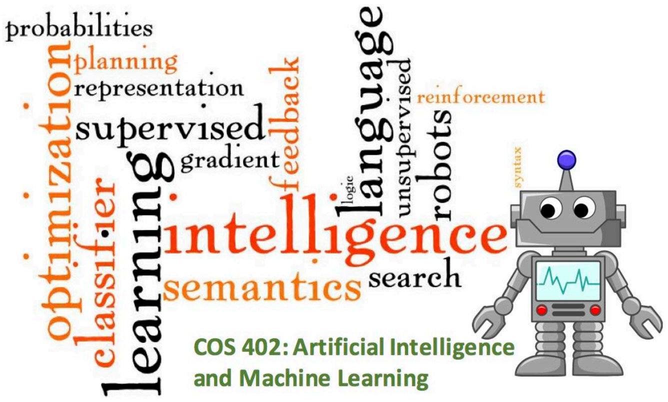


COS 402 – Machine Learning and Artificial Intelligence
Fall 2016

Lecture 15: MCMC

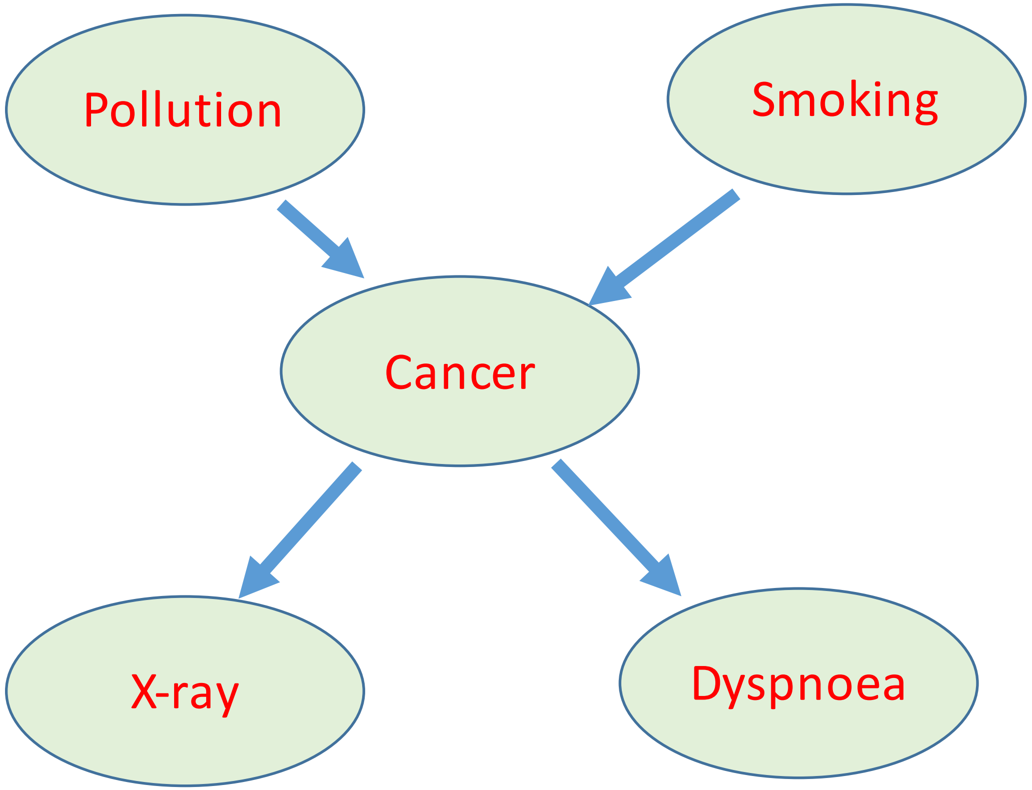
Sanjeev Arora Elad Hazan

PRINCETON UNIVERSITY

Course progress

- Learning from examples
  - Definition + fundamental theorem of statistical learning, motivated efficient algorithms/optimization
  - Convexity, greedy optimization – gradient descent
  - Neural networks
- Knowledge Representation
  - NLP
  - Logic
  - Bayes nets
  - Optimization: MCMC (TODAY)
- Next: reinforcement learning

Goal: inference in Bayes networks



Node name	Type	Values
Pollution	Binary	{*low*, *high*}
Smoker	Boolean	{*T*, *F*}
Cancer	Boolean	{*T*, *F*}
Dyspnoea	Boolean	{*T*, *F*}
X-ray	Binary	{*pos*, *neg*}

How to sample from a distribution?

- How to generate a random number?

- Von-Neumann's coin
  given a biased coin – turns up heads w.p. $p \neq \frac{1}{2}$
  how to generate a random bit?

How to sample from a distribution?

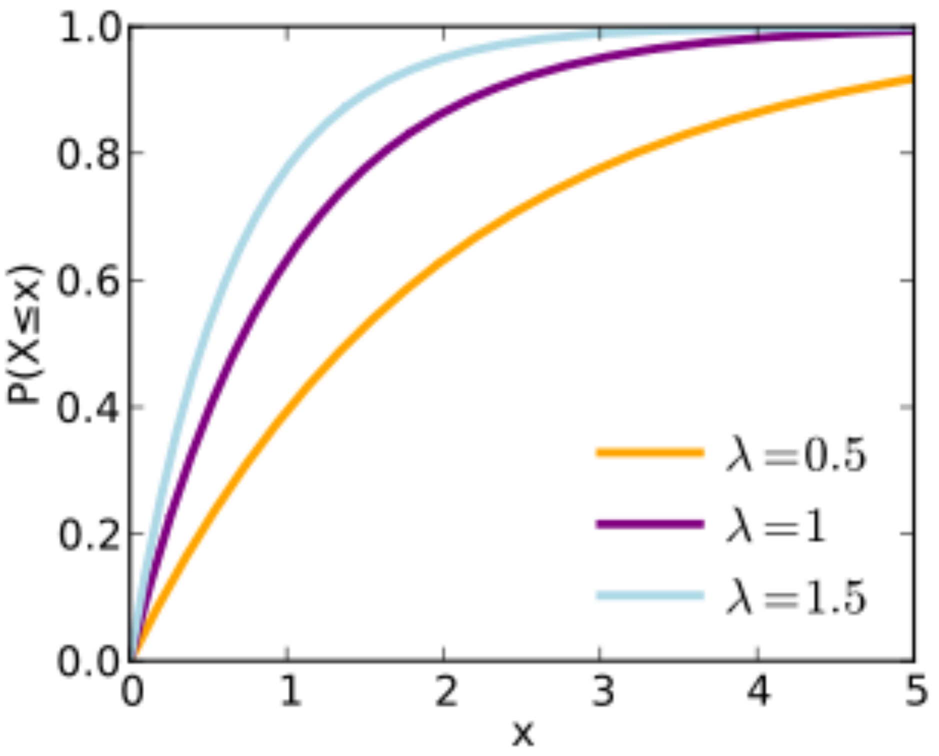
- How to generate a random number?

