Simulation et Monte Carlo: projets 2020-21

Nicolas Chopin

Vous devez former un groupe de trois étudiants au sein de votre groupe de TD, choisir un des projets suivants, et le traiter d'ici la dernière séance de TD, où vous ferez une présentation orale de 15 minutes devant les autres étudiants. Pas besoin de rendre un rapport rédigé: vous pouvez cependant rendre le jour de la soutenance un document contenant certains graphiques et résultats, mais surtout, vous devez envoyer vos programmes à votre chargé de TD, qui vérifiera que ce programme fonctionne bien.

Vous avez tout à fait le droit et même je vous encourage à chercher sur internet ou dans la littérature scientifique des inspirations pour effectuer votre projet. Une seule obligation: citez vos sources!

Si vous bloquez, contactez votre chargé de TD (qui me contactera si nécessaire).

Point Essentiel: toujours évaluer (d'une façon ou d'une autre: intervalles de confiance, box-plots, etc.) l'erreur de Monte Carlo de vos résultats. Dans le cas du MCMC, pensez aussi aux ACF (graphe de la fonction d'autocorrélation pour chaque composantes) et aux "traces" (valeur de chaque composante en fonction du temps, notamment pour déterminer le burn-in). Faites preuve d'un esprit scientifique!

Les questions bonus sont facultatives: elles sont réservées aux étudiants qui veulent s'investir plus dans leur projet. Leur bonne résolution sera récompensée par une meilleure note, mais uniquement si le reste du projet a été bien traité.

Données angulaires

Une loi classique pour des données correspondant à des angles observés sur l'intervalle $[-\pi, \pi]$ est la loi de von Mises, de densité:

$$\pi(\theta) = \frac{1}{Z(\kappa)} \exp\{\kappa \cos(\theta - \mu)\}$$

où $\kappa > 0$, $\mu \in [-\pi, \pi]$, et $Z(\kappa)$ est une constante de normalisation qui n'admet pas d'expression explicite.

Cette loi est souvent qualifiée de loi normale pour les angles, car $\cos(\theta) \approx 1 - \theta^2$.

1. Proposer un algorithme d'acceptation-rejet pour simuler ce type de loi, basé sur une loi de proposition uniforme (sur $[-\pi, \pi]$). Le mettre en oeuvre, et discuter comment sa performance dépend des paramètres μ et κ. (On pourra faire une représentation graphique par exemple.)

2. Proposer un algorithme de Metropolis simple pour simuler selon une telle loi; notamment proposer une règle simple pour que la performance de l'algorithme ne dépende pas des paramètres. (Expliquer ce que veut dire le mot "performance" pour un tel algorithme.)

3. Utiliser l'approche MCMC-MLE pour estimer les paramètres μ et κ à partir de données observées.

   Méthode MCMC-MLE: méthode qui revient à remplacer dans une fonction de log-vraisemblance une fonction des paramètres (ici Z) non calculable par une approximation basée sur de l'importance sampling, où la loi de proposition est la loi du modèle pour un paramètre fixe (bien choisi, à vous de réfléchir comment). On maximise ensuite la log-vraisemblance ainsi obtenue. Cette méthode est présentée notamment dans l'article de Geyer & Thompson (1992).

4. Bonus: proposer d'autres algorithmes de rejet plus efficaces pour simuler une loi de von Mises; quelques idées: prendre comme loi de proposition un mélange d'uniformes, ou une loi de Cauchy adaptée aux angles, comme expliqué p. 473 du livre de Devroye disponible à l'adresse suivante: http://luc.devroye.org/rnbookindex.html.

Prévalence de Listeria dans le lait cru

La listeria est une bactérie pathogène qui provoque la listeriose. Le fichier listeria.txt (disponible dans l'espace du cours sous Teams) recense les données de 91 études, reportant dans différents pays le nombre d'échantillons testés (seconde colonne) et le nombre de cas positifs (présence détectée de listeria, première colonne).

1. On suppose pour commencer une probabilité constante de présence de listeria dans toutes les études; soit $r_i \sim Bin(n_i, p)$, avec comme loi a priori $p \sim \alpha, \beta$, $\alpha = \beta = 1$. Déterminer la loi a posteriori, la représenter. Est-ce que ce modèle vous semble raisonnable pour ces données? (On pourra répondre en faisant un graphique.)

