Markov Chains and Random Walks

RA-MIRI

QT Curs 2020-2021

Stochastic Process

- A stochastic process is a sequence of random variables $\{X_t\}_{t=0}^{n}$.
- Usually the subindex t refers to time steps and if $t \in \mathbb{N}$, the stochastic process is said to be discrete.
- The random variable X_t is called the state at time t.
- If $n < \infty$ the process is said to be finite, otherwise it is said infinite.
- A stochastic process is used as a model to study the probability of events associated to a random phenomena.

An example: Gambler's Ruin

Model used to evaluate insurance risks.

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of 100€.
- You keep playing until you loose all your money or you arrive to have 1000€.

An example: Gambler's Ruin

Model used to evaluate insurance risks.

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of 100€.
- You keep playing until you loose all your money or you arrive to have 1000€.
- One goal is finding the probability of winning i.e. getting the 1000€.

An example: Gambler's Ruin

Model used to evaluate insurance risks.

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of 100€.
- You keep playing until you loose all your money or you arrive to have 1000€.
- One goal is finding the probability of winning i.e. getting the 1000€.

Notice in this process, once we get 0€ or 1000€, the process stops.

Markov Chain

One simple model of stochastic process is the Markov Chain:

- Markov Chains are defined on a finite set of states (S), where at time t, X_t could be any state in S, together with by the matrix of transition probability for going from each state in S to any other state in S, including the case that the state X_t remains the same at $t+1$.
- In a Markov Chain, at any given time t, the state X_t is determined only by X_{t-1}.
  memoryless: does not remember the history of past events,

Other memoryless stochastic processes are said to be Markovian.

An example: Gambler's Ruin

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of 100€.
- You keep playing until you loose all your money or you arrive to have 1000€.

An example: Gambler's Ruin

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of 100€.
- You keep playing until you loose all your money or you arrive to have 1000€.
- We have a state for each possible amount of money you can accumulate $S = \{0, 1, \ldots, 1000\}$.

An example: Gambler's Ruin

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of 100€.
- You keep playing until you loose all your money or you arrive to have 1000€.
- We have a state for each possible amount of money you can accumulate $S = \{0, 1, \dots, 1000\}$.
- The probability of losing/winning is independent on the state and the time, so this process is a Markov chain.

An example: Gambler's Ruin

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of 100€.
- You keep playing until you loose all your money or you arrive to have 1000€.
- We have a state for each possible amount of money you can accumulate $S = \{0, 1, \dots, 1000\}$.
- The probability of losing/winning is independent on the state and the time, so this process is a Markov chain.
- Observe that the number of states is finite.

Markov-Chains: An important tool for CS

- One of the simplest forms of stochastic dynamics.
- Allows to model stochastic temporal dependencies
- Applications in many areas
  - Surfing the web
  - Design of randomizes algorithms
  - Random walks
  - Machine Learning (Markov Decision Processes)
  - Computer Vision (Markov Random Fields)
  - etc. etc.

Formal definition of Markov Chains

A finite, time-discrete Markov Chain, with finite state $S = \{1, 2, \dots, k\}$ is a stochastic process $\{X_t\}$ s.t. for all $i, j \in S$, and for all $t \geq 0$,

$$\mathbf{Pr}\left[X_{t+1} = j \mid X_0 = i_0, X_1 = i_1, \dots, X_t = i\right] = \mathbf{Pr}\left[X_{t+1} = j \mid X_t = i\right].$$

Formal definition of Markov Chains

A finite, time-discrete Markov Chain, with finite state $S = \{1, 2, \dots, k\}$ is a stochastic process $\{X_t\}$ s.t. for all $i, j \in S$, and for all $t \geq 0$,

$$\mathbf{Pr}\left[X_{t+1} = j \mid X_0 = i_0, X_1 = i_1, \dots, X_t = i\right] = \mathbf{Pr}\left[X_{t+1} = j \mid X_t = i\right].$$

We can abstract the time and consider only the probability of moving from state i to state j, as $\mathbf{Pr}\left[X_{t+1} = j \mid X_t = i\right]$

MC: Transition probability matrix

For $v, u \in S$, let $p_{u,v}$ be the probability of going from $u \rightsquigarrow v$ in q steps i.e. $p_{u,v} = \mathbf{Pr}\left[X_{s+1} = v \,|\, X_s = u\right]$.

