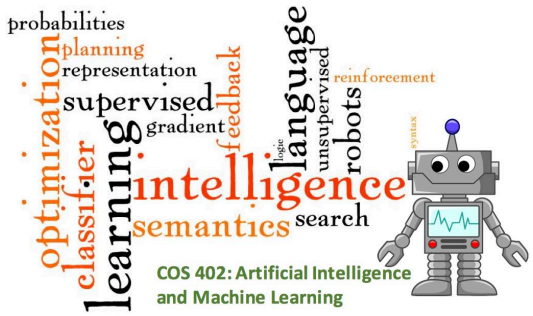


COS 402 – Machine Learning and Artificial Intelligence
Fall 2016

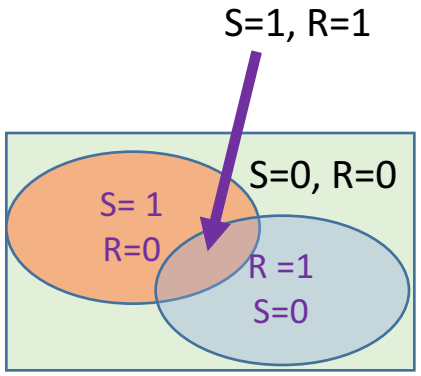
Lecture 14: Graphical Models (Part 2)

Sanjeev Arora Elad Hazan

PRINCETON UNIVERSITY

(Borrows from slides of Percy Liang, Stanford U. and Barnabas and Singh, CMU)

Review: Probabilities (example)



Random variables: Sunshine $S \in \{0, 1\}$; Rainy $R \in \{0, 1\}$.

Joint Distribution

$\mathbb{P}(S, R) =$

s	r	$\mathbb{P}(S = s, R = r)$
0	0	0.20
0	1	0.08
1	0	0.70
1	1	0.02

Marginal Distribution

$\mathbb{P}(S) =$

s	$\mathbb{P}(S = s)$
0	0.28
1	0.72

Conditional Distribution

$\mathbb{P}(S \mid R = 1) =$

s	$\mathbb{P}(S = s \mid R = 1)$
0	0.8
1	0.2

Review (contd)

Random variables:

$X = (X_1, \ldots, X_n)$ partitioned into (A, B)

Joint distribution:

$\mathbb{P}(X) = \mathbb{P}(X_1, \ldots, X_n)$

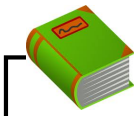
Marginal distribution:

$\mathbb{P}(A) = \sum_b \mathbb{P}(A, B = b)$

Conditional distribution:

