10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

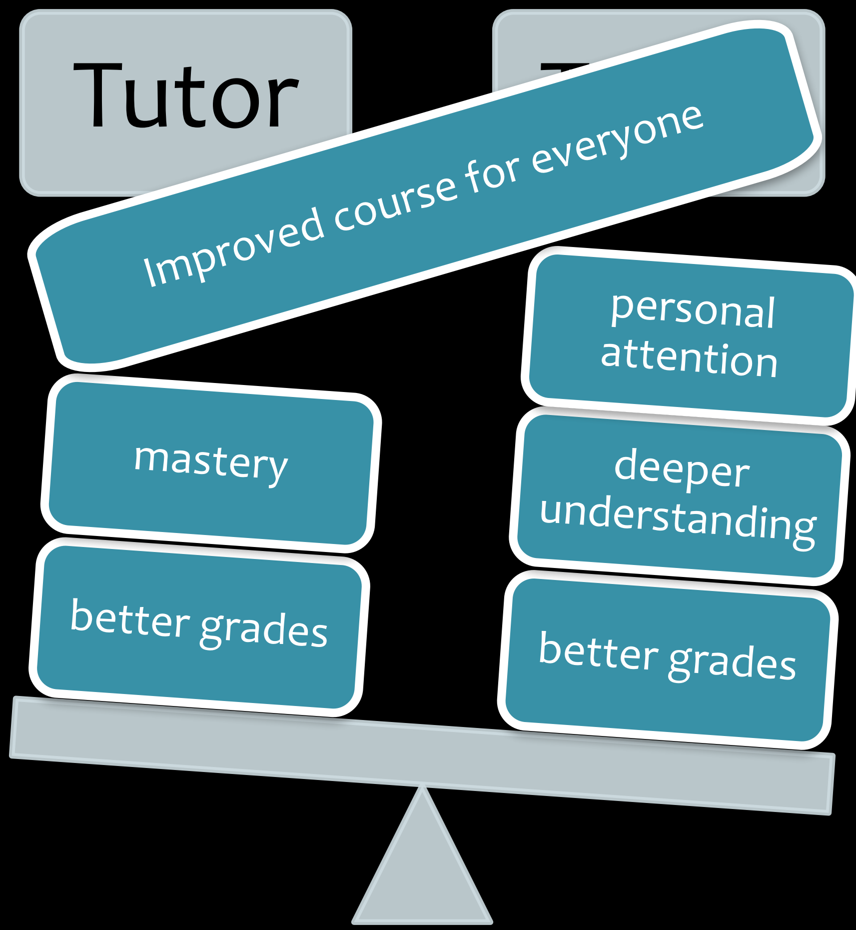
Bayesian Networks

Matt Gormley
Lecture 24
April 9, 2018

Reminders

- Homework 7: HMMs
  - Out: Wed, Apr 04
  - Due: Mon, Apr 16 at 11:59pm
- Schedule Changes
  - Lecture on Fri, Apr 13
  - Recitation on Mon, Apr 23

Peer Tutoring



HIDDEN MARKOV MODELS

Derivation of Forward Algorithm

Definition: $\alpha_t(k) \triangleq p(x_1, \ldots, x_t, y_t = k)$

Derivation:

Herein using "y_T" as shorthand for "$y_T = END$"

$$
\begin{aligned}
\alpha_T(END) &= p(x_1, \ldots, x_T, y_T = END) \\
&= p(x_1, \ldots, x_T \mid y_T)\, p(y_T) && \leftarrow \text{by def of joint} \\
&= p(x_T \mid y_T)\, p(x_1, \ldots, x_{T-1} \mid y_T)\, p(y_T) && \leftarrow \text{by cond. indep. of HMM} \\
&= p(x_T \mid y_T)\, p(x_1, \ldots, x_{T-1}, y_T) && \leftarrow \text{by def. of joint} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1}, y_{T-1}, y_T) && \leftarrow \text{by def. of marginal} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1}, y_T \mid y_{T-1})\, p(y_{T-1}) && \leftarrow \text{by def. of joint} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1} \mid y_{T-1})\, p(y_T \mid y_{T-1})\, p(y_{T-1}) && \leftarrow \text{by cond. indep. of HMM} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} p(x_1, \ldots, x_{T-1}, y_{T-1})\, p(y_T \mid y_{T-1}) && \leftarrow \text{by def. of joint} \\
&= p(x_T \mid y_T) \sum_{y_{T-1}} \alpha_{T-1}(y_{T-1})\, p(y_T \mid y_{T-1}) && \leftarrow \text{by def. of } \alpha_t(k)
\end{aligned}
$$

Forward-Backward Algorithm

<u>Define</u>: $\alpha_t(k) \triangleq p(x_1, \ldots, x_t, y_t = k)$

$\beta_t(k) \triangleq p(x_{t+1}, \ldots, x_T \mid y_t = k)$

<u>Assume</u> $y_0 = START$

$y_{T+1} = END$

① Initialize $\alpha_0(START) = 1 \qquad \alpha_0(k) = 0 \quad \forall k \neq START$

$\beta_{T+1}(END) = 1 \qquad \beta_{T+1}(k) = 0 \quad \forall k \neq END$

the alphas include the emission probabilities so we don't multiply them in separately

② For $t = 1, \ldots, T$:

For $k = 1, \ldots, K$:

$$\alpha_t(k) = p(x_t \mid y_t = k) \sum_{j=1}^{K} \alpha_{t-1}(j)\, p(y_t = k \mid y_{t-1} = j)$$

③ For $t = T, \ldots, T$:

For $k = 1, \ldots, K$:

$$\beta_t(k) = \sum_{j=1}^{K} p(x_{t+1} \mid y_{t+1} = j)\, \beta_{t+1}(j)\, p(y_{t+1} = j \mid y_t = k)$$

④ Compute $p(\vec{x}) = \alpha_{T+1}(END)$ [Evaluation]

⑤ Compute $p(y_t = k \mid \vec{x}) = \dfrac{\alpha_t(k)\, \beta_t(k)}{p(\vec{x})}$ [Marginals]

Viterbi Algorithm

Define:

$$\omega_t(k) \triangleq \max_{y_1,\dots,y_{t-1}} p(x_1,\dots,x_t, y_1,\dots,y_{t-1}, y_t = k)$$

"backpointers"

$$b_t(k) \triangleq \operatorname{argmax}_{y_1,\dots,y_{t-1}} p(x_1,\dots,x_t, y_1,\dots,y_{t-1}, y_t = k)$$

Assume $y_0 = \text{START}$

1. Initialize $\omega_0(\text{START}) = 1 \quad \omega_0(k) = 0 \ \forall k \neq \text{START}$

2. For $t = 1,\dots,T$:

   For $k = 1,\dots,K$:

$$\omega_t(k) = \max_{j \in \{1,\dots,K\}} p(x_t \mid y_t = k)\, \omega_{k-1}(j)\, p(y_t = k \mid y_{t-1} = j)$$

$$b_t(k) = \operatorname{argmax}_{j \in \{1,\dots,K\}} p(x_t \mid y_t = k)\, \omega_{k-1}(j)\, p(y_t = k \mid y_{t-1} = j)$$

3. Compute Most Probable Assignment

[Decoding]

$$\hat{y}_T = b_{T+1}(\text{END})$$

For $t = T-1,\dots,1$

$$\hat{y}_t = b_{t+1}(\hat{y}_{t+1})$$

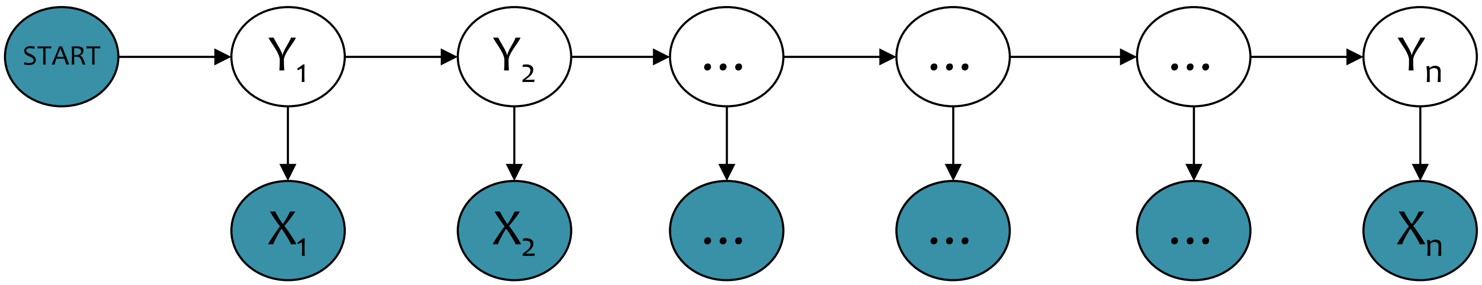
follow the "backpointers"

Inference in HMMs

What is the **computational complexity** of inference for HMMs?

- The **naïve** (brute force) computations for *Evaluation*, *Decoding*, and *Marginals* take **exponential time**, $O(K^T)$
- The **forward-backward** algorithm and **Viterbi** algorithm run in **polynomial time**, $O(T*K^2)$
  - Thanks to dynamic programming!

