RICM4 Probabilités et Simulation – Devoir n°2

Arnaud Legrand, Florence Perronnin, Jean-Marc Vincent

23 novembre 2016

Informations générales

Conseils et consignes importantes

- Avant de vous lancer dans la programmation, essayez de répondre aux différentes questions, l'une après l'autre (c'est important), en n'utilisant que votre intuition. Vous rédigerez votre opinion en début de document avant de détailler vos réponses finales. Le fait que cette intuition soit correcte ou pas n'aura aucune impact sur votre note finale. Décrire votre intuition a uniquement pour but que vous commenciez à réfléchir au problème et de réaliser à quel point votre intuition peut être correcte ou pas.

  Vous analyserez, en fin de DM, cette intuition au vu des résultats et des statistiques observés.
- Vous prendrez soin de donner des intervalles de confiance de chacune de vos estimations.
- Nous vous déconseillons fortement de chercher des solutions qui ressembleraient sur Internet ou dans des livres. Ce qu'on trouve sur Internet sur ces sujets est souvent bourré de fautes et les livres qui traitent formellement de ces sujets sont souvent bien au delà de notre programme. N'hésitez cependant pas à nous demander si vous avez un doute sur la signification de l'énoncé ou sur une question.
- Vous pouvez discuter du sujet entre vous mais le codage et la rédaction des analyses doivent être individuels. Vous citerez toute source d'information utilisée pour répondre aux questions.
- Pas la peine de chercher des choses compliquées. Pour la majorité des questions des différents sujets, le code à écrire n'excède pas quelques lignes.
- Votre devoir sera rédigé sous forme d'un document HTML généré à l'aide de `R/Markdown` et publié sur `rpubs` en prenant soin de bien laisser le code apparent et de fixer la graine de votre générateur à l'aide de la fonction `set.seed` au tout début du document afin qu'il soit possible de reproduire vos données avec exactitude. Vous enverrez l'url `rpubs` de votre devoir par mail à `arnaud.legrand@imag.fr` et à `florence.perronnin@imagr` en indiquant dans le sujet `[RICM4-PS] DM2` avant le 3 janvier à minuit.

Graphe de Erdös-Renyi Un graphe de Erdös-Renyi $G(N,p)$ est un graphe à N sommets, généré aléatoirement, et dans lequel les arêtes sont tirées indépendamment selon une loi de Bernoulli de paramètre p.

Calcul de Connexité Un certain nombre de sujets font appel à la notion de connexité dans un graphe. Le but n'est pas de faire des calculs de connexité avec une complexité optimale mais il faut quand même que ça soit calculé suffisamment rapidement pour que vous puissiez faire vos études... Une méthode simple consiste généralement à passer par une matrice d'adjacence, par exemple de la façon suivante :

```
N = 10;
A = matrix(data = c(F,T,F,T,F,F,F,F,T,F,T,F,F,T,F,F,F,F,F,F,T,T,T,
                    F,T,F,F,T,F,T,F,T,F,T,T,T ,T,T,T,T,T,F,F,F,F,F,F,
                    F,T,F,T,T,T,T,T,T,F,T,T,T,F,T,T,T,T,F,T,T,T,F,T,F,
                    T ,T,F,F,F,F,F,T,T,F,T,T,T,F,F,T,T,T,T,F,F,F,F,T,
                    T,F,T),nrow=N);
X1 = matrix(data = c(F,T,F,F,T,F,F,F,T,F), nrow=N); # un ensemble de sommets
X2 = ((A %*% X1) != 0) # permet d'obtenir l'ensemble des voisins de X1
as.vector(X2);
sum(X2);
```

```
 [1]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
[1] 7
```

Sujet 1 : Transmission de virus

Un modèle classique de diffusion d'épidémie consiste à modéliser le contact entre les individus. Si l'un d'eux est infecté alors il transmet le virus à celui qui ne l'est pas en une unité de temps. L'objectif de ce problème est d'étudier l'impact de la structure du graphe de connection entre les individus sur le temps de diffusion de l'épidémie, c'est à dire le temps nécessaire pour que toute la population soit infectée.
On modélise le système par un graphe représentant les contacts potentiels entre les individus, et la dynamique de propagation par le protocole :

- tirer uniformément un couple d'individus voisins sur le graphe ;
- si l'un des deux individus est infecté alors la maladie se propage à l'autre (si les 2 sont infectés ou si aucun n'est infecté alors rien ne change).