- Von-Neumann's coin:

- From now on: assume we have access to U[0,1]
- Uniformly at random on an interval?
- Exponential?

Inverse transform method

- Cumulative distribution function

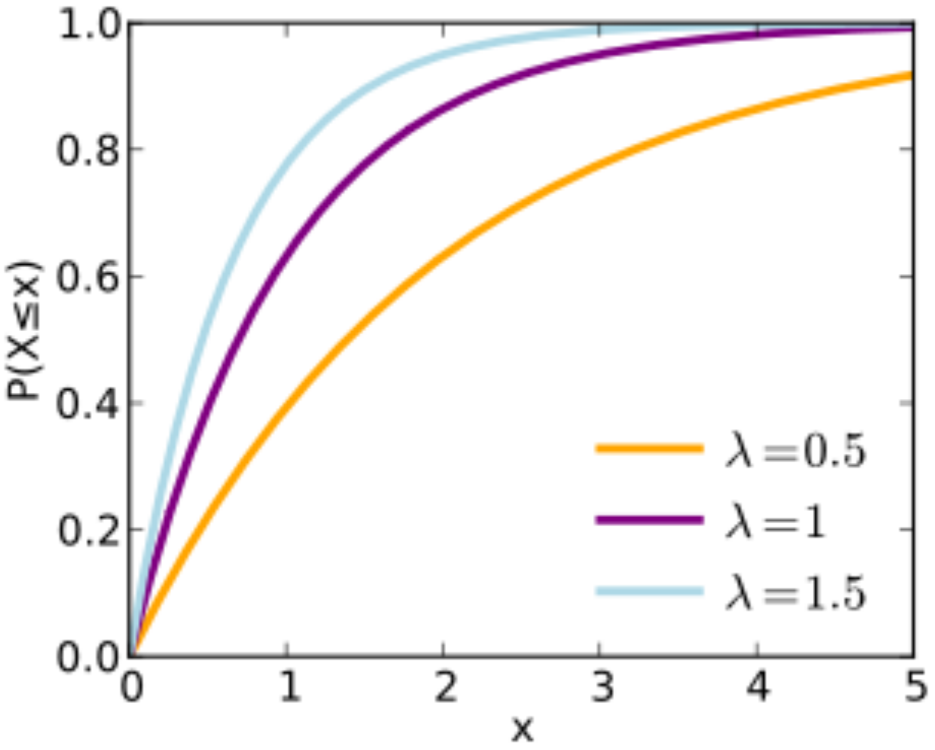


Inverse transform method

- Let $F: R \mapsto [0,1]$ be the CDF we want to sample from, let
  $F^{-1}: [0,1] \mapsto R$ be its inverse.

- Algorithm: sample $Y \sim U[0,1]$ and return $\mathrm{X} = F^{-1}(Y)$
- Theorem: $X \sim F$

- Exponential distribution: $\mathrm{F(x)} = 1 - \mathrm{e}^{-\lambda x}$ for $\mathrm{x} \geq 0$, so sample $\mathrm{y} \sim U[0,1]$, and return $-\frac{1}{\lambda} \ln(1-y)$

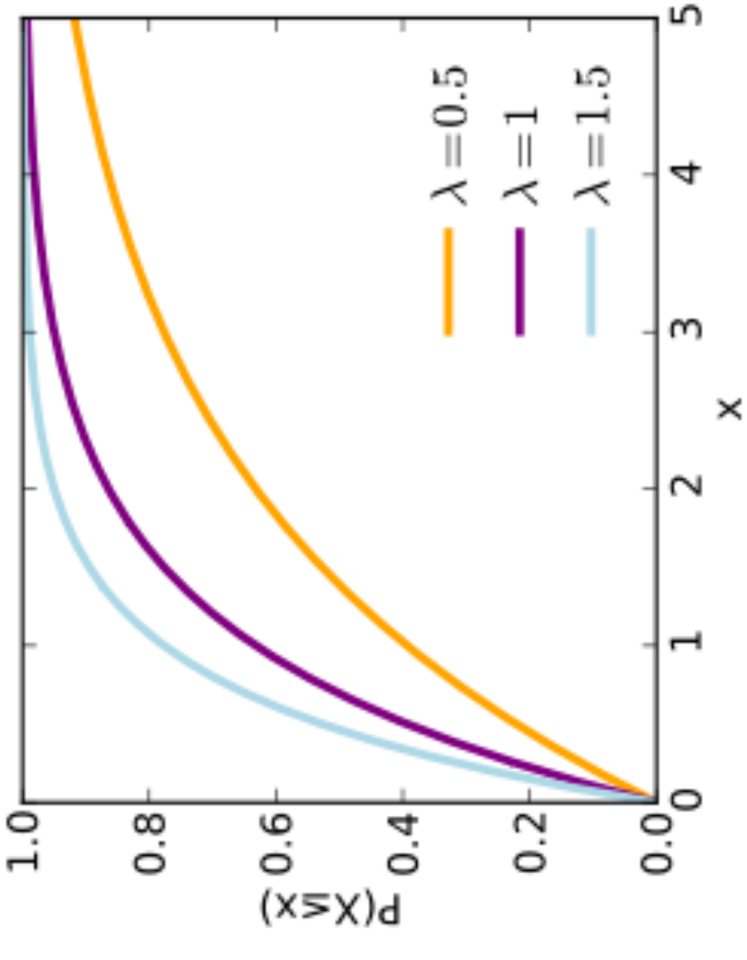


Inverse transform method

- Let $F\!: R \mapsto [0,1]$ be the CDF we want to sample from, let
  $F^{-1}\!: [0,1] \mapsto R$ be its inverse.