Shortcomings of Hidden Markov Models



- HMM models capture dependences between each state and only its corresponding observation
  - NLP example: In a sentence segmentation task, each segmental state may depend not just on a single word (and the adjacent segmental stages), but also on the (non-local) features of the whole line such as line length, indentation, amount of white space, etc.
- Mismatch between learning objective function and prediction objective function
  - HMM learns a joint distribution of states and observations P($\mathbf{Y}$, $\mathbf{X}$), but in a prediction task, we need the conditional probability P($\mathbf{Y}|\mathbf{X}$)

MBR DECODING

Inference for HMMs

Four

– ~~Three~~ Inference Problems for an HMM

1. Evaluation: Compute the probability of a given sequence of observations
2. Viterbi Decoding: Find the most-likely sequence of hidden states, given a sequence of observations
3. Marginals: Compute the marginal distribution for a hidden state, given a sequence of observations
4. MBR Decoding: Find the lowest loss sequence of hidden states, given a sequence of observations (Viterbi decoding is a special case)

Minimum Bayes Risk Decoding

- Suppose we given a loss function $l(\mathbf{y}', \mathbf{y})$ and are asked for a single tagging
- How should we choose just one from our probability distribution $p(\mathbf{y}|\mathbf{x})$?
- A minimum Bayes risk (MBR) decoder $h(\mathbf{x})$ returns the variable assignment with minimum **expected** loss under the model's distribution

$$\begin{aligned} h_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}} \; \mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})}[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y})] \\ &= \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}} \; \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) \end{aligned}$$

Minimum Bayes Risk Decoding

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{y}}}{\text{argmin}} \ \mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})}[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y})]$$

Consider some example loss functions:

The ***0-1* loss function** returns *1* only if the two assignments are identical and *0* otherwise:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = 1 - \mathbb{I}(\hat{\boldsymbol{y}}, \boldsymbol{y})$$

The MBR decoder is:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{y}}}{\text{argmin}} \ \sum_{\boldsymbol{y}} p_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})(1 - \mathbb{I}(\hat{\boldsymbol{y}}, \boldsymbol{y}))$$

$$= \underset{\hat{\boldsymbol{y}}}{\text{argmax}} \ p_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}} \mid \boldsymbol{x})$$

which is exactly the Viterbi decoding problem!

Minimum Bayes Risk Decoding

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underset{\hat{\boldsymbol{y}}}{\operatorname{argmin}} \; \mathbb{E}_{\boldsymbol{y} \sim p_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x})}[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y})]$$

Consider some example loss functions:

The **Hamming loss** corresponds to accuracy and returns the number of incorrect variable assignments:

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sum_{i=1}^{V} (1 - \mathbb{I}(\hat{y}_i, y_i))$$

The MBR decoder is:

$$\hat{y}_i = h_{\boldsymbol{\theta}}(\boldsymbol{x})_i = \underset{\hat{y}_i}{\operatorname{argmax}} \; p_{\boldsymbol{\theta}}(\hat{y}_i \mid \boldsymbol{x})$$

This decomposes across variables and requires the variable marginals.

BAYESIAN NETWORKS

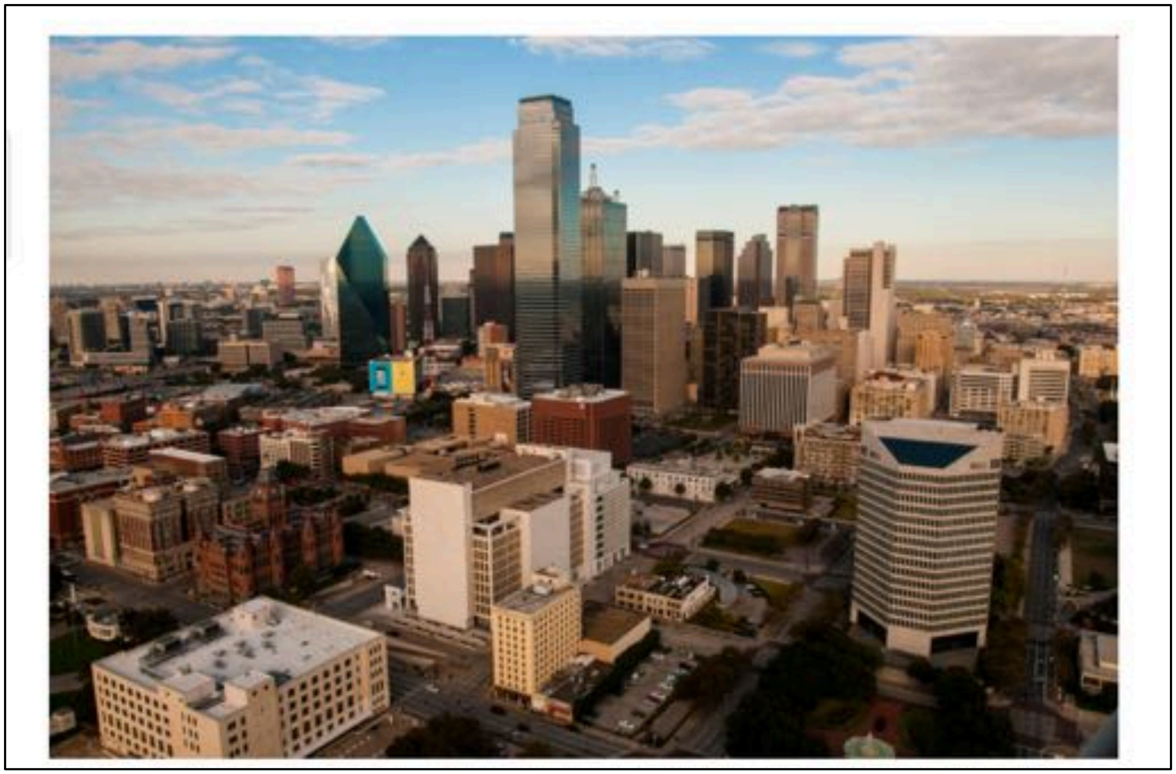
Bayes Nets Outline

- **Motivation**
  - Structured Prediction
- **Background**
  - Conditional Independence
  - Chain Rule of Probability
- **Directed Graphical Models**
  - Writing Joint Distributions
  - Definition: Bayesian Network
  - Qualitative Specification
  - Quantitative Specification
  - Familiar Models as Bayes Nets
- **Conditional Independence in Bayes Nets**
  - Three case studies
  - D-separation
  - Markov blanket
- **Learning**
  - Fully Observed Bayes Net
  - (Partially Observed Bayes Net)
- **Inference**
  - Background: Marginal Probability
  - Sampling directly from the joint distribution
  - Gibbs Sampling

Bayesian Networks

DIRECTED GRAPHICAL MODELS

Example: Tornado Alarms



1. Imagine that you work at the 911 call center in Dallas
2. You receive six calls informing you that the Emergency Weather Sirens are going off
3. What do you conclude?

Figure from https://www.nytimes.com/2017/04/08/us/dallas-emergency-sirens-hacking.html

Example: Tornado Alarms

Hacking Attack Woke Up Dallas With Emergency Sirens, Officials Say

By ELI ROSENBERG and MAYA SALAM APRIL 8, 2017

Warning sirens in Dallas, meant to alert the public to emergencies like severe weather, started sounding around 11:40 p.m. Friday, and were not shut off until 1:20 a.m. Rex C. Curry for The New York Times

1. Imagine that you work at the 911 call center in Dallas
2. You receive six calls informing you that the Emergency Weather Sirens are going off
3. What do you conclude?

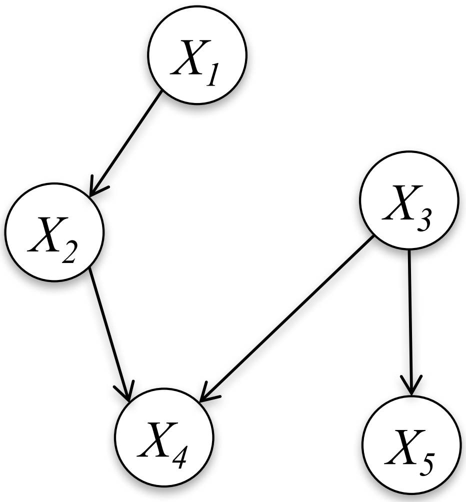
Figure from https://www.nytimes.com/2017/04/08/us/dallas-emergency-sirens-hacking.html

Directed Graphical Models (Bayes Nets)

Whiteboard

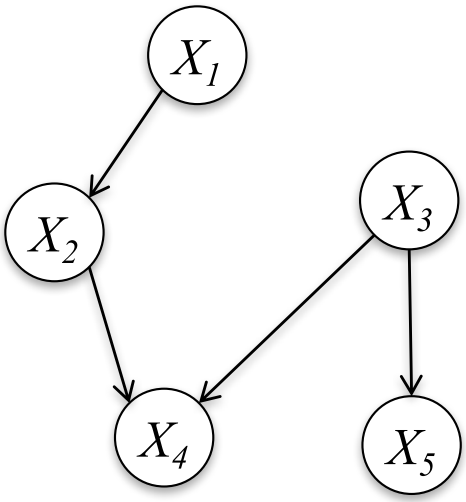
- Example: Tornado Alarms
- Writing Joint Distributions
  - Idea #1: Giant Table
  - Idea #2: Rewrite using chain rule
  - Idea #3: Assume full independence
  - Idea #4: Drop variables from RHS of conditionals
- Definition: Bayesian Network

Bayesian Network



$$
\begin{aligned}
p(X_1, X_2, X_3, X_4, X_5) = \\
p(X_5|X_3)p(X_4|X_2, X_3) \\
p(X_3)p(X_2|X_1)p(X_1)
\end{aligned}
$$

Bayesian Network



Definition:

$$P(X_1 \ldots X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$

- A Bayesian Network is a **directed graphical model**
- It consists of a graph **G** and the conditional probabilities **P**
- These two parts full specify the distribution:
  - Qualitative Specification: **G**
  - Quantitative Specification: **P**

Qualitative Specification

- Where does the qualitative specification come from?

