

Lecture Slides

[Help Center](#)

Click on the lecture titles to download the annotated slides for each lecture, or click on the slides link next to each section label to download the combined slides for the whole section. For further reading, we have also provided relevant references to the [class textbook](#) next to each lecture.

Introduction and Overview ([combined slides](#))

[Welcome!](#)

[Overview and Motivation](#) Chapter 1.

[Distributions](#) Chapters 2.1.1 to 2.1.3.

[Factors](#). Chapter 4.2.1.

Bayesian Network Fundamentals ([combined slides](#))

[Semantics and Factorization](#) Chapters 3.2.1, 3.2.2. If you are unfamiliar with genetic inheritance, please watch this short [Khan Academy video](#) for some background.

[Reasoning Patterns](#). Chapter 3.2.1.

[Flow of Probabilistic Influence](#). Chapter 3.3.1.

[Conditional Independence](#). Chapters 2.1.4, 3.1.

[Independencies in Bayesian Networks](#). Chapter 3.2.2.

[Naive Bayes](#). Chapter 3.1.3.

[Application - Medical Diagnosis](#) Chapter 3.2: Box 3.D (p. 67).

Template Models ([combined slides](#))

[Overview](#). Chapter 6.1.

[Temporal Models - DBNs](#). Chapters 6.2, 6.3.

[Temporal Models - HMMs](#). Chapters 6.2, 6.3.

[Plate Models](#). Chapter 6.4.1.

Octave Tutorial

[Octave Tutorial Code](#)

Structured CPDs ([combined slides](#))

[Overview](#). Chapters 5.1, 5.2.

[Tree-Structured CPDs](#). Chapter 5.3.

[Independence of Causal Influence](#). Chapter 5.4.

[Continuous Variables](#). Chapter 5.5.

Markov Network Fundamentals ([combined slides](#))

[Pairwise Markov Networks](#). Chapter 4.1.

[General Gibbs Distribution](#). Chapter 4.2.2.

[Conditional Random Fields](#). Chapter 4.6.1.

[Independencies in Markov Networks](#). Chapter 4.3.1.

[I-Maps and Perfect Maps](#). Chapter 3.3.4.

[Log-Linear Models](#). Chapter 4.4, p. 125.

[Shared Features in Log-Linear Models](#). Chapter 4: Box 4.B (p. 112), Box 4.C (p. 126), Box 4.D (p. 127).

Representation Wrapup: Knowledge Engineering ([combined slides](#))

[Knowledge Engineering](#).

Variable Elimination ([combined slides](#))

[Conditional Probability Queries](#). Chapter 9.3.

[MAP Queries](#). Chapter 13.2.1.

[Variable Elimination Algorithm](#). Chapter 9.2.

[Variable Elimination Complexity](#). Chapter 9.4 through 9.4.2.3.

[VE - Graph Based Perspective](#). Chapter 9.4.

[Finding Elimination Orderings](#). Chapter 9.4.3.

Belief Propagation ([combined slides](#))

[Belief Propagation](#). Chapter 11.3.2

[Properties of Cluster Graphs](#). Chapter 11.3.2

Belief Propagation, Part 2 ([combined slides](#))

[Properties of Belief Propagation](#). Chapter 11.3.3

[Clique Tree Algorithm - Correctness](#). Chapter 10.2.1

[Clique Tree Algorithm - Computation](#). Chapters 10.2.2, 10.3.3.1

[Clique Trees and Independence](#). Chapter 10.1.2

[Clique Trees and VE](#). Chapter 10.4.1

[BP in Practice](#). Box 11.C

[Loopy BP and Message Decoding](#). Box 11.A

MAP Estimation Part 1 ([combined slides](#))

[MAP Exact Inference](#). Chapter 13.2.1

[Finding a MAP Assignment](#). Chapter 13.2.2

MAP Estimation Part 2 ([combined slides](#))

[Tractable MAP Problems](#). Chapter 13.6.

[Dual Decomposition - Intuition](#). Dual Decomposition is not in the textbook, but for further information you may refer to the original paper: [MRF Energy Minimization and Beyond via Dual Decomposition](#) N. Komodakis, N. Paragios and G. Tziritas

[Dual Decomposition - Algorithm](#).

Sampling Methods ([combined slides](#))

[Simple Sampling](#). Chapter 12.1.

[Markov Chain Monte Carlo](#) . Chapter 12.3 up to 12.3.2.2.

[Using a Markov Chain](#). Chapter 12.3.5.

[Gibbs Sampling](#). Review of Chapter 12.3.2 as applied to Gibbs Sampling.

[Metropolis Hastings Algorithm](#). Chapter 12.3.4.2.

Inference In Temporal Models, Summary ([combined slides](#))

[Inference in Temporal Models](#)

[Inference - Summary](#)

Decision Making ([combined slides](#))

[Maximum Expected Utility](#) Chapter 22.1.1, 23.2.104, 23.4.1-2, 23.5.1

[Utility Functions](#) Chapter 22.2.1-3, 22.3.2, 22.4.2

[Value of Perfect Information](#) Chapter 23.7.1-2

Learning: Parameter Estimation, Part 1 ([combined slides](#))

[Overview](#). Chapter 16.1 and Intro to Chapter 17

[Maximum Likelihood Estimation](#). Chapter 17.1

[Maximum Likelihood Estimation for Bayesian Networks](#). Chapter 17.2 through 17.2.1

[Bayesian Estimation](#). Chapter 17.3.2

[Bayesian Prediction](#). Chapter 17.4

[Bayesian Estimation for Bayesian Networks](#)

Learning: Parameter Estimation, Part 2 ([combined slides](#))

[Maximum Likelihood Estimation for Log-Linear Models](#). Chapter 20.1 - 20.2

[Maximum Likelihood Estimation for Conditional Random Fields](#). Chapter 20.1 - 20.2

[MAP Estimation for Markov Random Fields and Conditional Random Fields](#). Chapter 20.1 - 20.2

Structure Learning ([combined slides](#))

[Structure Learning: Overview](#) Chapter 18.1

[Likelihood Scores](#) Chapter 18.3.1

[BIC and Asymptotic Consistency](#) Chapter 18.3.5

[Bayesian Score](#) Chapter 18.3.2-18.3.4, 18.3.6, 18.3.7

[Learning Tree Structured Networks](#) Chapter 18.4.1

[Learning General Graphs: Heuristic Search](#) Chapter 18.4.3.1-2

[Learning General Graphs: Heuristic Search and Decomposability](#) Chapter 18.4.3.3

Learning With Incomplete Data ([combined slides](#))

[Learning With Incomplete Data - Overview](#) Chapter 19.1.3 and 19.1.4

[Expectation Maximization - Intro](#) Chapter 19.2.2

[Analysis of EM Algorithm](#) Chapter 19.2.2

[EM in Practice](#) Box 19.B.

[Latent Variables](#)

Learning Summary

[Learning Summary](#)

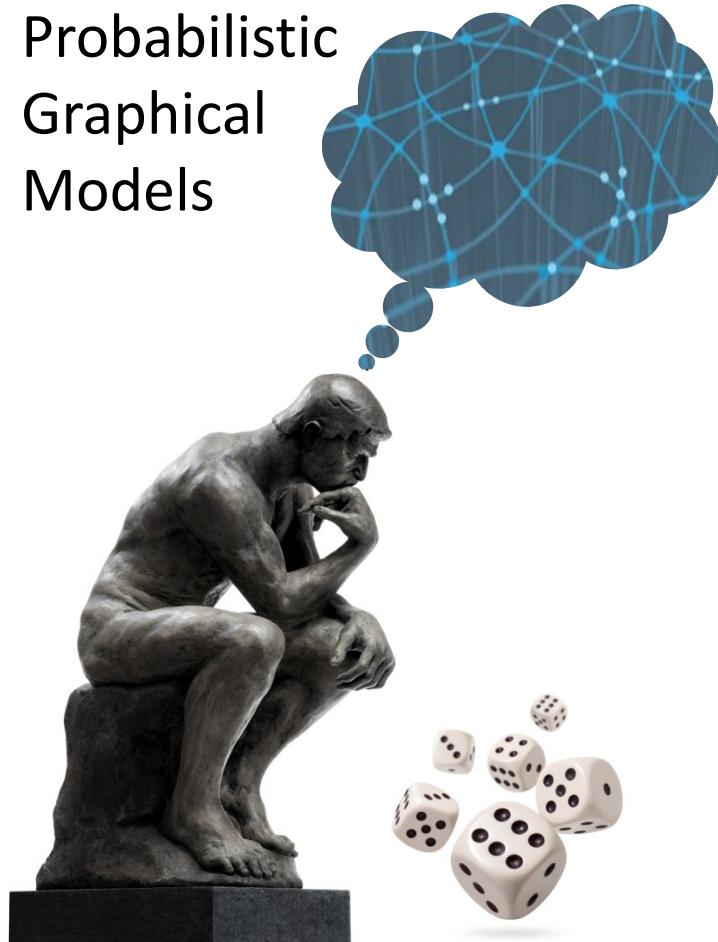
Final Summary

[Final Summary](#)

Created Sun 8 Jan 2012 12:51 AM PST

Last Modified Thu 10 May 2012 5:45 PM PDT

Probabilistic
Graphical
Models



Introduction

Welcome to the PGM Class

Probabilistic Graphical Models

Daphne Koller

Course Structure

- 10 weeks + final
- Videos + quizzes
- 9 problem sets
 - 25% of score
 - Multiple submissions

Course Structure

- 9 programming assignments
 - Genetically inherited diseases
 - Optical character recognition
 - Recognizing activities from Kinect sensor
 - $9 \times 7\% = 63\%$ of score
- Final exam
 - 12% of score

Background

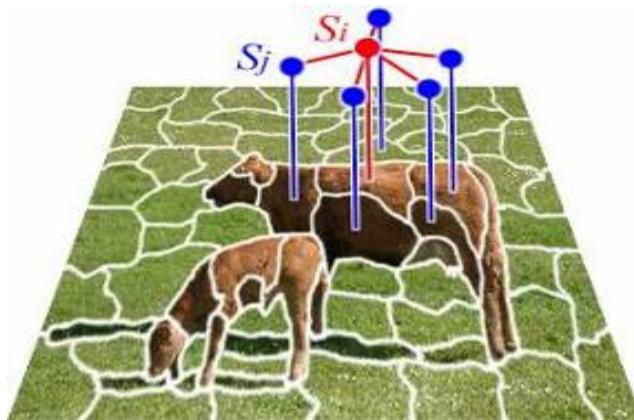
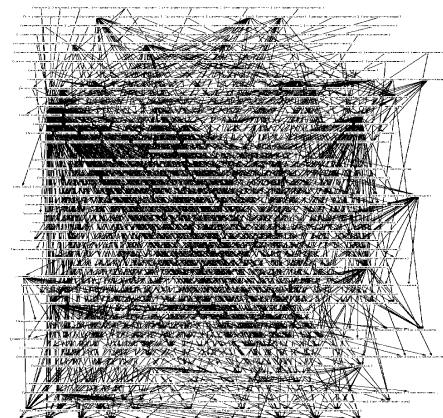
- Required
 - Basic probability theory
 - Some programming
 - Some algorithms and data structures
- Recommended
 - Machine learning
 - Simple optimization
 - Matlab or Octave

Other Issues

- Honor code
- Time management (10-15 hrs / week)
- Discussion forum & study groups

What you'll learn

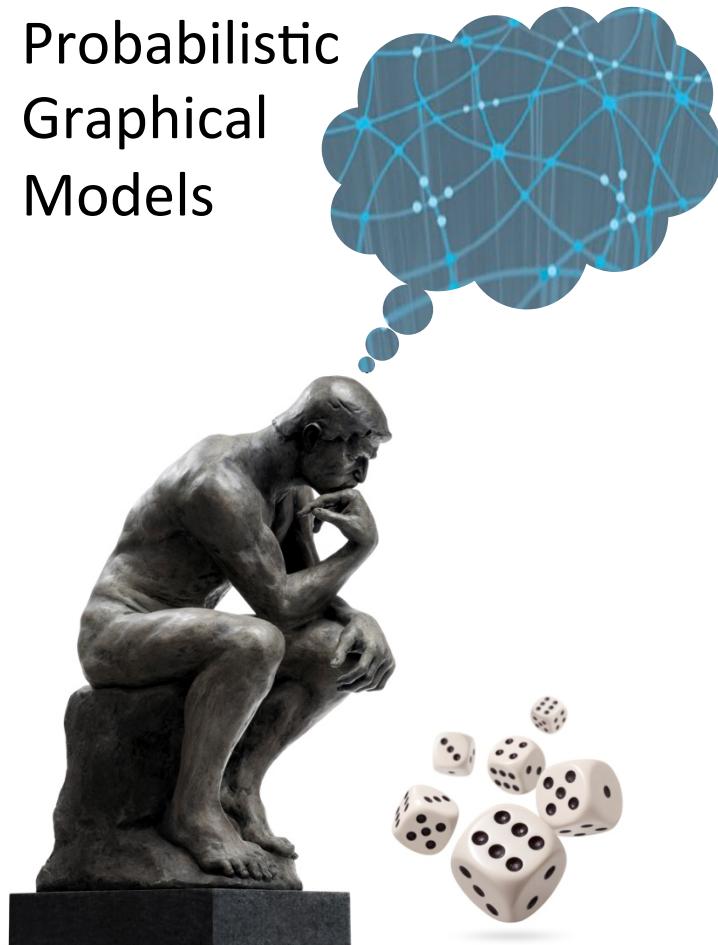
- Fundamental methods
- Real-world applications
- How to use these methods in your work



M. Pradhan , G. Provan , B. Middleton , M.Henrion, UAI 94

Daphne Koller

Probabilistic
Graphical
Models

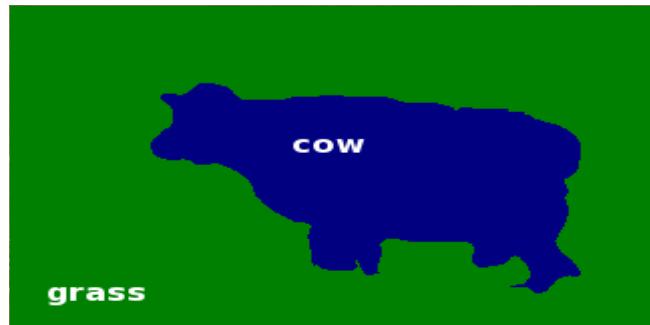


Introduction

Motivation and Overview



predisposing factors
symptoms
test results
diseases
treatment outcomes

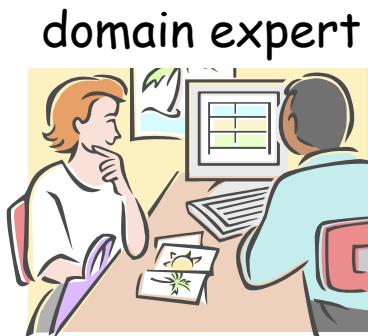


millions of pixels or
thousands of superpixels

each needs to be labeled
{grass, sky, water, cow, horse, ...}

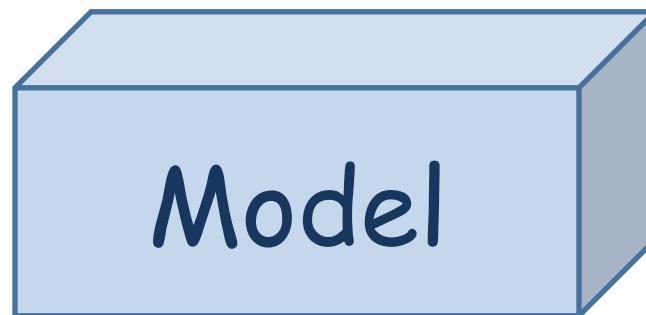
Probabilistic Graphical Models

Daphne Koller

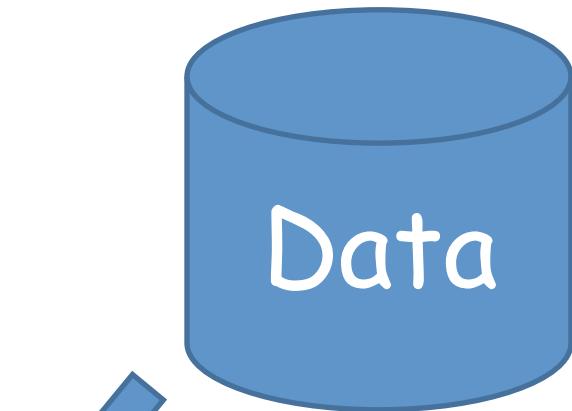


Models

Declarative representation



elicitation



Learning

Algorithm

Algorithm

Algorithm

Uncertainty

- Partial knowledge of state of the world
- Noisy observations
- Phenomena not covered by our model
- Inherent stochasticity

