COMP 480/580 — Probabilistic Algorithms and Data Structure March 31, 2020

Lecture 17

Lecturer: Anshumali Shrivastava *Scribe By: Renhao Lei*

1 Motivation: Sampling from State Space

We use the good old problem to motivate the lecture. Given a state space, $\Omega = \{X_1, X_2, X_3, ..., X_n\}$, we want to sample x from Ω such that the probability of getting any $x \in \Omega$ is proportional to the weight associated with x, w_x. In other words,

$$\Pr(x_i \text{ is sampled}) \ = \frac{w_i}{\sum_{j=1}^{n} w_j}$$

We will suppose that Ω is so large that $\sum_{j=1}^{n} w_j$ is not easy to compute.

1.1 Example: Random Permutation Problem

An example illustrating this concept is the random permutations problem. Essentially, given N numbers, we want to generate a random permutation of these numbers. Here, Ω is the set of all permutations. Initially, we will suppose that $w_x = 1$ for each permutation, x (i.e. each permutation is equally likely). So that $|\Omega| = N!$

A concrete example following this model is the act of shuffling a deck of N cards. We represent each permutation of cards as a number and we want to randomly sample from the numbers $[1, N!]$. We can solve the random permutation problem using the algorithm below.

Algorithm 1: Random Permutation Problem

```
Input   : N numbers
Output: A random permutation of N numbers
Start with the order initialized to 1, 2, ... N;
for ctn ← 1 to k do
    select i, j ∈ [1...N] randomly;
    swap the ith element and jth element;
end
return The current order of elements;
```

This algorithm swaps different integers within the list k times. As we increase the number k, in particular, as $k \to \infty$, the returned permutation becomes truly random. This algorithm works for this specific problem, but only because all of our weights w_x were equal to 1. In the general case when $w_x \neq 1$, we require Markov chains.

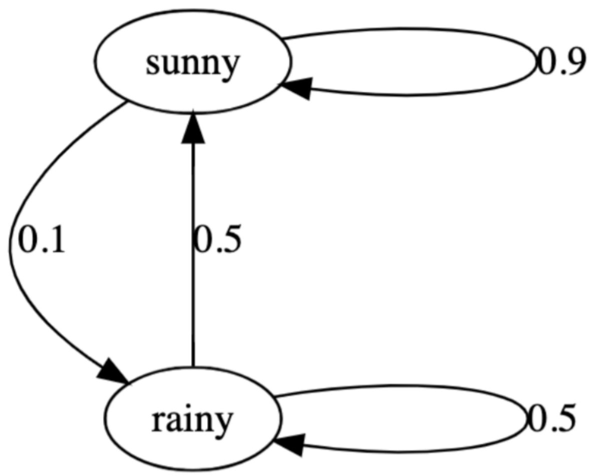
2 Markov Chains

A Markov chain is a stochastic process that consists of a sequence of states, $\{X_0, X_1, ..., X_t, ...\}$. Each state, X_i is from the set of all states, Ω. The interesting thing about Markov chains is that they are memory-less. This means that the next state is determined only by the current state, and no other state in the past matters. Thus, the probability of reaching state y at time t + 1, when we were at state X at time t can be written as:

$$Pr[X_{t+1} = y | X_t = x_t, X_{t-1} = x_{t-1}, ..., X_0 = x_0] = Pr[X_{t+1} = y | X_t = x_t] = P(y|x_t)$$

For the rest of our discussion of Markov chains, we will use $P(x, y)$ to denote the conditional probability $P(y|x)$. The following facts, which follow direction from the basic rules of probability, will be important in our analysis: $P(x, y) \geq 0$ for all $x, y \in \Omega$, and $\sum_{y \in \Omega} P(x, y) = 1$, for any $x \in \Omega$.

2.1 Markov Chains Example: Rainy or Sunny



A simple example of Markov chains: If it is sunny today, then the probability that it is sunny tomorrow is 0.9. If it is rainy today, then the probability that it is sunny tomorrow is 0.5. Similarly, if it is sunny today, then the probability that it is rainy tomorrow is 0.1. If it is rainy today, then the probability that it is rainy tomorrow is 0.5. In other words:

$$Pr(\text{S tomorrow}|\text{S today}) = 0.9$$

$$Pr(\text{S tomorrow}|\text{R today}) = 0.5$$

$$Pr(\text{R tomorrow}|\text{S today}) = 0.1$$

$$Pr(\text{R tomorrow}|\text{R today}) = 0.5$$

2.2 Non-Markov Chains Example: Draw a coin

Here is an example that is not a Markov Chains. Suppose we have 5 quarters, 5 dimes and 5 nickels in a box. Draw a coin randomly without replacement from the box. Let X_t be total value at time t. Then this process is not a Markov chain because X_t depends on all the previous draws and some coins might be unavailable at time t.

2.3 Markov Chains as a Graph

Based on the definition of a Markov chain, we can visualize the chain as a directed graph. Recall from graph theory that we can represent a graph g as a set of nodes V and edges $E : g = (V, E)$. The set of nodes is the set of states, so $V = \Omega$. The set of edges (x, y) is determined by the state transition probabilities of the Markov chain. A directed edge between two nodes represents a non-zero probability of transitioning from state x to state y. This edge has a weight of $P(x, y)$.

Due the laws of probability, our graph has some structural properties. For instance, the weights of all outgoing edges from any node must add up to 1. It should be noted that a node can also have an edge to itself. We can also see that representing the Markov chain as a graph captures the idea that Markov chains are memory-less, since it does not matter what path we took to arrive at a particular node when traversing the graph. To traverse the graph, all that matters are the relative probabilities of all outgoing edges from that node.

2.4 Markov Chains as a Matrix

Recall that all graphs can be represented as an adjacency matrix P, where $P_{i,j}$ is the weight of the directed edge between node i and node j. Therefore another way to represent a Markov chain is using a matrix. A Markov chain can be represented by an $N \times N$ matrix P, where $P_{x,y} = P(x, y)$ and $N = |\Omega|$. In other words, entry (i, j) in P gives us the probability of transitioning from state i to state j. This matrix is know as the stochastic matrix of a Markov chain. By representing Markov chains as matrices, we can use well-established linear algebra techniques in our analysis. In particular, we can easily compute the future distribution of states.

Given the distribution of states at time t, represented by a row vector π_t, where $\pi_{t[i]}$ gives us the probability of being in state i at time t, we can compute the distribution of states at time $t + k$ by:

$$\pi_{t+k} = \pi_t \times P^k$$

Here, P^k is the matrix product $P \times P \times ... \times P$, k times.