2. On suppose ensuite que cette probabilité varie d'une étude à l'autre; soit $r_i \sim Bin(n_i, p_i)$ où les p_i sont distincts mais suivent la même loi a priori Beta(α, β). De plus, on suppose que α et β sont eux-aussi inconnus, et re-paramétrés ainsi: $\mu = \alpha/(\alpha + \beta)$, avec $\mu \sim$ U$[0, 1]$, et $\kappa = \alpha + \beta$, avec $\kappa \sim$ Exp(0.1). Construire et mettre en oeuvre un algorithme de Metropolis-within-Gibbs pour simuler la loi a posteriori de $(\mu, \kappa, p_1, \ldots, p_n)$ sachant les données. Bien expliquer les détails.

3. On modifie le modèle comme suit: on suppose désormais que les p_i suivent, a priori, un mélange de deux lois Beta. Adapter l'algorithme précédent à ce nouveau modèle.

4. Bonus: proposer une méthode pour comparer les trois modèles. Pour ce faire, on pourra utiliser la vraisemblance marginale (la densité des données après intégration sur les paramètres), et par exemple la méthode suivante: importance sampling avec comme loi de proposition la loi a priori. Bien expliquer les détails.

Pricing d'options

On cherche à évaluer le prix d'une option asiatique:

$$C = \mathbb{E}\left[e^{-rT}\left(\frac{1}{k}\sum_{i=1}^{k} S(t_i) - K\right)^+\right]$$

avec $t_0 = 0 < t_1 < \ldots < t_k = T$, $r > 0$, K fixe, et S un processus défini sur $[0, T]$; notation: $x^+ = \max(x, 0)$.

On considère le modèle CIR (Cox, Ingersoll, Ross):

$$dS_t = \alpha(b - S_t)dt + \sigma\sqrt{S_t}dW_t$$

où W_t est le mouvement brownien. On pourra prendre par ex. $\alpha = 0.2$, $b = 0$, $\sigma = 0.3$, $T = 1$, $r = 0.05$, $K = 5$, $k = 20$, $t_i = i/20$. Notez que ce processus en temps continu doit être discrétisé pour pouvoir être simulé (comme expliqué en cours).

1. Comparer différentes méthodes de calcul de C en combinant les différentes méthodes de réduction de variance vues en cours (variables antithétiques, variables de contrôle). Vous pouvez essayer aussi de faire varier les différents paramètres (ainsi que le pas de discrétisation) pour voir à quel point vos conclusions en dépendent.

2. Expliquer et illustrer comment la méthode MLMC (multi-level Monte Carlo) vue en cours, et présentée dans l'article suivant http://statweb.stanford.edu/~owen/courses/362/readings/GilesMultilevel.pdf peut réduire le temps de calcul.

3. Reprendre la comparaison en se basant sur du Quasi-Monte Carlo.

Compression d'image

Le modèle de Potts est une généralisation du modèle d'Ising permettant un nombre de modalités supérieur à deux: On considère un vecteur aléatoire x à

valeurs dans $\{1, \ldots, K\}^n$, $K > 2$, dont la probabilité est:

$$\pi(x) = \frac{1}{Z(\beta)} \exp\left\{ \beta \sum_{i \sim j} \mathbf{1}[x_i = x_j] \right\}$$

où $\beta > 0$, $Z(\beta)$ est une constante de normalisation (donner son expression), et $i \sim j$ est une relation de voisinage (par exemple j est un des quatre pixels adjacents au pixel i, dans le cas où les x_i représentent les pixels d'une image rectangulaire).

1. Pour β et K fixé, proposer un Gibbs sampler pour simuler selon la loi du modèle. Déterminer la performance de l'algorithme en fonction de ces paramètres.
2. Un modèle classique en analyse d'image est de supposer que l'on observe à chaque pixel une valeur (i.e. niveau de gris) $y_i | x_i = k \sim N(\mu_k, \sigma_k^2)$, où les x_i forment un modèle de Potts. Construire et mettre en oeuvre un Gibbs sampler pour simuler la loi des paramètres (les μ_k, σ_k^2) et variables latentes (les x_i). On pourra prendre comme loi a priori des lois normales pour les μ_k et des inverse-gamma pour les σ_k^2.
3. Expliquer comment on peut utiliser cet algorithme pour effectuer de la compression d'image, et l'appliquer à différentes images (en niveaux de gris) de votre choix, en faisant varier éventuellement K, β, et les hyper-paramètres.
4. Bonus: quel problème se pose si on essaie de généraliser l'algorithme de façon à pouvoir aussi estimer (simuler) β? Une façon de régler le problème est de considérer une densité a priori particulière. Laquelle ? Mettre en oeuvre cet algorithme sur les mêmes images que la question précédente.