Probability Theory

- Declarative representation with clear semantics
- Powerful reasoning patterns *conditioning
decision making*
- Established learning methods

Complex Systems

predisposing factors
symptoms
test results
diseases
treatment outcomes

class labels for
thousands of superpixels

Random variables X_1, \dots, X_n

Joint distribution $P(X_1, \dots, X_n)$

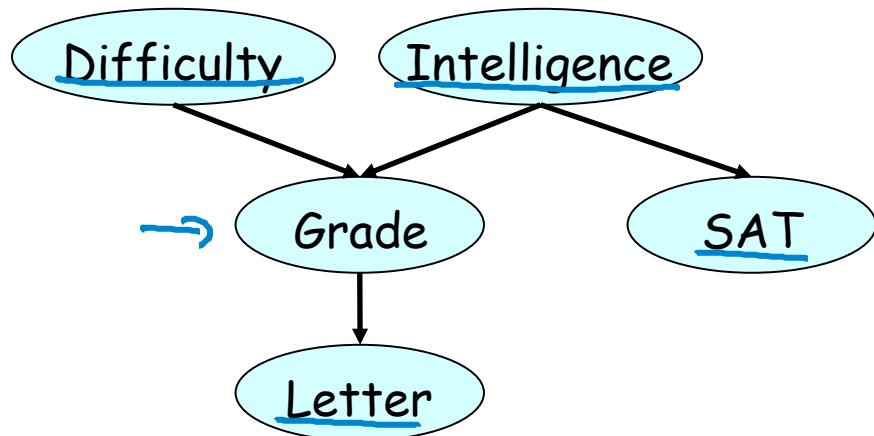
~ binary valued
distribution
over 2^n
possible states

~~x... nodes~~

Graphical Models

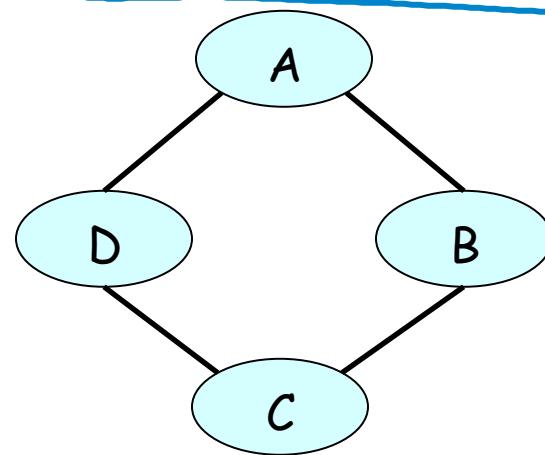
directed graph

Bayesian networks



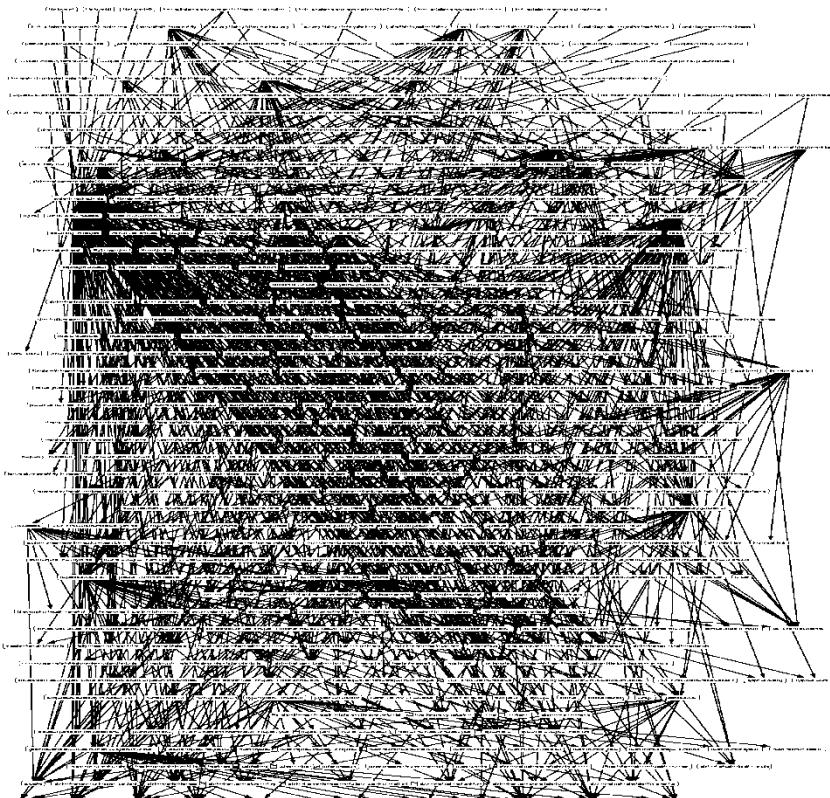
undirected graph

Markov networks

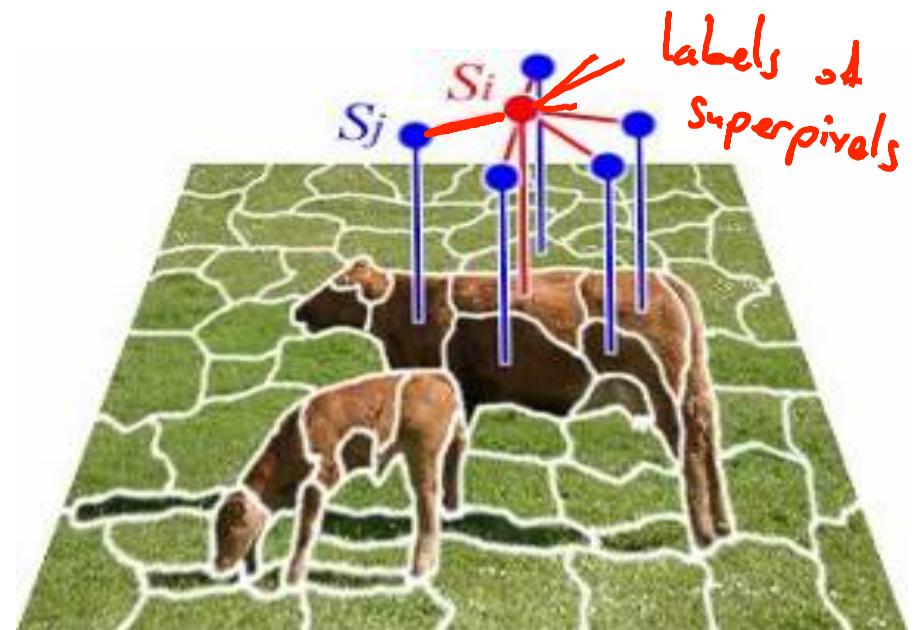


diag ~vis
CP CS

Graphical Models



M. Pradhan, G. Provan, B. Middleton, M. Henrion, UAI 94



Daphne Koller

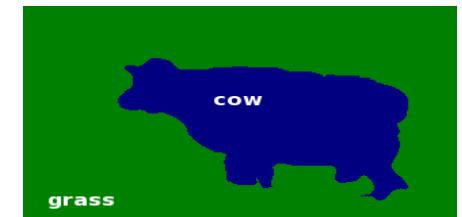
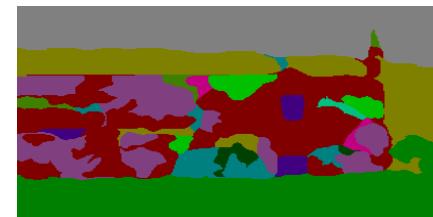
Graphical Representation

- Intuitive & compact data structure
- Efficient reasoning using general-purpose algorithms
- Sparse parameterization
 - feasible elicitation \leftarrow *by hand*
 - learning from data \leftarrow *automatically*

Many Applications

- Medical diagnosis
- Fault diagnosis
- Natural language processing
- Traffic analysis
- Social network models
- Message decoding
- Computer vision
 - Image segmentation
 - 3D reconstruction
 - Holistic scene analysis
- Speech recognition
- Robot localization & mapping

Image Segmentation



superpixels

machine learning
to separate superpixels

Daphne Koller

Thanks to: Eric Horvitz, Microsoft Research

Medical Diagnosis

Applet started

MS on □ ◀ ▶ X ?

ON STAGE ESSENTIALS COMMUNICATE FIND ✓ ?

OnParenting May 14 - May 20, 1997 Fidelity Investments® Our home on the web [is where] click here

cover contents news experts fun handbook talk find help feedback

There are two ways to search for specific information in OnParenting. In **Find by Word**, type the word(s) you want to find and get a list of titles relevant to that word. **Find by Symptom** will help you get information about children's symptoms. [Help](#) has tips to target your search.

Describe the child
in the drop-down boxes at the right. Relevant information will appear below.

Age: Toddler Sex: Female
Complaint: Abdominal pain

Localized pain: Can the child localize, or point to, the site of the pain?
 No, unable to localize
 Below the navel to the child's left
 Above the child's navel
 Either of the child's sides
 Below the navel to the child's right
 Above the navel to the child's right
 Above the navel to the child's left
 Don't Know

Results so far

Disorder	Relevance
Viral gastroenteritis	High
Psychosomatic pain	Medium
Urinary tract infection	Low
Other	Very Low

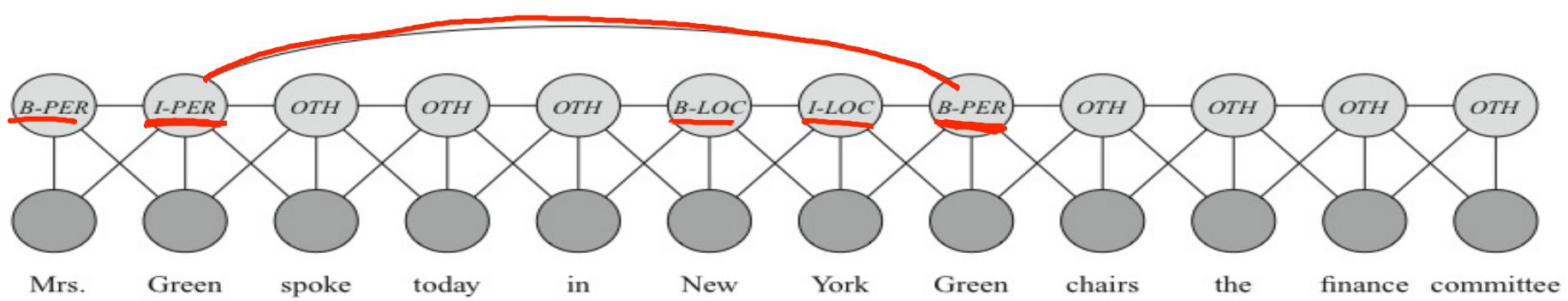
Start Over Review
Next>> Finish



Daphne Koller

Textual Information Extraction

Mrs. Green spoke today in New York. Green chairs the finance committee.
Person *Location* *Person* *Organization*



Multi-Sensor Integration: Traffic

Live Search Maps
http://maps.live.com/#JnE9eXaud2FzaGluZ3RvbikYyU3ZXNzdC4wTdlcGcuMSzIYj000S45NTEyMTk5MDg2NjIN2UtNjkuNzg1MTU2MjUIN2UyMC44

Live Search Maps | MSN | Windows Live

Live Search | Businesses | People | Collections | Locations | Web

Share | Print

Washington, D.C.

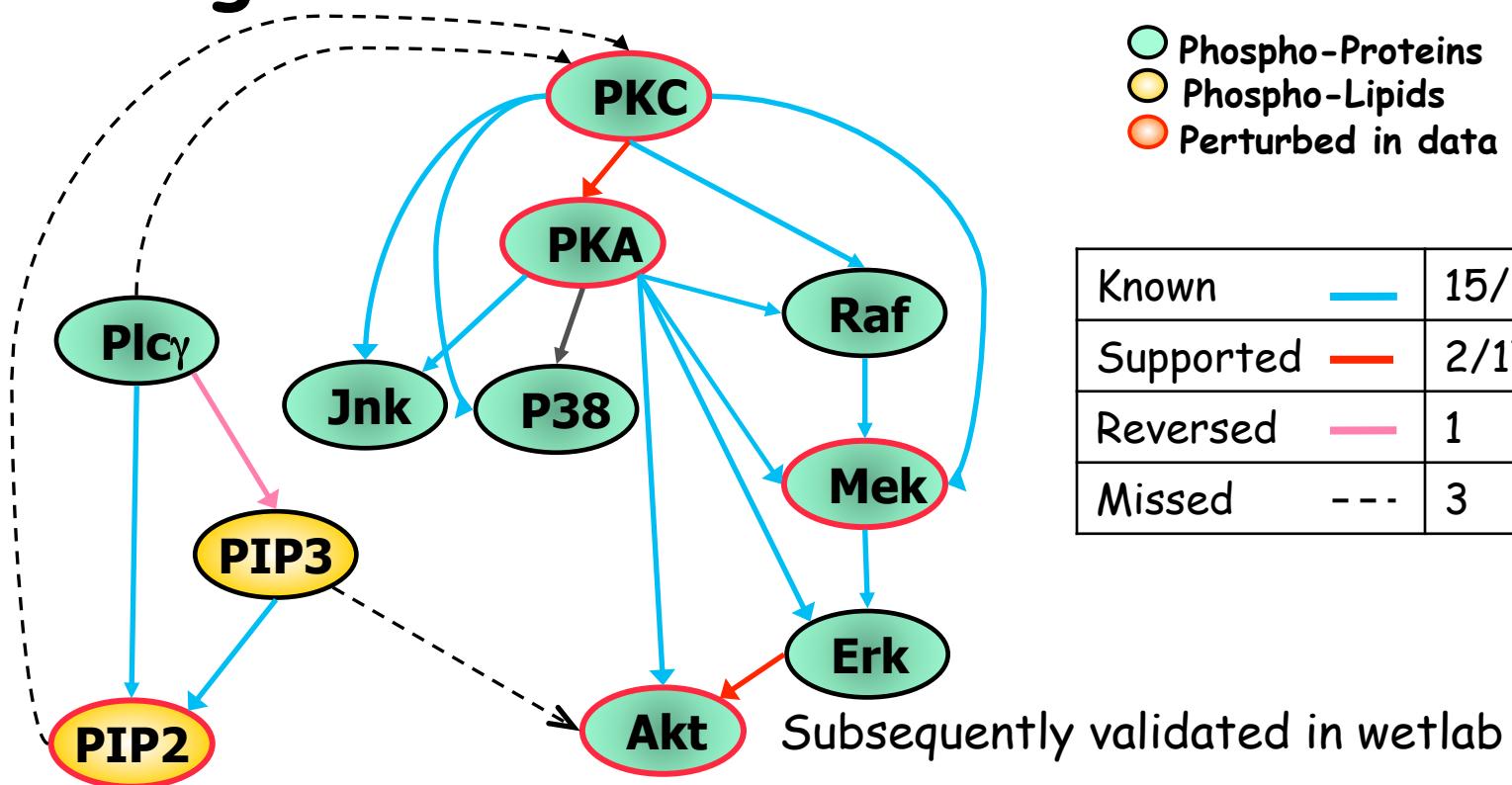
• I95 corridor experiment: accurate to ± 5 MPH in 85% of cases

• Fielded in 72 cities

Thanks to: Eric Horvitz, Microsoft Research

This figure may be used for non-commercial and classroom purposes only.
Any other uses require the prior written permission from AAAS

