CS 537: Randomness in Computation. Fall 2013	October 29, 2013

Random Walks

Lecturer: Steve Homer — Speaker: Yilei Chen

Last week

- Markov Chain
- Transition Matrix of Markov Chain
- 2-Satisfiability
- Hidden Markov Model (HMM)
- Markov decision process (MDP, example: ipod shuffle algorithm)
- Classification of states

Today

- Random Walk
- A randomized algorithm for S-T Connectivity

1 Motivation

We start by revisiting a fundamental problem in computer science: given a (directed or undirected) graph $G = (V, E)$, two vertices $s, t \in V$, is there a path from s to t?

The traditional Depth-first and Breadth-first searching algorithms are known to solve S-T connectivity efficiently. Let n be the number of vertices in G, m be the number of edges, the Depth-first searching (so does the Breadth-first searching) requires (in worst case) linear times and spaces to record the intermediate path.

One of the greatest questions is: can we do better? In this lecture we focus on the algorithms with smaller space complexity. It turns out that we know some algorithms that works with less spaces, even without randomness:

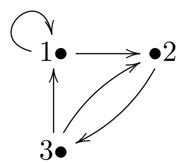
Theorem 1 *There's a deterministic recursive algorithm deciding S-T connectivity with space* $O(\log^2 n)$. [1]

However, with randomness, life is getting better. To analysis a randomized algorithm dealing with the S-T connectivity, we will introduce the notion of **Random Walk**.

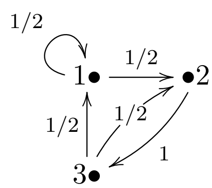
[1] But with time $O(n^{\log n})$, which is more than poly(n). See [3].

2 Random walk

A random walk on a graph $G = (V, E)$ is a special case of a Markov chain, where we pick the next state uniformly from the neighbors of the current state. For example, if we have the following graph



then the transition probabilities are fixed by



P	1	2	3
1	1/2	1/2	0
2	0	0	1
3	1/2	1/2	0

Formally, let $\deg_{\text{out}}(i)$ be the number of outedges from vertex i, we have

$$P(i,j) = \begin{cases} \frac{1}{\deg_{\text{out}}(i)} & \text{if } (i,j) \in E \\ 0 & \text{otherwise .} \end{cases}$$

2.1 Random walk on a graph

Before unveiling more properties of Random Walk, let's consider how it is related with the S-T connectivity problem. It turns out that, even without knowing any "deep" results in Random Walk, you might have roughly came up with an randomized algorithm to solve the problem. Let's look at the following algorithm:

Algorithm 1: deciding S-T connectivity

Input: $G = (V, E), s, t \in V$
(1) Initialized vertex $v = s$
(2) Repeat $T(n)$ times
  (2.1) If $v == t$, halt and accept
  (2.2) Else, let $v \overset{R}{\leftarrow} \{w : (v, w) \in E\}$
(3) Reject (if t is not visited after $T(n)$ times)

Compare with the traditional Depth-first or Breadth-first search, where maintaining a list of visited vertices is required (that's where linear space is coming from), this algorithm only requires $O(\log n)$ bits - the only thing needs to be recorded is the current vertex v as well as a counter for the number of steps taken.

Clearly, it never accepts when there is no path from s to t. The work remaining is to prove that the time complexity would not blow up too much, or at least in poly(n). We will show in the rest of the lecture that for an **undirected** [2] graph, a random walk of length $O(n^3)$ from one vertex will hit **any other** vertex with high probability. Applying this to the connected component of the graph containing s, it follows that the algorithm accepts with high probability when s and t are connected.

To prove this bound, we might revisit some properties of markov chain such as **existance of stationary distribution**.

2.2 Stationary distributions

A **stationary distribution** $\tilde{\pi}$ is one such that $\forall v \in \Omega$, we have

$$\tilde{\pi}(v) = \sum_u \tilde{\pi}(u) P(u, v) \ ,$$

or equivalently $\tilde{\pi} = \tilde{\pi} P$.

An important class of Markov chains is one in which a stationary distribution $\tilde{\pi}$ exists and is unique. It turns out ergodicity is sufficient to guarantee a stationary distribution, as stated by the following theorem.

Theorem 2 *Every ergodic Markov chain has a stationary distribution that is unique.*

Theorem 3 *Any finite, irreducible, and ergodic Markov Chain has a unique stationary distribution, and will be converged to from any starting state.*

For a graph which is finite, undirected and connected, we need only one more requirement to guarantee ergoticity (aperiodic and positive recurrent) - the graph cannot be bipartite. Note that if a graph is bipartite, you may never arrive at a stationary distribution simply due to oscillations between two sets. [3]

Theorem 4 *For a finite* undirected *graph G that is connected and not bipartite, the randon walk on G converges to a stationary distribution $\tilde{\pi}$:*

$$\tilde{\pi}(v) = \frac{d(v)}{2\,|E|} \tag{1}$$

where $d(v)$ is the degree of vertex v.

