COMPSCI 630: Randomized Algorithms March 26, 2018

Lecture 18

Lecturer: Rong Ge *Scribe: Jun Jiang*

1 Overview

In this first lecture, we will learn how to compute simple Markov chains using resister network.

2 Simple Markov Chains

A simple Markov chain is time-reversible, and thus can be represented by undirected graph.
A formal definition of time reversible is stated as follows:

Definition 1. *A Markov chain is time-reversible if there exists a stationary distribution π s.t.*

$$\forall i, j \in S, \pi_i P_{ij} = \pi_j P_{ji}$$

Here, π_i, π_j is the probability at state i and j, respectively. $P_i j$ is the transition probability from $i \to j$ when being at state i, and vice versa. A schematic illustration of a two-state simple Markov chain is shown below:

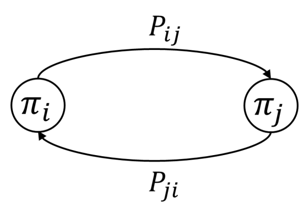

Figure 1: state diagram of simple two-state Markov chain

3 Representing Simple Markov Chain as Undirected Graph

For a simple Markov chain as defined above, we can use an undirected graph to represent it (i.e. we can use only one edge to connect two nodes). The **edge weight** for $e(i, j)$is defined by:

$$C_{ij} = \pi_i P_{ij} = \pi_j P_{ji}$$

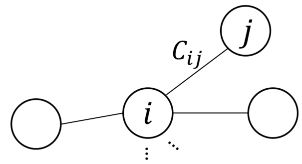

Figure 2: undirected graph representing node i and its surrounding nodes

Notice that $\sum_{j\in S(i)} C_{ij} = \sum_{j\in S(i)} \pi_i P_{ij} = \pi_i \sum_{j\in S(i)} P_{ij} = \pi_i \times 1 = \pi_i$, we can write P_{ij} as:

$$P_{ij} = \frac{C_{ij}}{\pi_i} = \frac{C_{ij}}{\sum_{j\in S(i)} C_{ij}}$$

Here, $S(i)$ denotes the space of all surrounding nodes of node i.

Note:

1. C_{ij} can be multiplied by a constant.
2. In general, $P_{ij} \neq P_{ji}$.
3. π_i is proportional to $\sum_{j\in S} C_{ij} = deg(i)$
4. In this lecture, we choose $C_{ij} = 1$ for all $(i, j) \in E$. As a result, $\pi_i = \frac{deg(i)}{2|E|}$, where $|E|$ denotes the total number of edges in this graph.

4 Compute Probability/Expectation of Markov Chains

4.1 Problem Description

Definition 2. *Define the following symbols as:*

$$\gamma_i = Pr\left[\textit{event happens, if we start random walk at state } i\right]$$

Note: Here ”event” refers to ”reach a certain state” or ”go through an edge” or other similar events.

In the following texts, we will show how to computer γ_i.

4.2 Solve the Problem Using Markov Property

The idea is to use Markov property and linearity of expectations.
Assuming the event cannot take place within one step. For node i, we have:

$$
\begin{aligned}
\gamma_i &= \sum_{j\in S(i)} Pr[event|j] \times Pr[state(j)|state(i)] \\
&= \sum_{j\in S(i)} \gamma_j P_{ij} \\
&= \frac{\sum_{j\in S(i)} \gamma_j}{deg(i)}
\end{aligned}
$$

If the graph has n nodes, we can write out n linear equations, i.e.

$$\gamma = P\gamma, \quad \gamma = (\gamma_1, \gamma_2, ..., \gamma_n)^T$$

There are two kinds of uncertainties for the solution of the above linear system:
If γ is a solution, then (1) $C \cdot \gamma$ is also a solution. (2) $\gamma + \lambda \vec{1}$ is also a solution.

4.3 Solve the Problem Using Resistor Network

Recall that in a resistor network, the voltage at node i is given by:

$$v_i = arg \min_{\mathrm{v}} \sum_{j\in S(i)} \frac{1}{2}(v_j - v)^2$$

Here, $\frac{1}{2}(v_j - v)^2$ is proportional to the energy generated by resistors between node i and j.
We can solve v_i by taking the derivative of this quadratic equation, and the result will be:

$$v_i = \frac{\sum_{j\in S(i)} v_j}{deg(i)}$$

We can see that the expression of v_i is the same as the expression of γ_i in the previous section. Therefore, by solving the voltages in a resistor network, we can solve the event probability in a simple Markov chain.

5 Examples

5.1 Example 1

Among the n states, pick two states a and b.
The event is defined as ”reaching a before reaching b”
And let $\gamma_i = Pr$ [event happens, if we start random walk at state i]

Solution:

In addition to the linear system equations $\gamma = P\gamma$, we notice that there are two boundary conditions:

$$\gamma_a = 1, \quad \gamma_b = 0$$

The boundary condition is analogous to applying a voltage source from a to b. In this way, we resolved the two uncertainties as described before, and we can get a unique solution.

5.2 Example 2

Among the n states, pick two states a and b.
The event is defined as "reaching a before returning to b"
And let $\gamma_i = Pr\,[\text{event happens, if we start random walk at state } i]$

The **escape probability** is defined as γ_b, i.e. the probability of reaching a before returning to b, if we start random walk at state b. Then the **escape probability** is given by:

Solution:

Notice that compared with Example 1, we only have one boundary condition $\gamma_a = 1$, which is not enough to solve the linear system.
But we can use the resistor network to solve γ_b.

$$\begin{aligned}\Pr[\text{escaping}] = \gamma_b &= \frac{\sum_{j \in S(b)} \gamma_j}{deg(b)} \\ &= \frac{1}{deg(b) \cdot R_{eff}}\end{aligned}$$

Here, $R_{eff} = \frac{1}{\text{current that goes into node b}}$ is the effective resistance between node a and node b.

5.3 Example 3

Among the n states, pick two states a and b.
Define the event as "get to state a".
Define X_i as: $X_i = E[\text{number of steps to event (get to state a), if starting from i}]$

Solution:

Using the same idea as in Section 4.2, we can get:

$$X_i = \frac{\sum_{j \in S(i)} X_j}{deg(i)} + 1$$

And we have one extra equation: $X_a = 0$.
As a result, we can solve the linear equations with unique solution. (Note: In comparison to Example 1 and 2, there is only one uncertain here. This is because in the equation above, there is an extra +1)