Biological Network Reconstruction



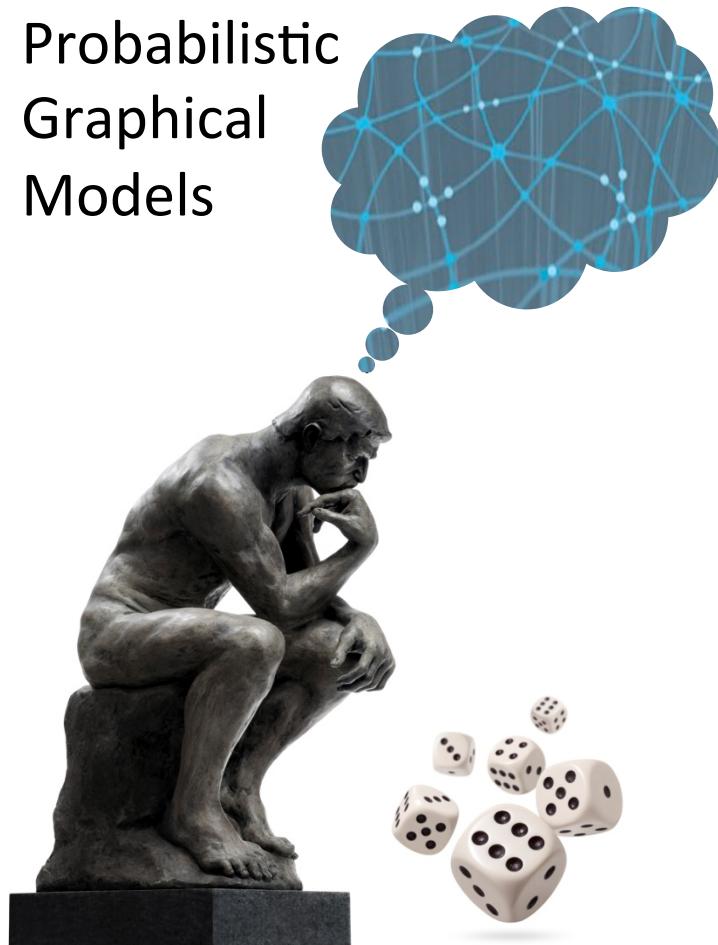
Causal protein-signaling networks derived from multiparameter single-cell data
Sachs et al., *Science* 2005

Daphne Koller

Overview

- Representation
 - Directed and undirected
 - Temporal and plate models
- Inference *reasoning*
 - Exact and approximate
 - Decision making
- Learning
 - Parameters and structure
 - With and without complete data

Probabilistic
Graphical
Models



Introduction

Preliminaries: Distributions

Joint Distribution $P(I, D, G)$

- Intelligence (I) $\leftarrow 2$
 - i^0 (low), i^1 (high),
 - Difficulty (D) $\leftarrow 2$
 - d^0 (easy), d^1 (hard)
 - Grade (G) $\leftarrow 3$
 - g^1 (A), g^2 (B), g^3 (C)
- $2 \times 2 \times 3 = 12$ *parameters*
- independent params
!!

I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^0	g^2	0.168
i^0	d^0	g^3	0.126
i^0	d^1	g^1	0.009
i^0	d^1	g^2	0.045
i^0	d^1	g^3	0.126
i^1	d^0	g^1	0.252
i^1	d^0	g^2	0.0224
i^1	d^0	g^3	0.0056
i^1	d^1	g^1	0.06
i^1	d^1	g^2	0.036
i^1	d^1	g^3	0.024

Conditioning

condition on g^1

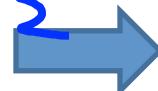
I	D	G	Prob.
i ⁰	d ⁰	g^1	0.126
i ⁰	d ⁰	g^2	0.168
i ⁰	d ⁰	g^3	0.126
i ⁰	d ¹	g^1	0.009
i ⁰	d ¹	g^2	0.045
i ⁰	d ¹	g^3	0.126
i ¹	d ⁰	g^1	0.252
i ¹	d ⁰	g^2	0.0224
i ¹	d ⁰	g^3	0.0056
i ¹	d ¹	g^1	0.06
i ¹	d ¹	g^2	0.036
i ¹	d ¹	g^3	0.024

Conditioning: Reduction

I	D	G	Prob.
i^0	d^0	g^1	0.126
i^0	d^1	g^1	0.009
i^1	d^0	g^1	0.252
i^1	d^1	g^1	0.06

Conditioning: Renormalization

I	D	G	Prob.
i ⁰	d ⁰	g ¹	0.126 <i>1/447</i>
i ⁰	d ¹	g ¹	0.009
i ¹	d ⁰	g ¹	0.252
i ¹	d ¹	g ¹	0.06



I	D	Prob.
i ⁰	d ⁰	0.282
i ⁰	d ¹	0.02
i ¹	d ⁰	0.564
i ¹	d ¹	0.134

$$\frac{P(I, D, g^1)}{0.447}$$

unnormalized measure

$$P(I, D | g^1)$$

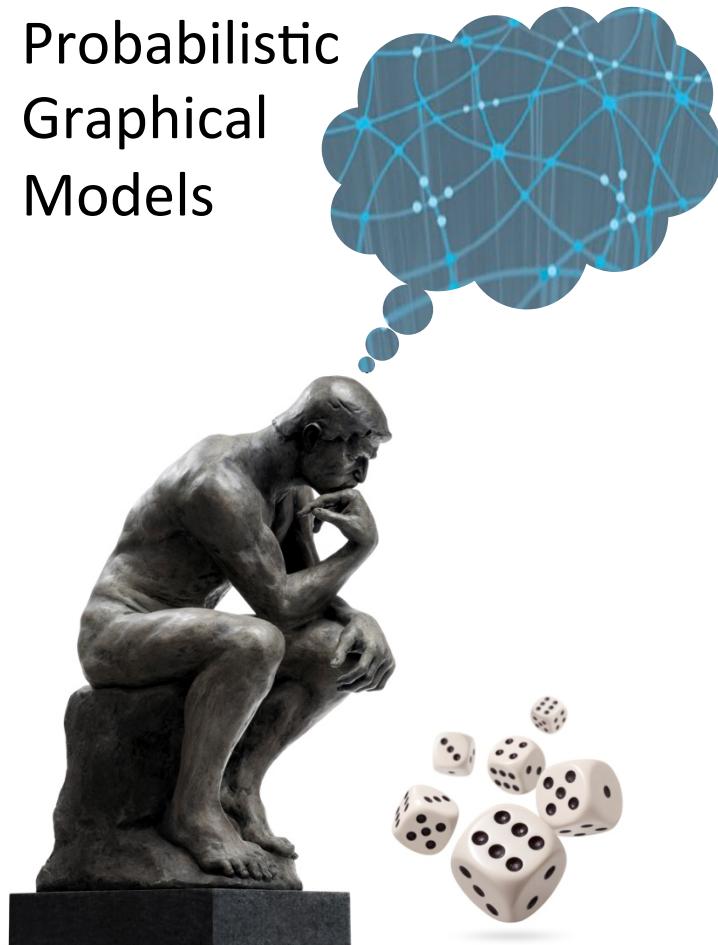
Marginalization

$P(I, D)$ Marginalize I

I	D	Prob.
i^0	d^0	0.282
i^0	d^1	0.02
i^1	d^0	0.564
i^1	d^1	0.134

D	Prob.
d^0	0.846
d^1	0.154

Probabilistic
Graphical
Models



Introduction

Preliminaries: Factors

Factors

- A factor $\phi(\underline{X_1}, \dots, \underline{X_k})$

$$\phi : \text{Val}(\underline{X_1}, \dots, \underline{X_k}) \rightarrow R$$

- Scope = $\{\underline{X_1}, \dots, \underline{X_k}\}$

Joint Distribution

$P(I, D, G)$

<u>I</u>	<u>D</u>	<u>G</u>	Prob.
i^0	d^0	g^1	0.126
i^0	d^0	g^2	0.168
i^0	d^0	g^3	0.126
i^0	d^1	g^1	0.009
i^0	d^1	g^2	0.045
i^0	d^1	g^3	0.126
i^1	d^0	g^1	0.252
i^1	d^0	g^2	0.0224
i^1	d^0	g^3	0.0056
i^1	d^1	g^1	0.06
i^1	d^1	g^2	0.036
i^1	d^1	g^3	0.024

Unnormalized measure $P(I, D, g^1)$

Scope = {I, D}

$P(I, D, g^1)$

I	D	G	Prob.
i ⁰	d ⁰	g ¹	0.126
i ⁰	d ¹	g ¹	0.009
i ¹	d ⁰	g ¹	0.252
i ¹	d ¹	g ¹	0.06

Conditional Probability Distribution (CPD)

$P(G | \underline{I}, \underline{D})$

context

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

\underline{A} \underline{B} \underline{C}

General factors

Scope = {A, φ}

A	B	φ
a ⁰	b ⁰	30
a ⁰	b ¹	5
a ¹	b ⁰	1
a ¹	b ¹	10

Factor Product

a^1	b^1	0.5
a^1	b^2	0.8
a^2	b^1	0.1
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.9

$q_1(a, b)$

b^1	c^1	0.5
b^1	c^2	0.7
b^2	c^1	0.1
b^2	c^2	0.2

$q_2(b, c)$

a^1	b^1	c^1	$0.5 \cdot 0.5 = 0.25$
a^1	b^1	c^2	$0.5 \cdot 0.7 = 0.35$
a^1	b^2	c^1	$0.8 \cdot 0.1 = 0.08$
a^1	b^2	c^2	$0.8 \cdot 0.2 = 0.16$
a^2	b^1	c^1	$0.1 \cdot 0.5 = 0.05$
a^2	b^1	c^2	$0.1 \cdot 0.7 = 0.07$
a^2	b^2	c^1	$0 \cdot 0.1 = 0$
a^2	b^2	c^2	$0 \cdot 0.2 = 0$
a^3	b^1	c^1	$0.3 \cdot 0.5 = 0.15$
a^3	b^1	c^2	$0.3 \cdot 0.7 = 0.21$
a^3	b^2	c^1	$0.9 \cdot 0.1 = 0.09$
a^3	b^2	c^2	$0.9 \cdot 0.2 = 0.18$

Supr A,B,C

Factor Marginalization

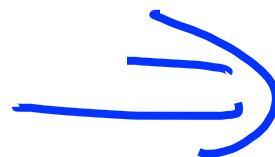
a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

A,C

a^1	c^1	0.33
a^1	c^2	0.51
a^2	c^1	0.05
a^2	c^2	0.07
a^3	c^1	0.24
a^3	c^2	0.39

Factor Reduction

a ¹	b ¹	c ¹	0.25
a ¹	b ¹	c ²	0.35
a ¹	b ²	c ¹	0.08
a ¹	b ²	c ²	0.16
a ²	b ¹	c ¹	0.05
a ²	b ¹	c ²	0.07
a ²	b ²	c ¹	0
a ²	b ²	c ²	0
a ³	b ¹	c ¹	0.15
a ³	b ¹	c ²	0.21
a ³	b ²	c ¹	0.09
a ³	b ²	c ²	0.18



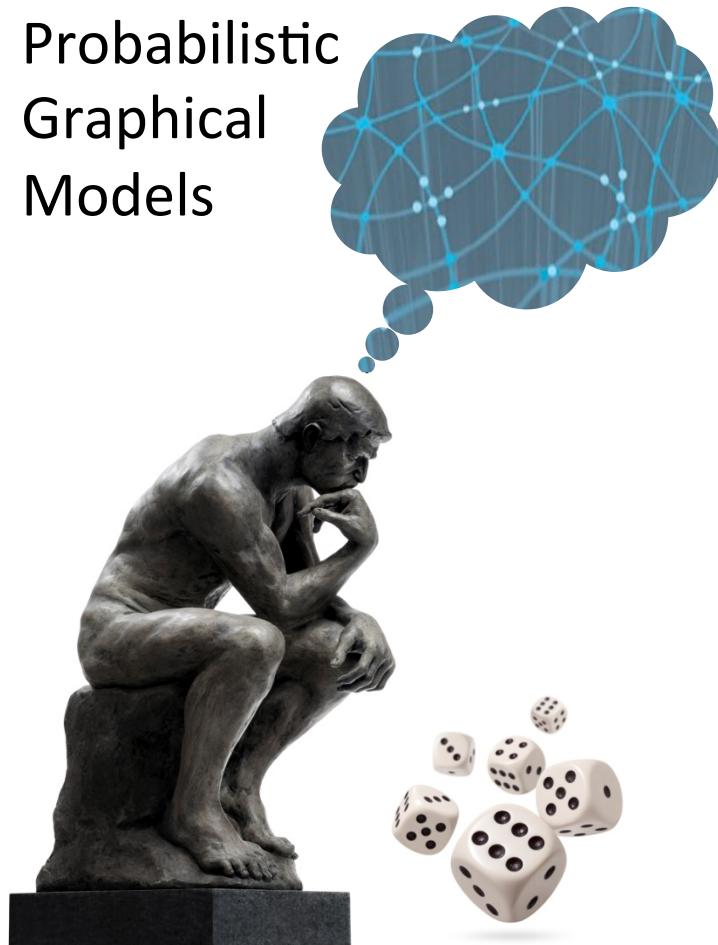
a ¹	b ¹	c ¹	0.25
a ¹	b ²	c ¹	0.08
a ²	b ¹	c ¹	0.05
a ²	b ²	c ¹	0
a ³	b ¹	c ¹	0.15
a ³	b ²	c ¹	0.09

A, B

Why factors?

- Fundamental building block for defining distributions in high-dimensional spaces
- Set of basic operations for manipulating these probability distributions