  - Prior knowledge of causal relationships
  - Prior knowledge of modular relationships
  - Assessment from experts
  - Learning from data (i.e. structure learning)
  - We simply link a certain architecture (e.g. a layered graph)
  - …

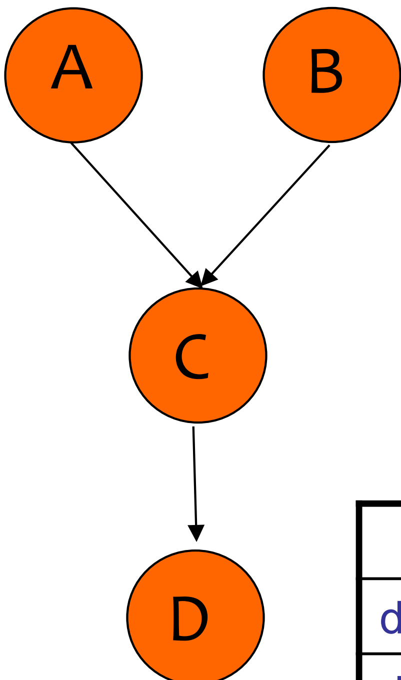
Quantitative Specification

Example: Conditional probability tables (CPTs) for discrete random variables

a^0	0.75
a^1	0.25

b^0	0.33
b^1	0.67

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$



	a^0b^0	a^0b^1	a^1b^0	a^1b^1
c^0	0.45	1	0.9	0.7
c^1	0.55	0	0.1	0.3

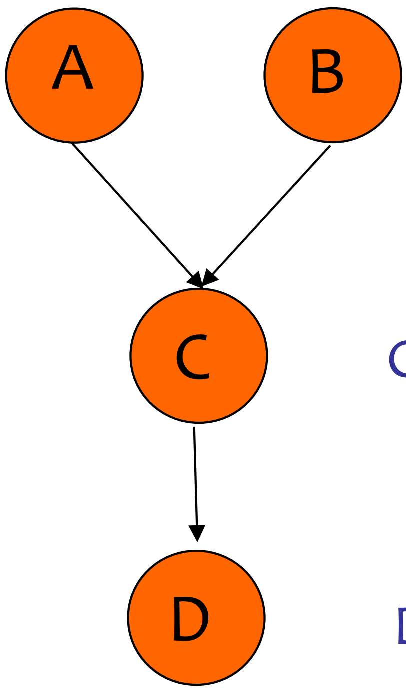
	c^0	c^1
d^0	0.3	0.5
d^1	07	0.5

Quantitative Specification

Example: Conditional probability density functions (CPDs) for continuous random variables

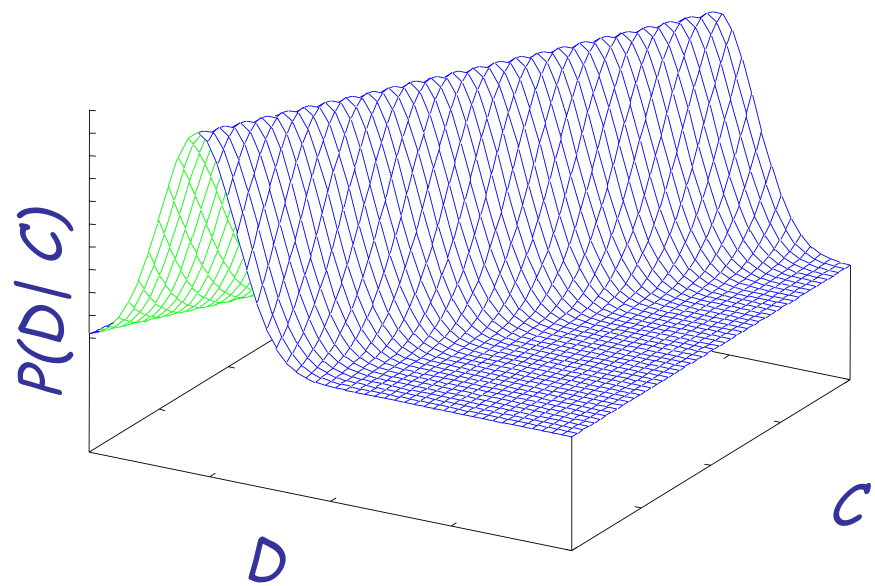
$A \sim N(\mu_a, \Sigma_a)$ $B \sim N(\mu_b, \Sigma_b)$

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$



$C \sim N(A+B, \Sigma_c)$

$D \sim N(\mu_d + C, \Sigma_d)$



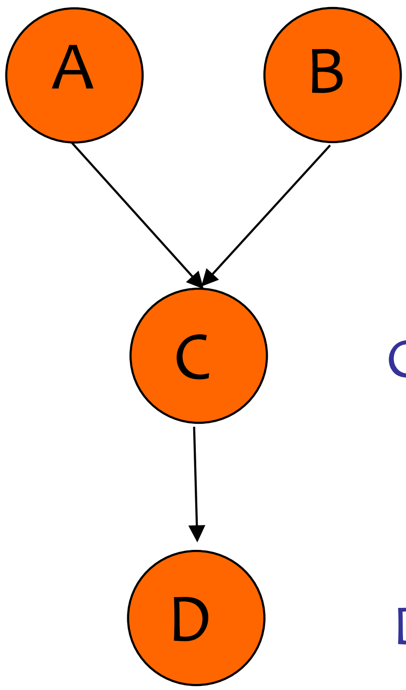
Quantitative Specification

Example: Combination of CPTs and CPDs for a mix of discrete and continuous variables

a^0	0.75
a^1	0.25

b^0	0.33
b^1	0.67

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$



$C \sim N(A+B, \Sigma_c)$

$D \sim N(\mu_d + C, \Sigma_d)$

Directed Graphical Models (Bayes Nets)

Whiteboard

- Observed Variables in Graphical Model
- Familiar Models as Bayes Nets
  - Bernoulli Naïve Bayes
  - Gaussian Naïve Bayes
  - Gaussian Mixture Model (GMM)
  - Gaussian Discriminant Analysis
  - Logistic Regression
  - Linear Regression
  - 1D Gaussian

GRAPHICAL MODELS: DETERMINING CONDITIONAL INDEPENDENCIES

Slide from William Cohen

What Independencies does a Bayes Net Model?

- In order for a Bayesian network to model a probability distribution, the following must be true:

  Each variable is conditionally independent of all its non-descendants in the graph given the value of all its parents.

- This follows from

$$P(X_1 \ldots X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$$

$$= \prod_{i=1}^{n} P(X_i \mid X_1 \ldots X_{i-1})$$

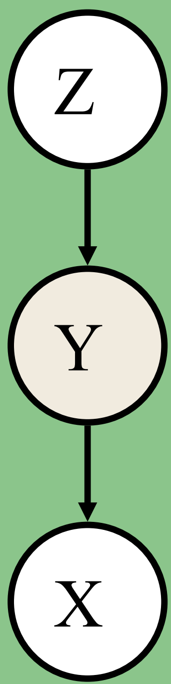
- But what else does it imply?

Slide from William Cohen

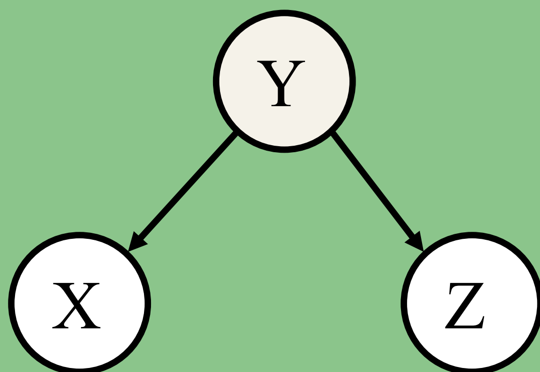
What Independencies does a Bayes Net Model?

Three cases of interest…

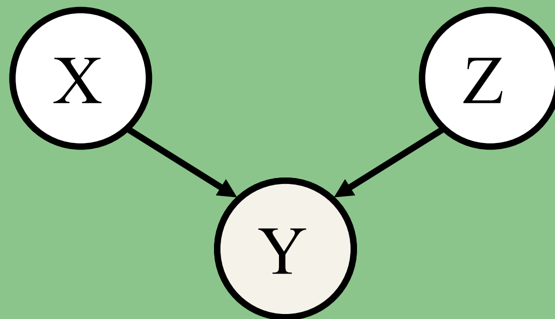
Cascade



Common Parent



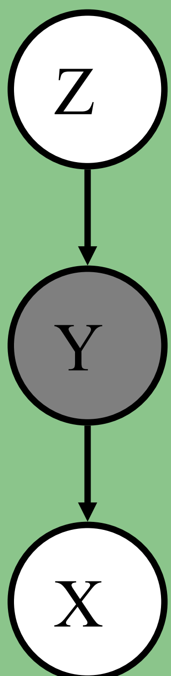
V-Structure



What Independencies does a Bayes Net Model?