- Algorithm: sample $Y \sim U[0,1]$ and return $\mathrm{X} = F^{-1}(Y)$
- Theorem: $X \sim F$

- Exponential distribution: $\mathrm{F}(\mathrm{x}) = 1 - \mathrm{e}^{-\lambda x}$ for $\mathrm{x} \geq 0$,
  so sample $\mathrm{y} \sim U[0,1]$, and return $-\frac{1}{\lambda}\ln\,(1-y)$



How to sample from a distribution?

- How to generate a random number?

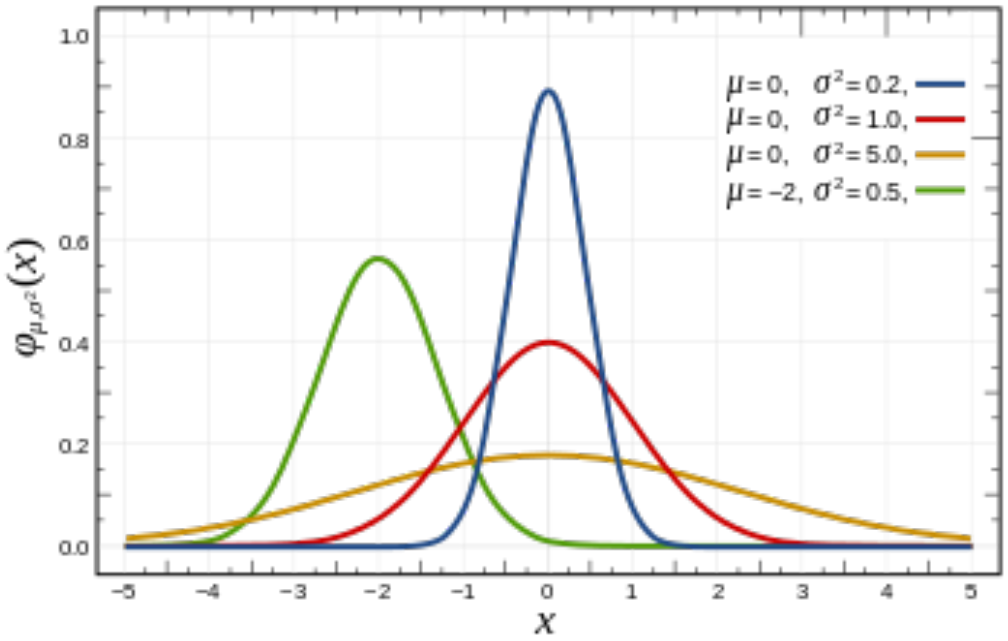
- Von-Neumann’s coin:

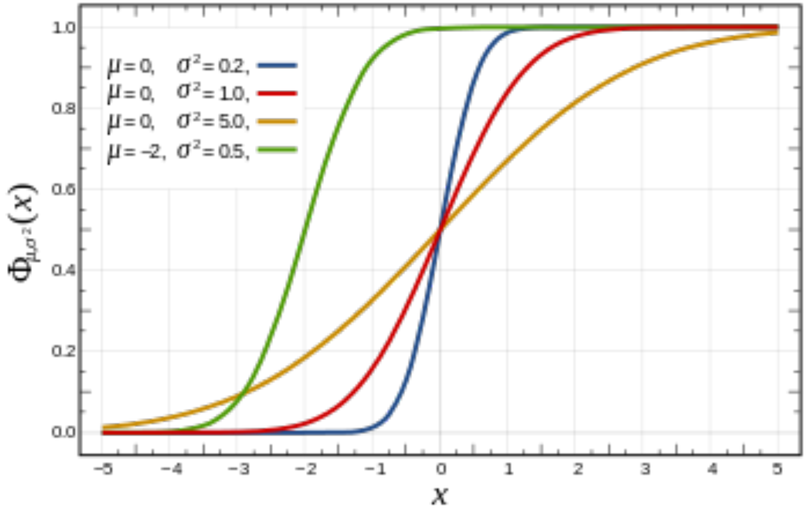
- From now on: assume we have access to U[0,1]
- Uniformly at random on an interval?
- Exponential?
- Gaussian/Normal?

Normal random variable (Gaussian)

PDF and CDF:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

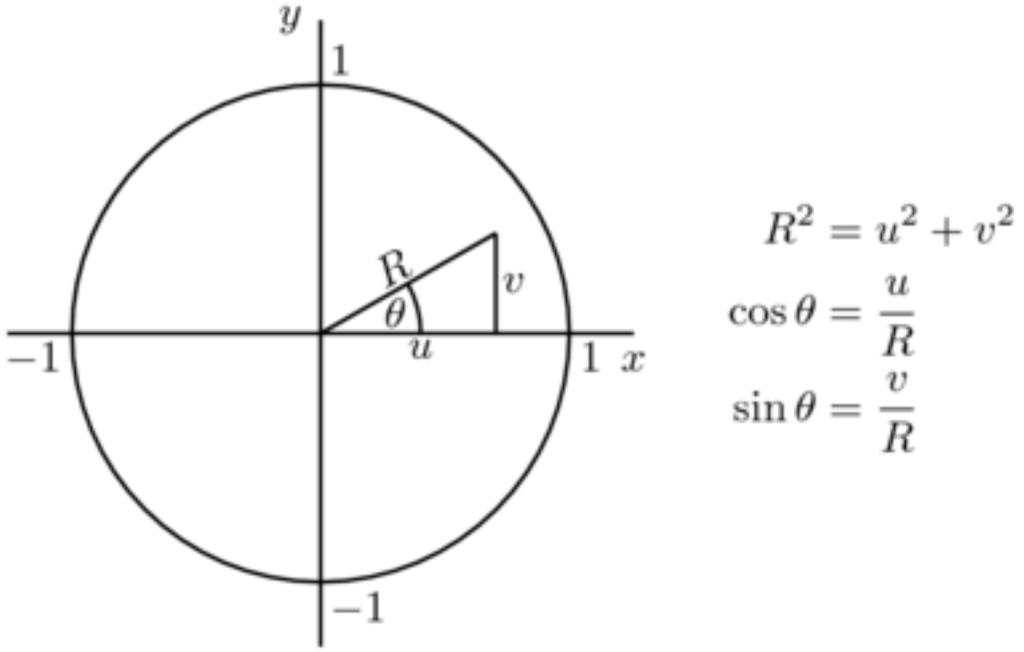




why can’t we use inverse transform?

