Il s'agira de répéter $N_T = 10000$ fois le fait de générer $N_s = 10000$ variables aléatoires distribuées selon la loi exponetielle. Pour chacun des tirages, il faudra enregistrer la valeur moyenne de l'ensemble des variables générées, ainsi que la valeur du minimum et la valeur du maximum.

3. Représenter la distribution obtenue pour ces trois observables. (Trois graphes distincts sont demandés)
4. En recentrant l'observable de la moyenne, par quelle quantité (valeur théorique) faut-il la multiplier pour obtenir une variable suivant une loi gaussienne de moyenne 0 et de variance 1 ?
5. Tracer l'histogramme en échelle logarithmiqute de l'observable du minimum. Que constatez-vous ?
6. Si vous avez identifié la distribution du minimum, afficher sur un graphe l'histogramme des valeurs avec en surimpression la distribution identifiée. (vous pouvez normaliser l'histogramme à l'aide du paramètre normed=True).
7. Recentrer les valeurs de l'observable du maximum (soustraire la moyenne et diviser par la déviation standard). Afficher sur le même graphe l'histogramme des valeurs maximums recentrées et l'histogramme des valeurs moyennes recentrées. Pouvez-vous commenter le comportement des deux histogrammes en échelle logarithmique ?

2 SVD et le jeu de données MNIST

Pour cette partie vous allez travailler avec la base de données MNIST (déjà utilisée au TP4). Cette base d'images représente des chiffres manuscrits scannés, allant de 0 à 9 en nuance de gris : chaque pixel prend une valeur réelle $p_i \; in [0 : 1]$.

On cherchera ici à étudier la base de données en utilisant deux chiffres : le 3 et le 6. On va d'abord explorer le jeu de données à l'aide de la SVD et ensuite, on verra si on peut trouver une solution simple afin de distinguer entre les deux chiffres.

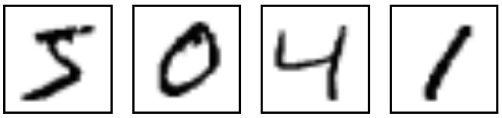

FIGURE 1 – Exemples d'images de la base MNIST

2.1 Décomposition en SVD sur l'ensemble des 3 et des 6

A l'aide des lignes suivantes, récupérer tous les chiffres 3 et 6 dans un unique tableau

```
X3 = train_set[0][np.where(train_set[1]==3)].T
X7 = train_set[0][np.where(train_set[1]==6)].T
X = np.concatenate((X3,X7),1)
X37 = np.random.permutation(X.T).T
print(X.shape)
```

On voudrait voir si il est possible d'utiliser les directions de la SVD pour distinguer entre un 3 et un 6.

1. Effectuer la décomposition SVD de $X37$.
2. Calculer la projection des 3 (en utilisant la matrice $X3$) et des 6 ($X6$) le long des directions propres de la svd calculer précédement.
3. Afficher le scatter plot des directions $1-2$, $3-4$ et $5-6$ des trois et des six.
4. Peut-on séparer les données des 3 et des 6 à l'aide d'une ou plusieurs de ces directions propres ?
5. En utilisant la première direction, combien de données serait classées correctement ? (pour les 3 et pour les 6)

2.2 Erreur de reconstruction

Cette fois-ci, on va effectuer la SVD sur chaque des chiffres séparément.

```
u3,s3,v3 = np.linalg.svd(X3/np.sqrt(X3.shape[1]))
u6,s6,v6 = np.linalg.svd(X6/np.sqrt(X3.shape[1]))
```

1. Pour une image de 3, calculer l'erreur de reconstruction (en utilias $u3$) en fonction du nombre de direction propres considérées. (Faire un graphe)
2. Pour une image de 6, calculer l'erreur de reconstruction (en utilias $u6$) en fonction du nombre de direction propres considérées. (Faire un graphe)
3. Pour chaque chiffre, calculer l'erreur de reconstruction (un graphe en utilisan $u3$ et un autre avec $u6$).