$\mathbb{P}(A \mid B = b) = \frac{\mathbb{P}(A, B=b)}{\mathbb{P}(B=b)}$

Bayesian Net: Formal Definition



Definition: Bayesian network

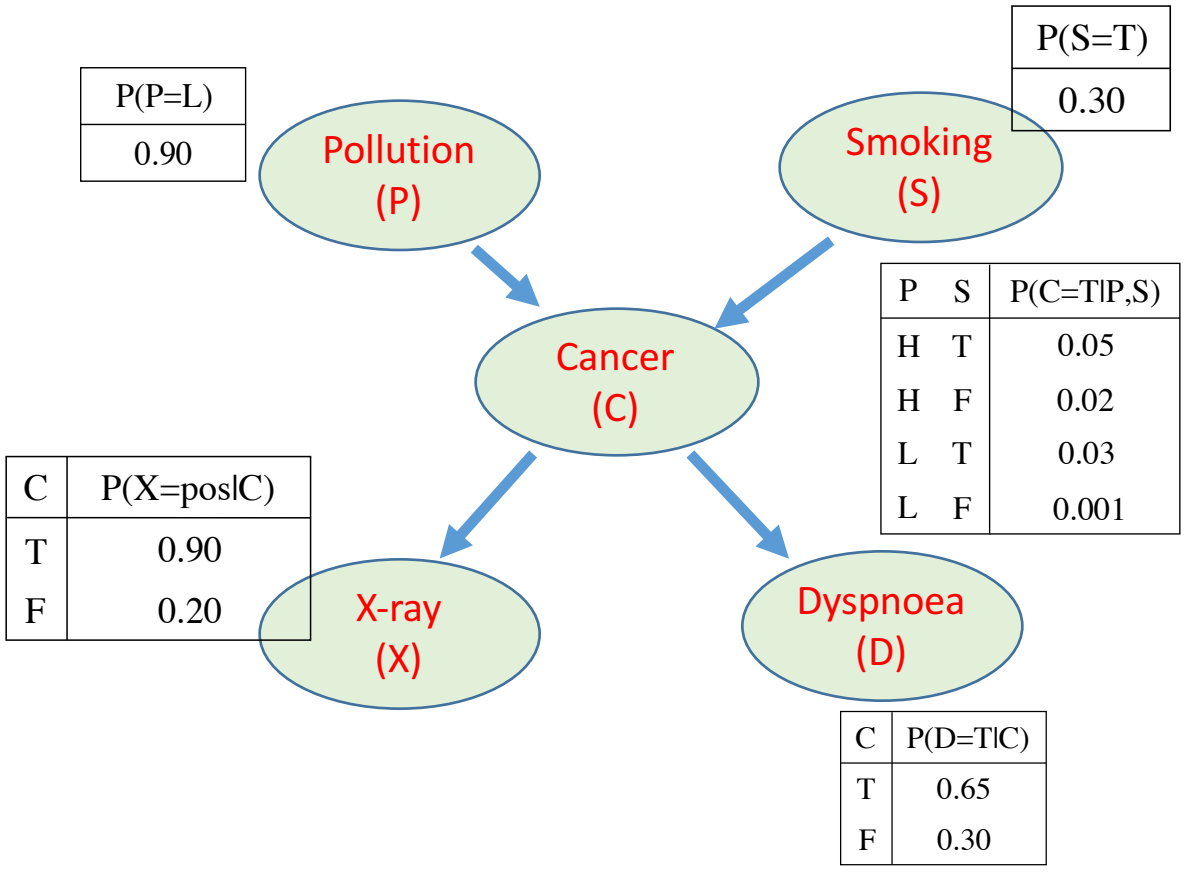
Let $X = (X_1, \ldots, X_n)$ be random variables.

A **Bayesian network** is a directed acyclic graph (DAG) that specifies a joint distribution over X as a product of local conditional distributions, one for each node:

$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} p(x_i \mid x_{\text{Parents}(i)})$$

(Will assume variables are boolean, for simplicity)

Example



$$\Pr[X, D, C, P, S]$$

$$= \Pr[X|C]\ \Pr[D|C]\ \ \Pr[C|P, S]\ \Pr[S]\ \Pr[C]$$

Note: Distribution on
5 boolean variables; specified
using only 10 numbers

(instead of the trivial
$2^5 - 1 = 31$ numbers.)

Key idea: locally normalized

All factors (local conditional distributions) satisfy:

$$\sum_{x_i} p(x_i \mid x_{\text{Parents}(i)}) = 1 \text{ for each } x_{\text{Parents}(i)}$$

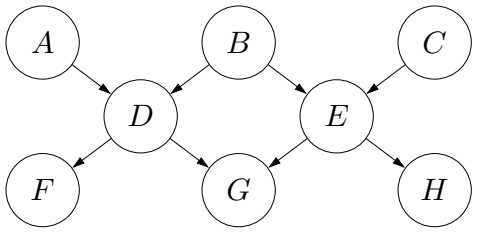
Implications:

- Consistency of sub-Bayesian networks
- Consistency of conditional distributions

I hinted but did not prove formally…

Bayes nets define proper distributions, in the sense that
all marginal distributions are well-defined (meaning probabilities sum to 1)
and behave as intuition suggests.