Gaussian sample: Box-Muller algorithm

- Idea – convert to radial basis
  Sample two variables $\mathrm{r} \sim \exp\left(\frac{1}{2}\right), \theta \sim U[0,1]$ and return the Cartesian coordinates:
  $\mathrm{X} = \mathrm{r}\cos\theta\,, Y = r\sin\theta$



Gaussian sample: Box-Muller algorithm

- Idea – convert to radial basis
  Sample two variables $\mathrm{r} \sim \exp\left(\frac{1}{2}\right), \theta \sim U[0,1]$ and return the Cartesian coordinates:
  $\mathrm{X} = \mathrm{r}\cos\theta\,, Y = r\sin\theta$

- Theorem: $\mathrm{X}, \mathrm{Y} \sim N[0,1]$ and are independent

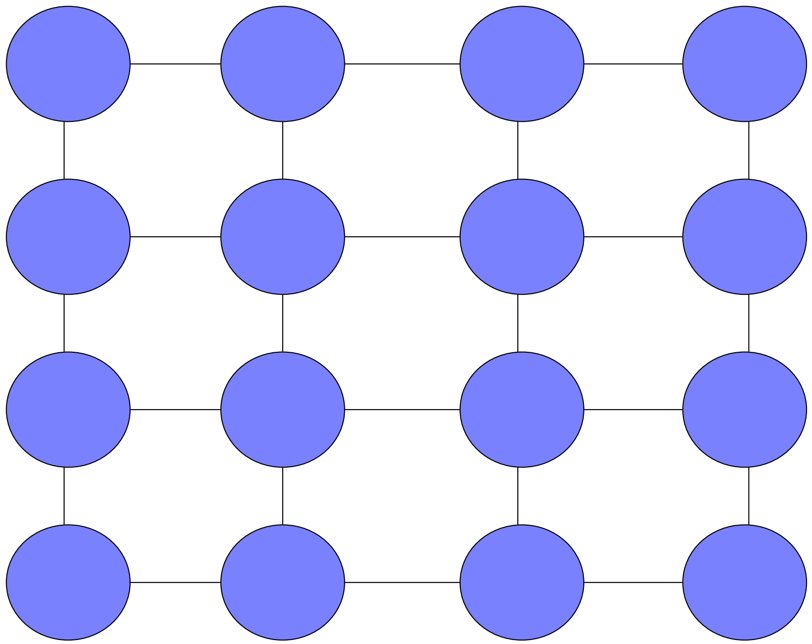
- Proof idea: sampling to i.i.d normal RV, is rotation symmetric, and radius distributed as exp(1/2).

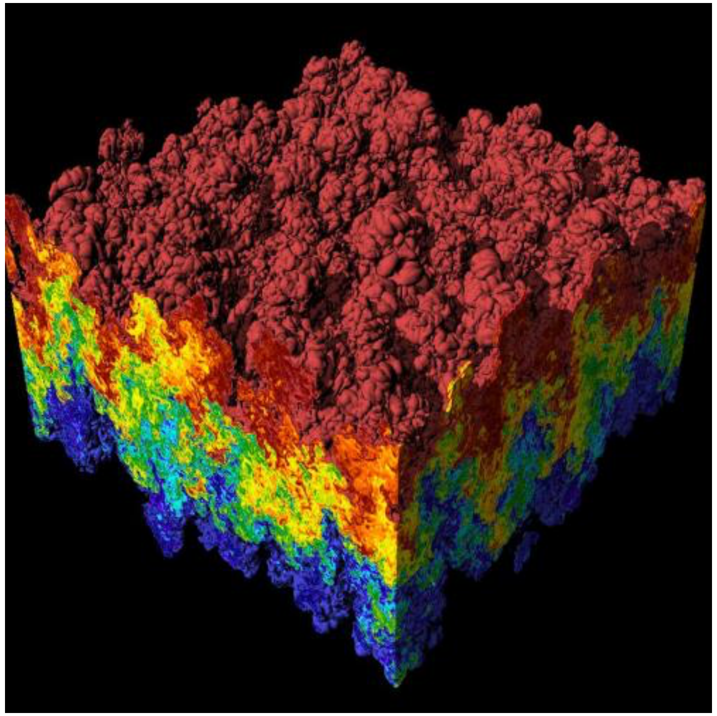
How to sample from a distribution?

- Sampling from a multi-dimensional distribution?
  - Inverse transform -> many times hard to compute
  - other methods (importance sampling, etc.) degrade exponentially with the dimension
  - Many times provably computationally hard
  - But also very important!

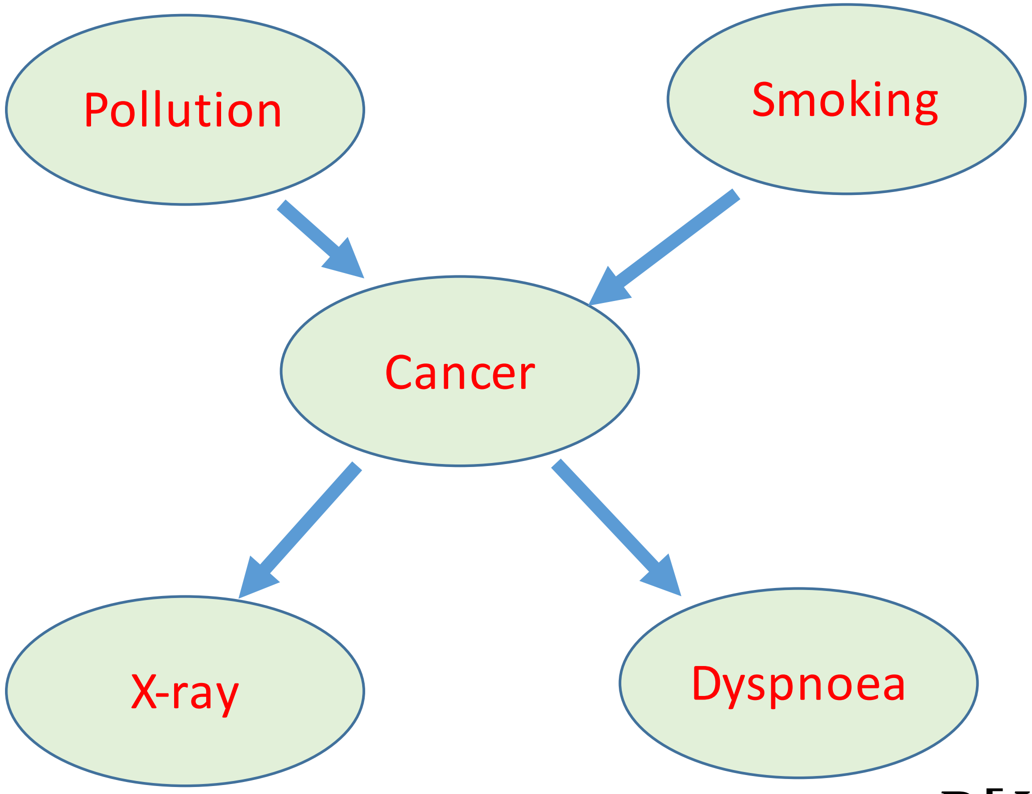
Los Alamos simulations

- Need to simulate complicated multi-particle experiments
- 0-1 assignment,
  valid configuration: "no neighboring 1's"





Sampling in Bayes networks??



