PROJET 19

Algorithme de Wilson pour la génération d'arbres couvrants uniformes

sujet proposé par L. Massoulié

laurent.massoulie@inria.fr

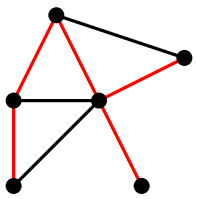

Figure 19.1 – Exemple de graphe avec un arbre couvrant en rouge.

Un graphe $G = (V, E)$ non orienté est défini par l'ensemble de sommets V et l'ensemble E d'arcs, qui sont des paires de sommets[1]. Un graphe est dit connexe si chaque paire de sommets i, j est reliée par un chemin (i, i_1), (i_1, i_2),... (i_k, j) d'arcs du graphe. Un graphe est un **arbre** s'il est connexe et ne contient aucun cycle non trivial, ou de manière équivalente, s'il est connexe et tel que $|E| = |V| - 1$.

Pour un graphe $G = (V, E)$, un **arbre couvrant** de G est un graphe arbre $T = (V(T), E(T))$, tel que $V(T) = V$, et $E(T) \subset E$. En d'autres termes, T est un arbre constitué d'arcs de G et connectant tous les sommets de G. Le nombre d'arbres couvrants d'un graphe G pouvant être exponentiel en le nombre de sommets de G, il n'est a priori pas évident d'obtenir un arbre couvrant choisi uniformément distribué au hasard.

L'algorithme de Wilson permet, pour un graphe non orienté connexe G, avec ensemble fini de sommets V, de simuler un arbre couvrant uniformément au hasard parmi tous les arbres couvrants de G. Il procède de la manière suivante.

Initialisation: $T_0 = \{r\}$, arbre réduit à un unique sommet $r \in V$ choisi arbitrairement.

Itération, étape i: ayant construit un arbre T_{i-1}, prendre un sommet $u \in V \setminus V(T_{i-1})$, où $V(T_{i-1})$ désigne l'ensemble des sommets de l'arbre T_{i-1}.

(a) Engendrer une marche aléatoire $\{U_0, \ldots, U_N\}$ sur G partant de u, avec probabilités de transition du sommet k vers le sommet l donnée par $P_{k,\ell} = d_k^{-1} 1_{(k,\ell)\in E}$, où d_k est le nombre de voisins de k dans G[2], issue de $U_0 = u$, et arrêtée au premier instant N auquel elle atteint l'un des sommets de T_{i-1}.

[1] les graphes considérés sont non orientés donc E vérifie la propriété de symétrie: si $(i, j) \in E$, alors $(j, i) \in E$.

[2] Autrement dit, il s'agit d'une marche qui se déplace en choisissant à chaque pas un voisin uniformément parmi les voisins de sa position actuelle.

(b) Appliquer la procédure suivante d'effacement de boucles à cette marche: pour

$$n = \inf\{t \in \{1, \ldots, N\} : \exists s < t \text{ tel que } U_s = U_t\},$$

on efface la boucle $U_s, \ldots, U_{n-1}$ pour conserver uniquement $U_0, \ldots, U_{s-1}, U_n, \ldots, U_N$. Itérer cette procédure d'effacement jusqu'à absence de telles boucles. Le chemin résultant est noté $V_0 = u, V_1, \ldots, V_M = U_N$. On définit alors l'arbre T_i comme l'union de T_{i-1} et du chemin $V_0, \ldots, V_M$.

L'algorithme termine lorsqu'un arbre couvrant est obtenu.

La partie théorique établit que l'algorithme de Wilson produit bien un arbre couvrant uniforme. La partie simulation consiste à mettre en oeuvre cet algorithme.

1 Partie théorique

Pour chaque sommet k de G distinct de r, on se donne une séquence i.i.d. $\{A_k^t\}_{t\geq 1}$ de voisins de k dans G choisis uniformément au hasard. Ces séquences sont mutuellement indépendantes entre elles.