e.g,

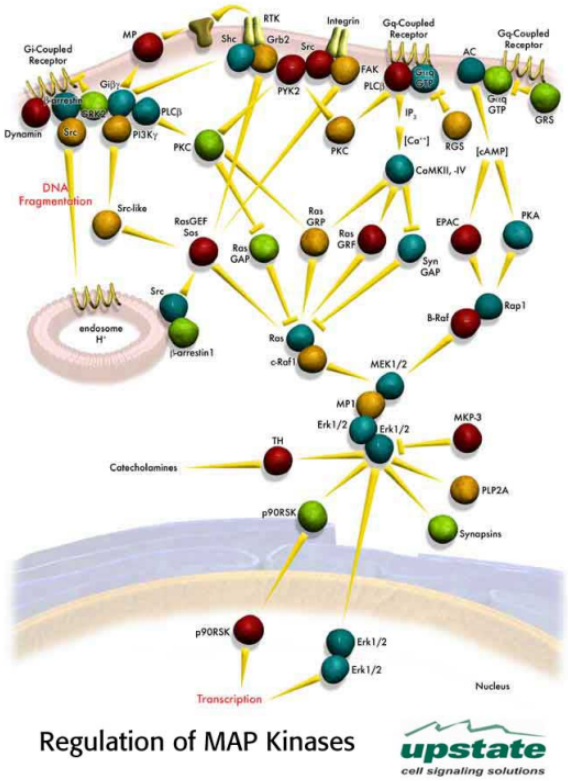


$$\underbrace{\mathbb{P}(D = d \mid A = a, B = b)}_{\text{From marginalization calculation}} = \underbrace{p(d \mid a, b)}_{\text{local factor}}$$

From CPD table

Applications

- Speech recognition
- Diagnosis of diseases
- Study Human genome
- Robot mapping
- Modeling fMRI data
- Fault diagnosis
- Modeling sensor network data
- Modeling protein-protein interact
- Weather prediction
- Computer vision
- Statistical physics
- Many, many more …

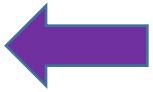


Today: Doing calculations/predictions with bayesian nets.

Types of interesting calculations

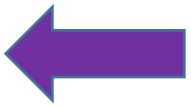
Marginal distribution:

$$\mathbb{P}(A) = \sum_b \mathbb{P}(A, B = b)$$



Conditional distribution:

$$\mathbb{P}(A \mid B = b) = \frac{\mathbb{P}(A, B=b)}{\mathbb{P}(B=b)} = \frac{P(A, B = b)}{\sum_a P(A = a, B = b)}$$



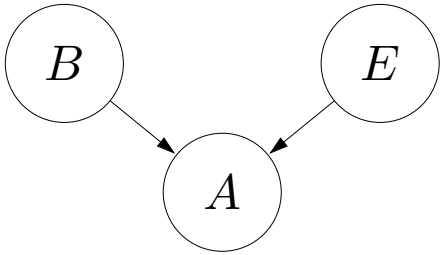
For both tasks, we need to marginalize out some variables.

Example:

Problem: alarm

You have an alarm that goes off if there's a burglary or an earthquake. You hear the alarm go off. What happened?

$$P(B = b, E = e, A = a) = p(b)p(e)p(a \mid b, e)$$

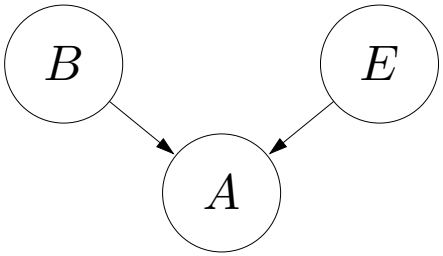


b	$p(b)$
1	ϵ
0	$1-\epsilon$

e	$p(e)$
1	ϵ
0	$1-\epsilon$

A = B ∨ E

$$P(B = b, E = e, A = a) = p(b)p(e)p(a \mid b, e)$$



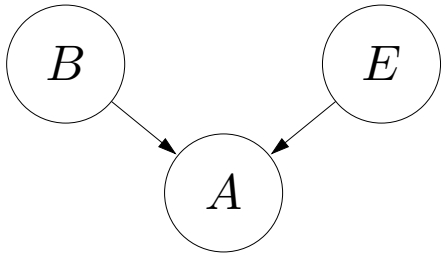
b	$p(b)$
1	ϵ
0	$1-\epsilon$

e	$p(e)$
1	ϵ
0	$1-\epsilon$

A = B ∨ E

b	e	a	$\mathbb{P}(B = b, E = e, A = a)$
0	0	0	$(1-\epsilon)^2$
0	0	1	0
0	1	0	0
0	1	1	$(1-\epsilon)\epsilon$
1	0	0	0
1	0	1	$\epsilon(1-\epsilon)$
1	1	0	0
1	1	1	ϵ^2