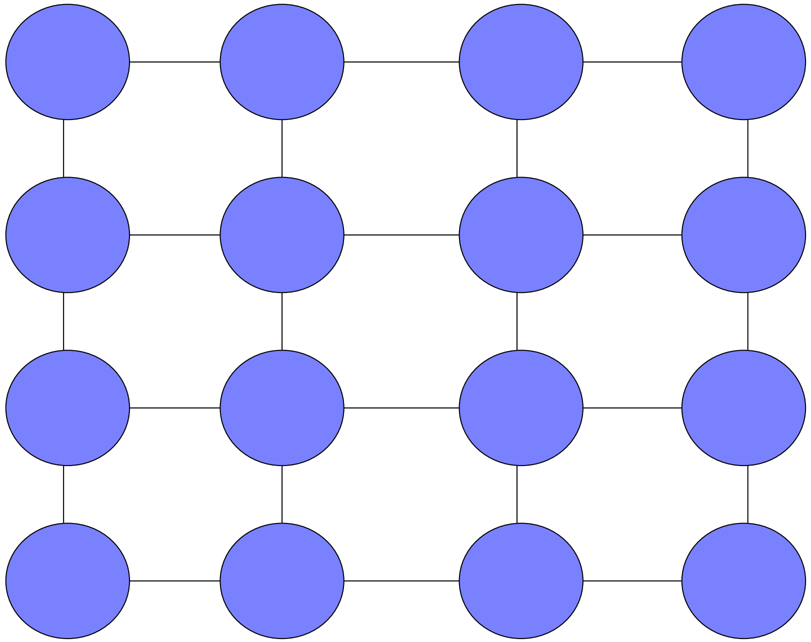
Node name	Type	Values
Pollution	Binary	{*low, high*}
Smoker	Boolean	{*T, F*}
Cancer	Boolean	{*T, F*}
Dyspnoea	Boolean	{*T, F*}
X-ray	Binary	{*pos, neg*}

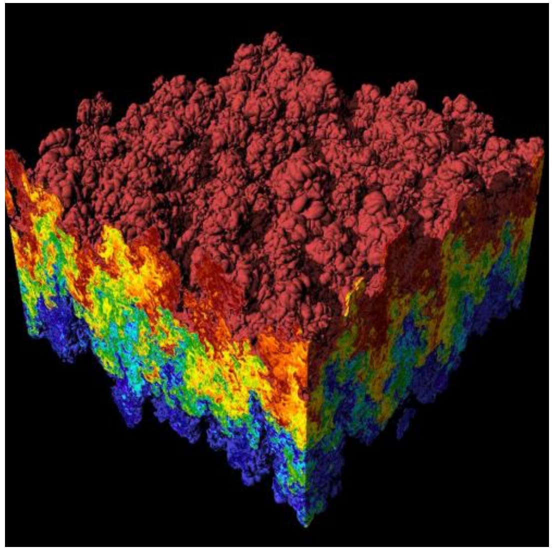
$P[X_i = a_i | X_1 = a_1, \ldots, X_n = a_n, X_j = ?] = ?$

$P[X_1 = a_1 | X_5 = a_5] = ?$

The MCMC paradigm

“to sample from a distribution p, **design** a Markov Chain whose stationary distribution is $\pi = p$. Then simulate the Markov Chain and sample from it after it has mixed (reached stationarity).“





The MCMC paradigm

“to sample from a distribution p, **design** a Markov Chain whose stationary distribution is $\pi = p$. Then simulate the Markov Chain and sample from it after it has mixed (reached stationarity).“

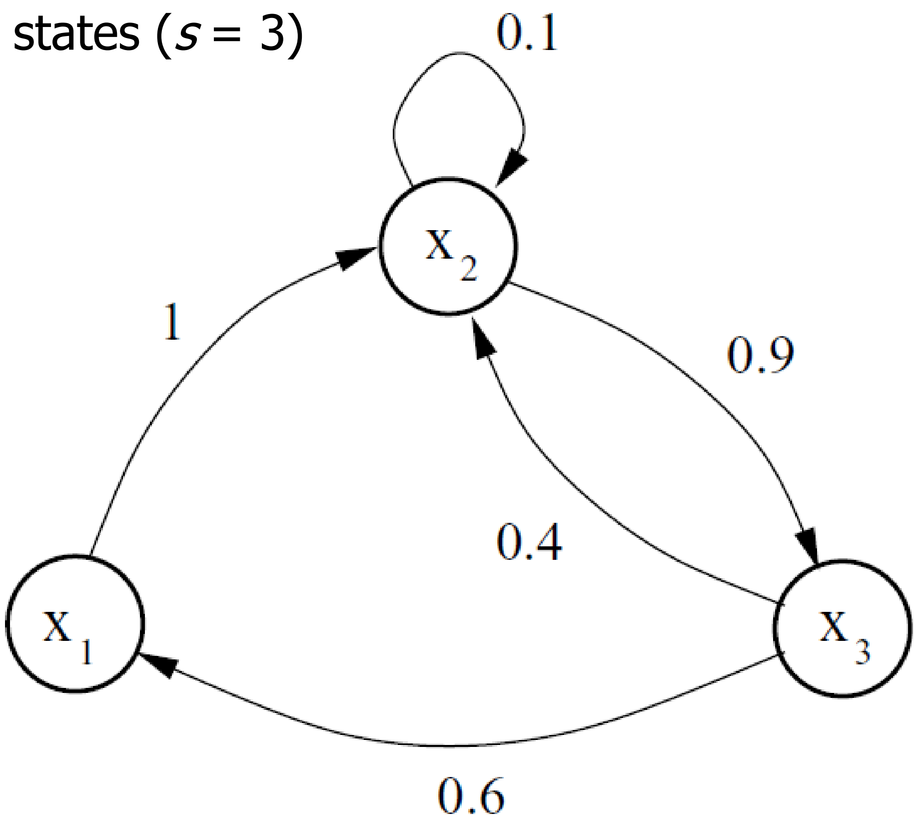
1. What is a Markov Chain & stationary dist. ?
2. When does it have a stationary distribution and how to find it / sample from it efficiently?
3. How to design a Markov Chain for a given distribution?

