Domaine Sciences et Technologies
Master 1 Informatique

Aix*Marseille université

Aspects probabilistes pour l'informatique : TP 2
Code UE: SINB19AL

Année 2018-19

Marches aléatoires et chaines de Markov

Observation : *Les exercices 1 et 3 sont obligatoires (pour obtenir une note de 3 sur le TP 2). Pour obtenir une note de 4/4.5 sur le TP 2, il vous faudra en plus effectuer un des exercices 2 ou 4.*

Exercice 1 (Marches aléatoires dans un graphe) Une marche aléatoire dans un graphe orienté $G = (V,E)$ est la création d'un chemin dans G depuis un sommet $u \in V$ donné où chaque successeur est choisi aléatoirement de manière uniforme sur l'ensemble de ses voisins : si on se trouve dans le sommet $v \in V$, on choisit comme successeur le sommet w avec probabilité $1/|\{(v,w) \in E\}|$. Le choix à une étape est donc indépendant des choix faits aux étapes précédentes.

Voici quelques questions qu'on peut se poser dans ce cadre :

— quel est le nombre moyen d'étapes nécessaires pour aller d'un sommet u à un sommet v ?

— à partir d'un sommet u, combien d'étapes sont nécessaires en moyenne pour visiter tous les sommets du graphe ?

Expérimentons sur le graphe complet $K_n = (\{0,1,2,\ldots,n-1\},\{(u,v) \mid u \neq v\})$ à n sommets.

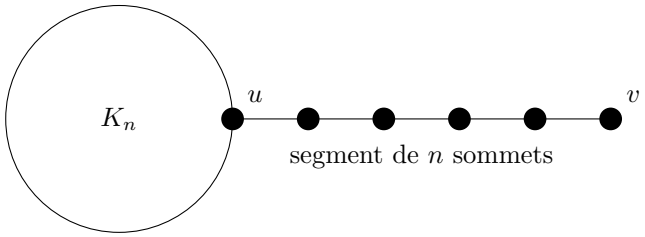
Question 1 Estimer, à l'aide d'un programme effectuant un grand nombre d'exécutions de la marche aléatoire, le nombre moyen d'étapes nécessaires pour aller d'un sommet u à un sommet $v \neq u$ fixés : on parle du *temps de premier passage* (*hitting time*, en anglais). Conjecturer une formule reliant ce nombre au nombre n de sommets.

Question 2 Estimer, à l'aide d'un programme effectuant un grand nombre d'exécutions de la marche aléatoire, le nombre moyen d'étapes nécessaires pour visiter tous les sommets de V à partir d'un sommet u fixé : on parle du *temps de couverture* (*cover time*, en anglais). Conjecturer une formule reliant ce nombre au nombre n de sommets.

D'autres graphes particuliers permettent de représenter un marcheur aléatoire sur une ligne ou sur un plan.

Question 3 Modéliser la marche aléatoire sur un chemin possédant n sommets alignés avec un graphe. Estimer, à l'aide d'un programme effectuant un grand nombre d'exécutions de la marche aléatoire, le nombre moyen d'étapes nécessaires pour aller d'une extrémité à l'autre du segment.

Question 4 Combinons le chemin avec une clique pour obtenir une « sucette » : c'est un graphe à $2n$ sommets, n sommets formant une clique (K_n) ainsi que n sommets qui partent en chemin depuis l'un des sommets u de la clique. À l'aide d'un programme, estimer le temps de couverture en partant de u ou en partant de v.



Question 5 Considérons désormais le problème d'un marcheur aléatoire qui se déplace à partir d'un sommet u d'une grille bidimensionnelle régulière. Estimer, à l'aide d'un programme effectuant un grand nombre d'exécutions d'une marche aléatoire sur un graphe à déterminer, le nombre moyen d'étapes nécessaires pour atteindre pour la première fois un sommet de la grille à distance (à vol d'oiseau) au moins n de u. Ici, en supposant que u a coordonnées $(0,0)$ dans la grille, on considèrera que la distance de u à un sommet v de coordonnées (x,y) est $\sqrt{x^2+y^2}$. En déduire une conjecture sur la distance moyenne atteinte par le marcheur aléatoire après p étapes.