$$P(B = b, E = e, A = a) = p(b)p(e)p(a \mid b, e)$$



b	$p(b)$
1	ϵ
0	$1-\epsilon$

e	$p(e)$
1	ϵ
0	$1-\epsilon$

A = B ∨ E

b	e	a	$\mathbb{P}(B = b, E = e, A = a)$
0	0	0	$(1-\epsilon)^2$
0	0	1	0
0	1	0	0
0	1	1	$(1-\epsilon)\epsilon$
1	0	0	0
1	0	1	$\epsilon(1-\epsilon)$
1	1	0	0
1	1	1	ϵ^2

Marginal distribution $\mathbb{P}(B)$?

$\mathbb{P}(B \mid A = 1)$?

$$P(B = 1|A = 1) = \frac{\epsilon}{2\epsilon - \epsilon^2}$$

$$P(B = 0|A = 1) = \frac{\epsilon - \epsilon^2}{2\epsilon - \epsilon^2}$$

$\mathbb{P}(B \mid A = 1, E = 1)$?

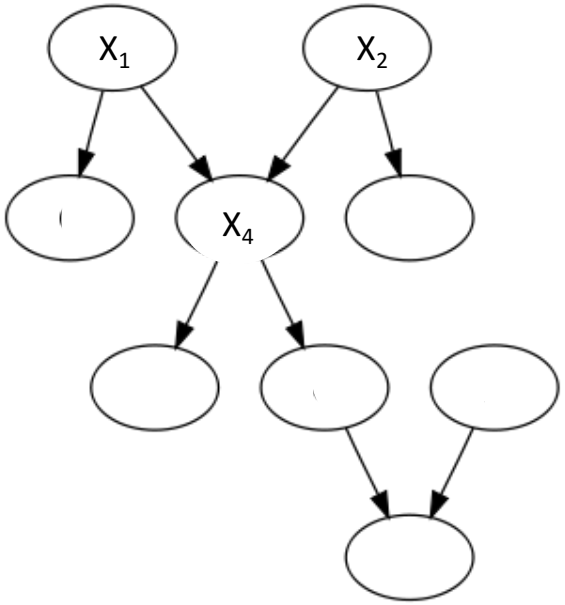
$$P(B = 1|A = 1, E = 1) = \frac{\epsilon^2}{\epsilon} = \epsilon$$

("Explaining away"!)

Computing P(A=a) where A consists of n-k variables requires sum over 2^k terms. Faster way?

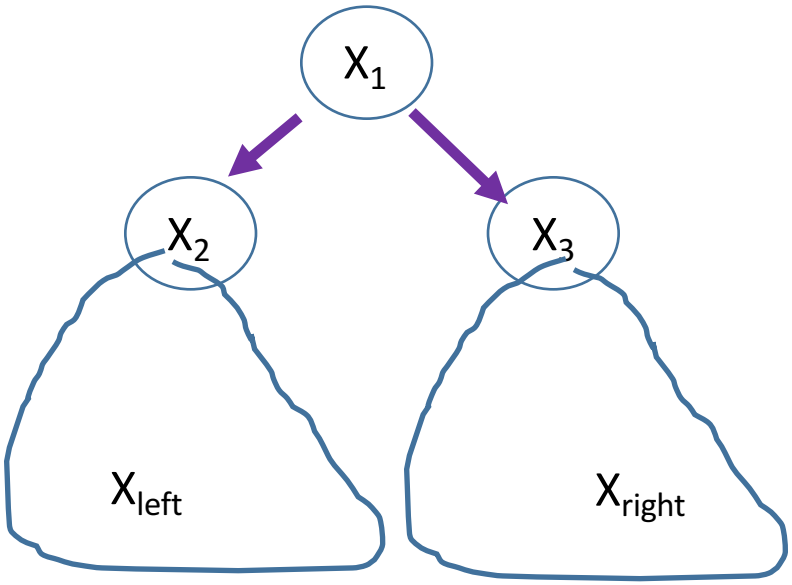
First algorithm today: Bayes net is a polytree

(= directed graph which is a acyclic when we make the edges undirected)



If graph has degree O(1), then an O(n) time algorithm to compute $\mathbb{P}(A=a)$ where A is some subsequence of X_i's. (Big win If n is much less than 2^k.)