MC: Transition probability matrix

For $v, u \in S$, let $p_{u,v}$ be the probability of going from $u \rightsquigarrow v$ in q steps i.e. $p_{u,v} = \mathbf{Pr}\left[X_{s+1} = v \,|\, X_s = u\right]$.

$P = (p_{u,v})_{u,v \in S}$ is a matrix describing the transition probabilities of the MC

P is called the transition matrix

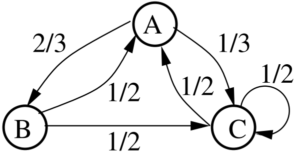
MC: Transition probability matrix

For $v, u \in S$, let $p_{u,v}$ be the probability of going from $u \rightsquigarrow v$ in q steps i.e. $p_{u,v} = \mathbf{Pr}[X_{s+1} = v \mid X_s = u]$.

$P = (p_{u,v})_{u,v \in S}$ is a matrix describing the transition probabilities of the MC

P is called the transition matrix P also defines digraph, possibly

with loops.

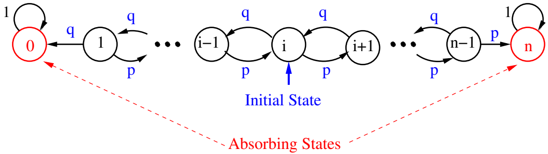


Gambler's Ruin: MC digraph

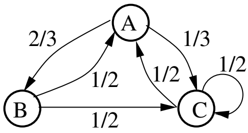
- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of i €and keep playing until you loose all your money or you arrive to have n€.
- We have a state for each possible amount of money you can accumulate $S = \{0, 1, \dots, n\}$.

Gambler's Ruin: MC digraph

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of i € and keep playing until you loose all your money or you arrive to have n€.
- We have a state for each possible amount of money you can accumulate $S = \{0, 1, \ldots, n\}$.



Transition matrix: Example



$$
\begin{array}{c} \\ A \\ B \\ C \end{array}
\begin{array}{c}
\begin{array}{ccc} A & B & C \end{array} \\
\begin{pmatrix} 0 & 2/3 & 1/3 \\ 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 \end{pmatrix}
\end{array} = P
$$

Notice the entry (u, v) in P denotes the probability of going from $u \to v$ in one step.

Notice, in a MC the transition matrix is stochastic, so sum of transitions out of any state must be 1.

Longer transition probabilities

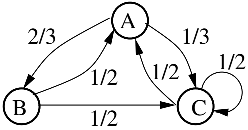
For $v, u \in S$, let $p^t_{u,v}$ be the probability of going from $u \rightsquigarrow v$ in exactly t steps i.e. $p^t_{u,v} = \mathbf{Pr}\left[X_{s+t} = v \,|\, X_s = u\right]$.

Formally for $s \geq 0$ and $t > 1$, $p^t_{u,v} = \mathbf{Pr}\left[X_{s+t} = v \,|\, X_s = u\right]$.

A times, we may use i $P^t_{u,v}$ to indicate entry (u, v)in the matrix P, i.e $p^t_{u,v} = P^t_{u,v} = \mathbf{Pr}\left[X_{s+t} = v \,|\, X_s = u\right]$.

How can we relate P^t with P?

The powers of the transition matrix



$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 0 & 2/3 & 1/3 \\ 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 \end{pmatrix} \end{array} = P$$

In ex. $\mathbf{Pr}[X_1 = C | X_0 = A] = P^1_{A,C} = 1/3$.
$\mathbf{Pr}[X_2 = C | X_0 = A] = P^1_{AB} P^1_{BC} + P^1_{AC} P^1_{CC} = 1/3 + 1/6 = P^2_{A,C}$

In general, assume a MC with k states and transition matrix P, let $u, v \in S$:

- What is the $\mathbf{Pr}[X_1 = u | X_0 = v]$, i.e. $= P_{v,u}$?
- What is the $\mathbf{Pr}[X_2 = u | X_0 = v] = P^2_{v,u}$?

