CS5314
Randomized Algorithms

Lecture 21: Markov Chains
(Definitions, Solving 2SAT)

Objectives

- Introduce Markov Chains
  - powerful model for special random processes
- Analyze a simple randomized algorithms for 2SAT and 3SAT problems

Stochastic Process

Definition: A collection of random variables $X = \{ X_t \mid t \in T \}$ is called a **stochastic process**. The index t often represents time; X_t is called the **state** of X at time t

E.g., A gambler is playing a fair coin-flip game: wins $1 if head, loses $1 if tail

Let X_0 = a gambler's initial money

X_t = a gambler's money after t flips

➔ $\{ X_t \mid t \in \{0,1,2,\ldots\} \}$ is a stochastic process

Stochastic Process (2)

Definition: If X_t assumes values from a finite set, then the process is a finite stochastic process

Definition: If T (where the index t is chosen) is countably infinite, the process is a discrete time process

Question: In the previous example about a gambler's money, is the process finite?

Is the process discrete time?

Markov Chain (Definition)

Definition: A discrete time stochastic process $X = \{X_0, X_1, X_2, \ldots\}$ is a Markov chain if

$$Pr(X_t = a \mid X_{t-1} = b, X_{t-2} = a_{t-2}, \ldots, X_0 = a_0)$$

$$= Pr(X_t = a \mid X_{t-1} = b) = P_{b,a}$$

That is, the value of X_t depends on the value of X_{t-1}, but not the history how we arrived at X_{t-1} with that value

Question: In the example about a gambler's money, is the process a Markov chain?

Markov Chain (2)

In other words, if X is a Markov chain, then

$\Pr(X_1 = a \mid X_0 = b) = P_{b,a}$

$\Pr(X_2 = a \mid X_1 = b) = P_{b,a}$

...

➔ $P_{b,a}$

$= \Pr(X_1 = a \mid X_0 = b)$

$= \Pr(X_2 = a \mid X_1 = b)$

$= \Pr(X_3 = a \mid X_2 = b) = \ldots$

Markov Chain (3)

- Next, we focus our study on Markov chain whose state space (the set of values that X_t can take) is finite

- So, without loss of generality, we label the states in the state space by 0,1,2,…,n

- The probability $P_{i,j} = Pr(X_t = j \mid X_{t-1} = i)$ is the probability that the process moves from state i to state j in one step

Transition Matrix

- The definition of Markov chain implies that we can define it using a one-step transition matrix P with

$$P_{i,j} = \Pr(X_t = j \mid X_{t-1} = i)$$

Question: For a particular i, what is $\sum_j P_{i,j}$?

Transition Matrix (2)

- The transition matrix representation of a Markov chain is very convenient for computing the distribution of future states of the process

- Let $p_i(t)$ denote the probability that the process is at state i at time t

Question: Can we compute $p_i(t)$ from the transition matrix P, assuming we know $p_0(t-1)$, $p_1(t-1)$, … ?

Transition Matrix (3)

The value of $p_i(t)$ can be expressed as:

$$p_0(t-1)\ P_{0,i} + p_1(t-1)\ P_{1,i} + \dots + p_n(t-1)\ P_{n,i}$$

In other words, let $\langle p(t) \rangle$ denote the vector

$$(p_0(t), p_1(t), \dots\ p_n(t))$$

Then, we have

$$\langle p(t) \rangle = \langle p(t-1) \rangle\ P$$

Transition Matrix (4)

- For any m, we define the m-step transition matrix $P^{(m)}$ such that

$$P^{(m)}_{i,j} = \Pr(X_{t+m} = j \mid X_t = i),$$

which is the probability that we move from state i to state j in exactly m steps

- It is easy to check that $P^{(2)} = P^2$, $P^{(3)} = P \cdot P^{(2)} = P^3$, and in general, $P^{(m)} = P^m$

