

# “Algorithmes stochastiques” - Examen final

Université de Paris - M2 MMA

3 janvier 2022 - durée : trois heures

- Ce sujet est composé d’une partie “théorique” (à composer sur papier) et d’une partie “pratique” (à composer sur ordinateur).
- Il faut rendre votre copie “théorique” avant de passer à la “pratique”. Cependant, vous disposez déjà du sujet “pratique” et pouvez y réfléchir à l’écrit.
- Les notes manuscrites sont autorisées pendant tout l’examen. Pendant la partie sur ordinateur, vous pouvez consulter la documentation Python et les corrigés des TP. Toute forme de communication est interdite.
- Pour la partie “théorique”, sauf indication contraire, toute réponse demande une argumentation mathématique qui peut être courte, mais précise.
- N’hésitez pas à poser des questions ni à signaler si vous croyez avoir repéré une erreur d’énoncé.
- Bon courage !

## Partie théorique

**Exercice 1** On rappelle que pour  $\lambda > 0$ , la loi exponentielle de paramètre  $\lambda$  est définie sur l'intervalle  $[0, +\infty)$  et a pour densité  $x \mapsto \lambda \exp(-\lambda x)$ .

On suppose que l'on dispose d'une fonction **Expo** qui simule une loi exponentielle de paramètre 1, et on veut l'utiliser pour simuler une loi exponentielle de paramètre  $\lambda \neq 1$  en utilisant la méthode du rejet.

1. Rappeler l'hypothèse faite sur les densités qui est nécessaire au bon fonctionnement de l'algorithme du rejet, et préciser pour quelles valeurs de  $\lambda$  cette hypothèse est vérifiée ici.
2. Écrire précisément (en pseudo-code) l'algorithme du rejet pour la situation présente.
3. Calculer le nombre théorique moyen de rejets (en fonction de  $\lambda$ ).

**Algorithme de Barker** L'algorithme de Barker est une méthode de type "Markov Chain Monte Carlo" qui est différente de Metropolis-Hastings et que l'on va brièvement étudier ici.

Soit  $E$  un espace d'états (qu'on pourra supposer fini et identifier à  $E = \{1, \dots, T\}$  pour un certain entier  $T$  qui désigne la taille de  $E$ ) et soit  $\mu$  une mesure de probabilité sur  $E$  que l'on souhaite simuler. Fixons une matrice de transition  $Q$  sur  $E \times E$  que l'on supposera **symétrique**. L'algorithme de Barker construit, à partir de  $Q$ , une nouvelle chaîne de Markov de la façon suivante : supposons qu'à l'étape  $n$  la chaîne se trouve en l'état  $X_n$ , alors :

- On fait une proposition  $\tilde{X}_{n+1}$  à l'aide de la matrice de référence  $Q$ .
- On accepte cette proposition (c'est-à-dire qu'on prend  $X_{n+1} = \tilde{X}_{n+1}$ ) avec probabilité

$$\frac{\mu(\tilde{X}_{n+1})}{\mu(\tilde{X}_{n+1}) + \mu(X_n)},$$

et on la refuse (c'est-à-dire qu'on prend  $X_{n+1} = X_n$ ) avec la probabilité complémentaire.

1. Donner l'expression des probabilités de transition  $P(\cdot, \cdot)$  pour cette nouvelle chaîne de Markov sur  $E$ .
2. Montrer que la mesure  $\mu$  est réversible pour  $P$ .

**Apprentissage d'ordonnement** Soit  $N$  échantillons étiquetés  $\{x_i\}_{i=1, \dots, N}$ , où chaque  $x_i$  est un vecteur dans  $\mathbb{R}^d$ . À chaque  $x_i$  correspond une étiquette  $y_i$  qui est un nombre réel.

Plutôt que d'apprendre directement à prédire les *étiquettes*, on cherche ici à apprendre les *préférences* c'est à dire qu'on cherche à construire un prédicteur  $f : x \in \mathbb{R}^d \mapsto f(x) \in \mathbb{R}$  tel que "plus  $y_i$  est grand, plus  $f(x_i)$  est grand".

On choisit un prédicteur linéaire  $f$  (c'est à dire qu'on cherche  $f$  sous la forme  $f(x) = \langle x, \mathbf{w} \rangle$  où  $\mathbf{w} \in \mathbb{R}^d$  est un vecteur de paramètres libres), et on se propose d'optimiser la fonction de coût suivante :

$$L(f) = \sum_{(i,j) | y_i > y_j} (y_i - y_j) \max(0, \gamma - (f(x_i) - f(x_j))), \text{ avec } \gamma = 1.$$

On peut comprendre  $L$  comme une somme du coût  $(y_i - y_j) \max(0, \gamma - (f(x_i) - f(x_j)))$  pour la paire d'entrées  $i$  et  $j$ . Insistons sur le fait que la somme définissant  $L$  porte sur les entrées pour lesquelles  $y_i > y_j$ .