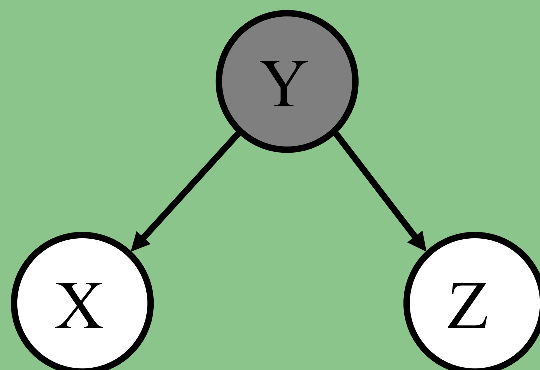
Three cases of interest…

Cascade



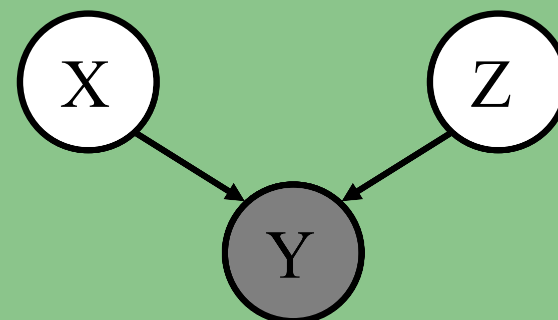
$$X \perp\!\!\!\perp Z \mid Y$$

Common Parent



$$X \perp\!\!\!\perp Z \mid Y$$

V-Structure



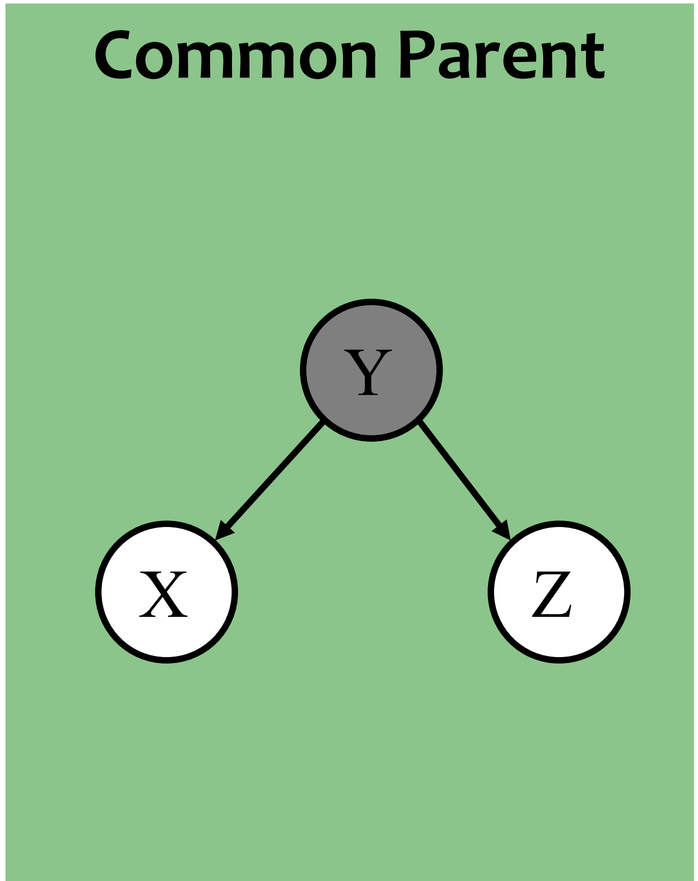
$$X \not\!\perp\!\!\!\perp Z \mid Y$$

Knowing Y
decouples X and Z

Knowing Y
couples X and Z

Whiteboard

Proof of conditional independence

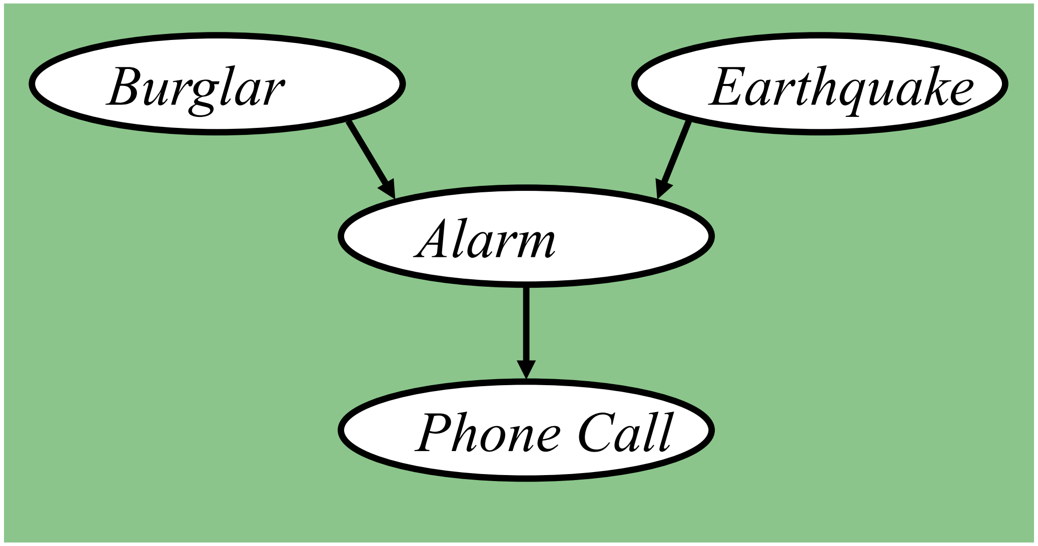


$$X \perp\!\!\!\perp Z \mid Y$$

(The other two cases can be shown just as easily.)

The "Burglar Alarm" example

- Your house has a twitchy burglar alarm that is also sometimes triggered by earthquakes.
- Earth arguably doesn't care whether your house is currently being burgled
- While you are on vacation, one of your neighbors calls and tells you your home's burglar alarm is ringing. Uh oh!



Quiz: True or False?

$$Burglar \perp\!\!\!\perp Earthquake \mid PhoneCall$$

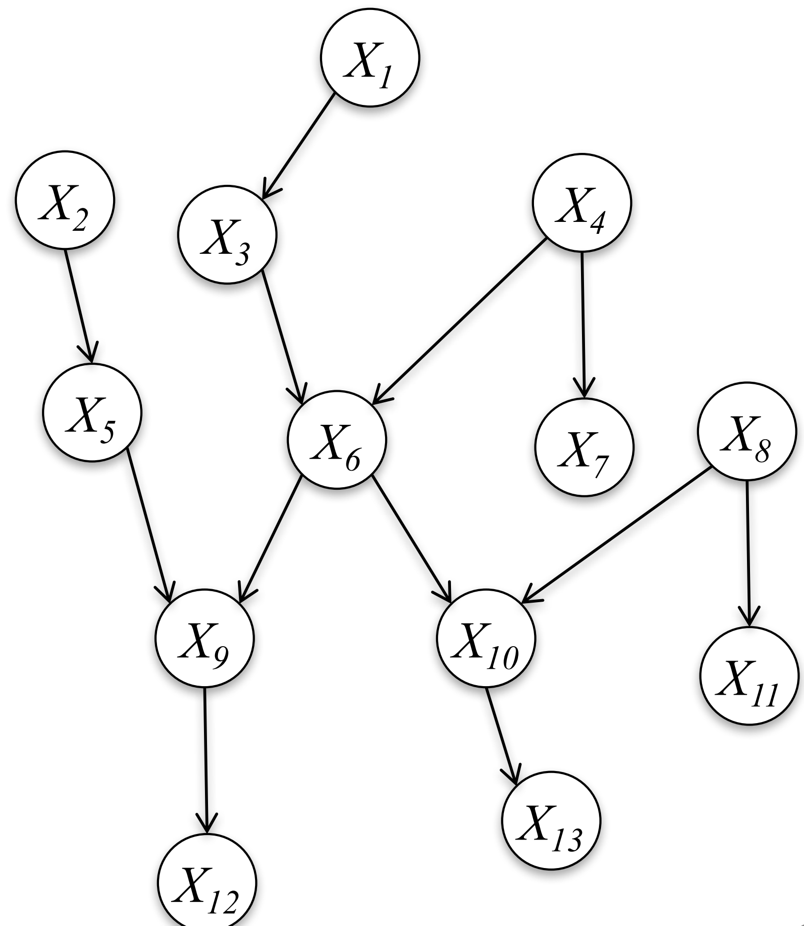
Slide from William Cohen

Markov Blanket

Def: the **co-parents** of a node are the parents of its children

Def: the **Markov Blanket** of a node is the set containing the node's parents, children, and co-parents.