The powers of the transition matrix

Use Law Total Probability+ Markov property:

$$\mathbf{Pr}\left[X_2 = u | X_0 = v\right] = \sum_{w=1}^{m} \mathbf{Pr}\left[X_1 = w | X_0 = v\right] \mathbf{Pr}\left[X_2 = u | X_1 = w\right]$$
$$= \sum_{w=1}^{m} P_{v,w} P_{w,u} = P^2_{v,u}.$$

In general $\mathbf{Pr}\left[X_t = w | X_0 = v\right] = P^t_{v,u}$ and $\mathbf{Pr}\left[X_{k+t} = w | X_k = v\right] = P^t_{v,u}$.

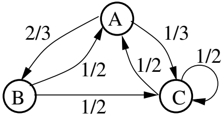
The argument can be generalized to

Given the transition matrix P of a MC, then for any $t > 1$,

$$P^t = P \cdot P^{t-1}.$$

Notice the entry (u, v) in P^t denotes the probability of going from $u \to v$ in t steps.

To fix the initial state, we consider a random variable X_0, assigning to S an initial distribution π_0, which is a row vector indicating at $t = 0$ the probability of being in the corresponding state.
For example, in the MC:



we may consider,

$$\begin{matrix} A & B & C \\ (0 & 0.3 & 0.6) \end{matrix} = \pi_0$$

Distributions at time t

Starting with an initial distribution π_0, we can compute the state distribution π_t (on S) at time t,

For a state v,

$$\begin{aligned}\pi_t[v] =& \mathbf{Pr}\left[X_t = v\right] \\ =& \sum_{u \in S} \mathbf{Pr}\left[X_0 = u\right] \mathbf{Pr}\left[X_t = v | X_0 = u\right] \\ =& \sum_{u \in S} \pi_0[u] P^t_{v,u}.\end{aligned}$$

i.e. $\pi_t[y]$ is the probability at step t the system is in state y.

Therefore, $\pi_t = \pi_0 P^t$ and $\pi_{s+t} = \pi_s P^t$.

Gambler's Ruin: Exercise

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of i € and keep playing until you loose all your money or you arrive to have n€.
- We have a state for each possible amount of money you can accumulate $S = \{0, 1, \ldots, n\}$.

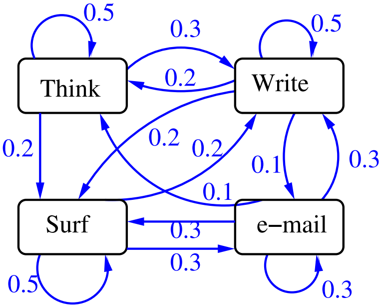
Gambler's Ruin: Exercise

- You place bets of 1€. With probability p, you gain 1€, and with probability $q = 1 - p$ you loose your 1€ bet.
- You start with an initial amount of i € and keep playing until you loose all your money or you arrive to have n€.
- We have a state for each possible amount of money you can accumulate $S = \{0, 1, \dots, n\}$.
- Which is the initial distribution π_0?
- And, the state distribution at time $t = 3$?

Example MC: Writing a research paper

Recall that Markov Chains are given either by a weighted digraph, where the edge weights are the transition probabilities,
or by the $|S| \times |S|$ transition probability matrix P,

Example: Writing a paper $S = \{r, w, e, s\}$



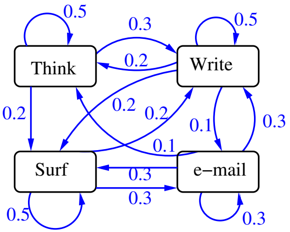
$$
\begin{array}{c} \\ r \\ w \\ e \\ s \end{array}
\begin{array}{c}
\begin{array}{cccc} r & w & e & s \end{array} \\
\begin{pmatrix}
0.5 & 0.3 & 0 & 0.2 \\
0.2 & 0.5 & 0.1 & 0.2 \\
0.1 & 0.3 & 0.3 & 0.3 \\
0 & 0.2 & 0.3 & 0.5
\end{pmatrix}
\end{array}
$$

More on the Markovian property

Notice the memoryless property does not mean that X_{t+1} is independent from $X_0, X_1, \dots, X_{t-1}$.

(For instance notice that intuitively we have:
$\mathbf{Pr}[\text{Thinking at } t+1] < \mathbf{Pr}[\text{Thinking at } t \mid \text{Thinking at } t-1]$).