Probabilistic
Graphical
Models



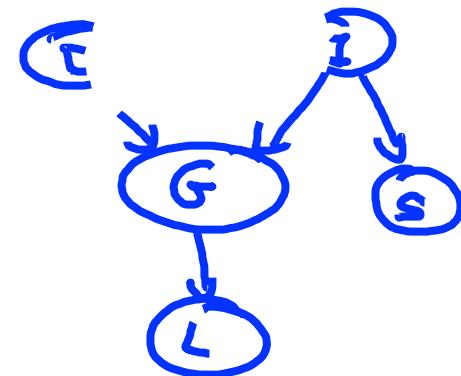
Representation

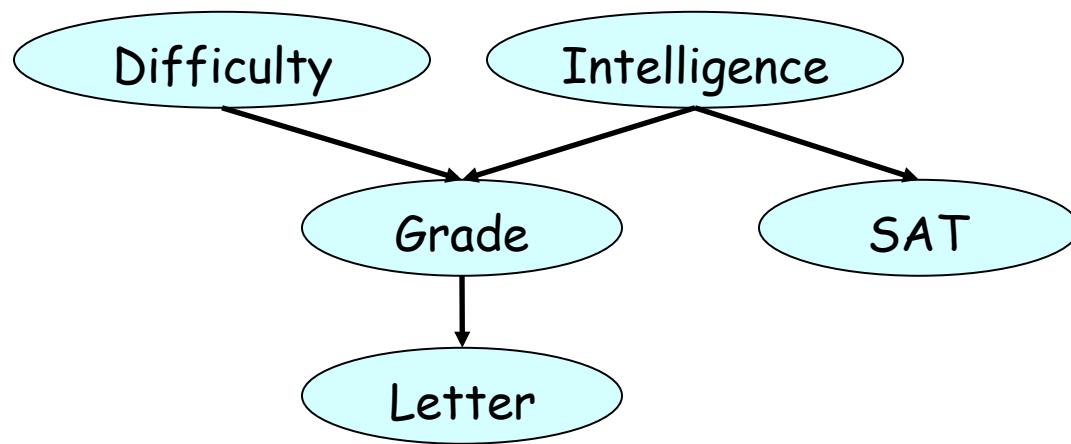
Bayesian Networks

Semantics &
Factorization

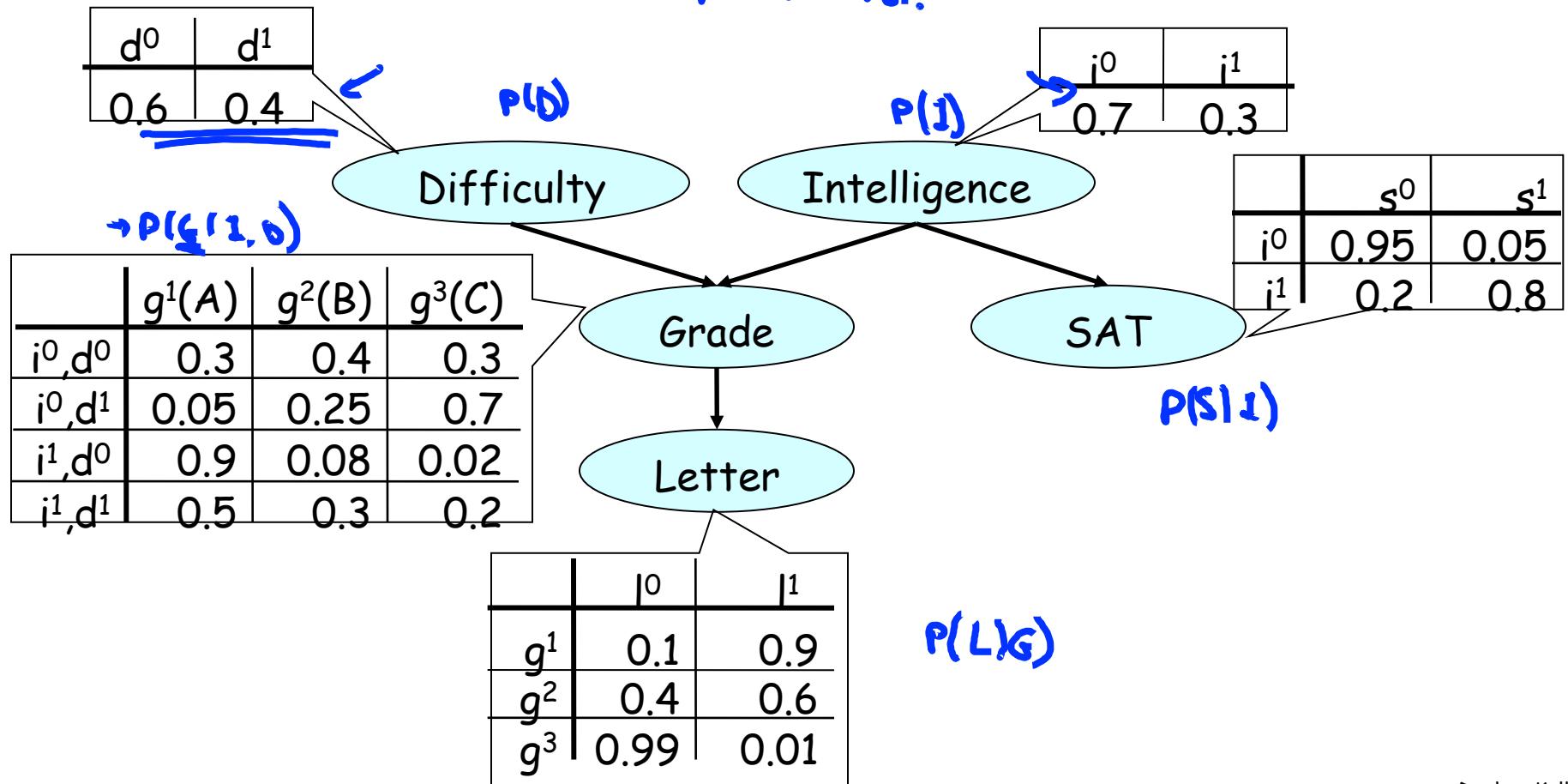
- Grade
- Course Difficulty
- Student Intelligence
- Student SAT
- Reference Letter

$$P(G, D, I, S, L)$$

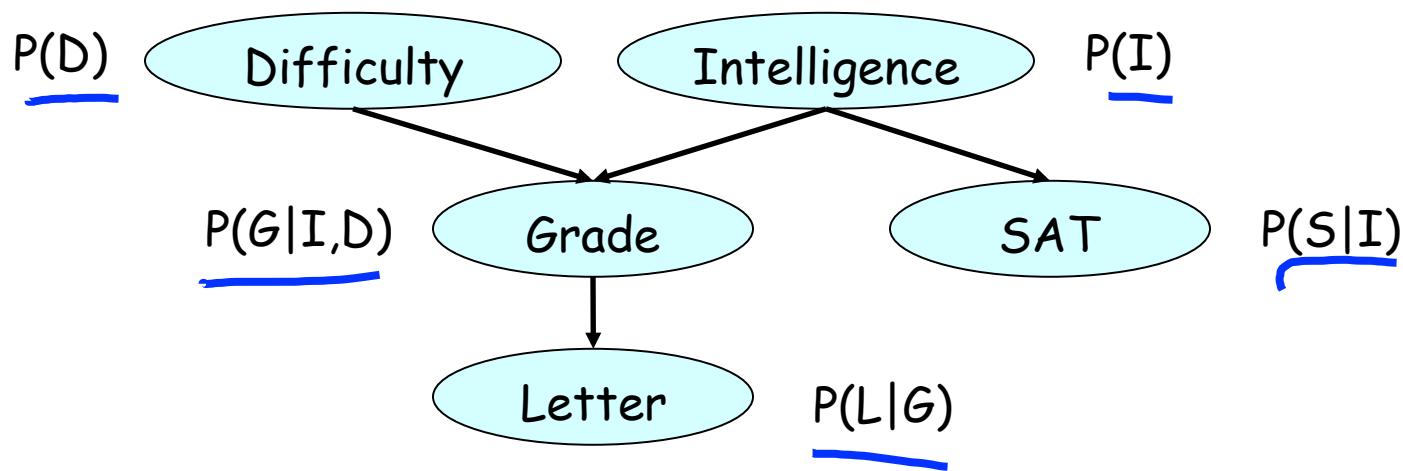




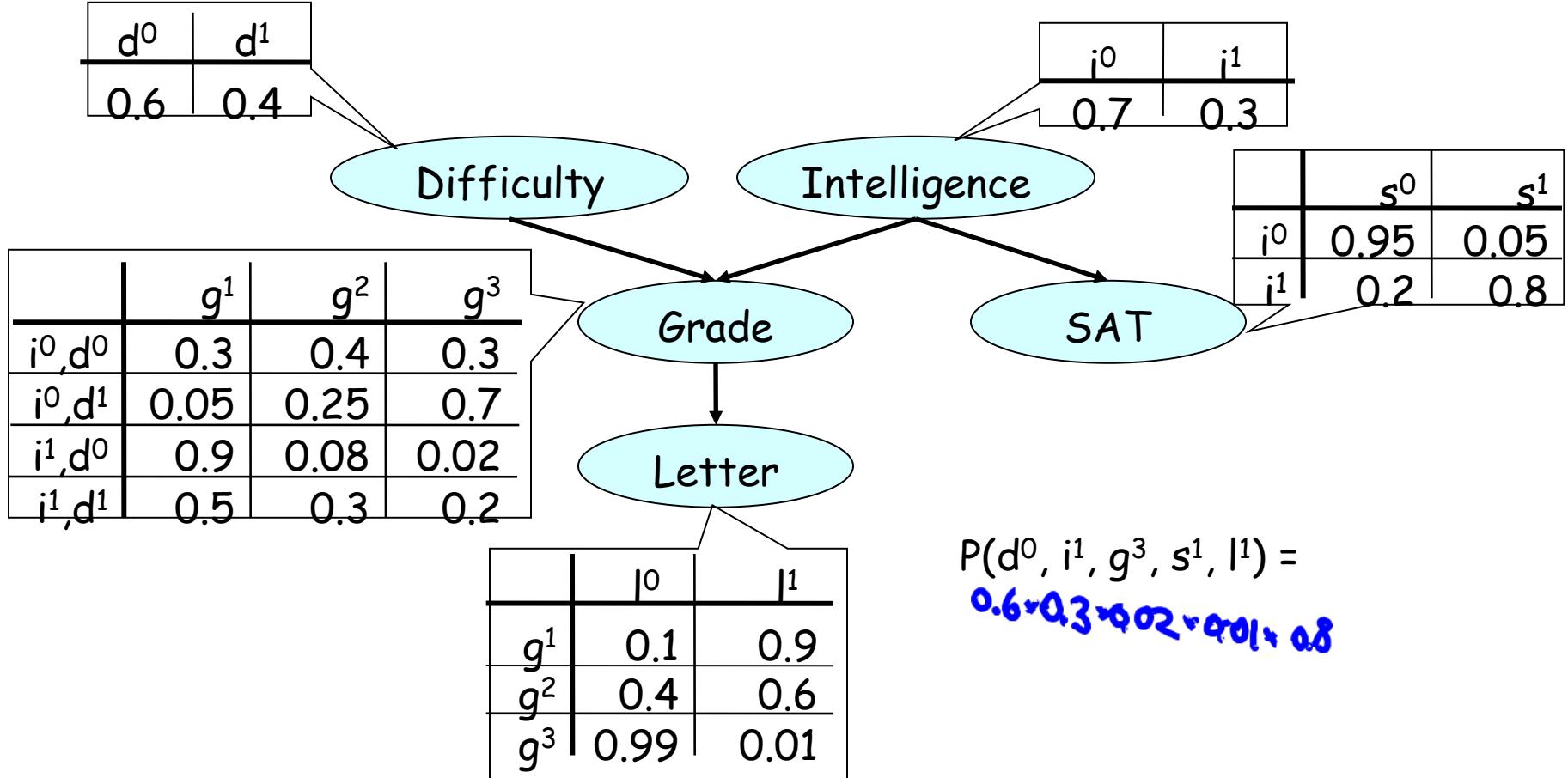
CPD = cond. prob. dist.



Chain Rule for Bayesian Networks



$$\underbrace{P(D, I, G, S, L)}_{\text{Distribution defined as a product of factors!}} = P(D) P(I) P(G|I,D) P(S|I) P(L|G)$$



Bayesian Network

- A Bayesian network is:
 - A directed acyclic graph (DAG) G whose nodes represent the random variables X_1, \dots, X_n
 - For each node $\underline{X_i}$ a CPD $P(\underline{X_i} \mid \underline{\text{Par}_G(X_i)})$
- The BN represents a joint distribution via the chain rule for Bayesian networks

$$P(X_1, \dots, X_n) = \prod_i P(X_i \mid \text{Par}_G(X_i))$$

BN Is a Legal Distribution: $P \geq 0$

P is a product of CPDs

CPDs are non-negative

BN Is a Legal Distribution: $\sum P = 1$

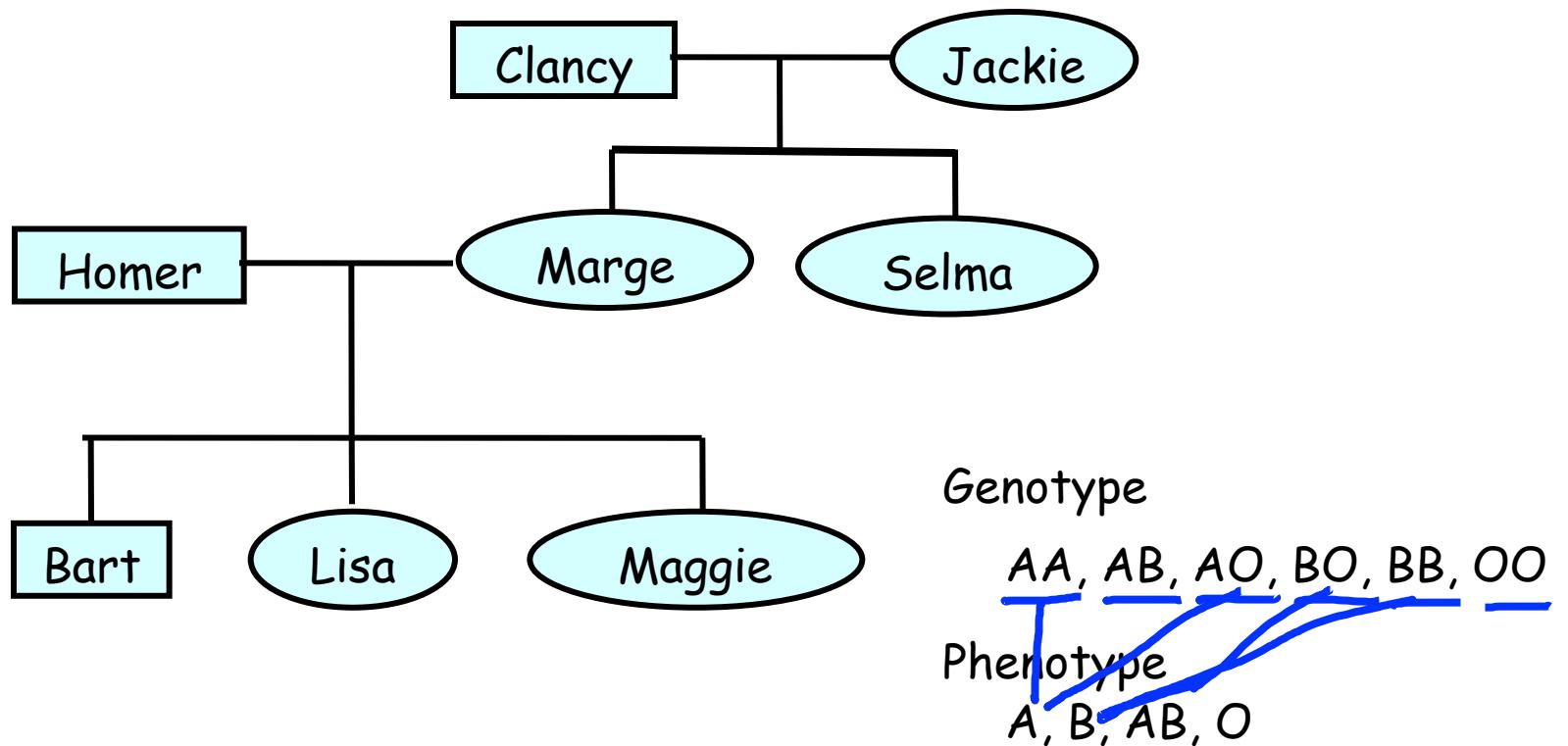
$$\begin{aligned}\sum_{D,I,G,S,L} P(D, I, G, S, L) &= \sum_{D,I,G,S,L} P(D) P(I) P(G|I,D) P(S|I) P(L|G) \\&= \sum_{D,I,G,S} P(D) P(I) P(G|I,D) P(S|I) \sum_L P(L|G) \\&= \sum_{D,I,G,S} P(D) P(I) P(G|I,D) P(S|I) \\&= \sum_{D,I,G} P(D) P(I) P(G|I,D) \sum_S P(S|I) \\&= \sum_{D,I} P(D) P(I) \sum_G P(G|I,D)\end{aligned}$$