Markov Chain

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.

Markov Chains – usage and examples

Common example: PageRank (google's webpages initial ranking system)

Webgraph:
Nodes = webpages , Edges = hyperlinks

T_{ij} = probability to move from page i to page j = $\begin{cases} \frac{1}{d_i} & (i,j) \sim E \\ 0 & o/w \end{cases}$

d_i = degree (outgoing links) from page i

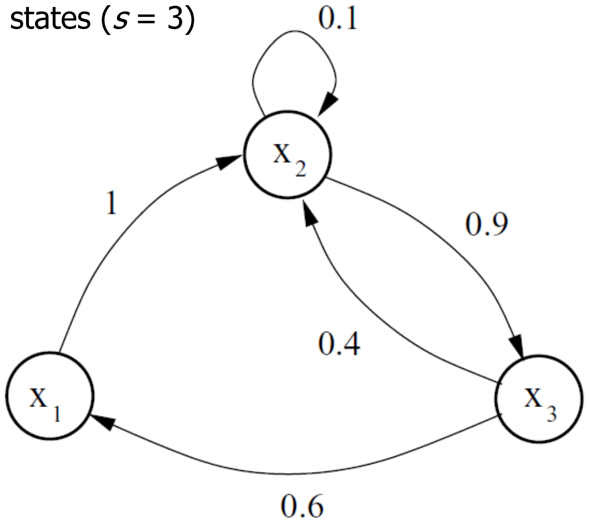
PageRank score for page = $\pi(i)$ = prob. in stationary distribution!

Random Walks in a Markov Chain

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

Starting from state i, the distribution after one step is given by

$$p_1 = e_i \,, \qquad p_2 = e_i T$$

After n steps:

$$p_n = e_i T * T * \cdots * T = e_i T^{n-1}$$

Let:

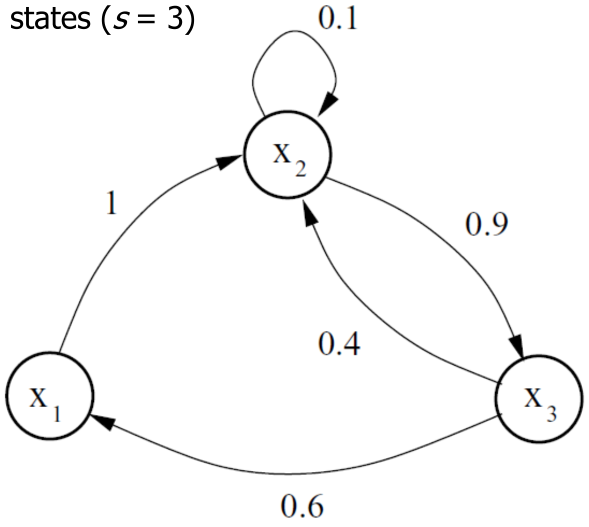
$$\pi = \lim_{n \mapsto \infty} e_i T^n$$

Random Walks in a Markov Chain

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

Let:

$$\pi = \lim_{n \mapsto \infty} e_i T^n$$

Thus,

$$\pi T = \pi$$

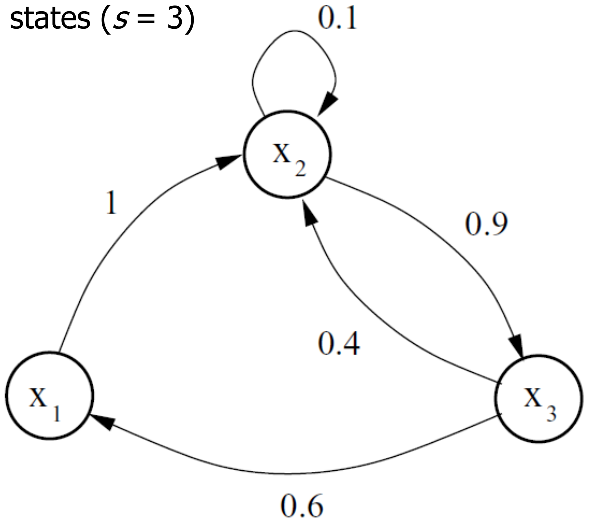
"stationary distribution"

Random Walks in a Markov Chain

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

For this MC:

$$p_1 = e_1 = (1,0,0)$$
$$p_2 = e_1 T = (0,1,0)$$
$$p_3 = p_2 T = (0,0.1,0.9)$$
$$p_4 = p_3 T = (0.54,0.37,0.09)$$

…

$$\pi = p_\infty \approx (0.22,0.41,0.37)$$

Stationary Distribution

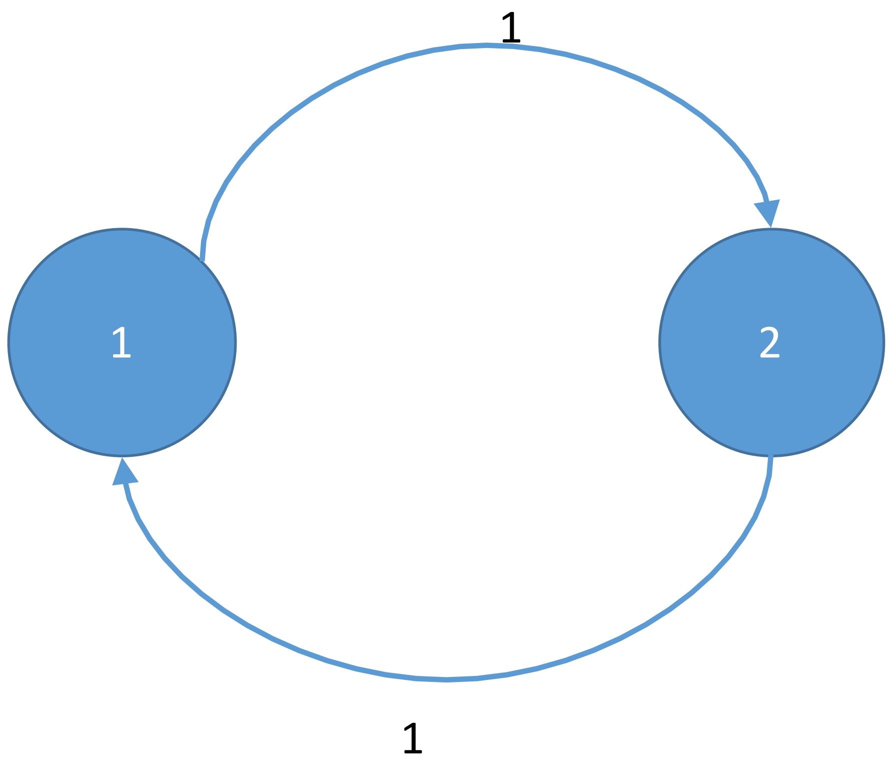
Distribution $\pi = (\pi_1, \ldots, \pi_m)$ is stationary if $\pi_i \geq 0 \; \forall i$,