1. À quoi correspond une valeur nulle du coût **pour une paire d'entrées** fixée ? On dira par la suite que deux telles entrées sont "bien rangées".
2. À quoi correspond une valeur positive du coût ? (On dira que deux telles entrées sont "mal rangées"). Comment se comporte le coût en fonction de l'écart entre  $y_i$  et  $y_j$  ?
3. Que se passe-t-il si on prend  $\gamma$  plus grand ? Et si l'on prend  $\gamma = 0$  ?
4. Calculer le gradient du coût **pour une paire d'entrées** par rapport au vecteur de paramètres  $\mathbf{w}$  (on pourra distinguer deux cas, selon que les entrées sont "bien rangées" ou non. Ce gradient est alors bien défini sauf dans une situation exceptionnelle qu'on pourra ignorer).
5. Afin d'optimiser  $L$ , on utilise une méthode de type "descente de gradient stochastique". Ici on ne tire pas une entrée au hasard mais **une paire d'entrées**  $i, j$  et on calcule le gradient du coût correspondant à cette paire d'entrées.
  - (a) Écrire un pseudo-code précis pour une telle descente de gradient stochastique.
  - (b) Exprimer l'évolution du vecteur de paramètres  $\omega$  entre l'étape  $t$  et l'étape  $t+1$  (on notera  $\epsilon$  le pas de temps).
  - (c) Supposons qu'on a tiré deux entrées "mal rangées". Montrer que :

$$f^{t+1}(x_i) - f^{t+1}(x_j) = f^t(x_i) - f^t(x_j) + \epsilon(y_i - y_j)\|x_i - x_j\|^2.$$

Est-ce cohérent avec ce que l'on veut faire ?

## Partie pratique

Vous enverrez votre copie/notebook Jupyter à l'adresse `thomasleble@gmail.com` avec l'objet **Partie pratique**. Pour chaque exercice, votre code doit être accompagné de quelques lignes de description/explication (quelle méthode avez-vous choisie, comment avez-vous choisi les différents paramètres et pourquoi, etc.).

### Simulations de variables aléatoires

1. On considère la fonction suivante définie sur l'intervalle  $I = [-1, 1]$ .

$$g : y \mapsto \frac{1}{10}y^2 + \frac{7}{15}.$$

Écrire un programme (en utilisant la méthode de votre choix) qui permet de simuler une variable aléatoire de densité  $g$ .

2. On considère l'ensemble  $E = \{1, \dots, 50\}$ , et une loi de probabilité  $\mu$  sur  $E$  définie de la façon suivante :

$$\forall k \in E, \quad \mu(k) = \frac{A}{k^2},$$

où  $A$  est la constante de normalisation telle que  $\sum_{k=1}^{50} \mu(k) = 1$ . Écrire un programme (en utilisant la méthode de votre choix) qui permet de simuler  $\mu$ .

3. Soit  $\alpha \in (0, 1)$ . On considère l'espace des états  $E = \{0, \dots, 50\}$  et la mesure  $\pi$  sur  $E$  définie par :

$$\forall i \in E, \pi(i) = \frac{1}{Z} \frac{\alpha^i}{i!},$$

où  $Z$  est la constante de normalisation telle que  $\sum_{i=0}^{50} \pi(i) = 1$ . Utiliser une méthode MCMC pour simuler  $\pi$ .

**Minimisation d'une fonction de coût** On considère la fonction de coût suivante, définie sur  $(-\infty, +\infty)$  :

$$f : x \mapsto f(x) = \frac{1}{2}x^2 - 10 \exp(-x^2) \sin(3x).$$

Utiliser la méthode de votre choix afin de trouver le minimum de  $f$ .

**Calcul de  $\pi$  par une méthode Monte Carlo** On admettra le fait suivant : si  $n$  est assez grand et que l'on tire deux entiers indépendamment dans  $\{1, \dots, n\}$  suivant la loi uniforme, la probabilité qu'ils soient premiers entre eux est proche de  $\frac{6}{\pi^2}$ . En utilisant cette information et l'esprit des méthodes de Monte-Carlo, écrire un programme qui calcule une valeur approchée de  $\pi$ .

(On rappelle que deux entiers sont dits "premiers entre eux" si leur plus grand diviseur commun est égal à 1. Vous pourrez utiliser la fonction `gcd` de la bibliothèque `math`, qui renvoie le plus grand diviseur commun de deux entiers.)