P Factorizes over G

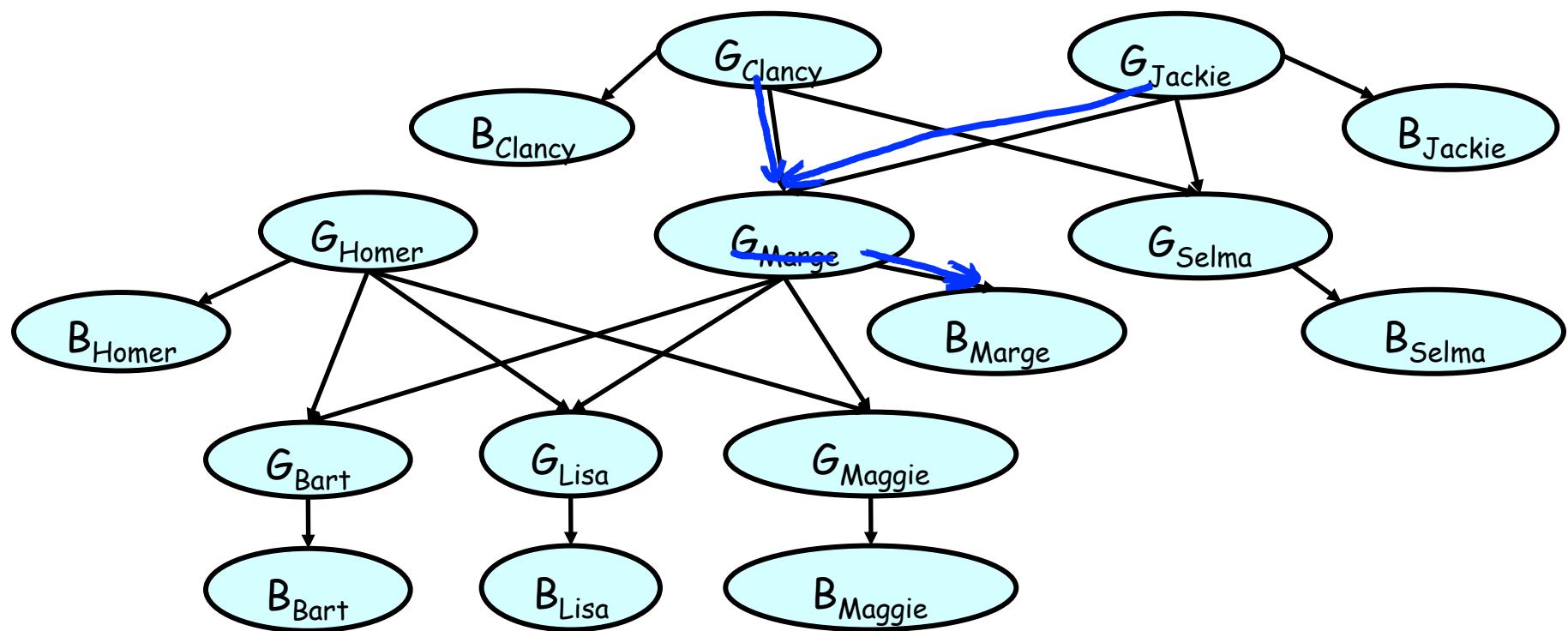
- Let G be a graph over X_1, \dots, X_n .
- P factorizes over G if

$$P(X_1, \dots, X_n) = \prod_i P(X_i | \text{Par}_G(X_i))$$

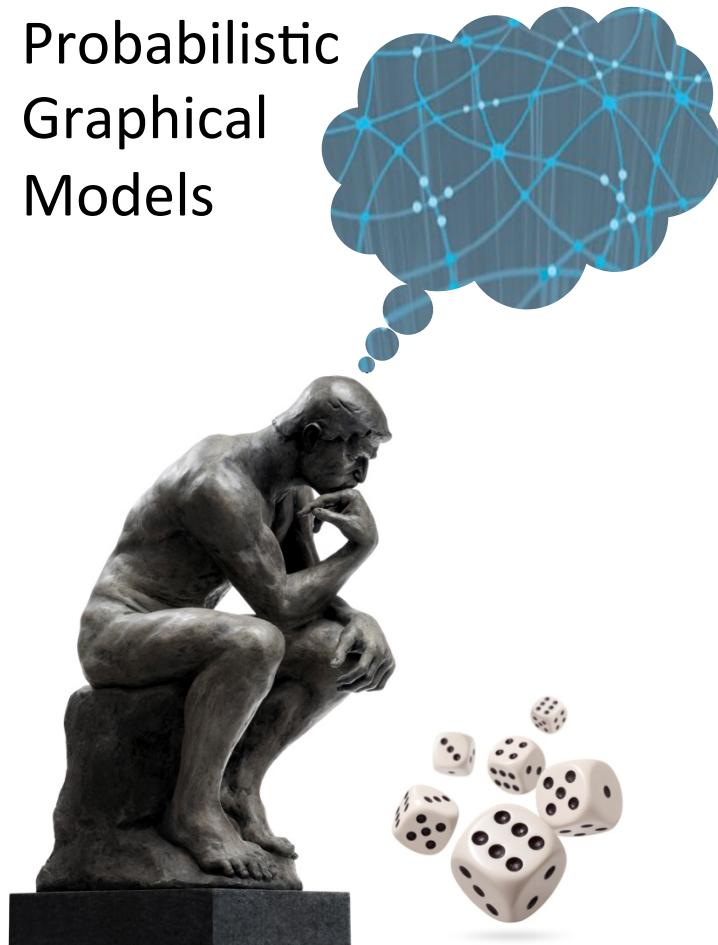
Genetic Inheritance



BNs for Genetic Inheritance



Probabilistic
Graphical
Models

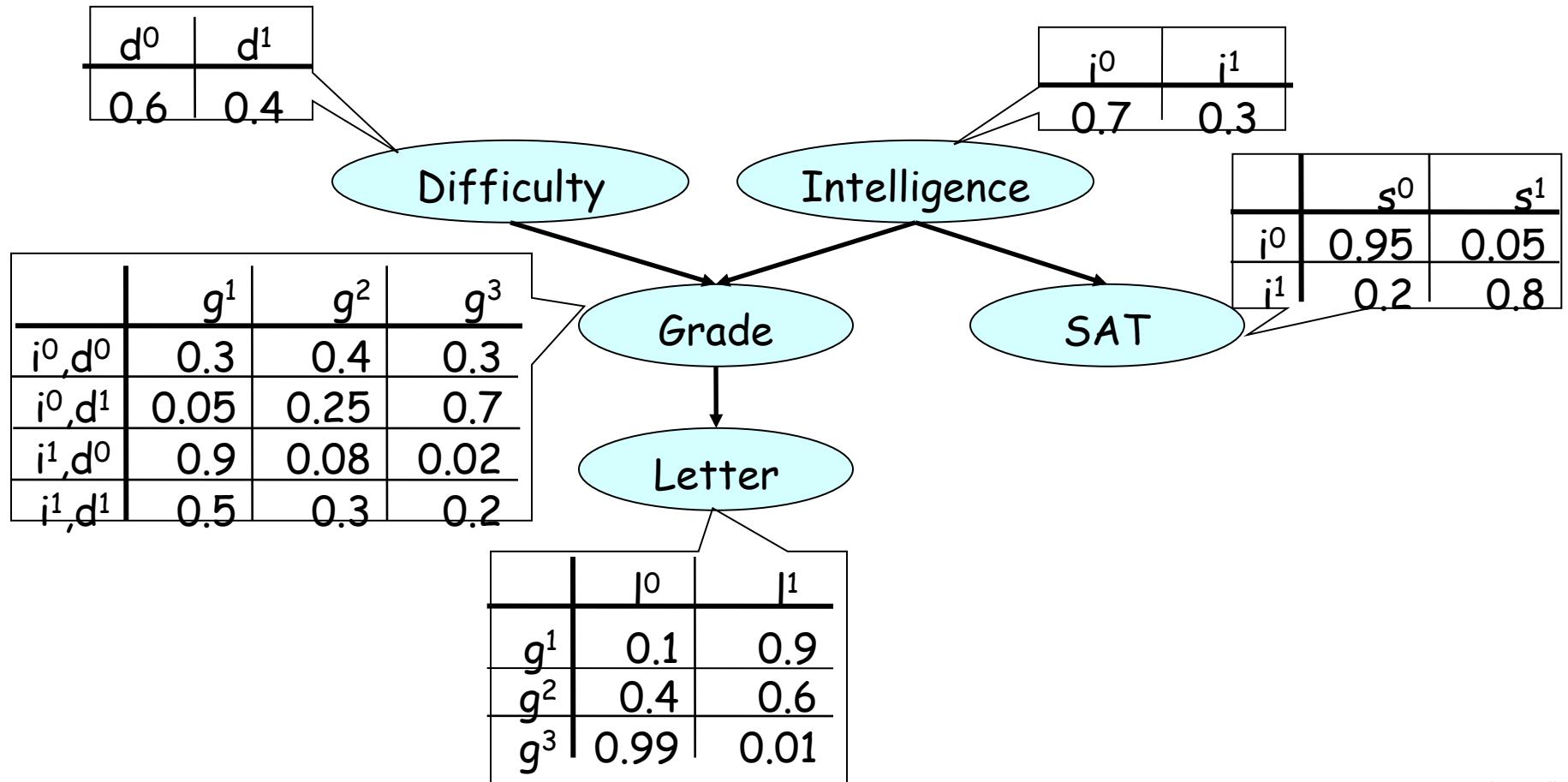


Representation

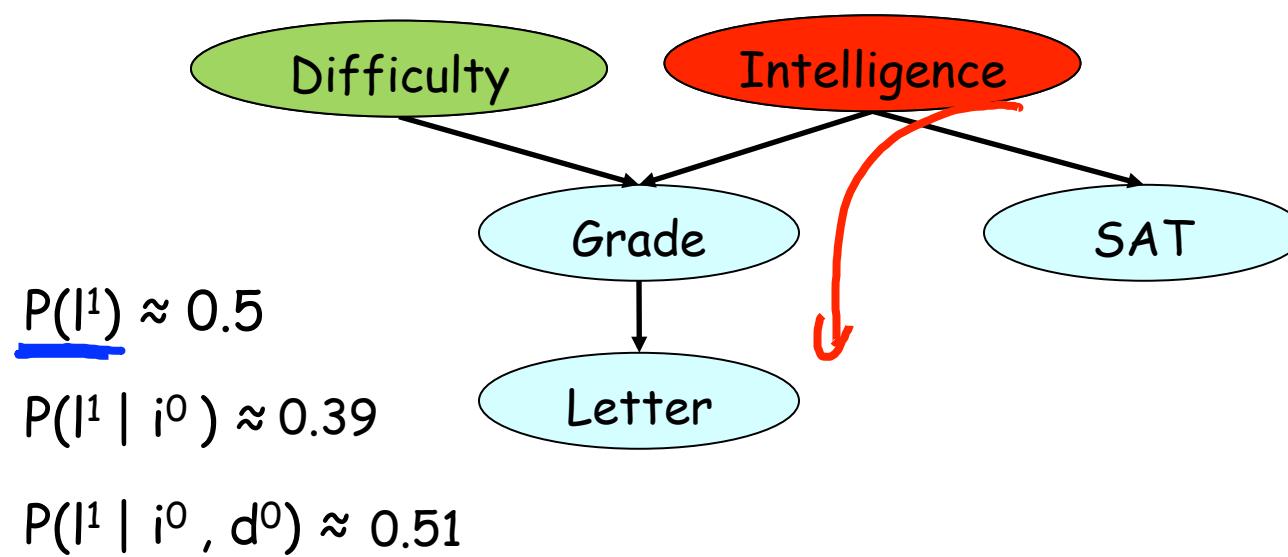
Bayesian Networks

Reasoning
Patterns

The Student Network



Causal Reasoning



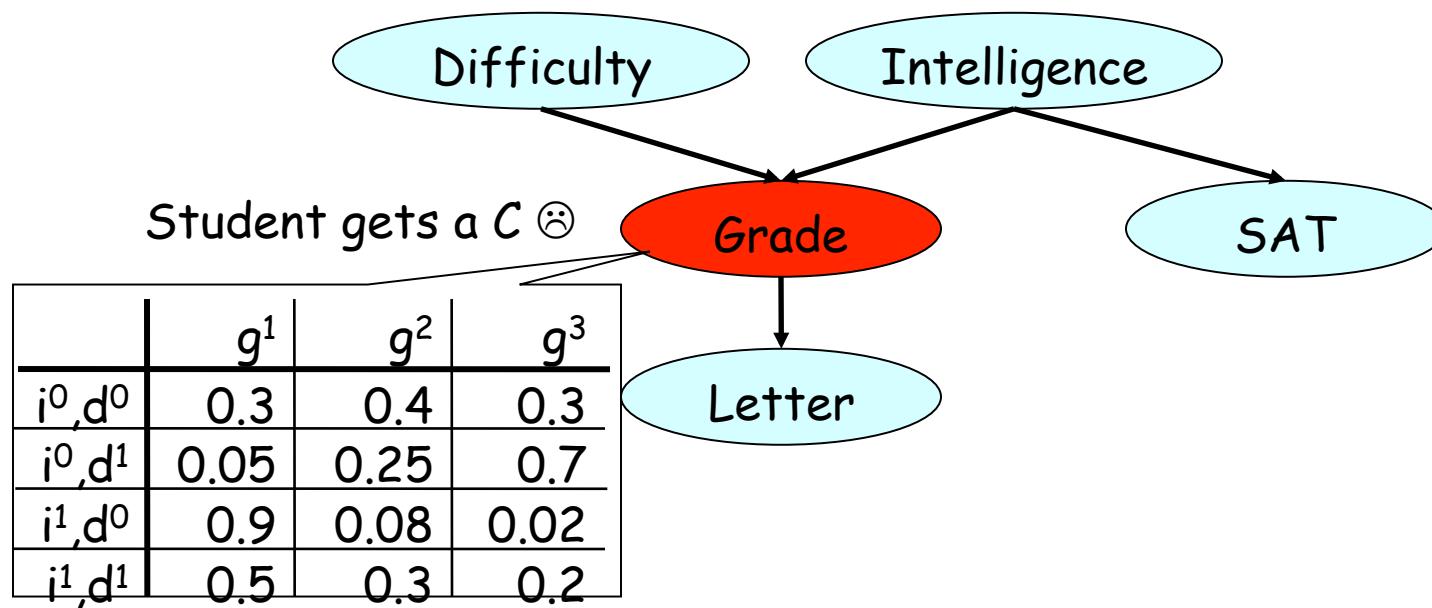
Evidential Reasoning

$$P(d^1) = 0.4$$

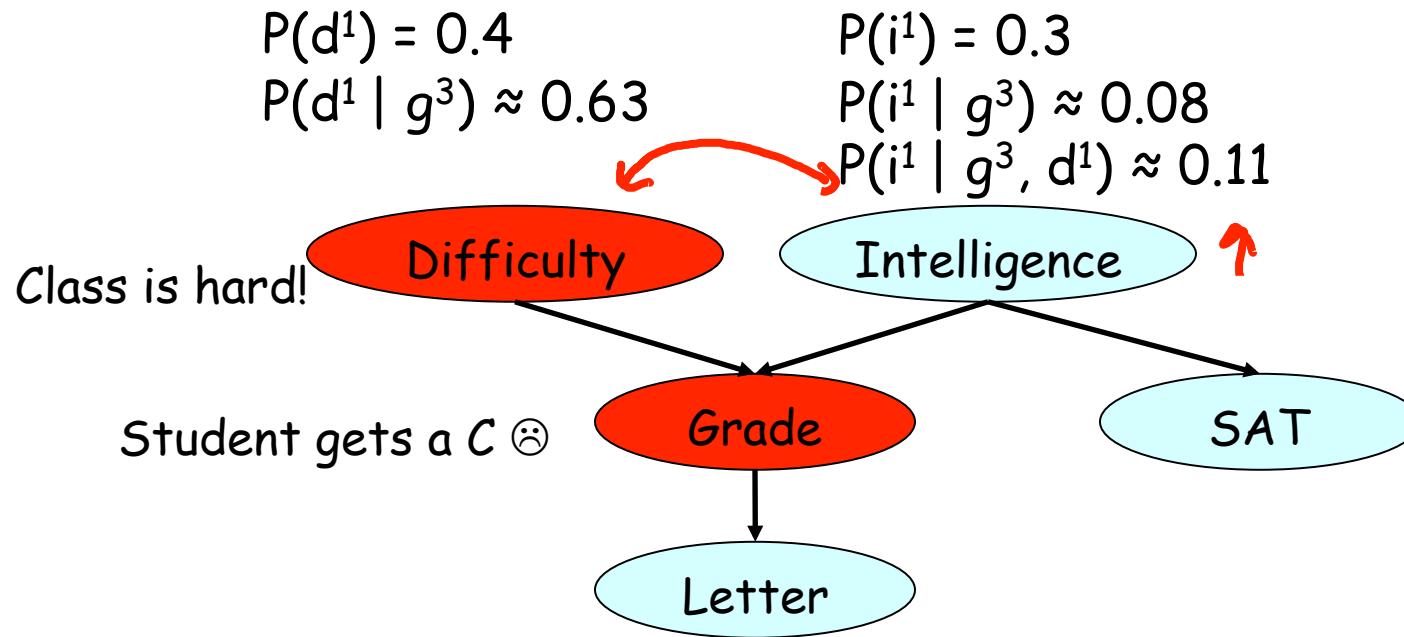
$$P(d^1 | g^3) \approx 0.63$$

$$P(i^1) = 0.3$$

$$P(i^1 | g^3) \approx 0.08$$

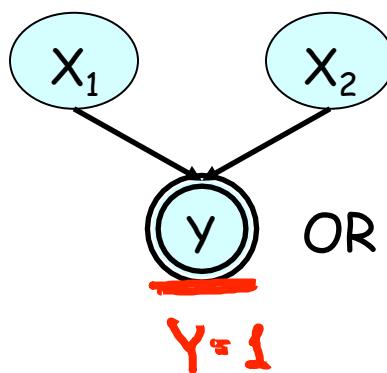


Intercausal Reasoning



Intercausal Reasoning Explained

explaining away



X_1	X_2	Y	Prob
0	0	0	0.25
0	1	1	0.25
1	0	1	0.25
1	1	1	0.25

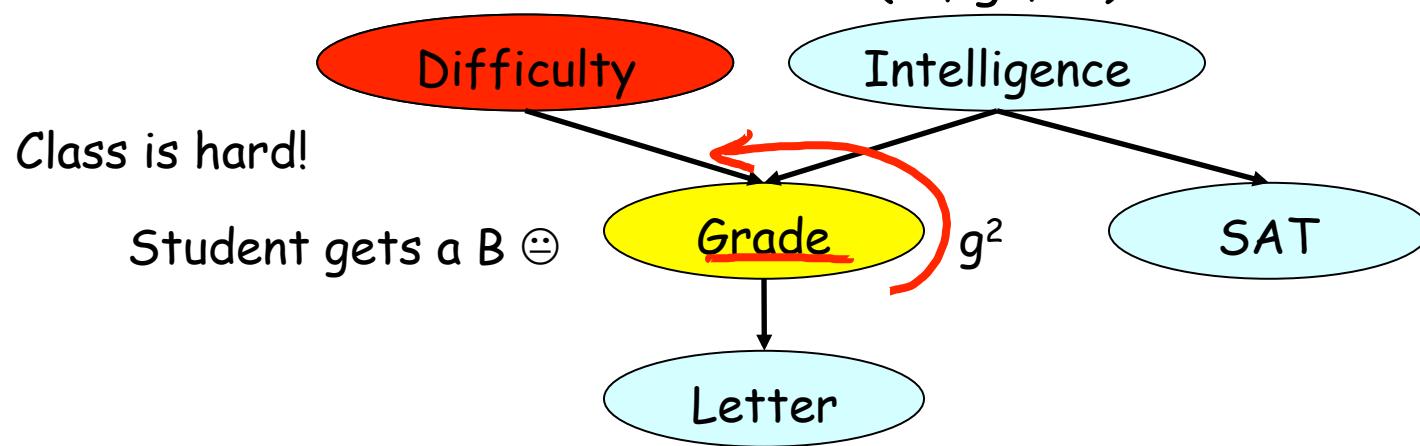
Annotations in red and blue highlight specific rows and columns of the table, corresponding to the causal paths from X_1 and X_2 to y .