Thm: a node is **conditionally independent** of every other node in the graph given its **Markov blanket**



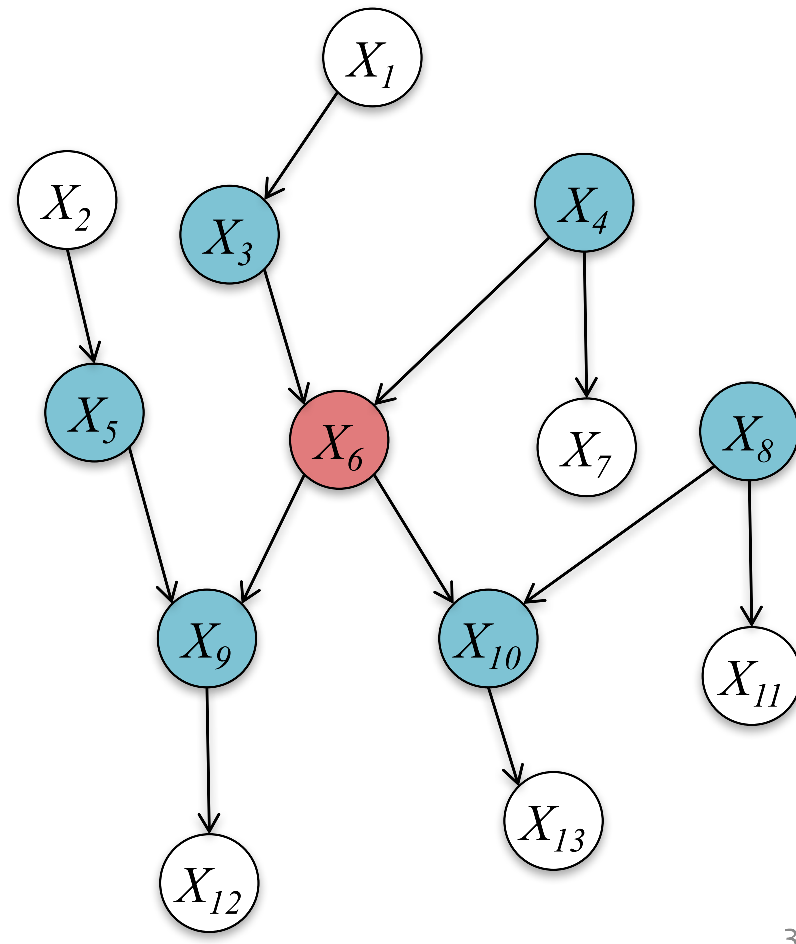
Markov Blanket

Def: the **co-parents** of a node are the parents of its children

Def: the **Markov Blanket** of a node is the set containing the node's parents, children, and co-parents.

Theorem: a node is **conditionally independent** of every other node in the graph given its **Markov blanket**

Example: The Markov Blanket of X_6 is $\{X_3, X_4, X_5, X_8, X_9, X_{10}\}$



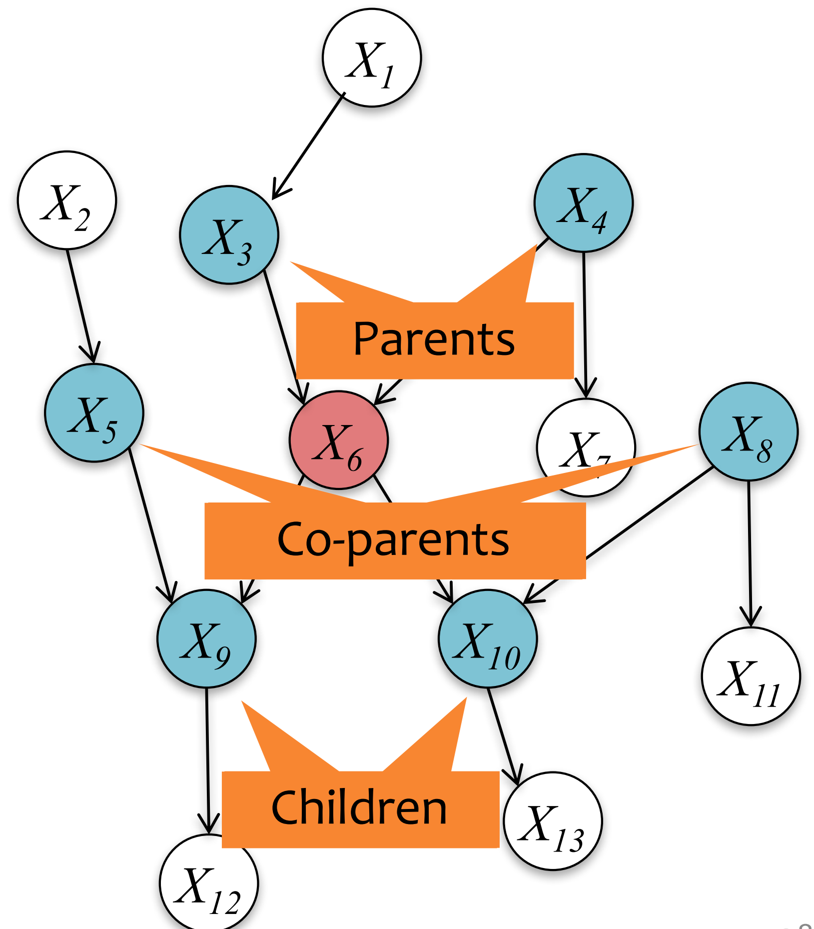
Markov Blanket

Def: the **co-parents** of a node are the parents of its children

Def: the **Markov Blanket** of a node is the set containing the node's parents, children, and co-parents.

Theorem: a node is **conditionally independent** of every other node in the graph given its **Markov blanket**

Example: The Markov Blanket of X_6 is $\{X_3, X_4, X_5, X_8, X_9, X_{10}\}$



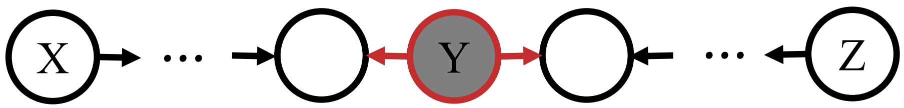
D-Separation

If variables X and Z are **d-separated** given a **set** of variables E
Then X and Z are **conditionally independent** given the **set** E

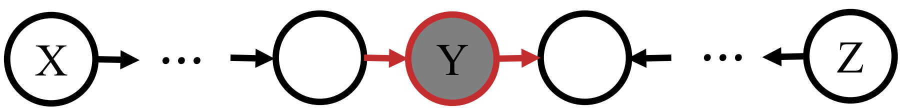
Definition #1:
Variables X and Z are **d-separated** given a **set** of evidence variables E iff every path from X to Z is "blocked".

A path is "blocked" whenever:

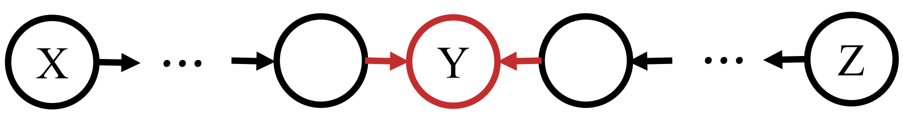
1. $\exists$ Y on path s.t. Y $\in$ E and Y is a "common parent"



2. $\exists$ Y on path s.t. Y $\in$ E and Y is in a "cascade"



3. $\exists$ Y on path s.t. {Y, descendants(Y)} $\notin$ E and Y is in a "v-structure"



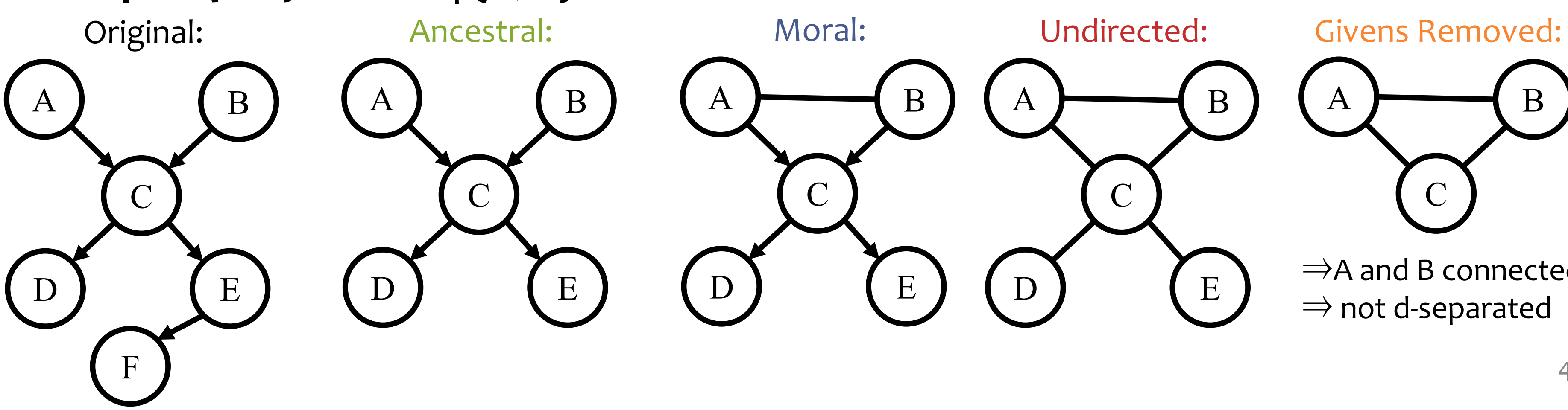
D-Separation

If variables X and Z are **d-separated** given a **set** of variables E
Then X and Z are **conditionally independent** given the **set** E

Definition #2:
Variables X and Z are **d-separated** given a **set** of evidence variables E iff there does **not** exist a path in the **undirected** **ancestral** **moral** graph **with E removed**.