But, the dependencies of X_t on $X_0, \dots, X_{t-1}$, are all captured by X_{t-1}.



Example of writing a paper

$\mathbf{Pr}[X_2 = s | X_0 = r]$ is the probability that, at $t = 2$, we are in state s, starting in state r.

$$\begin{pmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.2 & 0.5 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.3 & 0.3 \\ 0 & 0.2 & 0.3 & 0.5 \end{pmatrix} \begin{pmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.2 & 0.5 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.3 & 0.3 \\ 0 & 0.2 & 0.3 & 0.5 \end{pmatrix} = \begin{pmatrix} 0.31 & 0.34 & 0.09 & 0.26 \\ 0.21 & 0.38 & 0.14 & 0.27 \\ 0.14 & 0.33 & 0.21 & 0.32 \\ 0.07 & 0.29 & 0.26 & 0.38 \end{pmatrix} \begin{matrix} r \\ w \\ e \\ s \end{matrix}$$

$\mathbf{Pr}[X_1 = s | X_0 = r] = 0.07.$

Distribution on states

Recall π_t is the prob. distribution at time t over S.

For our example of writing a paper, if $t = 0$ (after waking up):

$$\pi_0 = \begin{matrix} r & w & e & s \\ (0.2 & 0 & 0.3 & 0.5) \end{matrix}$$

$$(0.2 \quad 0 \quad 0.3 \quad 0.5) \begin{pmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.2 & 0.5 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.3 & 0.3 \\ 0 & 0.2 & 0.3 & 0.5 \end{pmatrix} = (0.13 \quad 0.25 \quad 0.24 \quad 0.38) = \pi_1$$

Therefore, we have $\pi_t = \pi_0 \times P^t$ and $\pi_{k+t} = \pi_k \times P^t$

Notice $\pi_t = (\pi_t[r], \pi_t[w], \pi_t[e], \pi_t[s])$

An Example of MC analysis: The 2-SAT problem

Section 7.1 of [MU].

Given a Boolean formula ϕ, on

- a set X of n Boolean variables,
- defined by m clauses $C_1, \dots C_m$, where each clause is the disjunction of exactly 2 literals, (x_i or $\bar{x}_i$), on different variables.
- $\phi =$ conjunction of the m clauses.

The 2-SAT problem is to find an assignment $A^* : X \to \{0, 1\}$, which satisfies ϕ,
i.e, to find an A^* s.t. $A^*(\phi) = 1$.

Notice that if $|X| = n$, then $m \leq \binom{2n}{2} = O(n^2)$.

In general k-SAT$\in$ NP-complete, for $k \geq 3$. But 2-SAT$\in$ P.

A randomized algorithm for 2-SAT

Given a n variable 2-SAT formula ϕ, $\{C_j\}_{j=1}^m$
for all $1 \leq i \leq n$ **do**
  $A(x_i) = 1$
end for
$t = 0$
while $t \leq 2cn^2$ and some clause is unsatisfied **do**
  pick and unsatisfied clause C_j
  choose u.a.r. one of the 2 variables in C_j and flip its value
  if ϕ is satisfied **then**
    return A
  end if
end while
return ϕ is unsatisfiable

An example: unsat formula

If $\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$ does not has a $A^* \models \phi$.

t	x_1	x_2	sel clause
1	1	1	

An example: unsat formula

If $\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$ does not has a $A^* \models \phi$.

t	x_1	x_2	sel clause
1	1	1	2

An example: unsat formula

If $\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$ does not has a $A^* \models \phi$.

t	x_1	x_2	sel clause
1	1	1	2
2	1	0	

An example: unsat formula

If $\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$ does not has a $A^* \models \phi$.

t	x_1	x_2	sel clause
1	1	1	2
2	1	0	3

An example: unsat formula

If $\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$ does not has a $A^* \models \phi$.

t	x_1	x_2	sel clause
1	1	1	2
2	1	0	3
3	0	0	

An example: unsat formula

If $\phi = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2)$ does not has a $A^* \models \phi$.

t	x_1	x_2	sel clause
1	1	1	2
2	1	0	3
3	0	0	1
⋮	⋮	⋮	⋮

ϕ is unsat eventually the algorithm will stop after reaching the maximum number of steps.