$$\text{Ans: } P(X_1=1) = \frac{2}{3} \quad P(X_2=1) = \frac{2}{3}$$

Condition $X_1=1 \quad P(Y=1 | X_2=1) = 0.5$

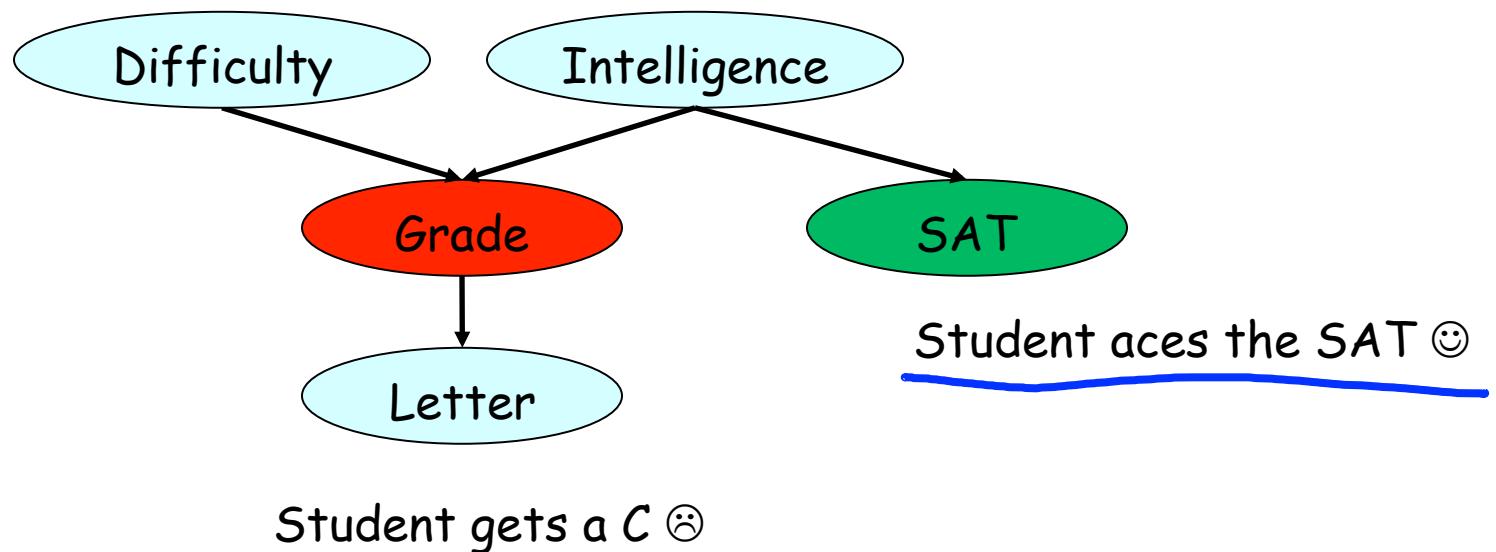
Intercausal Reasoning II

$$\begin{aligned}P(i^1) &= 0.3 \\P(i^1 | g^2) &\approx 0.175 \\P(i^1 | g^2, d^1) &\approx 0.34\end{aligned}$$



Student Aces the SAT

- What happens to the posterior probability that the class is hard?



Student Aces the SAT

$$P(d^1) = 0.4$$

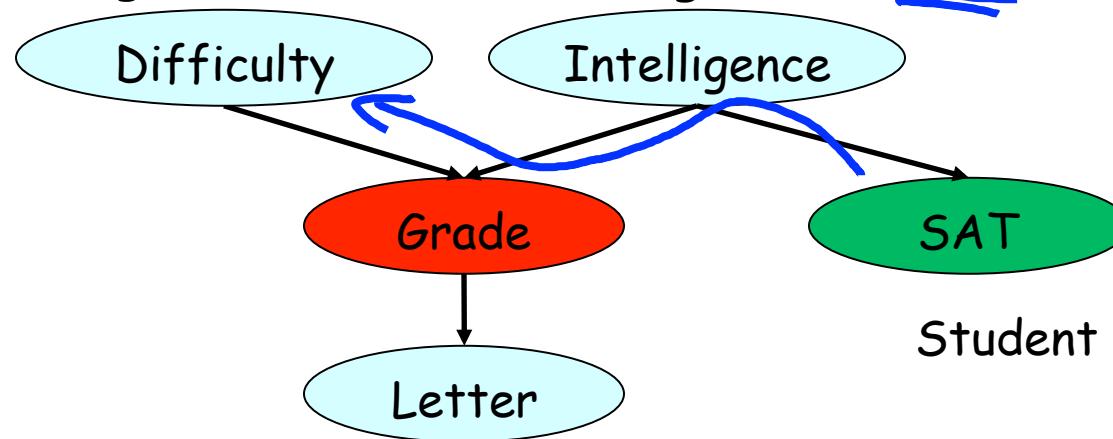
$$P(d^1 | g^3) \approx 0.63$$

$$P(d^1 | g^3, s^1) \approx \underline{0.76}$$

$$P(i^1) = 0.3$$

$$P(i^1 | g^3) \approx \underline{0.08}$$

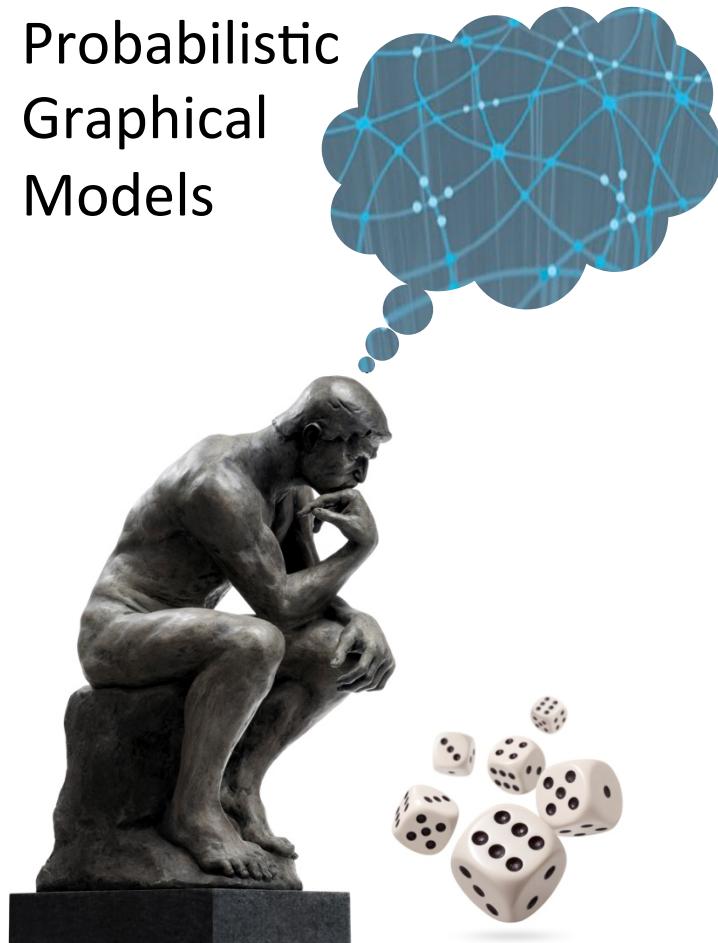
$$P(i^1 | g^3, s^1) \approx \underline{0.58}$$



Student aces the SAT ☺

Student gets a C ☹

Probabilistic
Graphical
Models



Representation

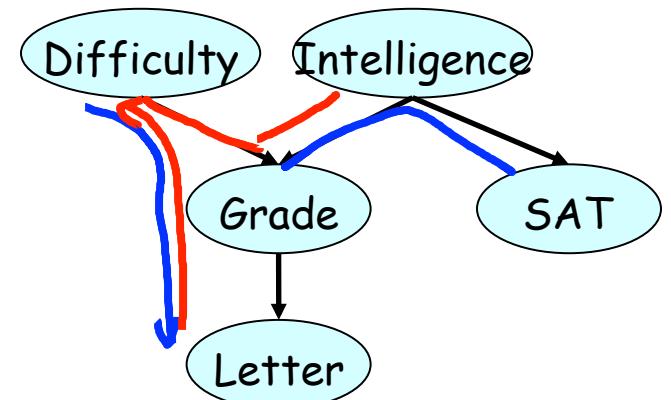
Bayesian Networks

Flow of
Probabilistic
Influence

When can X influence Y?

Condition on v-structure before inserting Y

- $X \rightarrow Y$ ✓
- $X \leftarrow Y$ ✓
- $X \rightarrow W \rightarrow Y$ ✓
- $X \leftarrow W \leftarrow Y$ ✓
- $X \leftarrow \underline{W} \rightarrow Y$ ✓
- $X \rightarrow \underline{W} \leftarrow Y$ ✗
v-structure



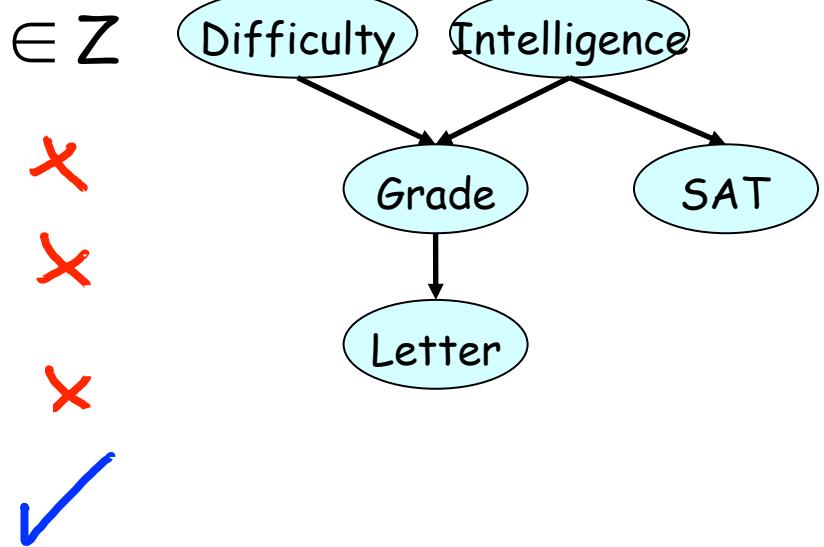
Active Trails

- A trail $X_1 - \dots - X_n$ is active if:
it has no v-structures $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$

When can X influence Y Given evidence about Z

- $X \rightarrow Y$
- $X \leftarrow Y$
- $X \rightarrow W \rightarrow Y$ $W \notin Z$ ✓
- $X \leftarrow W \leftarrow Y$ ✓
- $X \leftarrow W \rightarrow Y$ ✓
- $X \rightarrow W \leftarrow Y$ ✗

$W \in Z$



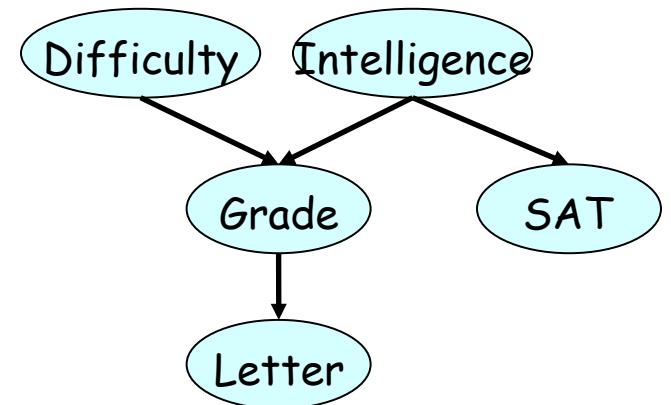
When can X influence Y given evidence about Z

- S – I – G – D allows influence to flow when:

I is observed X

I not observed,
nothing else X

I not observed
& G is observed

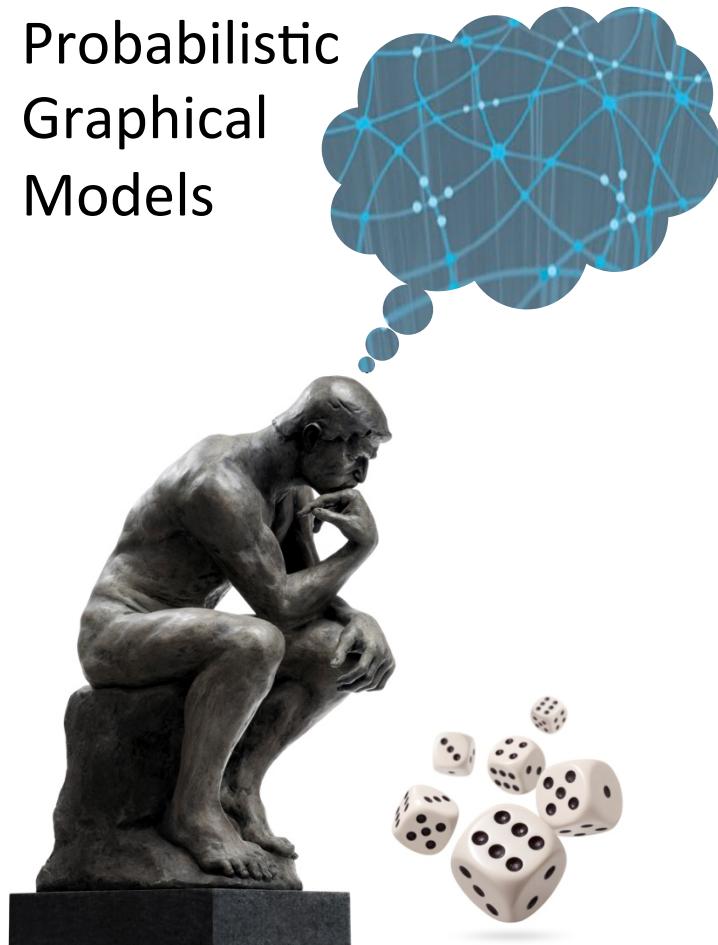


Active Trails

- A trail $X_1 - \dots - X_n$ is active given Z if:

