Probability and Statistics Introduction to Markov Chain

谢润烁 Nanjing University, 2023 Fall



Basics of Markov Chain

Random Process

- A collection of random variable $\{X(t) : t \in T\}$
 - With **time** dimension
- Property
 - Finite: dom(X) is finite
 - Discrete time: t takes on countable value (WLOG, $T = \{0, 1, 2, ...\}$)
- We only study finite and discrete time random process

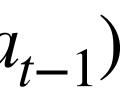
Markov Chain

•

Markov chain is a random process
$$\{X(t) : T \in \mathbb{N}\}$$
 that
• $\Pr(X_t = a_t | X_{t-1} = a_{t-1}, X_{t-2} = a_{t-2}, \dots, X_0 = a_0) = \Pr(X_t = a_t | X_{t-1} = a_t)$
• *Memoryless* or *Markovian* property

- We denote $Pr(X_t = a_t | X_{t-1} = a_t)$
 - Which we call as transition probability

$$P_{a_{t-1}}$$
) by P_{a_{t-1},a_t}



- Initially you have x, 0 < x < n
- Game rule:
 - Toss a coin each time
 - Win: add \$1
 - Lose: lose \$1

• When the game is over: you have n or 0 (you lose all your money!)

• X_t : money at time t

•
$$X_0 = x$$

•
$$1 \leq j \leq n-1$$

•
$$\Pr(X_t = j + 1 | X_{t-1} = j) = \frac{1}{2}$$

•
$$\Pr(X_t = j - 1 | X_{t-1} = j) = \frac{1}{2}$$

• j = 0 or n

• $\Pr(X_t = j | X_{t-1} = j) = 1$

- Question:
 - Probability of win/lose?
 - How many toss coins does it take?



Transition Matrix

$$P = \begin{pmatrix} P_{0,0} & P_{0,1} \\ P_{1,0} & P_{1,1} \\ \vdots & \vdots \\ P_{i,0} & P_{i,1} \\ \vdots & \vdots \end{pmatrix}$$

Why would we need it?

Transition Matrix

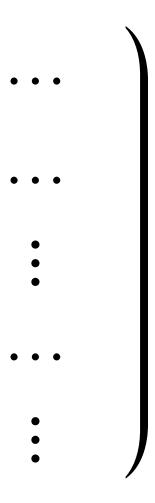
•
$$\Pr(X_t = j) = \sum_{i=0}^{n} \Pr(X_{t-1} = i) \cdot \Pr(X_t = j | X_{t-1} = i)$$

•
$$\operatorname{Let} p_t \text{ denote} \begin{bmatrix} \Pr(X_t = 0) \\ \Pr(X_t = 1) \\ \vdots \\ \Pr(X_t = n) \end{bmatrix}^T, \text{ we have } p_t(j) = \sum_{i=0}^{n} p_{t-1}(i) P_{ij}$$

•
$$p_t = p_{t-1}P$$

• $p_t = p_0 P^t$

$$P = \begin{pmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,j} \\ P_{1,0} & P_{1,1} & \dots & P_{1,j} \\ \vdots & \vdots & \ddots & \vdots \\ P_{i,0} & P_{i,1} & \dots & P_{i,j} \\ \vdots & \vdots & \ddots & \vdots \\ \end{pmatrix}$$
$$= j | X_{t-1} = i)$$



Example: A 2-SAT Algorithm SAT Problem

- *n* boolean variables: $x_1, x_2, ..., x_n \in \{T, F\}$
- Conjunctive Normal Form (CNF):

$$\Phi = (x_1 \lor \neg x_2 \lor x_3) \land (x_1 \lor \neg x_2 \lor x_3)$$

- k-CNF: each clause contains exactly variables
- *k*-SAT problem
 - Given k-CNF formula Φ
 - Determine whether Φ is satisfiable

 $(x_1 \lor x_2 \lor x_4) \land (x_3 \lor \neg x_4 \lor \neg x_5)$

- **#Possible assignment:** 2^n
- Are there fast algorithms?

2-SAT Algorithm:

- **1.** Start with an arbitrary truth assignment.
- 2. Repeat up to $2mn^2$ times, terminating if all clauses are satisfied: (a) Choose an arbitrary clause that is not satisfied.
 - value of its variable.
- **3.** If a valid truth assignment has been found, return it.
- 4. Otherwise, return that the formula is unsatisfiable.

(b) Choose uniformly at random one of the literals in the clause and switch the

Algorithm 7.1: 2-SAT algorithm.