➔ $\langle p(t+m) \rangle = \langle p(t) \rangle P^m$

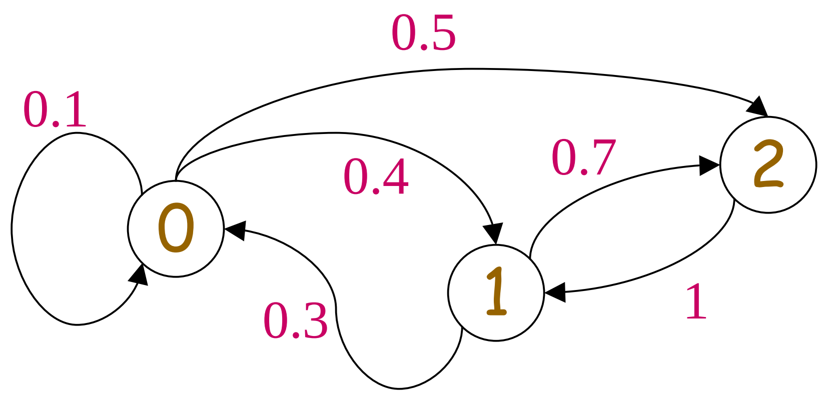
Directed Graph Representation

- Markov chain can also be expressed by a directed weighted graph (V,E), such that

  V = state space

  E = transition between states

  weight of edge (i,j) = $P_{i,j}$



Application: Solving 2SAT

- Given a Boolean formula F, with each clause consisting exactly 2 literals. Our task is to determine if F has satisfiable

  ➔ Can be solved in linear time ! (how??)

- Let n = # variables in F
- In the next slide, we describe a randomized algorithm for solving this problem, which is not efficient…
  - However, we can modify the algorithm a bit to solve the case when each clause has 3 literals instead (3SAT is NP-complete !)

1. Start with an arbitrary assignment
2. Repeat $2cn^2$ times, terminating with all clauses satisfied
   (a) Choose a clause that is currently not satisfied
   (b) Choose uniformly at random one of the literals in the clause and switch its value
3. If valid assignment found, return it
4. Else, conclude that F is not satisfiable

Application: Solving 2SAT (3)

Questions:

(1) When will the algorithm make a wrong conclusion?

Ans. … only when the formula is satisfiable, but the algorithm fails to find a satisfying assignment

(2) What is the success probability?

Ans. … let's study it using Markov chain ^_^

Application: Solving 2SAT (4)

- Firstly, suppose that the formula F is satisfiable (for the other case, we don't care much since the algorithm must give correct answer)

➔ That means, a particular assignment to the n variables in F can make F true

- Let A^* = this particular assignment
- Also, let A_t = the assignment of variables after the t^{th} iteration of Step 2
- Let X_t = the number of variables that are assigned the same value in A^* and A_t

Application: Solving 2SAT (5)

E.g., suppose that

$$F = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

and A^*: $x_1 = T, x_2 = T, x_3 = F$

- Also, suppose that after 4 iterations of Step 2 in the algorithm, we have

A_4: $x_1 = F, x_2 = T, x_3 = F$

➔ X_4 = # variables that are assigned the same value in A^* and A_4

= 2

Application: Solving 2SAT (6)

- So, when $X_t = n$, the algorithm terminates with a satisfying assignment

  … in fact, the algorithm may terminate before X_t reaches n, as it is possible that we find another satisfying assignment

  … but for our analysis, we are very pessimistic, and we consider the algorithm only stops when $X_t = n$

- Let us take a closer look of how X_t changes over time, so that we can tell how long it takes for X_t to reach n

Application: Solving 2SAT (7)

- First, when $X_t = 0$, any change in the current assignment A_t must increase the # of matching assignment with A^* by 1. So,

$$\Pr(X_{t+1} = 1 \mid X_t = 0) = 1$$

- When $X_t = j$, with $1 \leq j \leq n-1$, we will choose a clause that is false with the current assignment A_t, and change the assignment of one of its variable next …

Application: Solving 2SAT (8)

Question: What can be the value of X_{t+1}?

Ans. … it can either be j-1 or j+1

Question: Which is more likely to be X_{t+1}?

Ans. … j+1. It is because the assignment A^* will make this clause true, which must mean that either one, or both the variables in this clause is assigned differently in A_t ➔ If we change one variable randomly, at least 1/2 of the time A_{t+1} will match more with A^*

Application: Solving 2SAT (9)

- So, for j, with $1 \leq j \leq n-1$ we have

$$\Pr(X_{t+1} = j+1 \mid X_t = j) \geq 1/2$$
$$\Pr(X_{t+1} = j-1 \mid X_t = j) \leq 1/2$$

- Note: the stochastic process $X_0, X_1, X_2, \ldots$ is not necessarily a Markov chain…
  - Reason : the transition probabilities, e.g., $\Pr(X_{t+1} = j+1 \mid X_t = j)$, is not a constant