On note N la taille de la population et T_N le temps nécessaire pour que toute la population soit infectée.

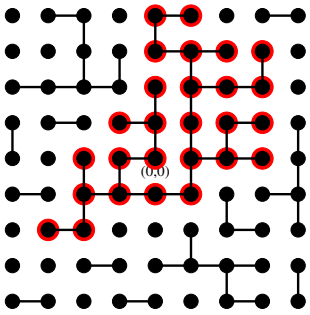
▷ **Q1. Cas du graphe complet**

- Étudier T_N en fonction de N lorsque le graphe des contacts potentiels est complet.
- Calculez l'espérance de T_N (on pourra commencer par étudier la probabilité de contaminer une nouvelle personne sachant qu'il y a déjà i personnes contaminées).
- Vous vérifierez la cohérence de ce résultat avec vos observations.

▷ **Q2. Graphe ligne 1D et grille 2D**

- Reprendre l'étude lorsque le graphe de contact est une ligne unidimensionnelle (attention T_N dépendra du premier individu infecté).
- Reprendre l'étude quand ce graphe est une grille 2D de coté $\sqrt{N}$.

Sujet 2 : Communication dans une grille non fiable



L'objectif de ce sujet est d'évaluer la connexité d'un réseau non fiable. Le réseau est constitué d'une grille 2D où chaque nœud interne est connecté à 4 voisins. Malheureusement les liens entre les voisins ne sont pas fiables et peuvent tomber en panne.
On suppose qu'un lien est en panne avec la probabilité $1-p$ et on s'intéresse à l'ensemble des nœuds connectés au nœud $(0,0)$.

▷ **Q1. Algorithme de génération** Écrire un algorithme qui génère une grille $[-n,n]\times[-n,n]$ et dont les liens sont en panne avec la probabilité $(1-p)$.

▷ **Q2. Connectivité** Étudier pour un n fixé et suffisamment grand, en fonction de p le nombre de nœuds connectés au nœud $(0,0)$.
En particulier, vous vous poserez la question de l'existence d'une probabilité critique p_c telle que si $p<p_c$ le nombre de nœuds connectés au nœud $(0,0)$ est supérieur à n^2 (i.e., moralement aussi grand que le graphe tout entier).

Sujet 3 : Connexité d'un réseau pair à pair

L'objectif de ce sujet est d'étudier la connexité d'un réseau pair à pair non structuré et que l'on modélisera donc à l'aide d'un graphe d'Erdös-Renyi $G(n,p)$.

▷ **Q1. Sommets isolés**

- Vérifiez expérimentalement que $\mathbb{E}(\text{points isolés de } G(n,(c+\log(n))/n)) \xrightarrow[n\to\infty]{} e^{-c}$.
- Démontrez le.
- Vérifier expérimentalement que le nombre X_n de sommets isolés du graphe $G(n,(c+\log(n))/n)$ converge en loi vers une loi de Poisson de paramètre e^{-c}.

▷ **Q2. Connexité**

- Vérifier expérimentalement que $\mathbb{P}(G(n,(c+\log(n))/n) \text{ est connexe}) = e^{-e^{-c}}$
- Quel est l'espérance du degré d'un graphe d'Erdös-Renyi ? Qu'en concluez vous sur le degré moyen nécessaire à maintenir la connexité d'un graphe pair à pair non structuré ?