- We can view an assignment as a vector $v \in \{0,1\}^n$
- S: target assignment which is satisfiable
- A_t : assignment at step t
- X_t : the number of variables in A_t that have the same value in S
 - i.e. $n ||A_t S||_1$
 - $X_t \in \{0, 1, ..., n\}$

• $1 \le j \le n - 1$ • $\Pr(X_t = j + 1 | X_{t-1} = j) \ge \frac{1}{2}$ • $\Pr(X_t = j - 1 | X_{t-1} = j) \le \frac{1}{2}$

- Other
 - $\Pr(X_t = 1 | X_{t-1} = 0) = 1$
 - $\Pr(X_t = n | X_{t-1} = n) = 1$

Example: A 2-SAT Algorithm A Pessimistic Version

•
$$1 \le j \le n - 1$$

• $\Pr(X_t = j + 1 | X_{t-1} = j) \ge \frac{1}{2}$
• $\Pr(X_t = j - 1 | X_{t-1} = j) \le \frac{1}{2}$

- Other
 - $\Pr(X_t = 1 | X_{t-1} = 0) = 1$
 - $\Pr(X_t = n | X_{t-1} = n) = 1$

•
$$1 \le j \le n - 1$$

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

- Other
 - $\Pr(Y_t = 1 \mid Y_{t-1} = 0) = 1$
 - $\Pr(Y_t = n \mid Y_{t-1} = n) = 1$

- It takes more time for Y to reach n than X
- Z_j : the number of steps to reach n from state j of Y
- $h_j: \mathbb{E}[Z_j]$
 - $Z_j = \frac{1}{2}(Z_{j+1} + 1) + \frac{1}{2}(Z_{j-1} + 1), 2 \le j \le n 1$
 - $Z_n = 0$
 - $Z_0 = Z_1 + 1$

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

•
$$h_n = 0, h_0 = h_1 + 1$$

• Solve the recurrence

•
$$h_j = h_{j+1} + 2j + 1$$

• $h_0 = \sum_{i=0}^{n-1} (2i+1) = n^2$

- It takes more time for Y to reach n than X
- Z_i : the number of steps to reach *n* from state *j* of *Y*
- $h_i = \mathbb{E}[Z_i]$

•
$$h_0 = \sum_{i=0}^{n-1} (2i+1) = n^2$$

• The expected steps to find a satisfiable solution is bounded by n^2

Long-term Behavior

Classification of States

- Communicate
 - Irreducible
- Recurrence
- Aperiodic & Periodic
- Ergodic

Communication

- State *j* is **accessible** from state $i: i \rightarrow j$
 - $\exists n > 0. P_{i,i}^n > 0$
- State *i* and state *j* communicate: $i \leftrightarrow j$
- - Strongly Connected Graph

A Markov chain is irreducible if all states belong to one communicating class

Recurrence

- Let $r_{i,j}^t$ denote the probability that
 - starting at state *i*, the first transition to state *j* occurs at time *t*

• i.e.
$$r_{i,j}^t = \Pr(X_t = j \text{ and, for } 1 \le s \le t - 1, X_s \ne j | X_0 = i)$$

A state is **recurrent** if $\sum_{t \ge 1} r_{i,i}^t = 1$, and it is **transient** if $\sum_{t \ge 1} r_{i,i}^t < 1$

A Markov chain is recurrent if every state in the chain is recurrent

- Initially you have x, 0 < x < n
- Game rule:
 - Toss a coin each time
 - Win: add \$1
 - Lose: lose \$1

• When the game is over: you have n or 0 (you lose all your money!)

• X_t : money at time t

•
$$X_0 = x$$

•
$$1 \leq j \leq n-1$$

•
$$\Pr(X_t = j + 1 | X_{t-1} = j) = \frac{1}{2}$$

•
$$\Pr(X_t = j - 1 | X_{t-1} = j) = \frac{1}{2}$$

• j = 0 or n

• $\Pr(X_t = j | X_{t-1} = j) = 1$

 What is the probability of winning and losing?