    (sometimes, it can be 1, sometimes, it can be 1/2 … in fact, this value depends on which j variables are matching with A^*, which in fact depends on the history of how we obtain A_t)

Application: Solving 2SAT (10)

- To simplify the analysis, we invent a true Markov chain $Y_0, Y_1, Y_2, \ldots$ as follows:

$$Y_0 = X_0$$

$$\Pr(Y_{t+1} = 1 \mid Y_t = 0) = 1$$

$$\Pr(Y_{t+1} = j+1 \mid Y_t = j) = 1/2$$

$$\Pr(Y_{t+1} = j-1 \mid Y_t = j) = 1/2$$

- When compared with the stochastic process $X_0, X_1, X_2, \ldots$, it takes more time for Y_t to increase to n … (why??)

Application: Solving 2SAT (11)

- Thus, the expected time to reach n from any point is larger for Markov chain Y than for the stochastic process X

➔ So, we have

E[time for X to reach n starting at X_0]

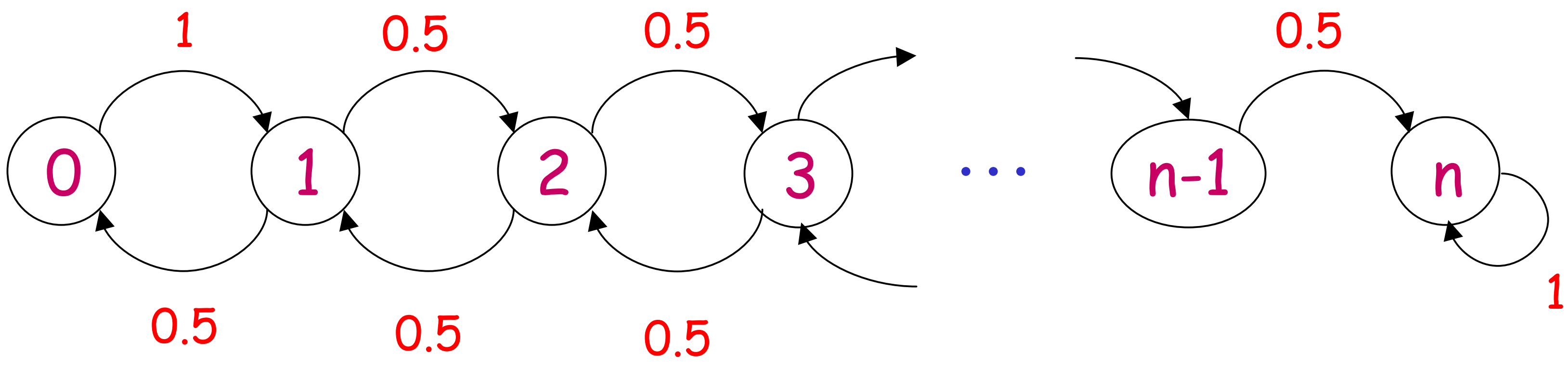
$\leq$ E[time for Y to reach n starting at Y_0]

Question: Can we upper bound the term E[time for Y to reach n starting at Y_0]?

Application: Solving 2SAT (12)

Let us take a look of how the Markov chain Y looks like in the graph representation

- Recall that vertices represents the state space, which are the values that any Y_t can take on:



Application: Solving 2SAT (13)

Let h_j = E[time to reach n starting at state j]

Clearly,

$$h_n = 0 \quad \text{and} \quad h_0 = h_1 + 1$$

Also, for other values of j, we have

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

By induction, we can show that for all j,

$$h_j = n^2 - j^2 \leq n^2$$

Application: Solving 2SAT (13)

- Combining with previous argument :

  E[time for X to reach n starting at X_0]

  $\leq$ E[time for Y to reach n starting at Y_0]

  $\leq n^2$, which gives the following lemma:

Lemma: Assume that F has a satisfying assignment. Then, if the algorithm is allowed to run until it finds a satisfying assignment, the expected number of iterations is at most n^2

Application: Solving 2SAT (13)

- Since the algorithm runs for $2cn^2$ iterations, we can show the following:

Theorem: The 2SAT algorithm answers correctly if the formula is unsatisfiable.
Otherwise, with probability $\geq 1 - 1/2^c$, it returns a satisfying assignment

How to prove?

(Hint: Break down the $2cn^2$ iterations into c groups, and apply Markov inequality)