On interprète ces variables de la manière suivante: dans notre implémentation de l'algorithme de Wilson, la première fois qu'un sommet k est atteint par une des marches construites, son étape suivante est A_k^1; la deuxième fois, l'étape suivante est A_k^2; la t-ème fois, c'est A_k^t. L'ensemble des arcs orientés (k, A_k^1) définit un graphe orienté.

T1. On envisage de telles séquences comme des piles (d'épaisseur infinie) de choix aléatoires de voisins pour chaque sommet $k \neq r$. Considérant le graphe dirigé défini par les choix A_k^1 au sommet des piles, si celui-ci contient un cycle orienté $C = k_0, k_1 = A_{k_0}^1, \ldots, k_0 = A_{k_\ell}^1$, on s'autorise à le supprimer, c'est à dire à supprimer du haut des piles associées les choix correspondants $A_{k_0}^1, \ldots, A_{k_\ell}^1$. Un tel cycle est dit **directement suppressible**.

On appelle cycle toute séquence $C = (k_0, k_1 = A_{k_0}^{t_0}, \ldots, k_0 = A_{k_\ell}^{t_\ell})$. On dira qu'un cycle $C = (k_0, k_1 = A_{k_0}^{t_0}, \ldots, k_0 = A_{k_\ell}^{t_\ell})$ est **suppressible** si il existe une séquence de cycles $C_1, \ldots, C_k = C$ tels qu'on peut supprimer séquentiellement les cycles C_1, puis C_2, etc. jusqu'à $C_k = C$.

Prouver que pour tout cycle C suppressible, et tout cycle C' directement suppressible, alors on peut supprimer C via une séquence de cycles qui commence par le cycle C'.

T2. En déduire que seuls deux scénarios sont possibles: soit après chaque suppression d'un cycle arbitraire, il reste toujours un cycle qu'on peut encore supprimer (le processus de suppression des cycles ne termine jamais), soit le processus de suppression des cycles termine, auquel cas les cycles enlevés ne dépendent pas de l'ordre dans lequel on les a enlevés, et les valeurs au sommet des piles dans la configuration terminale ne dépendent pas non plus de cet ordre.

T3. Montrer qu'avec probabilité 1, l'algorithme de Wilson termine en un nombre d'étapes fini. En interprétant l'algorithme de Wilson comme un enlèvement de cycles, en déduire que avec probabilité 1 le processus de suppression des cycles termine.

T4. Soit $C_1, \ldots, C_M$ l'ensemble de cycles enlevables étant données les piles $\{A_k^t\}_{t\geq 1}$, $k \in V \setminus \{r\}$, et soit $\vec{T}$ le graphe orienté, libre de cycles, obtenu dans la configuration terminale après suppression des cycles. Justifier que la collection de cycles $\{C_1, \ldots, C_M\}$ est indépendante de $\vec{T}$. En déduire que l'arbre T obtenu en retirant l'orientation des arcs de $\vec{T}$ est uniformément distribué parmi les arbres couvrants de G.

2 Partie simulation

S1. Implémenter l'algorithme de Wilson: en prenant comme entrée un graphe non orienté G, obtenir en sortie un échantillon uniforme d'un arbre couvrant de G.

S2. Tester l'algorithme avec $G = (V, E)$ une grille à deux dimensions, i.e. $V = \{(x, y), x \in \{1, \ldots, n\}, y \in \{1, \ldots, n\}\}$ et $E = \{\{(x, y), (x', y')\} \in V^2 : |x - x'| + |y - y'| = 1\}$. Engendrer des images de 2 arbres couvrants résultants pour la grille avec $n = 10$, puis avec $n = 50$.

S3. Tester l'algorithme avec $G = (V, E)$ une grille triangulaire, i.e. $V = \{(x, y), x \in \{1, \ldots, n\}, y \in \{1, \ldots, x\}\}$ et

$$E = \{\{(x, y), (x', y')\} \in V^2 : \text{ soit } x = x' \text{ et } |y - y'| = 1, \text{ soit } x' = x + 1 \text{ et } y' \in \{y, y + 1\}\}.$$

Engendrer des images de 2 arbres couvrants résultants pour la grille triangulaire avec $n = 10$, puis avec $n = 50$.