Sujet 4 : Connexité d'un réseau de capteurs

L'objectif de ce sujet est d'évaluer la couverture d'un espace sur lequel on a déposé des capteurs de manière aléatoire. Pour simplifier l'approche on considère que l'espace est représenté par un disque de rayon 1 et que les n capteurs sont répartis selon une loi uniforme sur ce disque. Les capteurs ont une portée de communication θ, c'est à dire que 2 capteurs situés à une distance l'un de l'autre inférieure à θ peuvent échanger des informations.

▷ **Q1. Génération du réseau**

- Écrire un algorithme qui génère n points dans le disque. Calculer la matrice des distances entre les différents capteurs et par suite le graphe de connexion.
- Représentez graphiquement trois graphes (d'une taille n de votre choix) afin d'illustrer l'impact de θ sur la structure du graphe.

▷ **Q2. Connexité** Étudier en fonction de n et de θ la probabilité $p_c(n,\theta)$ que l'ensemble des capteurs soient connectés (il existe un chemin entre 2 capteurs quelconques). Pour cela tracer $p_c(n,\theta)$ en fonction de θ pour $n = 10, 20, 100$. On estimera l'erreur à l'aide d'intervalles de confiance.

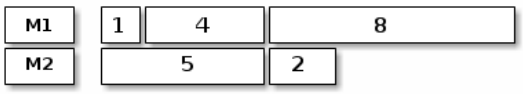
Sujet 5 : Ordonnancement

On dispose de N tâches de durée $t = t_1, \ldots, t_N$ que l'on souhaite répartir au mieux entre P machines de façon à minimiser leur temps de traitement. Une allocation a est donc une fonction de $\{1,\ldots,N\}$ dans $\{1,\ldots,P\}$ qui indique sur quelle machine chaque tâche est traitée et le coût d'une allocation C_a est donc égal à

$$C_a(t) = \max_{p\in\{1,\ldots,P\}} \left(\sum_{j \text{ tel que } a(j)=p} t_j \right)$$

Une approche gloutonne possible consiste, à partir d'une liste de priorité sur les tâches (une permutation de $\{1,\ldots,N\}$), à allouer chacune des tâches sur la machine lui permettant de finir le plus tôt possible.

Ainsi, si on dispose de $P = 2$ machines et que l'ensemble des durées est $t = (1, 4, 5, 8, 2)$, l'allocation associée à la liste de priorité $\sigma = (1, 3, 2, 4, 5)$ sera la suivante et son coût $C(\sigma, t)$ sera donc de 13.



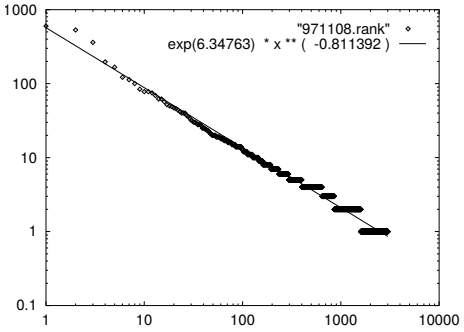
▷ **Q1. Génération de l'ensemble des allocations**

- Combien y a-t-il de liste de priorités possibles pour le petit exemple précédent.
- Écrire une fonction qui génère l'ensemble des listes de priorités possibles puis utiliser cette fonction pour étudier la répartition des $C(\sigma, t)$ sur l'exemple précédent.
- Se fixer maintenant $N = 10$ et $P = 3$ et générer une instance t telle que $t_j \sim U(1, 2)$ pour tout j. Étudier la répartition des $C(\sigma, t)$.

▷ **Q2. Génération uniforme d'allocation** On suppose maintenant que $t_j \sim U(1, 2)$ pour tout j et que la liste de priorité σ sera tirée uniformément sur sur l'ensemble des listes de priorités possibles. On s'intéresse à la variable aléatoire $C(\sigma, t)$.

- Écrire une fonction qui génère une liste de priorité σ uniformément.
- Utiliser cette fonction pour étudier la distribution de $C(\sigma, t)$ (on supposera par exemple que $N = 30$ et $P = 4$).
- Étudier la distribution de $C(\sigma^{\downarrow}, t)$ où $\sigma^{\downarrow}$ est la liste de priorité consistant à trier les durées des tâches par ordre décroissant.
- Pouvez vous conclure que l'allocation résultant de $\sigma^{\downarrow}$ obtient en moyenne de meilleurs résultats qu'une allocation aléatoire ?