Computing Marginals in polytrees.
Will show how to compute $\mathbb{p}(A=a)$.



Intuition: Suppose polytree looks like this

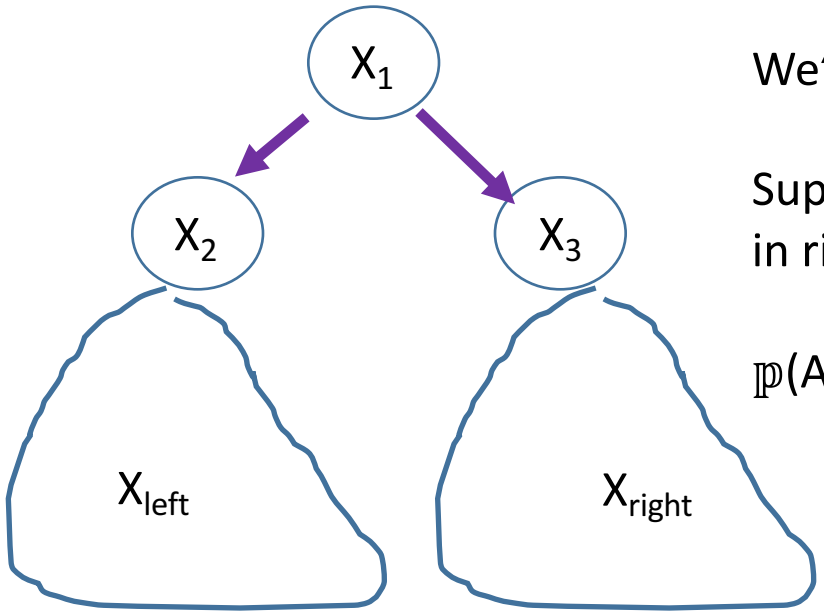
Defines distribution of the form
$P(X_1\ X_2\ X_3\ ,\ldots,X_n) = p(X_1)\ F_1(X_{left})\ F_2(X_{right})$
Where F_1, F_2 describe prob. distributions for the Left and Right subtrees.

We're computing $\mathbb{p}(A=a)$.

Let A_1 =subset of A in left subtree and
A_2= subset of A in right subtree
(a_1, a_2 are their values in a)

$$\mathbb{p}(A=a) = p(X_1=0)\ \mathbb{p}_1(A_1=a_1 \mid X_1=0)\ \mathbb{p}_2(A_2=a_2 \mid X_1=0)$$
$$+\ p(X_1=1)\ \mathbb{p}_1(A_1=a_1 \mid X_1=1)\ \mathbb{p}_2(A_2=a_2 \mid X_1=0)$$

Main observation: Left subtree only affected by X_1 via X_2
(and Right subtree only affected by X_1 via X_3)



We're computing $\mathbb{p}(A=a)$.

Suppose A_1 is subset of A in left subtree and A_2 in right subtree. (a_1, a_2 are their values in a)

$$\mathbb{p}(A=a) = p(X_1=0)\,\mathbb{p}_1(A_1=a_1 \mid X_1=0)\,\mathbb{p}_2(A_2=a_2 \mid X_1=0) + p(X_1=1)\,\mathbb{p}_1(A_1=a_1 \mid X_1=1)\,\mathbb{p}_2(A_1=a_1 \mid X_1=0)$$

Inductive Algorithm: Left subtree computes $\mathbb{p}_1(A_1=a_1, X_2=c)$ for c = 0,1.
Right Subtree computes $\mathbb{p}_1(A_1=a_1, X_3=d)$ for d = 0,1