- i = 1, 2, ..., n 1 are all transient states
 - $\lim p_t(i) = 0$ $t \rightarrow \infty$
 - $\lim_{t \to \infty} p_t(n) = q, \lim_{t \to \infty} p_t(0) = 1 q$

$$\mathbb{E}(X_t) = \sum_{i=0}^n ip_t(i) = x$$

X • $\lim \mathbb{E}(X_t) = nq = x \Rightarrow q =$ $t \rightarrow \infty$ n

Periodicity

- Periodic: a state *j* is periodic if
 - $\exists \Delta > 1.s \nmid \Delta \rightarrow \Pr(X_t = j \mid X_0 = j) = 0$
- A Markov chain is **periodic** if any state in the chain is **periodic**
- A state or chain that is not periodic is **aperiodic**

Stationary Distribution

- Stationary distribution: a probability distribution $\vec{\pi}$ such that
 - $\vec{\pi} = \vec{\pi}P$
- Question
 - Does stationary distribution exist?
 - If exists, is it unique?
 - What can we do using stationary distribution?

Fundamental Theorem of Markov Chain

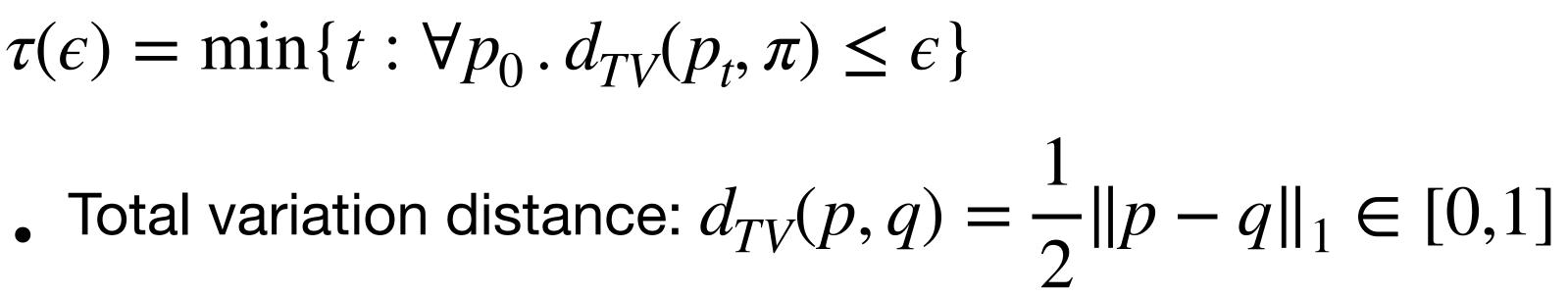
- The chain has a unique stationary distribution $\pi = (\pi_0, \pi_1, \dots, \pi_n)$.
- For all j and i, the limit $\lim_{t \to \infty} P_{ij}^t$ exists and it is independent of j.

•
$$\pi_i = \lim_{t \to \infty} P_{ji}^t = \frac{1}{h_{ii}}$$

Any finite, irreducible, and aperiodic Markov chain has the following properties:

Mixing Time Important but difficult problems

- Sometimes asymptotic is not enough when designing algorithms
- ϵ -mixing time:
 - $\tau(\epsilon) = \min\{t : \forall p_0 . d_{TV}(p_t, \pi) \le \epsilon\}$
- Methods
 - Probabilistic: coupling
 - Algebraic: spectral analysis



Random Walks on Undirected Graphs



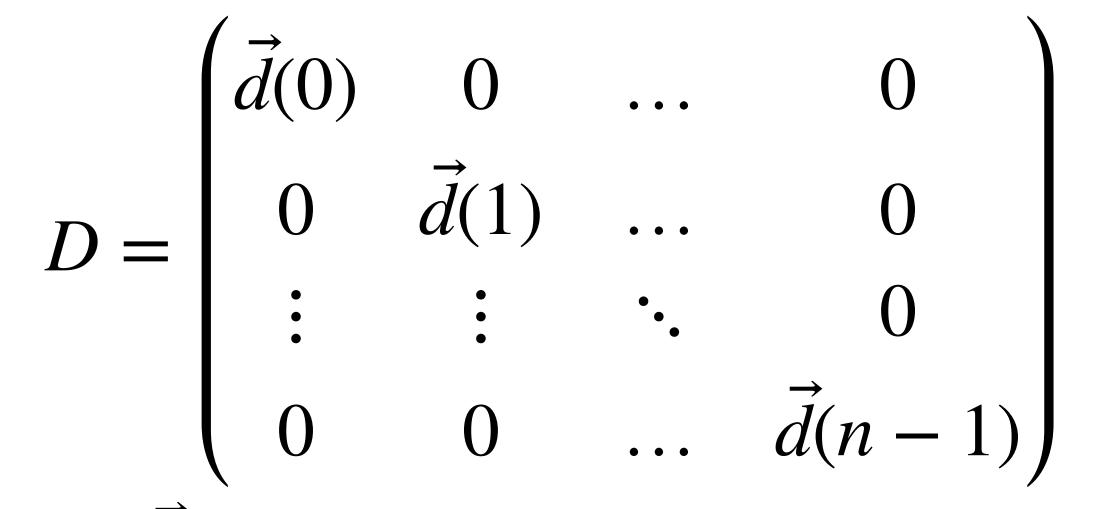
Overview

- We will see some interesting results without going into details
- And we will see an interesting algorithm