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

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

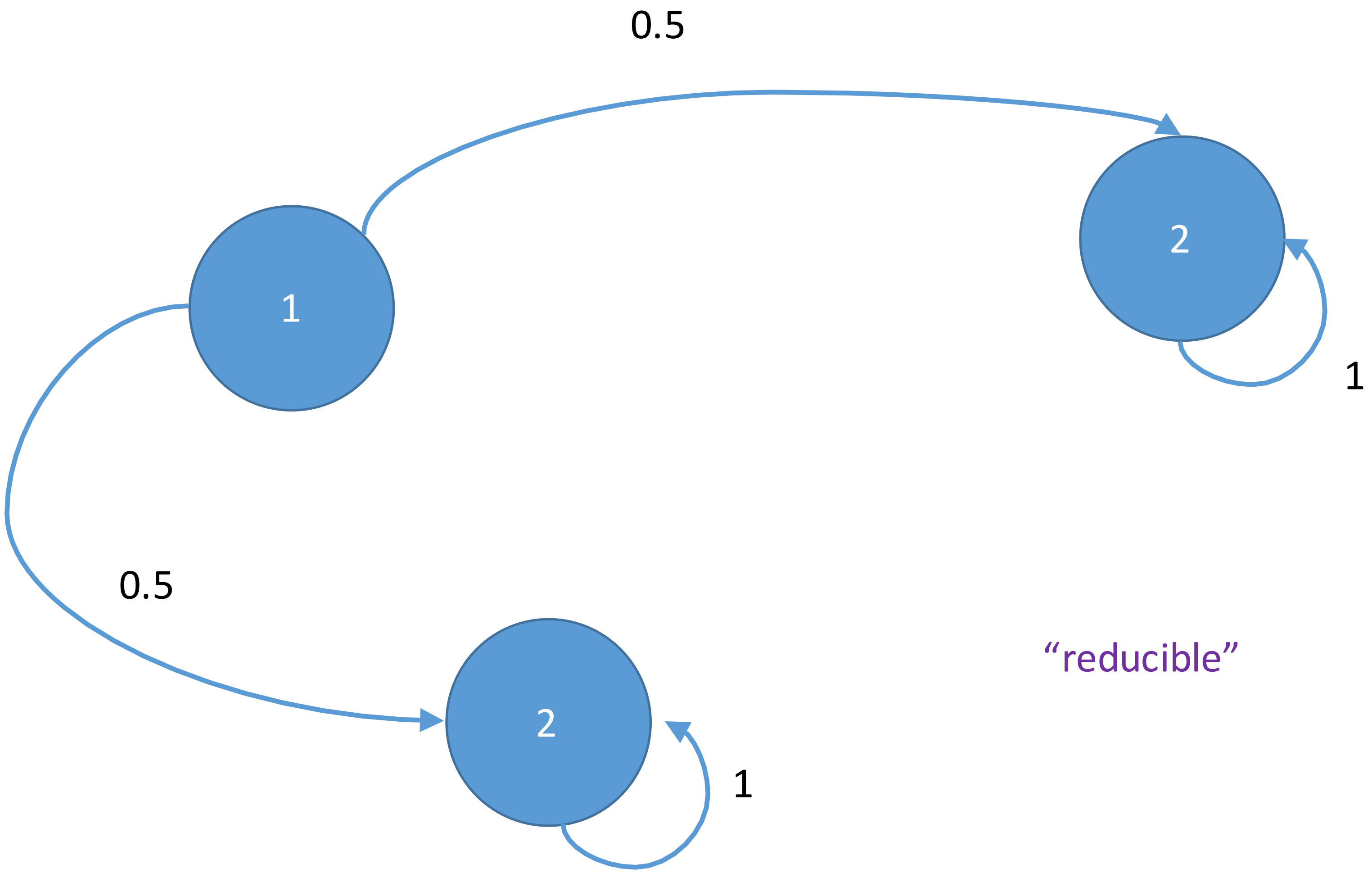
$$(0.22, 0.41, 0.37) \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix} = (0.22, 0.41, 0.37)$$

Non-stationary Markov chains



"periodic"

Non-stationary Markov chains



"reducible"

Ergodic theorem

Amazingly, every irreducible and a-periodic Markov chain has a unique stationary distribution, and every random walk starting from any node converges to it!

→ implication to PageRank…

Mixing time is:

$$n_\epsilon \quad s.t. \quad |e_i T^{n_\epsilon} - \pi| \leq \epsilon$$

In general – depends polynomially in #nodes, very hard to bound.
Many times in practice – depends logarithmically in #nodes!