Putting it together:

$$\mathbb{p}(A=a) = \sum_{b,c,d \text{ in } \{0,1\}} p(X_1=b)\,p(X_2=c \mid X_1=b)\,p(X_3=d \mid X_1=b)\,\mathbb{p}_1(A_1=a_1, X_2=c)\,\mathbb{p}_2(A_2=a_2, X_3=d)$$

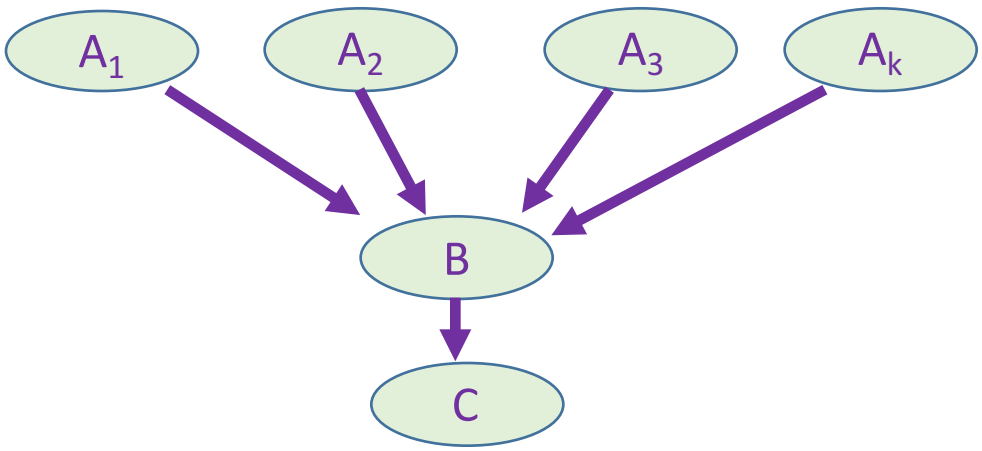
(** For simplicity am assuming

What about doing inference/marginals in bayesian nets that are not polytrees?

Polytree algorithm can be extended, but running time goes way up.
(and computing marginals for completely general bayes nets is NP-hard).

Next: A randomized algorithm (Metropolis-Hastings) to approximate marginals
(Works well in practice; though some bad cases are known)

Recap: Bayes nets as models of probabilistic processes



Step 1: Coins tossed at each A_i node to decide if A_i happened.
($Pr[Heads] = Pr[A_i = 1]$)

Step 2: Coins tossed at B node to decide if B =1. (Pr[Heads] looked up from CPD table)

Step 3: Coins tossed at C node to decide if C =1.

Moral for today: Using some random bits we can efficiently generate a random sample from the distribution defined by the Bayes net.

Randomized approximation algorithm (warmup)

Suppose bayes net (not a polytree) defines a distribution $p(X_1, X_2, .. X_n)$.
How to approximate marginal $\mathbb{p}(X_7 = 1)$?

Generate random samples $(b_1, b_2, \ldots, b_n)$ from bayes net.
Keep track of fraction of times $b_7 = 1$.
(Law of large numbers implies this fraction converges to $\mathbb{p}(X_7 = 1)$ quite fast.)

What goes wrong if we try to compute complicated marginals $\mathbb{p}(X_7 = 1 | A = a)$ (where A has say n/2 variables)?

Answer: If we just produce random samples, the event A=a may be very very unlikely and may not show up for a long time. We need a different way to sample.

Metropolis Hastings Sampling Algorithm

- A recent survey places the **Metropolis algorithm** among the

  10 algorithms that have had the *greatest influence* on the development and practice of science and engineering in the 20th century (Beichl&Sullivan, 2000).

- The Metropolis algorithm is an instance of a large class of sampling algorithms, known as **Markov chain Monte Carlo** (MCMC).

Bread and butter of statistical calculations!

It will let us sample from the sub-distribution P(A=a)

Random walk

Drunk man leaves a bar in Manhattan. Whenever arrives at any street corner, goes N or S or E or W with probability ¼. How long before he gets to his apartment?