Exercice 2 (2-SAT par marches aléatoires) Utilisons enfin une marche aléatoire pour résoudre le problème 2-SAT de logique. Il s'agit d'un cas particulier du problème SAT, de satisfaisabilité d'une formule de la logique propositionnelle, au cas de formules en forme canonique conjonctive $\bigwedge_i(\ell_{i,0} \vee \ell_{i,1} \vee \cdots \vee \ell_{i,n_i})$ (avec $\ell_{i,j}$ un littéral, c'est-à-dire une variable propositionnelle ou sa négation), où chaque clause admet deux littéraux (c'est-à-dire $n_i = 2$). Le problème SAT est NP-complet, mais le problème 2-SAT est lui résoluble en temps polynomial. Appliquons une sorte de marche aléatoire dans l'hypercube de dimension n (où n est le nombre de variables dans la formule) pour le résoudre très simplement. Les sommets du graphe que nous considérons alors consistent en les valuations des variables propositionnelles. Les arêtes du graphe consistent à modifier de la manière suivante une telle valuation :

— à partir d'une valuation ν des variables ne satisfaisant pas la formule, on choisit aléatoirement uniformément une clause de la formule qui n'est pas satisfaite ;
— on choisit alors aléatoirement uniformément l'un des deux littéraux de la clause et on inverse sa valuation pour trouver la nouvelle valuation.

On poursuit la marche aléatoire tant qu'on n'a pas rencontré une valuation satisfaisant la formule.

Question 1 Écrire un programme qui prend une formule de 2-SAT en argument, ainsi qu'un entier H servant d'horizon, et applique au plus H étapes de la marche aléatoire précédente à la formule en partant d'une valuation arbitraire (par exemple, falsifiant toutes les variables).

Question 2 Si la formule est satisfaisable, le temps moyen pour que la marche aléatoire précédente ait trouvé une valuation la satisfaisant est en $O(n^2)$ où n est le nombre de variables. En déduire un algorithme de Monte-Carlo en temps polynomial qui retourne *Insatisfaisable* si la formule n'est pas satisfaisable, et retourne avec forte probabilité (qui doit être un argument de l'algorithme) une valuation satisfaisant la formule si celle-ci est satisfaisable. L'implémenter et le tester sur des formules de votre choix. *En l'absence d'implémentation pour vérifier si votre algorithme réussit ou échoue, vous pouvez utiliser l'information suivante : parmi l'ensemble des formules ayant 10 variables et 20 clauses, environ* 45% *d'entre elles sont satisfaisables (ce même ratio tombe à* 24% *environ pour des formules ayant 20 variables et 40 clauses). Vous pouvez donc générer aléatoirement un grand nombre de formules aléatoirement satisfaisant cette propriété et vérifier que votre algorithme déduit bien que la bonne proportion est satisfaisable.*

Exercice 3 (Pile ou Face) Considérons un jeu pile ou face à deux joueurs, avec une pièce équilibré. On attribue à chaque joueur une séquence de pile ou face, par exemple la séquence PFP pour le joueur 1 et PPP pour le joueur 2. On répète alors le lancer de la pièce jusqu'à ce que l'une des deux séquences apparaissent dans les derniers lancers : ainsi, si on obtient la séquence PFFFPFFPPP, c'est le joueur 2 qui a gagné.

Question 1 Quelle est la probabilité que les trois premiers lancers soient PFP ? Quelle est la probabilité que les trois premiers lancers soient PPP ? Ce jeu vous semble-t-il équitable ?

Question 2 Simulons ce jeu. En réalisant 500 parties, renvoyer le pourcentage de victoires de chacun des deux joueurs (on pourra remarquer qu'on a besoin de ne retenir que les deux derniers lancers pour décider si un joueur a gagné...) Qu'en déduire sur l'équité de jeu ?

Question 3 Essayons de comprendre le paradoxe précédent. Pour cela, modéliser ce jeu à l'aide d'une chaîne de Markov qui aura, entre autres, trois états ε (pour représenter le début du jeu), J1 (pour signifier que le joueur 1 a gagné) et J2 (pour signifier que le joueur 2 a gagné). Donner son graphe et sa matrice de probabilité de transitions. En particulier, depuis l'état J1, la seule transition sortante va vers J1 avec probabilité 1 (même chose pour J2).

Question 4 Iterer la matrice de probabilité de transitions de la chaîne de Markov pour conjecturer à nouveua la probabilité que le joueur 1 gagne la partie.

Question 5 Observer que la chaîne de Markov $\mathcal{M}$ précedente n'est pas irréductible. Une façon classique de la rendre irréductible est de remplacer les transitions J1 et J2 par une transition vers l'etat ε : cela signifie qu'une fois que l'un des joueurs a gagné, on reprend une nouvelle partie à zero. Cette nouvelle chaîne de Markov $\mathcal{M}'$ est-elle irréductible ? Est-elle apériodique ?