- for any v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$ we have that X_i or one of its descendants $\in Z$
 - no other X_i is in Z
not in v-structure

Probabilistic
Graphical
Models



Representation

Independencies

Preliminaries

Independence

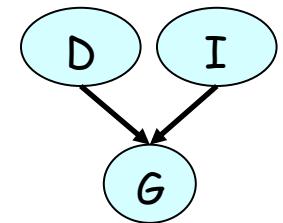
- For events α, β , $P \models \alpha \perp \beta$ if:
 - $P(\alpha \cap \beta) = P(\alpha) \cdot P(\beta)$ satisfies independence
 - $P(\alpha | \beta) = P(\alpha)$
 - $P(\beta | \alpha) = P(\beta)$
- For random variables X, Y , $P \models X \perp Y$ if:
 - $P(X, Y) = P(X) P(Y)$
 - $P(X | Y) = P(X)$ universal $P(x,y) = P(x) \cdot P(y)$
 - $P(Y | X) = P(Y)$

Independence

I	D	G	Prob.
i ⁰	d ⁰	g ¹	0.126
i ⁰	d ⁰	g ²	0.168
i ⁰	d ⁰	g ³	0.126
i ⁰	d ¹	g ¹	0.009
i ⁰	d ¹	g ²	0.045
i ⁰	d ¹	g ³	0.126
i ¹	d ⁰	g ¹	0.252
i ¹	d ⁰	g ²	0.0224
i ¹	d ⁰	g ³	0.0056
i ¹	d ¹	g ¹	0.06
i ¹	d ¹	g ²	0.036
i ¹	d ¹	g ³	0.024

$P(I, D) =$

I	D	Prob
i ⁰	d ⁰	0.42
i ⁰	d ¹	0.18
i ¹	d ⁰	0.28
i ¹	d ¹	0.12



$P(I)$

I	Prob
i ⁰	0.6
i ¹	0.4

$P(D)$

D	Prob
d ⁰	0.7
d ¹	0.3

Conditional Independence

- For (sets of) random variables X, Y, Z

$P \models (X \perp Y \mid Z)$ if:

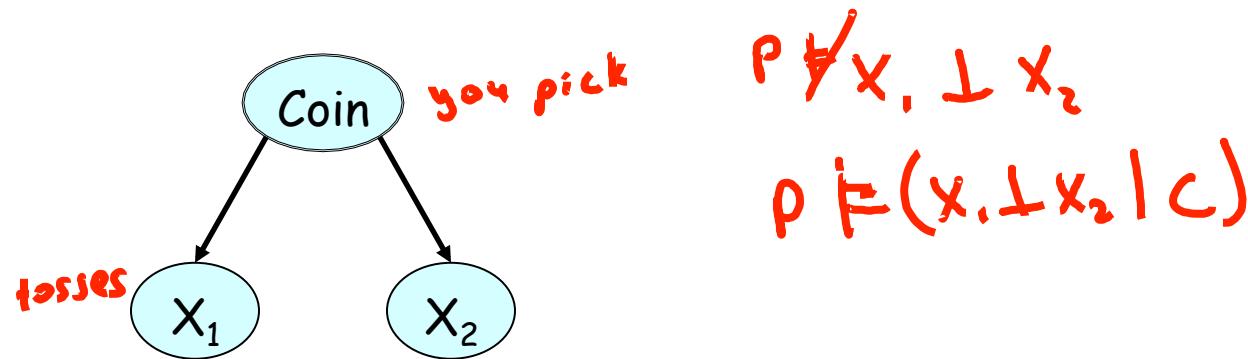
$$- P(X, Y \mid Z) = P(X \mid Z) P(Y \mid Z)$$

$$- P(X \mid Y, Z) = P(X \mid Z)$$

$$- P(Y \mid X, Z) = P(Y \mid Z)$$

$$- P(X, Y, Z) \propto \phi_1(X, Z) \phi_2(Y, Z)$$

Conditional Independence



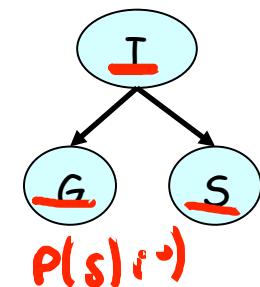
Conditional Independence

$P(I, S, G)$

I	S	G	Prob.
i^0	s^0	g^1	0.114
i^0	s^0	g^2	0.1938
i^0	s^0	g^3	0.2622
i^0	s^1	g^1	0.006
i^0	s^1	g^2	0.0102
i^0	s^1	g^3	0.0138
i^1	s^0	g^1	0.252
i^1	s^0	g^2	0.0224
i^1	s^0	g^3	0.0056
i^1	s^1	g^1	0.108
i^1	s^1	g^2	0.0096
i^1	s^1	g^3	0.0024

$P(S, G | \underline{i^0})$

S	G	Prob.
s^0	g^1	0.19
s^0	g^2	0.323
s^0	g^3	0.437
s^1	g^1	0.01
s^1	g^2	0.017
s^1	g^3	0.023



S	Prob.
s^0	0.95
s^1	0.05

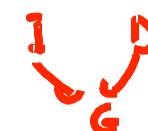
$P(g | i^0)$

G	Prob.
g^1	0.2
g^2	0.34
g^3	0.46

Daphne Koller

Conditioning can Lose Independences

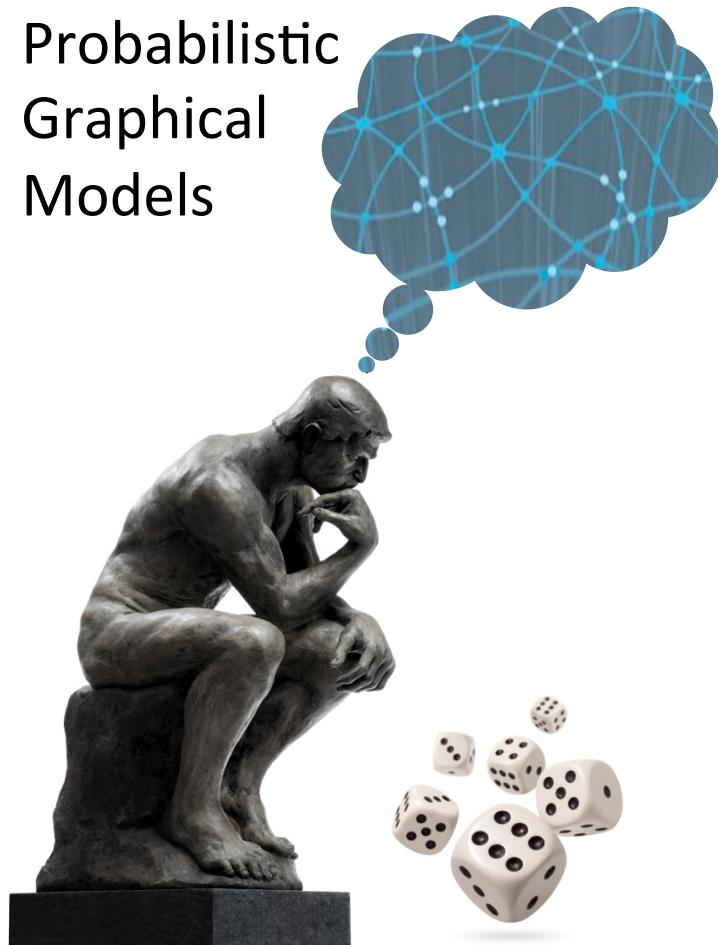
I	D	G	Prob.
i ⁰	d ⁰	g ¹	0.126
i ⁰	d ⁰	g ²	0.168
i ⁰	d ⁰	g ³	0.126
i ⁰	d ¹	g ¹	0.009
i ⁰	d ¹	g ²	0.045
i ⁰	d ¹	g ³	0.126
i ¹	d ⁰	g ¹	0.252
i ¹	d ⁰	g ²	0.0224
i ¹	d ⁰	g ³	0.0056
i ¹	d ¹	g ¹	0.06
i ¹	d ¹	g ²	0.036
i ¹	d ¹	g ³	0.024



$P(I, D | g^1)$

I	D	Prob.
i ⁰	d ⁰	0.282
i ⁰	d ¹	0.02
i ¹	d ⁰	0.564
i ¹	d ¹	0.134

Probabilistic
Graphical
Models



Representation

Independencies

Bayesian Networks

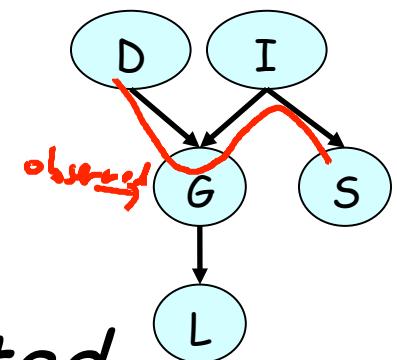
Independence & Factorization

$$P(X,Y) = P(X) P(Y) \quad X, Y \text{ independent}$$

$$P(X,Y,Z) \propto \phi_1(X,Z) \phi_2(Y,Z) \quad (X \perp Y \mid Z)$$

- Factorization of a distribution P implies independencies that hold in P
- If P factorizes over G , can we read these independencies from the structure of G ?

Flow of influence & d-separation



Definition: X and Y are d -separated in G given Z if there is no active trail in G between X and Y given Z

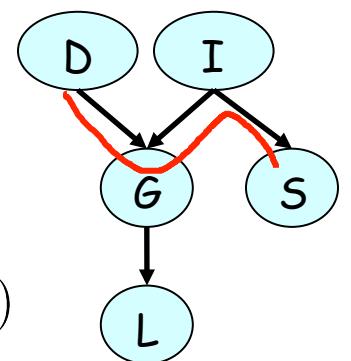
Notation: $d\text{-sep}_G(X, Y \mid Z)$

Factorization \Rightarrow Independence: BNs

Theorem: If P factorizes over G , and $d\text{-sep}_G(X, Y \mid Z)$
 then P satisfies $(X \perp Y \mid Z)$

$$P(D, I, G, S, L) = P(D)P(I)P(G \mid D, I)P(S \mid I)P(L \mid G) \quad \text{chain rule}$$

$P \models D \perp S$



$$\begin{aligned} P(D, S) &= \sum_{\substack{G, I}} P(D)P(I)P(G \mid D, I)P(S \mid I)P(L \mid G) \\ &= \sum_I P(D)P(I)P(S \mid I) \sum_G (P(G \mid D, I) \sum_L P(L \mid G)) \\ &= P(D) \left(\sum_I P(I)P(S \mid I) \right) \end{aligned}$$

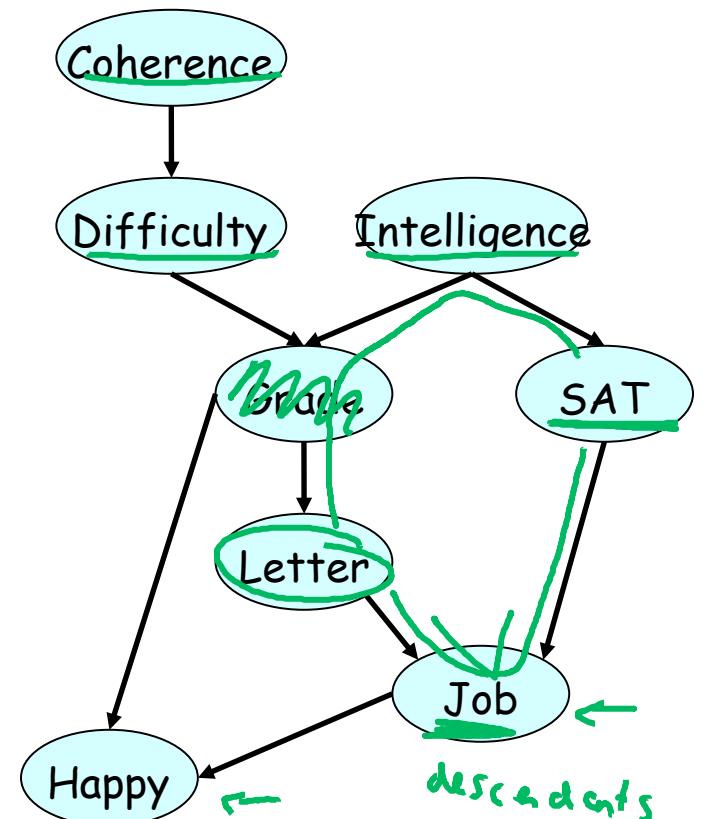
$\cancel{\phi_2(s)}$

J

Any node is d-separated from its non-descendants given its parents



If P factorizes over G, then in P, any variable is independent of its non-descendants given its parents



Daphne Koller

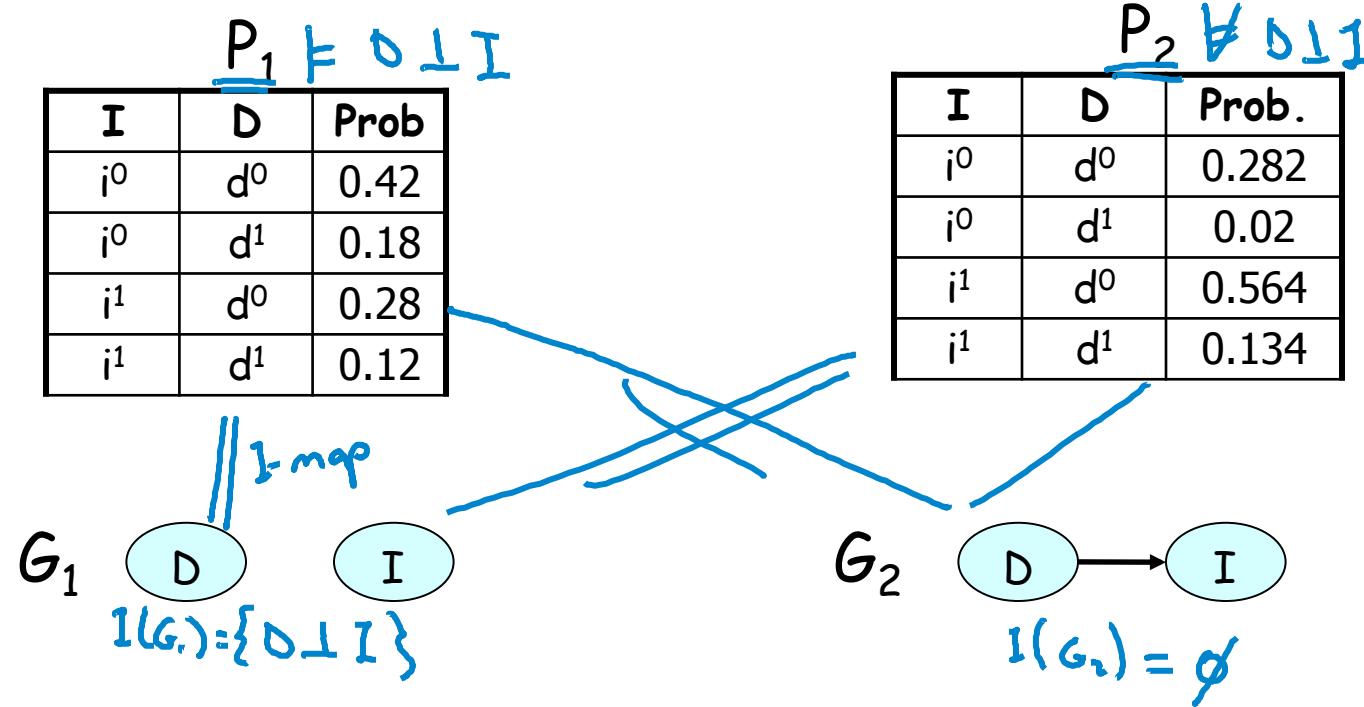
I-maps

- d-separation in $G \Rightarrow P$ satisfies corresponding independence statement

$$I(G) = \{(\underline{X} \perp \underline{Y} \mid \underline{Z}) : d\text{-sep}_G(\underline{X}, \underline{Y} \mid \underline{Z})\}$$

- Definition: If P satisfies $I(G)$, we say that G is an I-map (independency map) of P