Bayesian Lasso

Soit un problème de régression, avec des variables expliquées Y_n (à valeur dans $\mathbb{R}$), et des prédicteurs X_n (à valeur dans $\mathbb{R}^p$). Pour faire simple, on supposera que les Y_n et les composantes des X_n ont été préalablement normalisées; cela permet notamment de ne pas avoir à ajouter de constante dans le modèle; expliquer pourquoi.

L'estimateur du LASSO est obtenu par minimisation (en $\beta \in \mathbb{R}^p$) du critère suivant:

$$\sum_{n=1}^{N} (Y_n - \beta^T X_n)^2 + \lambda \sum_{i=1}^{p} |\beta_i|$$

1. Expliquer le lien entre cet estimateur et la loi a posteriori qui correspondrait à (1) un modèle de régression avec bruit gaussien (de variance connue); (2) une loi a priori avec des β_i suivant une loi de Laplace (double-exponentielle).

2. Ce lien entre LASSO et estimation bayésienne a engendré un certain intérêt pour une version bayésienne de Lasso, qui nécessite donc de simuler selon la loi a posteriori mentionnée dans la question précédente. Proposer et mettre en oeuvre une approche basée sur l'échantillonnage d'importance pour simuler cette loi. Pour la loi de proposition, on pourra remarquer que, pour $\lambda = 0$, on retrouve une gaussienne (que l'on calculera).

3. On propose maintenant de simuler selon une loi a posteriori qui prend en compte le fait que la variance du bruit n'est pas connue (lui attribuer une loi a priori inverse-gamma). Un article récent a proposé un Gibbs sampler pour cette loi, basée sur la propriété suivante de la loi de Laplace: si $\tau_j \sim \text{Exp}(\lambda^2/2)$, et $\beta_j|\tau_j \sim N(0, \tau_j\sigma^2)$, alors $\beta_j \sim \text{Laplace}(\lambda/\sigma^2)$. (Noter qu'il s'agit d'une loi conditionnelle en σ^2). Décrire et mettre en oeuvre un Gibbs sampler basé sur cette propriété, et qui itérativement simule les lois conditionnelles des composantes suivantes: le vecteur β; le vecteur des τ_j; et σ^2. (Pour les τ_j, on pourra calculer la densité de $1/\tau_j$, et consulter la page wikipedia sur l'"Inverse Gaussian distribution".)

4. Bonus: Proposer différentes façons de comparer ce Gibbs sampler et la méthode proposée en 2, dans le cas où σ est fixe; vous pouvez considérer différents jeux de données standards, tels que "Boston Housing". (On a vu en cours la notion d'ESS pour l'importance sampling; vous pouvez voir comment définir une notion équivalente pour du MCMC. Vous avez le droit de vous inspirer de la littérature scientifique.)

File d'attente

Une file d'attente a les caractéristiques suivantes: le temps entre deux arrivées de client suit une loi $\text{Exp}(\lambda_1)$, et le temps de service d'un client suit une loi $\text{Exp}(\lambda_2)$. On souhaite dans un premier temps estimer la probabilité que le nombre de clients dans la salle d'attente dépasse un certain seuil critique n (entre le temps 0 et le temps t).

1. Simuler la file d'attente, et montrer numériquement comment varie cette probabilité en fonction de λ_1, λ_2, n et t.

2. Quel problème pratique se pose-t-il si on essaye d'évaluer cette probabilité avec du quasi-Monte Carlo ? Proposer une méthode hybride, mélangeant (randomized) quasi-Monte Carlo et Monte Carlo pour évaluer cette probabilité, et évaluer sa performance par rapport à la méthode standard mise en œuvre dans la question précédente.

3. On suppose désormais que l'on observe uniquement les temps de sortie des clients. Proposer un algorithme de type ABC pour estimer les paramètres λ_1 et λ_2 à partir d'un jeu de données. Bien préciser notamment la statistique résumé utilisée.

4. Bonus: reprendre l'exercice pour des lois de Weibul (à la place de la loi exponentielle), et commenter les différences observées.