Sujet 6 : Topologie d'Internet



En 1999, les frères Faloutsos ont publié dans la conférence SIGCOMM le résultat surprenant suivant. Le graphe d'Internet dont la croissance est le fruit d'actions non coordonnées et d'apparence aléatoire semble avoir une structure particulière : La distribution des degrés de nœuds semble suivre une loi de puissance.
Dans le graphique ci-contre, les nœuds ont été trié par degré décroissant et les chercheurs ont tracé les points de coordonnées (r_v, d_v) où d_v est le degré du nœud v et r_v est son rang dans le tableau des nœuds une fois trié par degré décroissant.

▷ **Q1. Estimation de la structure d'Internet à partir du graphique**

- Environ combien de nœuds le graphe d'Internet analysé comportait il ?
- À partir de la formule donnée sur le graphe, estimez le degré moyen des nœuds du graphe.

▷ **Q2. Distribution du degré pour Graphe Erdös-Renyi** Générer des graphes $G(n, p)$ selon le modèle d'Erdös-Renyi. Vous utiliserez pour n et pour p les valeurs obtenues précédemment.
En refaisant un graphique similaire à celui de l'article, peut on considérer que les graphes d'Erdös-Renyi sont un bon modèle de la topologie d'Internet ?

▷ **Q3. Modèle de Barabási-Albert** Quelques années plus tard, Barabási et Albert ont proposé un nouveau modèle de graphe aléatoire. Le graphe est construit de façon incrémentale.

- On part d'un réseau à deux nœuds connectés l'un à l'autre.
- Chaque nouveau nœud se connecte à chacun des nœuds i précédents avec une probabilité $p_i = \frac{k_i}{\sum_j k_j}$ où k_i est le degré courant du noeud i (un nouveau nœud peut donc se connecter à plusieurs nœuds déjà existant).

Intuitivement, les nouveaux nœuds se rattachent en priorité à des nœuds populaires qui agissent comme des sortes de "hubs".
Générer des graphes selon ce modèle et refaire une analyse similaire à celle de l'article. Peut-on considérer que ce modèle de graphe est un meilleur modèle de la topologie d'Internet ?

Sujet 7 : Génération aléatoire de labyrinthes

Un labyrinthe de taille $n \times n$ est constitué par des obstacles placés aléatoirement sur une grille, on note p la probabilité d'avoir un obstacle en un point de la grille. L'entrée du labyrinthe se fera par le point A (en bas à gauche de la grille) et la sortie par le point B (en haut à droite de la grille)

▷ **Q1. Algorithme de génération de labyrinthe**

- Écrire un algorithme à base de rejet qui génère uniformément, en fonction de p, un terrain acceptable, c'est-à-dire tel qu'il existe un chemin entre A et B.
- Représentez graphiquement quelques labyrinthes afin de tester votre algorithme de génération.

▷ **Q2. Complexité de l'algorithme**

- Estimer la probabilité de rejet quand $n = 10$ en fonction de p.
- Même question quand $n = 20$.

Sujet 8 : Complexité algorithmique

L'objectif de ce sujet est d'évaluer une propriété simple des arbres binaires générés uniformément.

▷ **Q1. Algorithme de génération** Écrire un algorithme de génération uniforme d'arbre binaire de taille n (vous utiliserez l'approche basée sur les nombre de Catalan vue en cours).

▷ **Q2. Hauteur moyenne** On note H_n la hauteur d'un un arbre binaire de taille n généré uniformément.

- Étudier pour différentes valeurs de n la distribution H_n.
- Vérifier expérimentalement que $\mathbb{E}(H_n) \approx 2\sqrt{\pi n} + c.\log(n) + c' + o(1)$ Vous essayerez de déterminer c.