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	2

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	2
2	0	1	1	1	

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	2
2	0	1	1	1	1

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	2
2	0	1	1	1	1
3	0	0	1	1	

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	2
2	0	1	1	1	1
3	0	0	1	1	4

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	2
2	0	1	1	1	1
3	0	0	1	1	4
4	0	0	1	0	

An example: sat formula

If $\phi = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

t	x_1	x_2	x_3	x_4	sel clause
1	1	1	1	1	2
2	0	1	1	1	1
3	0	0	1	1	4
4	0	0	1	0	–

$(0, 0, 1, 0)$ satisfies ϕ

Analysis for 2-SAT algorithm

Given $\phi, |X| = n, \{C_j\}_{i=1}^m$

assume that there is A^* such that $\phi(A^*) = 1$

- Let A_i be the assignment at the i-th iteration.
- Let $X_i = |\{x_j \in X \mid A_i(x_j) = A^*(x_j)\}|$.
- Notice $0 \leq X_i \leq n$. Moreover, when $X_i = n$, we found A^*.
- Analysis: Starting from $X_i < n$, how long to get $X_i = n$?
- Note that $\mathbf{Pr}[X_{i+1} = 1 \mid X_i = 0] = 1$.

Analysis for 2-SAT algorithm

- As A^* satisfies ϕ and A_i no, there is a clause C_j that A^* satisfies but A_i not.
- So A^* and A_i disagree in the value of at least one variable.
- It is also possible to flip the value of the variable în C_j in which A and A^* agree.
- Therefore,

  For $1 \leq k \leq n-1$, $\mathbf{Pr}\left[X_{i+1} = k+1 \,|\, X_i = k\right] \geq 1/2$ and $\mathbf{Pr}\left[X_{i+1} = k-1 \,|\, X_i = k\right] \leq 1/2$.

Analysis for 2-SAT

The process $X_0, X_1, \ldots$ is not necessarily a MC,

- The probability that $X_{i+1} > X_i$ depends on whether A_i and A^* disagree in 1 or 2 variables in the selected unsatisfied clause C.
- If A^* makes true both literals in C,
  $\mathbf{Pr}[X_{i+1} = k+1 \mid X_i = k] = 1$, otherwise
  $\mathbf{Pr}[X_{i+1} = k+1 \mid X_i = k] = 1/2$
- This difference might depend on the clauses and variables selected in the past, so the transition probabilities are not memoryless.
- X_t is not a Markov chain.

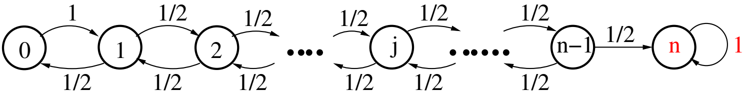
Analysis for 2-SAT

The process $X_0, X_1, \dots$ is not necessarily a MC,

- The probability that $X_{i+1} > X_i$ depends on whether A_i and A^* disagree in 1 or 2 variables in the selected unsatisfied clause C.
- If A^* makes true both literals in C,
  $\mathbf{Pr}[X_{i+1} = k+1 \mid X_i = k] = 1$, otherwise
  $\mathbf{Pr}[X_{i+1} = k+1 \mid X_i = k] = 1/2$
- This difference might depend on the clauses and variables selected in the past, so the transition probabilities are not memoryless.
- X_t is not a Markov chain. Can we bound the process by a MC?.

Define a MC $Y_0, Y_1, Y_2, \ldots$ which is a pessimistic version of process $X_0, X_1, \ldots$, in the sense that Y_i measures exactly the same quantity than X_i but the probability of change (up or down) will be exactly $1/2$.

- $Y_0 = X_0$ and $\mathbf{Pr}\left[Y_{i+1} = 1 \mid Y_i = 0\right] = 1$;
- For $1 \leq k \leq n-1$, $\mathbf{Pr}\left[Y_{i+1} = k+1 \mid Y_i = k\right] = 1/2$;
- $\mathbf{Pr}\left[Y_{i+1} = k-1 \mid Y_i = k\right] = 1/2$.