Proof Sketch: The stationary distribution $\tilde{\pi}$ is unique and will be arrived follows "connected" and "not bipartite". We only need to "check" if $\tilde{\pi}(v) = \frac{d(v)}{2|E|}$ is correct or not. Let $N(v)$ denote the neighbors of v. Since $\tilde{\pi} = \tilde{\pi} P$, we have

$$\tilde{\pi}(v) = \sum_{u \in N(v)} \frac{d(u)}{2\,|E|} \frac{1}{d(u)} = d(v) \cdot \frac{1}{2\,|E|}$$

[2] For directed graph it may not work. Indeed it's a good exercise.

[3] Or, look at the proof of lemma 7.12 in [1], page 175

□

Recall that $h_{v,u}$ denotes the expected number of steps to reach u from v. For the graph where the stationary distribution is unique, we have the following corollary:

Corollary 5 *If the stationary distribution $\tilde{\pi}$ is unique, then for any vertex v in G,*

$$h_{u,u} = \frac{1}{\tilde{\pi}(u)}$$

Proof Sketch:

The expected time between visits to u is $h_{u,u}$, and therefore state u is visited $\frac{1}{h_{u,u}}$ of the time, which is equal to $\tilde{\pi}$ if it is unique. □

2.3 Cover time

Finally, we define the **cover time** of a graph (undirected) to be

$$\begin{aligned} \mathcal{C}_u(G) &= E[\text{number of steps to reach all nodes in } G \text{ starts from } u] \\ \mathcal{C}(G) &= \max_u \mathcal{C}_u(G) \end{aligned}$$

Let's consider the cover time of some typical graphs:

- $\mathcal{C}(K_n^*) = \Theta(n \log n)$, where K_n^* is the complete graph on n nodes that included self loops. The bound follows from the coupon collector problem.
- $\mathcal{C}(L_n) = \Theta(n^2)$, where L_n is the line graph on n nodes (prove or google it).
- $\mathcal{C}(n\text{-node lollipop}) = \Theta(n^3)$, where an n-node lollipop is a $L_{n/2}$ with a $K_{n/2}^*$ at one of the ends. For intuition, the worst thing to do is start in the clique. Look at how many times you must hit the start of the line before getting all the way to the end. Roughly speaking, it's $\Theta(n^2)$ times, and you only escape the clique with probability $1/n$.

It turns out that the $\Theta(n^3)$ bound is the worst possible for cover time. We will prove something stronger in a moment.

Lemma 6 *For all $(u, v) \in E$, we have $h_{v,u} \le 2|E|$.*

Proof Sketch: Let $N(u)$ be the set of neighbors of u. We have

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

Therefore,

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

and we conclude that $h_{v,u} \le 2|E|$ □

Theorem 7 *For any graph $G = (V, E)$, we have $\mathcal{C}(G) \le 4\,|V|\,|E|$.*

Proof Sketch: Notice that $\frac{1}{h_{v,u}}$ can be thought as how frequently the edge (u, v) is visited. The idea of construction is to pick any start vertex v_0, and construct arbitrary spanning tree T of G rooted at v_0. Note that the number of **edges** in the tree is $n - 1$.

Let $v_0, v_1, v_2, \ldots, v_{2n-2}$ be a Depth-first traversal of the spanning tree T. Notice that $v_{2n-2} = v_0$, and each edge of T appears exactly twice, once in each direction.

We conclude that

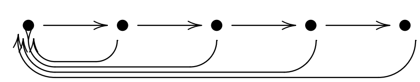
$$\begin{aligned}
\mathcal{C}(G) &\le \sum_{j=0}^{2n-3} h_{v_j v_{j+1}} \\
&= \sum_{(u,v)\in T} h_{v,u} \\
&\le \sum_{(u,v)\in T} 2\,|E| && \text{from Lemma 6} \\
&\le 4\,|V|\,|E|
\end{aligned}$$

□

3 Summary

Let $T(n)$ in algorithm 1 be $4n^3$, then if there's a path between s and t, the algorithm accepts with probability higher than 1/2 (which follows the Markov inequality); if there're no paths between s and t, our algorithm accepts with probability 0.

It's worth to know that this theorem does not hold for directed graphs (as mentioned in footnote 2). In particular, consider the graph



Here, the cover time $\mathcal{C}(G) = \Theta(2^n)$ which can be exhibited by starting at the leftmost node.

References

[1] Michael Mitzenmacher, Eli Upfal: Probability and computing - randomized algorithms and probabilistic analysis. Cambridge University Press 2005, ISBN 978-0-521-83540-4, page 167-177

[2] Ronitt Rubinfeld: Lecture notes 15 of Randomness and Computation, Spring 2008. `http://people.csail.mit.edu/ronitt/COURSE/S08/download/notes15.pdf`

[3] Salil Vadhan: Pseudorandomness. Chapter 2, page 22-27. `http://people.seas.harvard.edu/~salil/pseudorandomness/power.pdf`