1. **Ancestral graph**: keep only X, Z, E and their ancestors
2. **Moral graph**: add undirected edge between all pairs of each node's parents
3. **Undirected graph**: convert all directed edges to undirected
4. Givens Removed: delete any nodes in E

Example Query: $A \perp\!\!\!\perp B \mid \{D, E\}$



⇒A and B connected
⇒ not d-separated

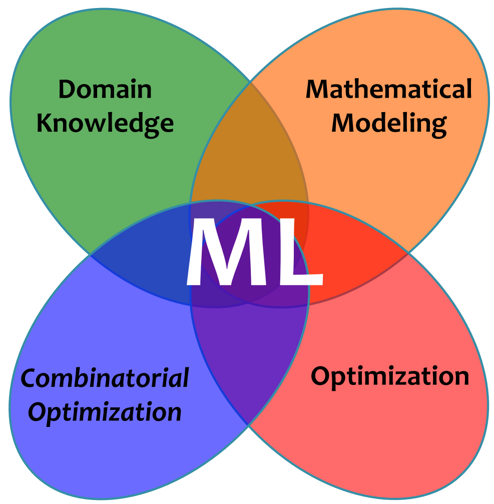
SUPERVISED LEARNING FOR BAYES NETS

Machine Learning

The **data** inspires the structures we want to predict

Our **model** defines a score for each structure

It also tells us what to optimize



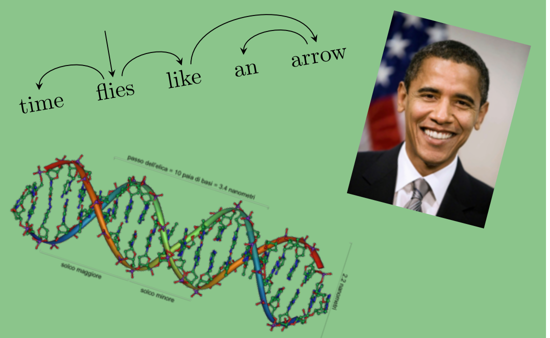
Learning tunes the parameters of the model

Inference finds {best structure, marginals, partition function} for a new observation

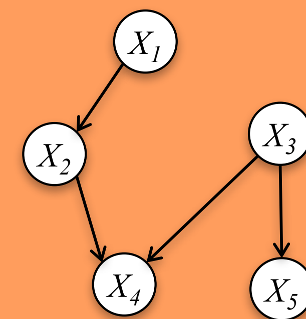
(**Inference** is usually called as a subroutine in learning)

Machine Learning

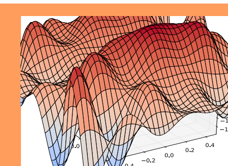
Data



Model



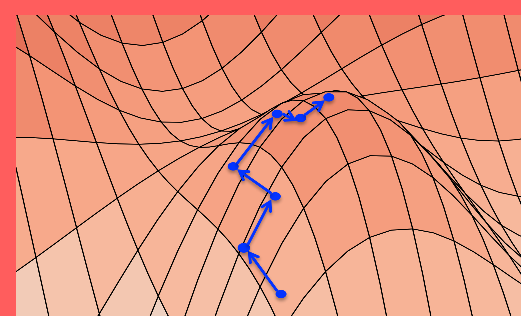
Objective



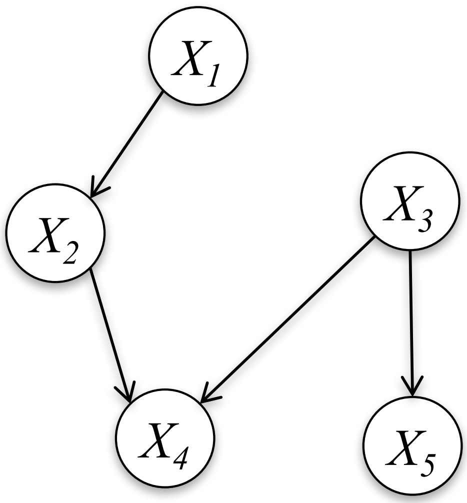
Inference

(**Inference** is usually called as a subroutine in learning)

Learning

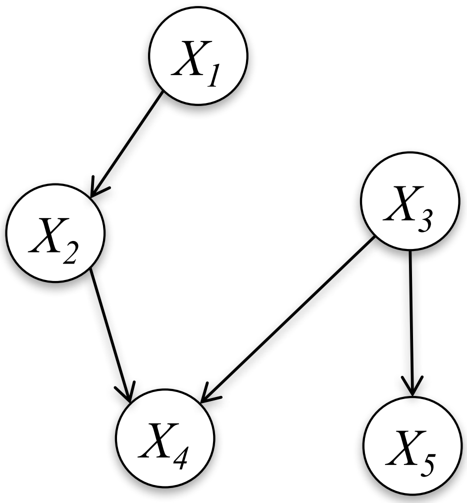


Learning Fully Observed BNs



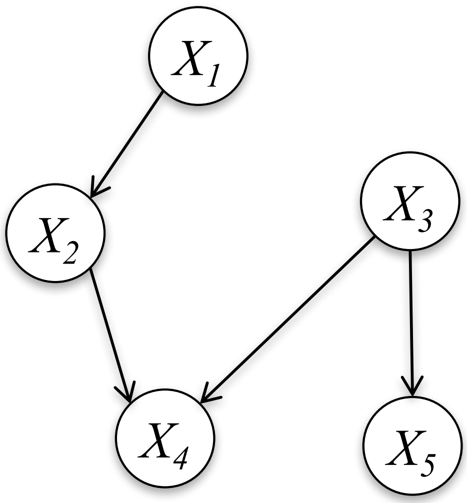
$$
\begin{aligned}
p(X_1, X_2, X_3, X_4, X_5) = \\
p(X_5|X_3)p(X_4|X_2, X_3) \\
p(X_3)p(X_2|X_1)p(X_1)
\end{aligned}
$$

Learning Fully Observed BNs



$$p(X_1, X_2, X_3, X_4, X_5) =$$
$$p(X_5|X_3)p(X_4|X_2, X_3)$$
$$p(X_3)p(X_2|X_1)p(X_1)$$

Learning Fully Observed BNs



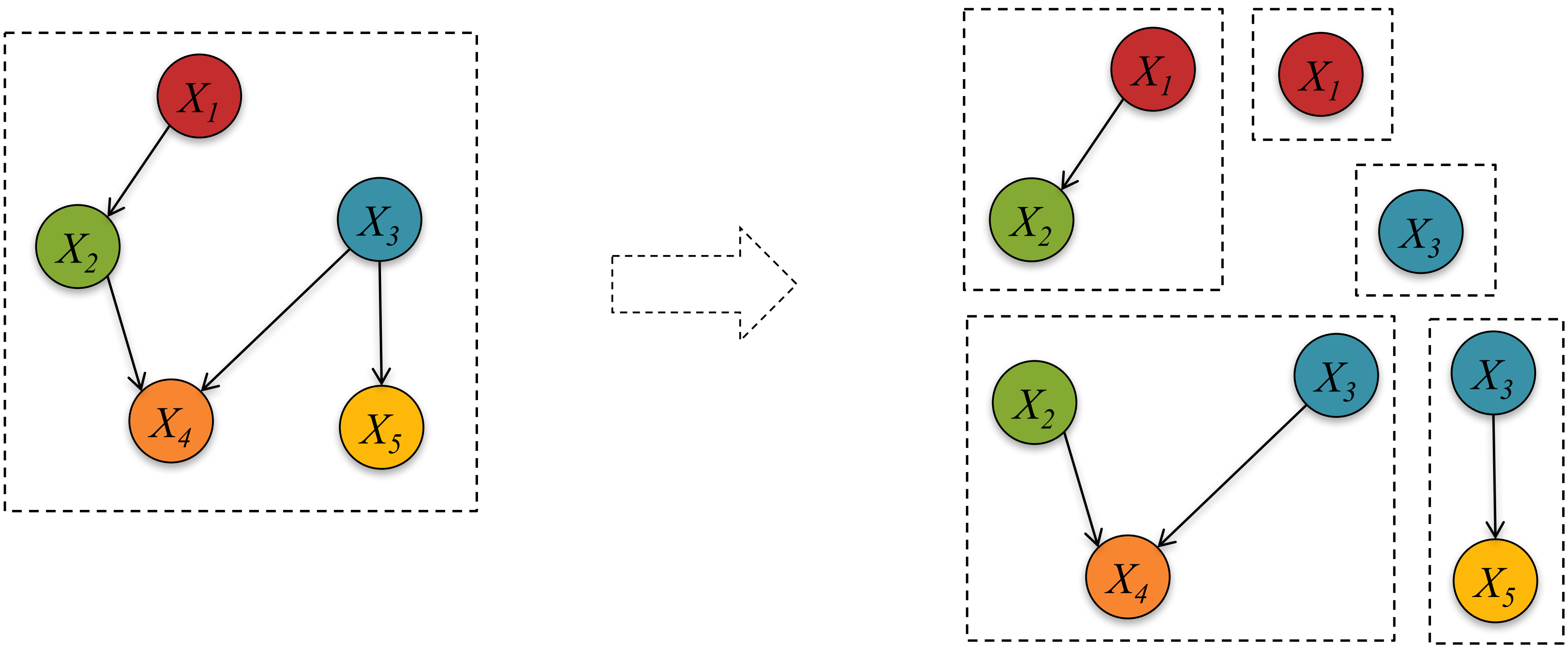
$$p(X_1, X_2, X_3, X_4, X_5) =$$
$$p(X_5|X_3)p(X_4|X_2, X_3)$$
$$p(X_3)p(X_2|X_1)p(X_1)$$

How do we learn these conditional and marginal distributions for a Bayes Net?