MC for 2–SAT

The time to reach n from $j \geq 0$ in $\{Y_i\}_{i=0}^{n}$ is $\geq$ that in $\{X_i\}_{i=0}^{n}$.

Upper Bound on the time to arrive state n

Lemma

If a 2-CNF ϕ on n variables has a satisfying assignment A^*, the 2-SAT algorithm finds one in expected time $\leq n^2$.

Proof

- Let h_j be the expected time, for process Y, to go from state j to state n.
- It suffices to prove that, when Y starts in state j the time to arrives to n is $\leq 2cn^2$.
- We devise a recurrence to bound h

Upper Bound on the time to arrive state n

- $h_n = 0$ and $h_1 = h_0 + 1$;
- We want a general recurrence on h_j, for $1 \leq j < n$
- Define a rv Z_j counting the steps to go from state $j \to n$ in Y.
- With probability $1/2$, $Z_j = Z_{j-1} + 1$ and, with probability $1/2$, $Z_j = Z_{j+1} + 1$.
- So $h_j = \mathbf{E}[Z_j]$.

$$\mathbf{E}[Z_j] = \mathbf{E}\left[\frac{Z_{j-1}+1}{2} + \frac{Z_{j+1}+1}{2}\right] = \frac{\mathbf{E}[Z_{j-1}]+1}{2} + \frac{\mathbf{E}[Z_{j+1}]+1}{2}.$$

So, $h_j = \frac{h_{j-1}}{2} + \frac{h_{j+1}}{2} + 1$.

Upper Bound on the time to arrive state n

From the previous bound we get $h_j = \frac{h_{j-1}}{2} + \frac{h_{j+1}}{2} + 1$.

The recurrence has the $n + 1$ equations,

$$\begin{aligned} h_n &= 0 \\ h_0 &= h_1 + 1 \\ h_j &= \frac{h_{j-1}}{2} + \frac{h_{j+1}}{2} + 1 \qquad 0 \leq j \leq n-1 \end{aligned}$$

Let us prove, by induction that

$$h_j = h_{j+1} + 2j + 1.$$

For $0 \leq j \leq n-1$, $h_j = h_{j+1} + 2j + 1$.

Proof

Base case: If $j = 0$, $2j + 1 = 1$, and we were given $h_0 = h_1 + 1$.

Upper Bound on the time to arrive state n

For $0 \leq j \leq n-1$, $h_j = h_{j+1} + 2j + 1$.

Proof

IH: for$j = k-1$, $h_{k-1} = h_k + 2(k-1) + 1$.
Now consider $j = k$. By the "middle case" of our system of equations,

$$
\begin{aligned}
h_k &= \frac{h_{k-1} + h_{k+1}}{2} + 1 \\
&= \frac{h_k + 2(k-1) + 1}{2} + \frac{h_{k+1}}{2} + 1 \qquad \text{by IH} \\
&= \frac{h_k}{2} + \frac{h_{k+1}}{2} + \frac{2k+1}{2}
\end{aligned}
$$

Subtracting $\frac{h_k}{2}$ from each side, we get the result.

Upper Bound on the time to arrive state n

As

$$h_j = h_{j+1} + 2j + 1.$$

$$h_0 = h_1 + 1 = h_2 + 3 + 1 = h_3 + 5 + 3 + 1 \cdots$$
$$= \underbrace{h_n}_{=0} + \sum_{i=0}^{n-1}(2i+1) = n^2.$$

Error probability for 2-SAT algorithm

Theorem

The 2-SAT algorithm gives the correct answer NO if ϕ is not satisfiable. Otherwise, with probability $\geq 1 - \frac{1}{2^c}$ the algorithm returns a satisfying assignment.

Proof

- Let ϕ be satisfiable (otherwise the theorem holds).
- Break the $2cn^2$ iterations into c blocks of $2n^2$ iterations.
- For each block i, define a r.v. Z = number of iterations from the start of the i-block until a solution is found.
- Using Markov's inequality:

$$\mathbf{Pr}\left[Z > 2n^2\right] \leq \frac{n^2}{2n^2} = \frac{1}{2}.$$

- Therefore, the probability that the algorithm fails to find a satisfying assignment after c segments (no block includes a solution) is at most $\frac{1}{2^c}$.