TD 09 – Random walks and Monte Carlo Markov Chain (corrigé)

Exercice 1. *Cover time in graphs*

Given a finite, undirected non-bipartite and connected graph $G = (V, E)$, recall that the *cover time* of G is the maximum over all vertices $v \in V$ of the expected time to visit all of the nodes in the graph by a random walk starting from v.

1. Recall that $h_{v,u}$ is the expected number of steps to reach u from v and $h_{u,u} = \frac{2|E|}{d(u)}$. Show that

$$\sum_{w \in N(u)} (1 + h_{w,u}) = 2|E|.$$

☞ Let $N(u)$ be the set of neighbors of vertex u in G. We compute $h_{u,u}$ in two different ways:

$$h_{u,u} = \frac{2|E|}{d(u)}$$

and

$$h_{u,u} = \frac{1}{d(u)} \sum_{w \in N(u)} (1 + h_{w,u}).$$

Hence

$$2|E| = \sum_{w \in N(u)} (1 + h_{w,u}).$$

2. Let T be a *spanning tree* of G (i.e. T is a tree with vertex set V). Show that there is a *tour* (i.e. a walk with the same starting and ending points) passing each edge of T exactly twice, once for each direction.

☞ Following Depth-first Search.

3. Let $v_0, v_1, ..., v_{2|V|-2} = v_0$ be the sequence of vertices of such tour. Prove that

$$\sum_{i=0}^{2|V|-3} h_{v_i, v_{i+1}} < 4|V| \times |E|.$$

☞ From Q1, we have $h_{v,u} < 2|E|$ for any edge uv. The result trivially follows.

4. Conclude that the cover time of G is upper-bounded by $4|V| \times |E|$.

☞ $\sum_{i=0}^{2|V|-3} h_{v_i,v_{i+1}}$ is the expected time to go through the tour, i.e the expected time for any vertex v_i visit all nodes in the graph in a strict order. Hence it is longer than the expected time for any vertex v_i to visit all nodes in the graph without such order restriction, i.e. the cover time of G is at most $\sum_{i=0}^{2|V|-3} h_{v_i,v_{i+1}} < 4|V| \times |E|$.

Exercice 2. *$s - t$ connectivity in graph*

Suppose we are given a finite, undirected and non-bipartite graph $G = (V, E)$ and two vertices s and t in G. Let $n = |V|$ and $m = |E|$. We want to determine if there is a path connecting s and t. This is easily done in linear time using a standard breadth-first search or depth-first search. Such algorithms, however, require $\Omega(n)$ space. Here we develop a randomized algorithm that works with only $O(\log n)$ bits of memory. This could be even less than the number of bits required to write the path between s and t. The algorithm is simple:

Perform a random walk from s on G. If the walk reaches t within $4n^3$ steps, return that there is a path. Otherwise, return that there is no path.

1. Show that the algorithm returns the correct answer with probability at least 1/2, and it only makes errors by returning that there is no path from s to t when there is such a path.

☞ If there is no path then the algorithm returns the correct answer. If there is a path, the algorithm errs if it does not find the path within $4n^3$ steps of the walk. The expected time to reach t from s (if there is a path) is bounded from above by the cover time of their shared component, which by Lemma 7.16 is at most $4nm < 2n^3$. By Markov's inequality, the probability that a walk takes more than $4n^3$ steps to reach s from t is at most 1/2.

Exercice 3. *2ColorNonMonochTriangles*

Une k-coloration d'un graphe est un assignement pour chaque sommet d'une couleur parmi k couleurs au total. Elle est *propre* si deux sommets adjacents ne reçoivent jamais la même couleur. Un graphe est k-colorable s'il existe une k-coloration propre. Soit G un graphe 3-colorable.

1. Prouver qu'il existe une 2-coloration (non propre) telle qu'aucun triangle n'est monochromatique (un triangle est monochromatique si les trois sommets qui le composent reçoivent la meme couleur).

☞ Il existe une coloration propre Rouge, Bleu, Vert. On recolorie les sommets verts en rouge. Chaque triangle contenait déjà un sommet rouge et un sommet bleu avant la recoloration, et c'est toujours le cas.

On considère maintenant l'algorithme suivant dont le but est de trouver une telle 2-coloration: on commence avec une 2-coloration arbitraire. Tant qu'il y a un triangle monochromatique, on choisit uniformément un sommet parmi les trois sommets qui le composent, et on change sa couleur. On veut étudier l'espérance du nombre de recolorations avant de s'arrêter.

Comme G est 3-colorable, il existe une coloration propre Rouge, Bleu, Vert (mais que l'on ne connaît pas). On note R (resp. B, V) l'ensemble des sommets colorés rouge (resp. bleu, vert) dans cette 3-coloration. Soit R_t (resp. B_t) l' ensemble des sommets de R qui participent à un triangle (ou plus!). Soit $n' = |R_t| + |B_t|$. Donc, $n' \leq n$. Considérons maintenant une 2-coloration arbitraire c de G (en rouge et bleu, disons). Soit $m(c)$ le nombre de sommets de R_t qui ne sont pas colorés rouge dans c , plus le nombre de sommets de B_t qui ne sont pas colorés bleu dans c.