Learning Fully Observed BNs

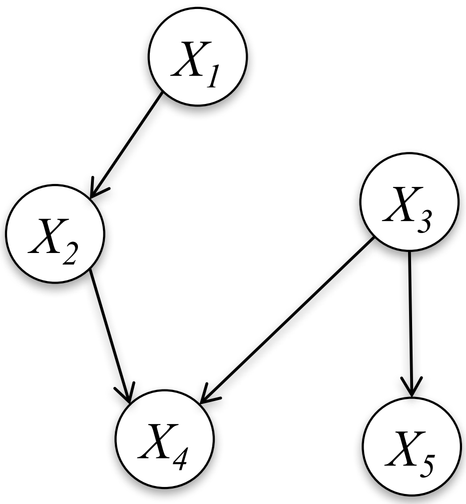
Learning this fully observed Bayesian Network is **equivalent** to learning five (small / simple) independent networks from the same data

$$p(X_1, X_2, X_3, X_4, X_5) = \\ p(X_5|X_3)p(X_4|X_2, X_3) \\ p(X_3)p(X_2|X_1)p(X_1)$$



Learning Fully Observed BNs

How do we **learn** these conditional and marginal distributions for a Bayes Net?



$$\begin{aligned}\boldsymbol{\theta}^* &= \operatorname*{argmax}_{\boldsymbol{\theta}} \log p(X_1, X_2, X_3, X_4, X_5) \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}} \log p(X_5 | X_3, \theta_5) + \log p(X_4 | X_2, X_3, \theta_4) \\ &\qquad + \log p(X_3 | \theta_3) + \log p(X_2 | X_1, \theta_2) \\ &\qquad + \log p(X_1 | \theta_1)\end{aligned}$$

$$\theta_1^* = \operatorname*{argmax}_{\theta_1} \log p(X_1 | \theta_1)$$

$$\theta_2^* = \operatorname*{argmax}_{\theta_2} \log p(X_2 | X_1, \theta_2)$$

$$\theta_3^* = \operatorname*{argmax}_{\theta_3} \log p(X_3 | \theta_3)$$

$$\theta_4^* = \operatorname*{argmax}_{\theta_4} \log p(X_4 | X_2, X_3, \theta_4)$$

$$\theta_5^* = \operatorname*{argmax}_{\theta_5} \log p(X_5 | X_3, \theta_5)$$

Learning Fully Observed BNs

Whiteboard

- Example: Learning for Tornado Alarms

INFERENCE FOR BAYESIAN NETWORKS

A Few Problems for Bayes Nets

Suppose we already have the parameters of a Bayesian Network…

1. How do we compute the probability of a specific assignment to the variables?
   P(T=t, H=h, A=a, C=c)

2. How do we draw a sample from the joint distribution?
   t,h,a,c ~ P(T, H, A, C)

3. How do we compute marginal probabilities?
   P(A) = …

4. How do we draw samples from a conditional distribution?
   t,h,a ~ P(T, H, A | C = c)

5. How do we compute conditional marginal probabilities?
   P(H | C = c) = …

Can we use samples?

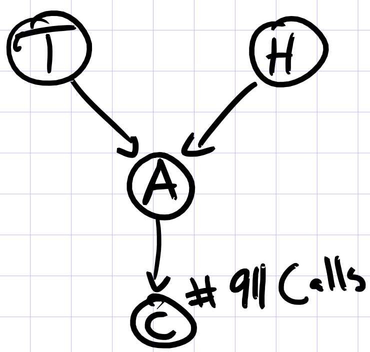
Inference for Bayes Nets

Whiteboard

– Background: Marginal Probability
– Sampling from a joint distribution
– Gibbs Sampling

Sampling from a Joint Distribution

Ex: Tornado



$T \sim \text{Bernoulli}(\tau)$ $\quad \tau = 1/2$

$H \sim \text{Bernoulli}(\eta)$ $\quad \eta = 1/3$

$A \sim \text{Bernoulli}(\alpha_{H,T})$ $\quad \alpha = \begin{array}{c|cc} & T=0 & T=1 \\ \hline H=0 & 0 & 1/2 \\ H=1 & 1/2 & 1 \end{array}$

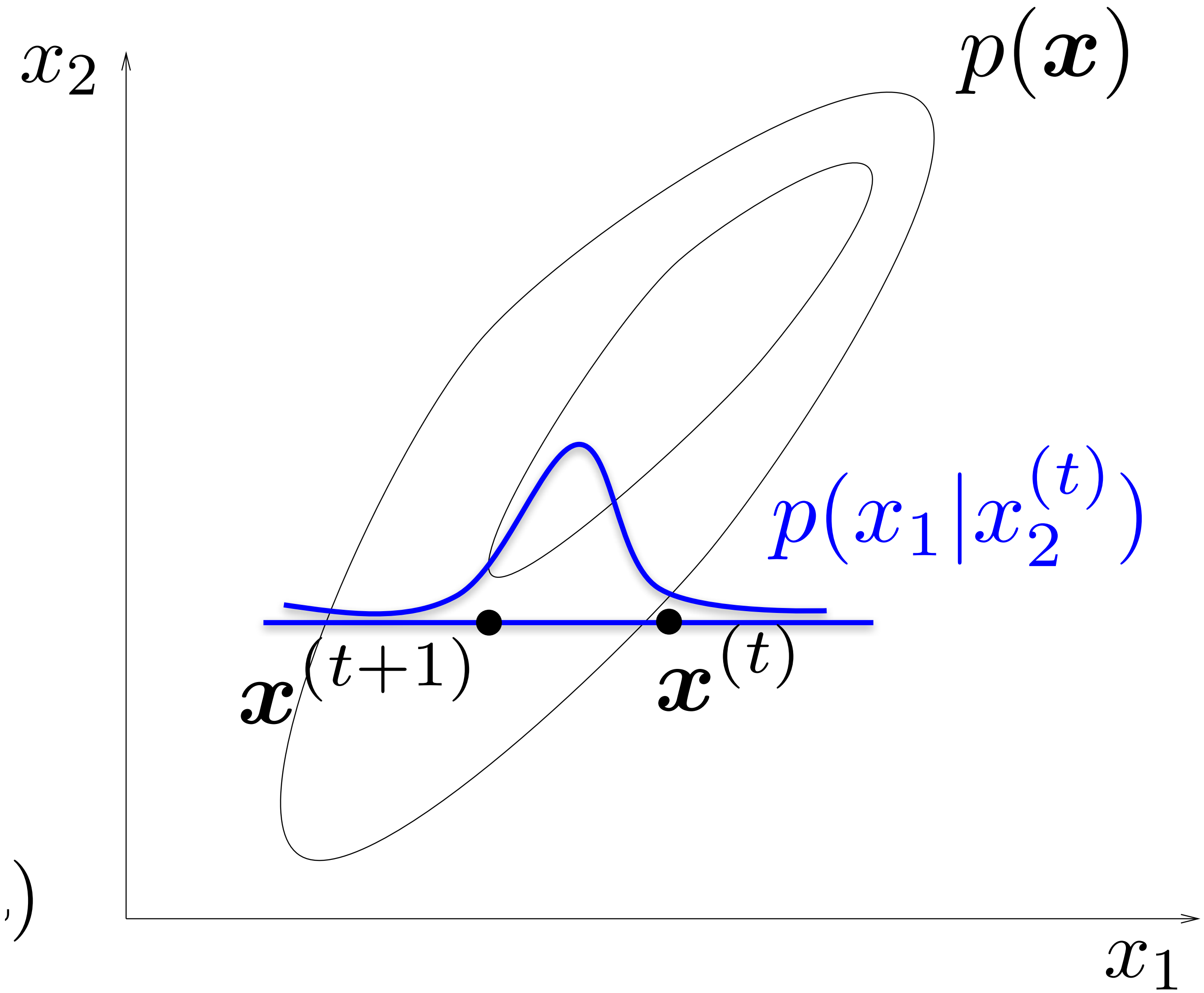
$C \sim \text{Unif}(\{1,\dots,63\}) + A * \text{Unif}(\{1,\dots,63\})$