(Answer: Walks $O(n^4)$ blocks if apartment is n blocks away. Talk to me if you want to know how to do such calculations.)

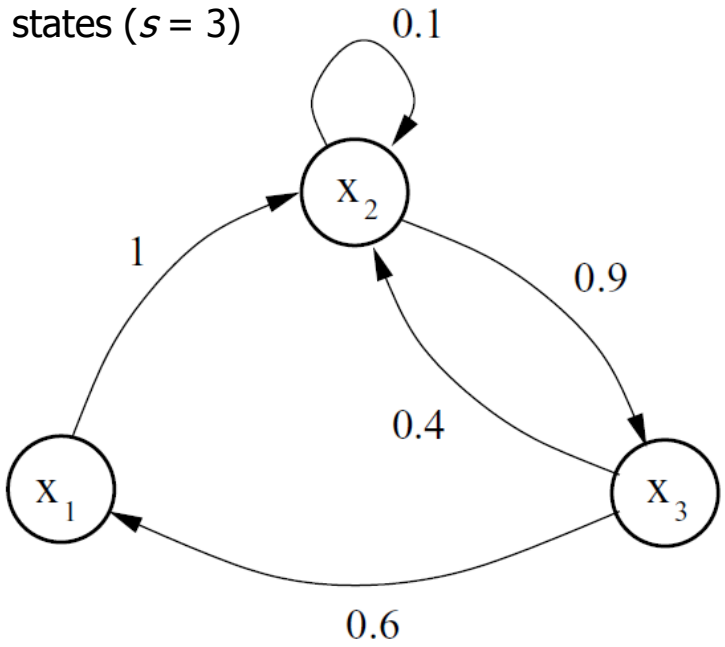
2-D random walk
https://upload.wikimedia.org/wikipedia/commons/a/a9/2D_Random_Walk_400x400.ogv

Markov Chain (drunkard's walk on weighted directed graph)

Markov chain with three states ($s = 3$)

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$

Transition matrix



Transition graph

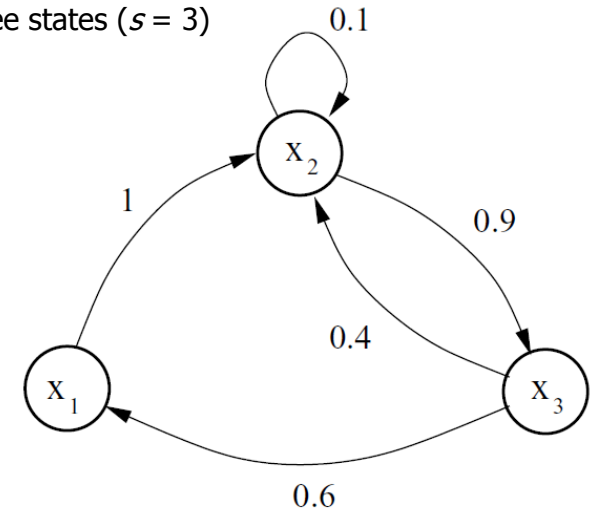
Directed graph, and a transitition matrix giving, for each i, j the probability of stepping to j when at i.

At step n, his whereabouts are given by some vector

p_i = Prob he is at node i.

Markov Chain (drunkard's walk on weighted directed graph)

Markov chain with three states ($s = 3$)



$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$

Transition matrix

Transition graph

Transition matrix T giving, for each i, j the probability of stepping to j when at i.

Fact: Evolution of probability distribution given by = Vector x Matrix.

Suppose, at step n

p_i = prob. he is at node x_i

Then prob. he is at x_k at step n+1

$$= \sum_i p_i T_{ik}$$

"Distribution at step n +1" is pT

(Note: row vector times matrix is a row vector)

Stationary Distribution

$$\text{Distribution } \pi = (\pi_1, \ldots, \pi_m) \text{ is stationary if } \pi_i \geq 0 \ \forall i,$$

$$\sum_i \pi_i = 1 \quad \text{and } \pi T = \pi$$

(Taking one step according to the markov chain leaves this distribution unchanged)

$$(0.22, 0.41, 0.37) \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix} = (0.22, 0.41, 0.37)$$

Under some reasonable conditions (ergodicity) this distribution is unique, and reached in finite time from any starting position.