2. Que dire si $m(c) = n'$ ou $m(c) = 0$?

☞ Dans ces cas, aucun triangle n'est monochromatique et l'on a terminé.

3. En s'inspirant de l'exemple du cours sur 2-SAT, modéliser l'évolution de $m(c)$ par une chaîne de Markov sur $\{0, \ldots, n'\}$. Quels sont le ou les sommets à atteindre pour terminer? Que pouvez-vous dire de l'état j par rapport à l'état $n' - j$ pour $j \in \{0, \ldots, n'\}$?

☞ Supposons $m(c) = j \neq 0, n'$ et regardons le triangle monochromatique choisi. Sans perte de généralité, on peut supposer qu'il est entièrement rouge. Avec proba 1/3, on tire le sommet de V et on le recolore: cela laisse $m(c)$ inchangé; avec proba 1/3, on tire le sommet de B et on le recolore: on passe à $m(c) + 1$; et enfin avec proba 1/3, on tire le sommet de R et on le recolore: on passe de $m(c)$ à $m(c) - 1$. La chaîne de Markov est donc ainsi: pour $j \neq 0, n'$, avec proba 1/3 on passe à $j - 1$, avec proba 1/3 on reste sur j et avec proba 1/3 on passe à $j + 1$. Pour $j = 0$ ou n', on reste sur l'état courant avec proba 1. Le but est d'atteindre le sommet 0 ou le sommet n'. La chaîne est complètement symétrique entre l'état j et l'état $n' - j$.

Attention: The algorithm might stop even if $m(c) \neq \{0, n'\}$. Consider a single triangle, with all vertices red. Initially, $m(c) = 1$. Assume we colour the "green" vertex blue. Then $m(c)$ remains 1, but it is not monochromatic anymore. Hence, algorithm $A = \{$ randomly fix a monochromatic triangle $\}$ can terminate at more states than just $\{m(c) = 0, m(c) = n + 1\}$. Let B the algorithm that follows the Markov-Chain i.e terminates only if $m(c) = 0$ or $m(c) = n'$. Then $time(A) \leq time(B)$. Intuitively: if A reaches $m(c) = 0, m(c) = n'$ it stops. Hence, it is at least as fast as B. We thus focus on the analysis of B to upper bound $time(A)$.

4. Soit h_j l'espérance du nombre de recolorations à effectuer pour terminer, en partant d'une 2-coloration c pour laquelle $m(c) = j$. Exprimer h_j en fonction de h_{j-1} et h_{j+1} pour $j = 1 \ldots (n' - 1)$. Determiner h_0 et $h_{n'}$.

☞ On a $h_0 = 0$ et $h_{n'}$. De plus, on a:

$$h_j = \frac{1}{3}(1 + h_{j-1} + 1 + h_j + 1 + h_{j+1})$$

autrement dit

$$h_j = \frac{3}{2} + \frac{1}{2}(h_{j-1} + h_{j+1})$$

5. Montrer que $h_j = h_{j+1} + f(j)$ pour une certaine fonction f à déterminer, avec $f(0) = -h_1$.

☞ On a $h_0 = h_1 + f(0) = h_1 - h_1 = 0$: ok.

Puis

$$h_j = \frac{3}{2} + \frac{1}{2}(h_{j-1} + h_{j+1}) = \frac{3}{2} + \frac{1}{2}(h_j + f(j-1) + h_{j+1})$$

donc

$$\frac{1}{2}h_j = \frac{3}{2} + \frac{1}{2}f(j-1) + \frac{1}{2}h_{j+1}$$

donc $h_j = h_{j+1} + 3 + f(j-1) = h_{j+1} + f(j)$ avec $f(j) = 3 + f(j-1)$ donc $f(j) = 3j - h_1$.

6. Prouver que $h_{n'/2} = \mathcal{O}(n^2)$ et conclure. *(On pourra utiliser la relation $h_1 = h_{n'-1}$ que l'on obtient par symétrie, pour finir de résoudre la récurrence).*

☞ On a $h_{n'-1} = h_{n'} + f(n'-1) = 0 + f(n'-1)$. Comme $h_1 = h_{n'-1}$ et $f(n'-1) = 3(n'-1) - h_1$, on obtient: $h_1 = 3(n'-1) - h_1$ donc $h_1 = 3(n'-1)/2$. Donc $h(j) = h_{j+1} + 3j - 3(n-1)/2$.

Donc

$$h_j = h_n + \sum_{k=j}^{n'-1}(3j - \frac{3}{2}(n'-1)) = 0 + 3\sum_{k=j}^{n'-1} j - \frac{3(n'-1)(n'-j)}{2} = 3\frac{(n'-1+j)(n'-j)}{2} - \frac{3(n'-1)(n-j)}{2} = 3\frac{(n'-j)}{2}(n-1+j-(n'-1)) = \frac{3j(n'-j)}{2}$$

donc

$$h_{n'/2} = \frac{3(n'/2)^2}{2} = \mathcal{O}(n^2).$$

(since $n' \in O(n)$)

Comme $h_{n'/2}$ est le "milieu" de la chaine, c'est le pire cas (on peut vérifier que c'est le maximum de h_j) et donc l'espérance du nombre de recolorations est quadratique.