← integer

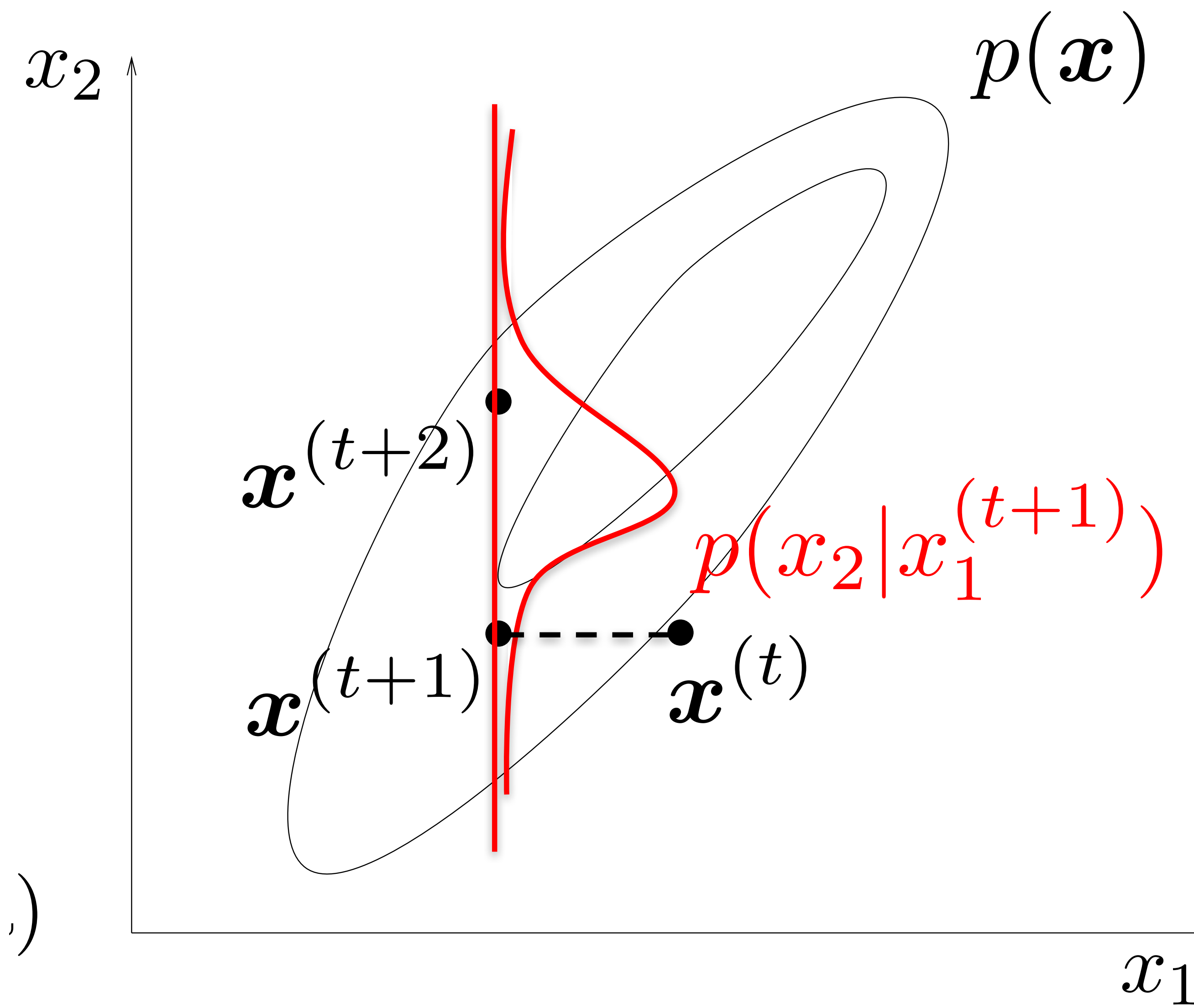
T	H	A	C

We can use these samples to estimate many different probabilities!

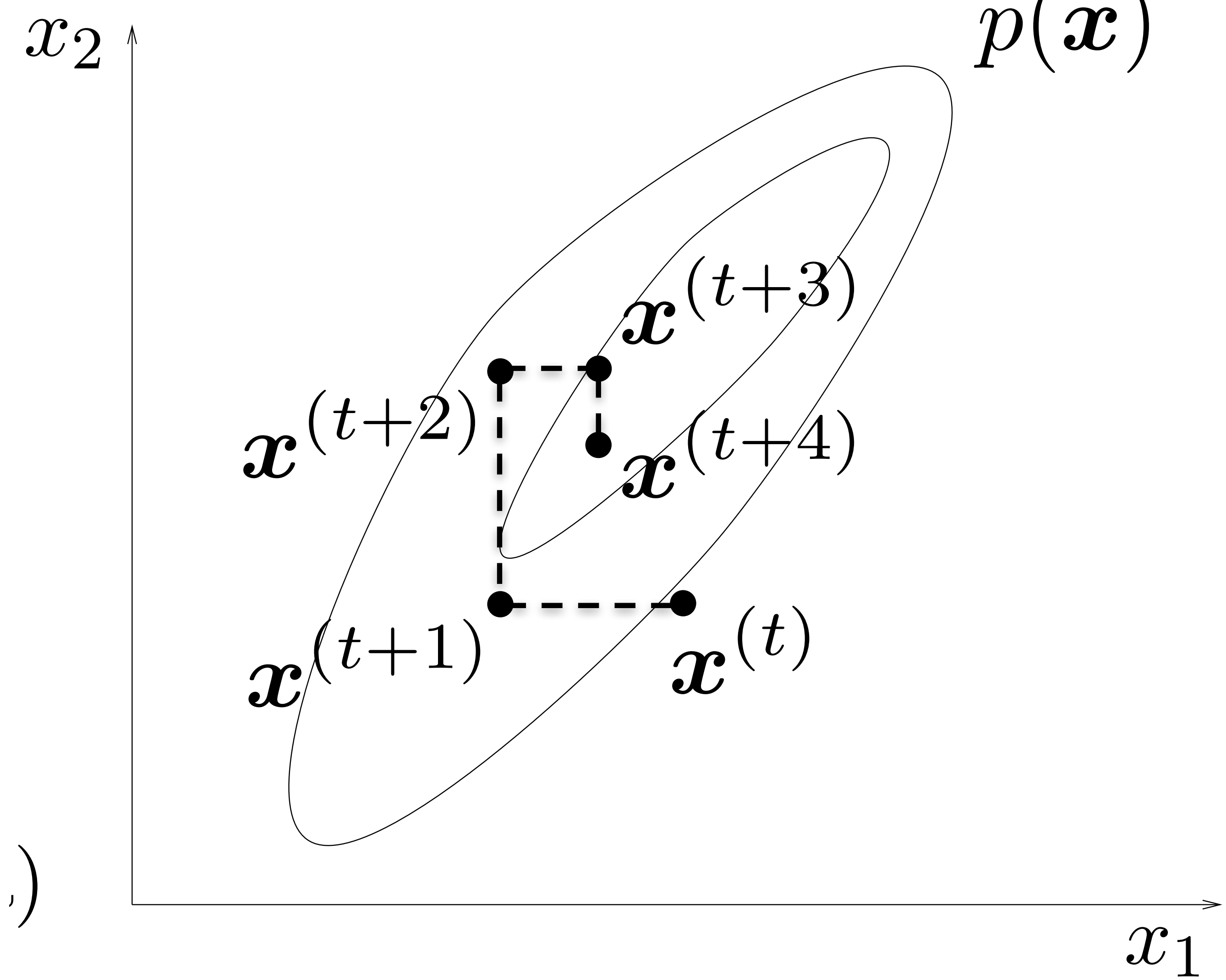
Gibbs Sampling



Gibbs Sampling



Gibbs Sampling



Gibbs Sampling

Question:
How do we draw samples from a conditional distribution?
$y_1, y_2, \ldots, y_J \sim p(y_1, y_2, \ldots, y_J \mid x_1, x_2, \ldots, x_J)$

(Approximate) Solution:

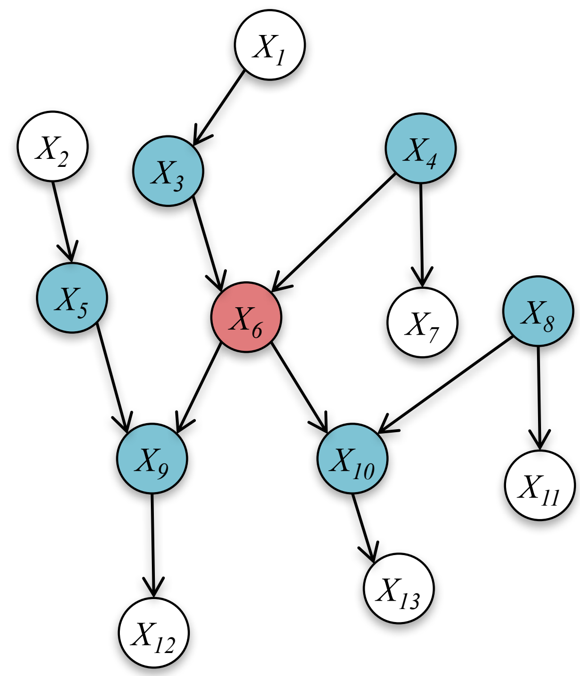
– Initialize $y_1^{(0)}, y_2^{(0)}, \ldots, y_J^{(0)}$ to arbitrary values
– For t = 1, 2, … :
  - $y_1^{(t+1)} \sim p(y_1 \mid y_2^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J)$
  - $y_2^{(t+1)} \sim p(y_2 \mid y_1^{(t+1)}, y_3^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J)$
  - $y_3^{(t+1)} \sim p(y_3 \mid y_1^{(t+1)}, y_2^{(t+1)}, y_4^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J)$
  - …
  - $y_J^{(t+1)} \sim p(y_J \mid y_1^{(t+1)}, y_2^{(t+1)}, \ldots, y_{J-1}^{(t+1)}, x_1, x_2, \ldots, x_J)$

Properties:

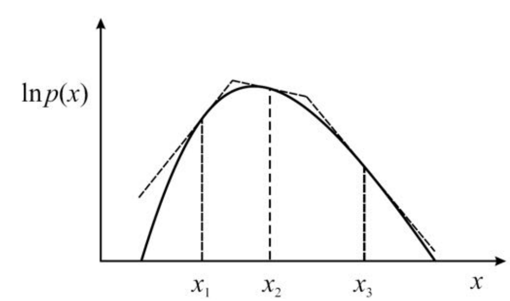
– This will eventually yield samples from $p(y_1, y_2, \ldots, y_J \mid x_1, x_2, \ldots, x_J)$
– But it might take a long time -- just like other Markov Chain Monte Carlo methods

Gibbs Sampling

Full conditionals only need to condition on the **Markov Blanket**



- Must be "easy" to sample from conditionals
- Many conditionals are log-concave and are amenable to adaptive rejection sampling



Learning Objectives

Bayesian Networks

You should be able to…

1. Identify the conditional independence assumptions given by a generative story or a specification of a joint distribution
2. Draw a Bayesian network given a set of conditional independence assumptions
3. Define the joint distribution specified by a Bayesian network
4. User domain knowledge to construct a (simple) Bayesian network for a real-world modeling problem
5. Depict familiar models as Bayesian networks
6. Use d-separation to prove the existence of conditional indenpendencies in a Bayesian network
7. Employ a Markov blanket to identify conditional independence assumptions of a graphical model
8. Develop a supervised learning algorithm for a Bayesian network
9. Use samples from a joint distribution to compute marginal probabilities
10. Sample from the joint distribution specified by a generative story
11. Implement a Gibbs sampler for a Bayesian network