Interesting Results

- Undirected graph G = (V, E) with $V = \{0, 1, 2, ..., n 1\}$
- Transition matrix $P = D^{-1}A$

• A is adjacency matrix, d is the degree vector



Interesting Results

- Connected ⇔ Irreducible
- Aperiodic ⇔ Non-bipartite
- It is always positive recurrent

The fundamental theorem of Markov chain becomes

• For any finite, connected, non-bipartite graph

 $p_t \rightarrow$

$$\vec{\pi} = \frac{\vec{d}}{2m}$$

Interesting Results

- Bounding cover time
 - The cover time of a connected graph is at most 2m(n-1)
- *Bounding mixing time

•
$$\tau(\epsilon) \leq \frac{1}{\lambda} \log(\frac{n}{\epsilon})$$
, where **spectra**

• $\alpha_1 \ge \alpha_2 \ge \ldots \ge \alpha_n$ is the eigenvalues of the transition matrix P

al gap $\lambda = \min\{1 - \alpha_2, 1 - |\alpha_n|\}$

Example: an s-t Connectivity Algorithm

- Determine if there is a path connecting s and t
- What we have learned: BFS/DFS
 - requires $\Omega(n)$ space
 - Not possible when *n* is too large

• Given an undirected graph G = (V, E) and two different vertices $s, t \in V$

Example: an s-t Connectivity Algorithm

s–*t* Connectivity Algorithm:

- **1.** Start a random walk from *s*.
- 2. If the walk reaches t within $2n^3$ steps, return that there is a path. Otherwise, return that there is no path.

Algorithm 7.4: *s*–*t* Connectivity algorithm.

- Space complexity: $O(\log n)$
- Pr(return not connected | s-t is connected) $\leq \frac{1}{2}$
 - By the upper bound on cover time
- Use **union bound** to bound the error rate

Markov Chain Monte Carlo

Sampling of Complex Objects

- Consider uniform distribution of the following objects
- These are easy to sample
 - *n*
 - { $(x, y, z) | x^2 + y^2 + z^2 \le 100$ }
- How about these?
 - All independent sets in a graph G
 - All k-colorings of a graph G



Markov Chain Monte Carlo (MCMC)

- We want to sample a certain distinct distribution p
- We construct a Markov chain $\{X_t\}$ with a unique stationary distribution p
- We bound the $\tau(\epsilon)$ by finding a T such that $T \ge \tau(\epsilon)$
- Then we sample $X_T, X_{2T}, X_{3T}, \ldots$
- How to construct?

Markov Chain Monte Carlo (MCMC)

Lemma 11.7: For a finite state space Ω and neighborhood structure $\{N(X) \mid x \in \Omega\}$, let $N = \max_{x \in \Omega} |N(x)|$. Let M be any number such that $M \ge N$. Consider a Markov chain where

$$P_{x,y} = \begin{cases} 1/M & \text{if} \\ 0 & \text{if} \\ 1 - N(x)/M & \text{if} \end{cases}$$

If this chain is irreducible and aperiodic, then the stationary distribution is the uniform distribution.

```
f x \neq y \text{ and } y \in N(x),
fx \neq y \text{ and } y \notin N(x),
x = y.
```

Markov Chain Monte Carlo (MCMC)

Consider now the following simple Markov chain, whose states are independent sets in a graph G = (V, E).

- **1.** X_0 is an arbitrary independent set in G.
- **2.** To compute X_{i+1} :
 - (a) choose a vertex v uniformly at random from V;
 - (b) if $v \in X_i$ then $X_{i+1} = X_i \setminus \{v\}$;
 - (c) if $v \notin X_i$ and if adding v to X_i still gives an independent set, then $X_{i+1} =$ $X_i \cup \{v\};$
 - (d) otherwise, $X_{i+1} = X_i$.