Stationary Distribution

Distribution $\pi = (\pi_1, \ldots, \pi_m)$ is station

$$\sum_i \pi_i = 1 \quad \text{and } \pi T = \pi$$

(Taking one step according to the markov chain leaves this distribution unchan

$$(0.22, 0.41, 0.37) \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix} = (0.22, 0.41, 0.37)$$

Alternative take: Drunkard's walk run for this # of steps is a way to draw a sample according to this stationary distribution.
Important: # of nodes m can be large; drunkard moves node to node in this large graph.

Under some reasonab conditions (ergodicity) this distribution is unique and reached in finite time from any starting position.

Where we are headed.

Suppose probability distribution is $p(X_1, X_2, .. X_n)$; we desire $\mathbb{p}(X_7=1|A= a)$

We will do a random walk on a Markov chain where vertices are all bit strings $(X_1, X_2, .. X_n)$ in which A=a.
(Graph has size 2^{n-k} where k is the size of A. Too large to write down; but drunkard's walk only needs to know the local edges out of each node it reaches.)

Markov chain constructed such that stationary distribution is the distribution conditional on A= a.

➔ Drunkard's walk gives us a sample. Repeat a few times and estimate $\mathbb{p}(X_7=1|A= a)$

Metropolis Hastings Algorithm

Let $b_1, \ldots, b_m > 0$, and $B = \sum_{j=1}^{m} b_j$

(This kind of quantity is of interest in computing marginals!)

Assume that m is so big, that it is difficult to calculate B.

Our goal:

Generate samples from the following **discrete** distribution:

$$P(X = j) = \pi_j = \frac{b_j}{B}$$

We don't know *B*!

The main idea is to construct a time-reversible Markov chain with $(\pi_1,\ldots,\pi_m)$ limit distributions

Goal

Generate samples from the following **discrete** distribution:

$$P(X = j) = \pi_j = \frac{b_j}{B}$$ **We don't know *B*!**

j=1, 2, .. , m (m is big)

In our heads, create a graph with m nodes. (remember, m is big)

For all nodes i, P_{ii} = 0.5. (i.e., stay in place with prob. ½)

For i ≠j, where i has an edge to j: $P_{ij} = \frac{0.5}{degree(i)} \min\{1, \frac{b_j}{b_i}\}$

Claim: The desired distribution is the unique stationary distribution for this markov chain if all b_j's are nonzero. ➔ Drunkard walk gives us samples from this distrib.

Using Metropolis-Hastings for computing marginals

Suppose probability distribution is $p(X_1, X_2, .. X_n)$; we desire $\mathbb{P}(X_7=1 | A= a)$

Graph in our head: Nodes are all possible samples where A=a;
Edges correspond to pairs of samples that differ in exactly 1 bit.
b_i = Probability of the sample represented by node i.

We run Metropolis-Hastings to generate samples from the distribution where A =a. (Graph is too big to write down, but can do drunkard walk in it)

For i ≠j, where i has an edge to j: $P_{ij} = \frac{0.5}{degree(i)} \min\{1, \frac{b_j}{b_i}\}$ (This is ratio of probabilities!)

Using Metropolis-Hastings for computing marginals

Suppose probability distribution is $p(X_1, X_2, .. X_n)$; we desire $\mathbb{P}(X_7=1 | A= a)$

Graph in our head: Nodes are all possible samples where A=a;
Edges correspond to pairs of samples that differ in exactly 1 bit.
b_i = Probability of the sample represented by node i.

We run Metropolis-Hastings to generate samples from the distribution where A =a. (Graph is too big to write down, but can do drunkard walk in it)

For i ≠j, where i has an edge to j: $P_{ij} = \frac{0.5}{degree(i)} \min\{1, \frac{b_j}{b_i}\}$ (This is ratio of probabilities!)

Easy to check: There is a simple algorithm that given two samples that differ by 1 bit, computes the ratio of their probabilities.