Designing a Markov chain: Metropolis-Hastings

Input: distribution we wish to sample from, probability of even i is given by p_i

Output: sample from Markov Chain whose stationary distribution is $\pi = p$

MH algorithm: for t=1,2,…,T

1. Start in arbitrary state i, and let $s_1 = i$
2. At time t, pick state j from [n] uniformly at random (or some other "Reasonable" distribution).
3. Update the step according to the rule:

$$s_{t+1} = \begin{cases} j & w.p.\ \min\{1, \frac{p_j}{p_i}\} \\ s_t & o/w \end{cases}$$

4. Return to (2), unless t=T, in which case stop and return s_t

Designing a Markov chain: Metropolis-Hastings

Theorem: Markov Chain below is always stationary with stationary distribution being $\pi = p$

MH algorithm: for t=1,2,…,T

1. Start in arbitrary state i, and let $s_1 = i$
2. At time t, pick state j from [n] uniformly at random (or some other "Reasonable" search rule).
3. Update the step according to the rule:

$$s_{t+1} = \begin{cases} j & w.p.\ \min\{1, \frac{p_j}{p_i}\} \\ s_t & o/w \end{cases}$$

4. Return to (2), unless t=T, in which case stop and return s_t

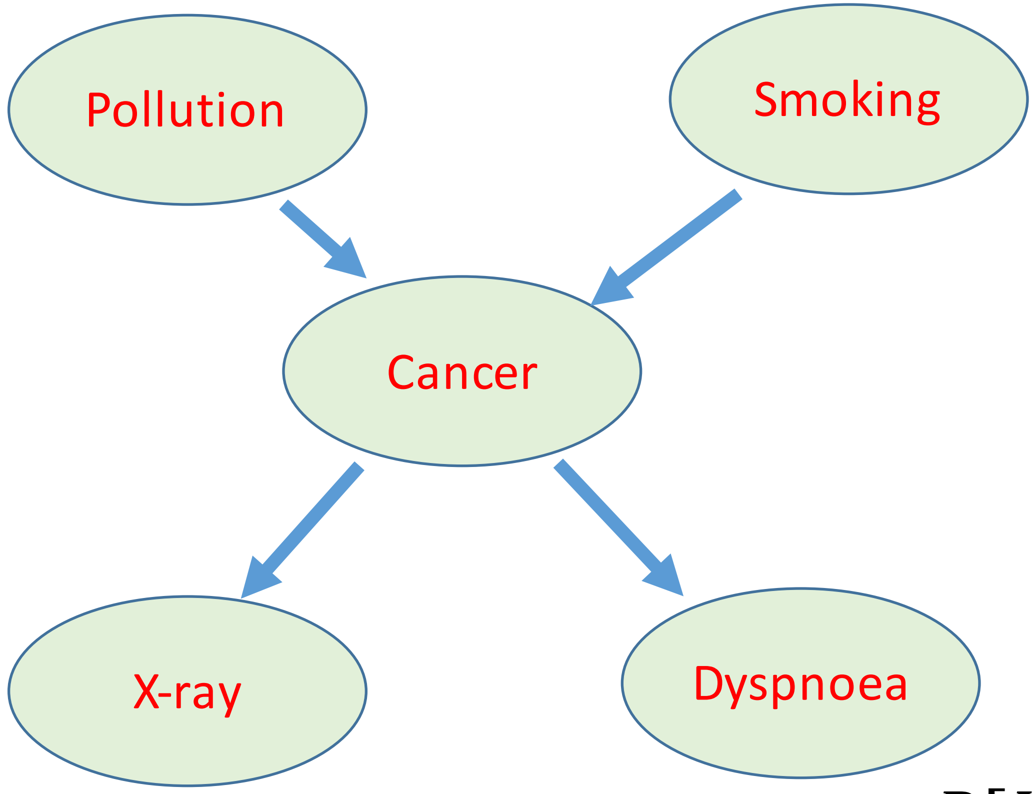
Metropolis Hastings Algorithm

- Create Markov chain
- start from any state
- simulate MC many many times (mixing time)
- return final state

By theorem it is a sample from p!

Next: applying it to Bayesian inference!

Sampling in Bayes networks by MCMC



Node name	Type	Values
Pollution	Binary	$\{low, high\}$
Smoker	Boolean	$\{T, F\}$
Cancer	Boolean	$\{T, F\}$
Dyspnoea	Boolean	$\{T, F\}$
X-ray	Binary	$\{pos, neg\}$

$$P[X_i = a_i | X_1 = a_1, \ldots, X_n = a_n, X_j =?] = ?$$

$$P[X_1 = a_1 | X_2 = a_2, X_5 = a_5] = ?$$

Sampling in Bayes networks by MCMC

Goal: estimate $P[X_1 = a_1 | X_2 = a_2, X_5 = a_5]$

Method: sample from $P[X_1 | X_2 = a_2, X_5 = a_5]$ and estimate the probability of value $X_1 = a_1$. Assume binary values, $a_i \in \{0,1\}$.

Graph: all possible assignments to all variables (2^n nodes!).

The MCMC algorithm:

1. Start in arbitrary state $(b_1, b_2, \dots, b_n)$, such that $b_2 = a_2, b_5 = a_5$
2. Pick random variable $X_j \neq X_2, X_5$, move to state $(b_1, b_2, \dots, 1 - b_j, \dots, b_n)$ with probability

$$\frac{P[b_1, b_2, \dots, 1 - b_j, \dots, b_n]}{P[b_1, b_2, \dots, b_n]}$$

3. Return to (2), unless reached limit, in which case return current state

Different search rule (in Metropolis-Hastings)

Goal: estimate $P[X_1 = a_1 | X_2 = a_2, X_5 = a_5]$

Method: sample from $P[X_1 | X_2 = a_2, X_5 = a_5]$ and estimate the probability of value $X_1 = a_1$. Assume binary values, $a_i \in \{0,1\}$.

Graph: all possible assignments to all variables (2^n nodes!).

The MCMC algorithm:

1. Start in arbitrary state $(b_1, b_2, \ldots, b_n)$, such that $b_2 = a_2, b_5 = a_5$
2. Pick two variable $X_{j_1}, X_{j_2} \neq X_2, X_5$, move to state $\left(b_1, \ldots, 1 - b_{j_1}, \ldots, 1 - b_{j_2}, \ldots, b_n\right)$ with probability

$$\frac{P[\left(b_1, \ldots, 1 - b_{j_1}, \ldots, 1 - b_{j_2}, \ldots, b_n\right)]}{P[b_1, b_2, \ldots, b_n]}$$

3. Return to (2), unless reached limit, in which case return current state

The MCMC paradigm - summary

“to sample from a distribution p, **design** a Markov Chain whose stationary distribution is $\pi = p$. Then simulate the Markov Chain and sample from it after it has mixed (reached stationarity).“

1. Metropolis-Hastings – general methodology for designing Markov chains for a given distribution
2. Can be applied to Bayes networks, since only ratio of local probabilities needed
3. Mixing time – the hard quantity to bound (usually poly-graph-size, which is exponential in theory)