Exercice 4. *PiMonteCarlo*

The goal of this exercise is to develop two algorithms that approximate the number π. The first is based on the so-called *direct-sampling*, the second uses *Markov-chain sampling*.

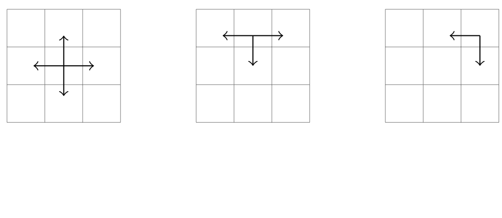
1. Let (x, y) be a point chosen uniformly at random from in a 2×2 square centred in the origin. Assume you have m such points. Describe an algorithm with a lower-bound on m that approximates the value π with error ε with probability larger than $1 - \delta$. In other words, your algorithm should return a value $\bar{\pi}$ s.t. $\mathbf{P}\{|\bar{\pi} - \pi| \leq \varepsilon\pi\} > 1 - \delta$.

☞ MU, p.252 Define a r.v. Z that takes value 1 if $\sqrt{x^2 + y^2} \leq 1$ (we landed in the circle), and 0 otherwise. Clearly $\mathbf{P}\{Z = 1\} = \frac{\textbf{area of the circle}}{\textbf{area of the square}} = \frac{\pi}{4}$. Averaging over m trials, gives $\mathbf{E}[\sum_{i=1}^{m} Z_i] = m\pi/4$. Set $W' := W \cdot (\frac{4}{m}) = \sum_{i=1}^{m} Z_i(\frac{4}{m})$ as an estimate for π. Chernoff bound leads to

$$\mathbf{P}\{|W' - \pi| \geq \varepsilon\pi\} = \mathbf{P}\{|W - \mathbf{E}[W]| \geq \varepsilon\mathbf{E}[W]\} \leq 2\exp(-m\pi\varepsilon^2/4 * 3)$$

so $m \geq 12\ln(2/\delta)/\pi\varepsilon^2$ for the above probability to be less than $1 - \delta$.

2. The above algorithm relies on the fact that you have a direct access to a uniform distribution on a square. In the following, we shall approximate a (rather coarse) discretized version of it. Consider the following moves inside a 3×3 grid. From the central position we move up/left/right/left each with probability $1/4$, in one of three directions from the edges or stay at the position with probability $1/4$; and in one of the two directions from the corners again with probability $1/4$ and stay in the corner with probability $1/2$. We start the walk from the upper-right corner.



Describe the transition matrix P for this Markov process. Find the eigenvectors and the corresponding eigenvalues of P. Conclude on the stationary distribution. Using the second largest eigenvalue, determine the rate of convergence (Hint: Express the starting state π^0 in terms of P's eigenvectors. What is the λ_1 and which eigenvector corresponds to it?) ☞ The question is taken from the book http://blancopeck.net/Statistics.pdf

The Markov chain on 9 states is represented by the following matrix:

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2} \end{pmatrix}$$

The convergence of the walk can be analysed by examining the eigenvectors $\{\pi_e^1, \ldots, \pi_e^9\}$ and the eigenvalues $\{\lambda_1, \ldots, \lambda_2\}$ of the transfer matrix.

Intuition: Let us write probability distribution π^0 as sum of the eigenvectors: $\pi^0 = \sum_{k=1}^{9} \alpha_k \pi_e^k$

After one iteration of Random Walk:

$$P\pi^0 = \sum_{k=1}^{9} \alpha_k \pi_e^k P = \sum_{k=1}^{9} \alpha_k \lambda_k \pi_e^k$$

After i steps :

$$P^i \pi^0 = \pi^i = \sum_{k=1}^{9} \alpha_k {\lambda_k}^i \pi_e^k$$

Assuming our random walks start from state 9: $\pi^{(0)} = (0, \ldots, 0, 1) = \sum_{i=1}^{9} \alpha_k \pi_e^k$, after the i-step we obtain

$$\pi^{(i)} = 1 \cdot (1/9, \ldots, 1/9, 1/9) + \alpha_2(\mathbf{0.75})^{\mathbf{i}}(-0.21, \ldots, -0.21) + \alpha_3(0.5)^i() + \ldots$$

- Note that stationary distribution is $\pi_e^1 = (1/9, \ldots, 1/9)$
- When $i \to \infty$ only π_e^1 remains (all other gradually turn 0)
- **Intuition**: At each step, influence of λ_k decreases by λ_k.
- Since $\lambda_2 > \lambda_3 > ... > \lambda_9$, the value λ_2 affects the convergence of π the most
- The convergence rate is **thus** $(0.75)^i = \exp(-i/3.476)$.