Question 6 Itérer la nouvelle matrice de probabilité de transitions de la chaîne $\mathcal{M}'$. Que remarquez-vous ?

Question 7 On cherche à calculer de manière exacte la distribution stationnaire π de la chaîne $\mathcal{M}'$ de matrice P, c'est-à-dire trouver une distribution de probabilité π telle que $\pi P = \pi$. En terme de calcul matriciel, cela veut dire que π est un vecteru propre (à gauche) associé à la valeur propre 1 de la matrice P. Calculer la distribution stationnaire de $\mathcal{M}'$.

Question 8 La probabilité que le joueur 1 gagne dans le jeu initial est égal à la probabilité de terminer dans l'état J1 dans la chaîne $\mathcal{M}$. Dans la chaîne $\mathcal{M}'$ cela devient la probabilité d'être dans l'état J1 **sachant** qu'un des deux joueurs gagne. En déduire la valeur de la probabilité que le joueur 1 gagne dans le jeu de pile ou face.

Question 9 On peut retenter l'expérience avec d'autres séquences. Par exemple, vous pourriez essayer de voir qui des séquences FFF et PPFF gagne avec la plus grande probabilité...

Indication. Pour les operations matricielles vous pouvez utilser SageMath. En SageMath, on peut créer une matrice à l'aide de la classe Matrix. par exemple, la matric identité de taile 2×2 est obtenu à l'aide du code M=Matrix([[1,0],[0,1]]). On peut multiplier un vecteur ligne x à une matrice M par x*M. On peut élever la matrice M à la puissance n via Mn̂.

On peut aussi calculer les valeurs/vecterus propres à l'aide de la méthode eigenmatrix_left(). Pour cela, il est cependant nécessaire de dire à SageMath dans quel espace sont les coefficients de la matrice : on peut par exemple lui dire de considérer que les coefficients sont des rationnels en précisant la définition de la matrice grâce à Matrix(QQ,[[1,0],[0,1]]). La méthode eigenmatrix left() renvoie alors deux matrices : une matrice D diagonale dont les coefficients diagonaux sont les valeurs propres et une matrice V dont les lignes sont des vecteurs propres (à gauche) associés à chaque valeur propre. Une fois récupéré le vecteur propre x associé à une valeur propre particulière, on peut diviser le vecteur par sa norme 1 (c'est-à-dire la somme des valeurs absolues de ses coefficients) à l'aide de x/x.norm(1) : si x est un vecteur à coordonnées positives, on obtient ainsi un vecteur décrivant une distribution de probabilités. On peut changer de format pour les coefficients d'un vecteur ou d'une matrice en réutilisant Matrix ou vector : typiquement, pour transformer un vecteur x de rationnels en vecteur de réels (de nombres flottants), on utilise vector(RR,x).

Exercice 4 (PageRank) Supposons qu'il y a N pages web dans le monde, nommées $p_1, \ldots, p_N$. L'algorithme de PageRank crée une chaîne de Markov avec N états $\{1, \ldots, N\}$. Si la page p_i a des liens vers k autres pages, alors on pose $P_{i,j} = \frac{0.85}{k} + \frac{0.15}{N}$ si $p_i (i \neq j)$ a un lien vers la page p_j et $P_{i,j} = \frac{0.15}{N}$ sinon. Cela reflète qu'avec probabilité 0.85 on continue le surf aléatoire et avec probabilité 0.15 on répart à zéro en tapant une nouvelle adresse dans la barre d'URL du navigateur.

Considerons un graphe G à 5 sommets A,B,C,D,E et 9 arcs $A \to B$,$B \to A$,$A \to C$,$C \to A$,$B \to E$,$C \to E$,$E \to C$,$E \to D$ et $D \to A$.

Question 1 Commencer par calculer la matrice P de probabilité de transitions de la marche aléatoire associé à ce graphe. Est-elle irreductible ? Est-elle apériodique ? Calculer les itérés de cette matrice pour observer la périodicité.

Question 2 Modifier ensuite la matrice pour encoder la chaîne de Markov correspondant au PageRank : il suffit de remarquer que la nouvelle matrice vaut $0.85 \times P + \frac{0.15}{N} \times M$ avec N le nombre des pages et M une matrice pleine de 1. Vérifier que la nouvelle chaîne est irréductible et apériodique. Calculer la distribution stationnaire pour en déduire l'ordre de popularité PageRank.

Indication. En SageMath on peut récupérer une matrice à a lignes et b colonnes, à coefficients rationnels, à l'aide de la commande ones_matrix(QQ,a,b).