I-maps



Factorization \Rightarrow Independence: BNs

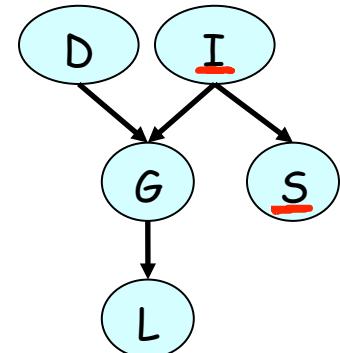
Theorem: If P factorizes over G, then G is
an I-map for P

Can read from G independencies in P
regardless of parameters

Independence \Rightarrow Factorization

Theorem: If G is an I-map for P, then P factorizes over G

IID



P(I,D) chain rule for probabilities

$$P(D, I, G, S, L) = \underbrace{P(D)}_{\text{P(I,D) chain rule for probabilities}} \underbrace{P(I | D)}_{\cancel{\text{P(I,D)}}} \underbrace{P(G | D, I)}_{\cancel{\text{P(G,I,D)}}} \underbrace{P(S | D, I, G)}_{\cancel{\text{P(S,I,G,D)}}} \underbrace{P(L | D, I, G, S)}_{\cancel{\text{P(L,I,G,D,S)}}}$$

$$P(D, I, G, S, L) = P(D)P(I)P(G | D, I)P(S | I)P(L | G)$$

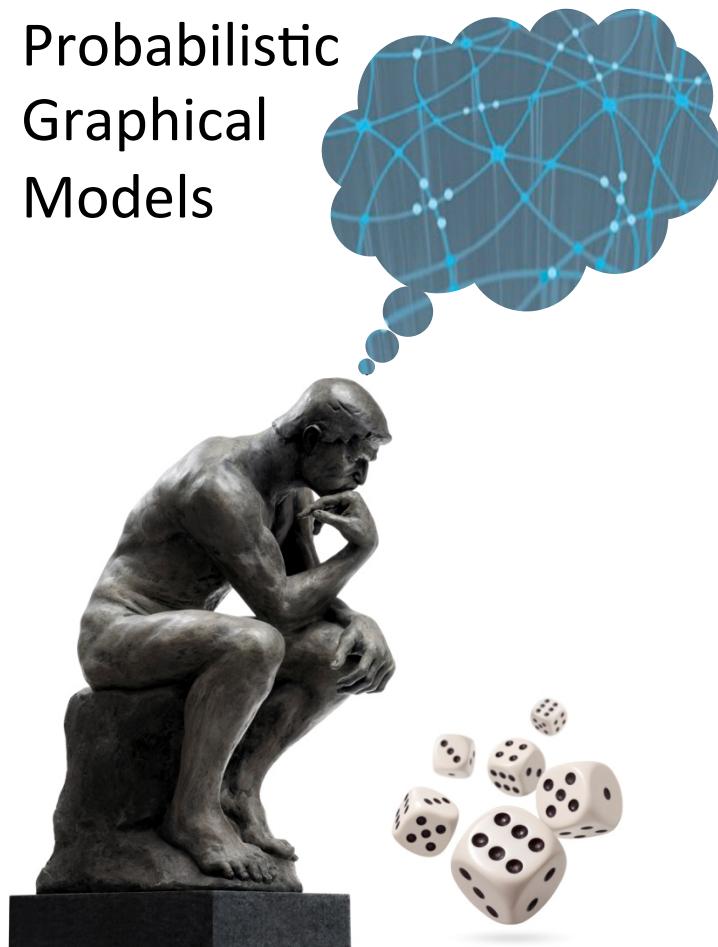
Summary

Two equivalent views of graph structure:

- Factorization: G allows P to be represented
- I-map: Independencies encoded by G hold in P

If P factorizes over a graph G , we can read from the graph independencies that must hold in P (an independency map)

Probabilistic
Graphical
Models

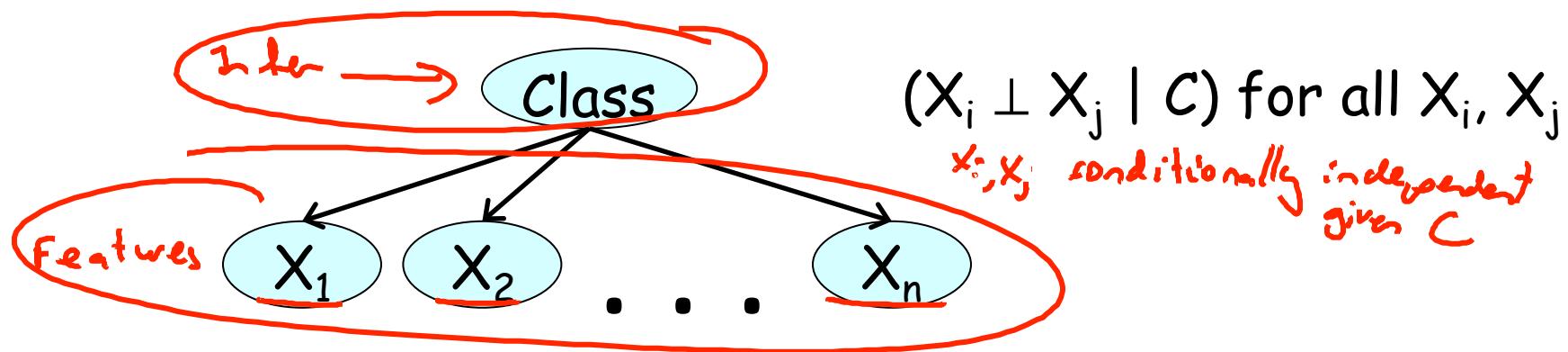


Representation

Bayesian Networks

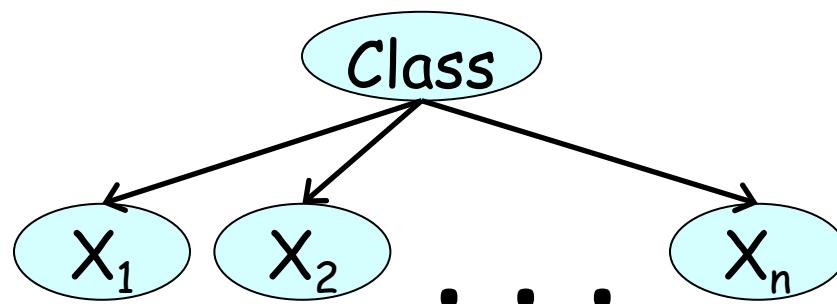
Naïve Bayes

Naïve Bayes Model



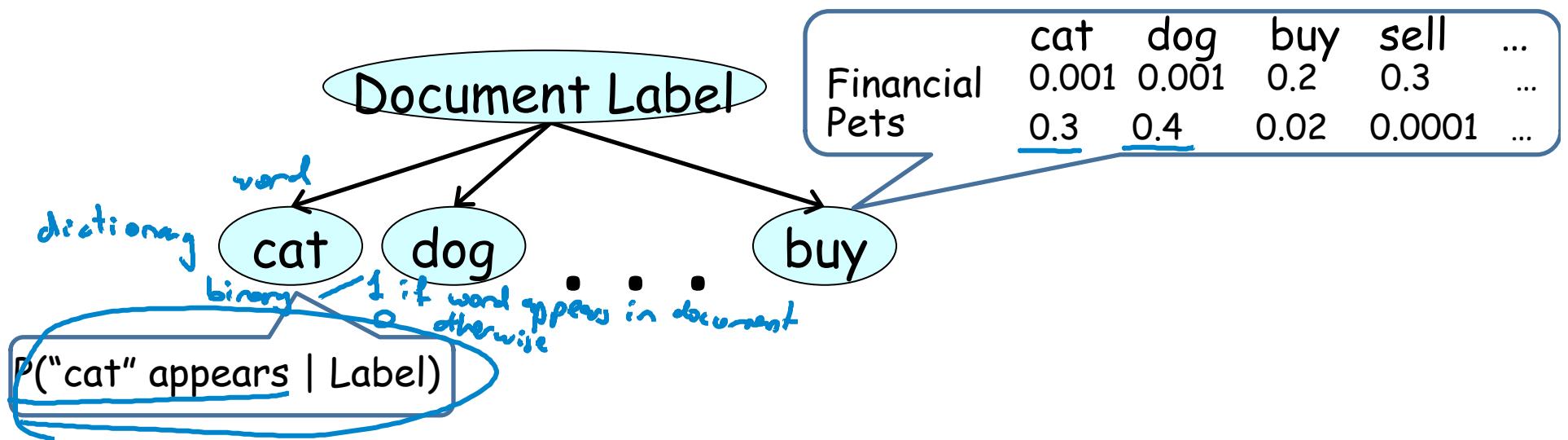
$$\underline{P(C, X_1, \dots, X_n)} = \underbrace{P(C)}_{i=1} \prod_{i=1}^n P(X_i \mid C)$$

Naïve Bayes Classifier



$$\frac{P(C = c^1 \mid x_1, \dots, x_n)}{P(C = c^2 \mid x_1, \dots, x_n)} = \underbrace{\frac{P(C = c^1)}{P(C = c^2)}}_{\text{odds ratios}} \prod_{i=1}^n \frac{P(x_i \mid C = c^1)}{P(x_i \mid C = c^2)}$$

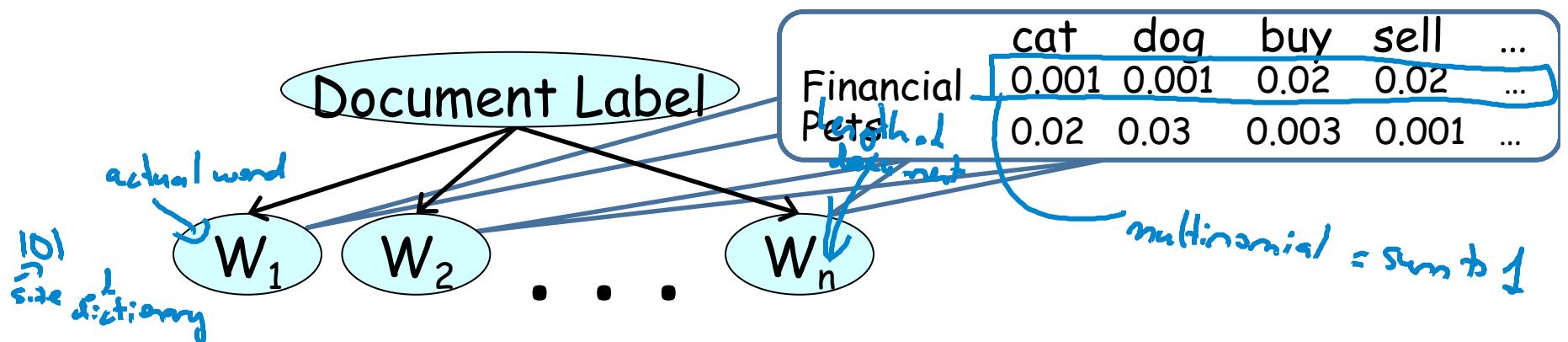
Bernoulli Naïve Bayes for Text



$$\frac{P(C = c^1 \mid x_1, \dots, x_n)}{P(C = c^2 \mid x_1, \dots, x_n)} = \frac{P(C = c^1)}{P(C = c^2)} \prod_{i=1}^n \frac{P(x_i \mid C = c^1)}{P(x_i \mid C = c^2)}$$

Daphne Koller

Multinomial Naïve Bayes for Text



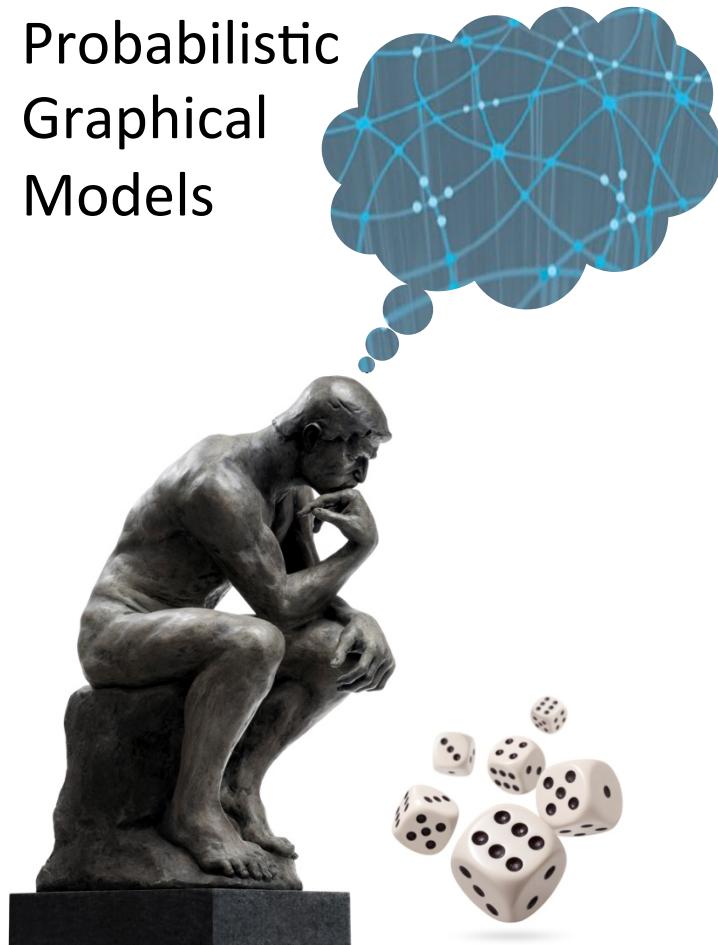
$$\frac{P(C = c^1 \mid x_1, \dots, x_n)}{P(C = c^2 \mid x_1, \dots, x_n)} = \frac{P(C = c^1)}{P(C = c^2)} \prod_{i=1}^n \frac{P(x_i \mid C = c^1)}{P(x_i \mid C = c^2)}$$

Daphne Koller

Summary

- Simple approach for classification
 - Computationally efficient
 - Easy to construct
- Surprisingly effective in domains with many weakly relevant features
- Strong independence assumptions reduce performance when many features are strongly correlated

Probabilistic
Graphical
Models



Representation

Bayesian Networks

Application:
Diagnosis

Medical Diagnosis: Pathfinder (1992)

- Help pathologist diagnose lymph node pathologies (60 different diseases)
- Pathfinder I: Rule-based system
- Pathfinder II used naïve Bayes and got superior performance

Heckerman et al.

Daphne Koller

Medical Diagnosis: Pathfinder (1992)

- Pathfinder III: Naïve Bayes with better knowledge engineering
- No incorrect zero probabilities
- Better calibration of conditional probabilities
 - $P(\text{finding} \mid \text{disease}_1)$ to $P(\text{finding} \mid \text{disease}_2)$
 - Not $P(\text{finding}_1 \mid \text{disease})$ to $P(\text{finding}_2 \mid \text{disease})$

Heckerman et al.

Daphne Koller

Medical Diagnosis: Pathfinder (1992)

- Pathfinder IV: Full Bayesian network
 - Removed incorrect independencies
 - Additional parents led to more accurate estimation of probabilities
- BN model agreed with expert panel in 50/53 cases, vs 47/53 for naïve Bayes model
- Accuracy as high as expert that designed the model

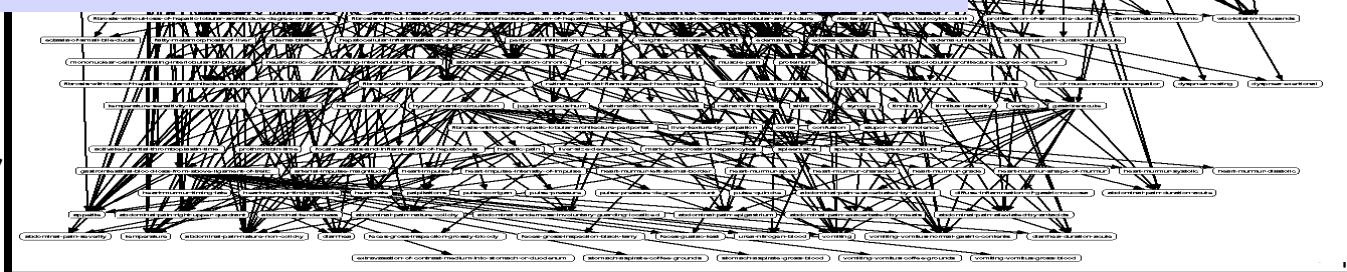
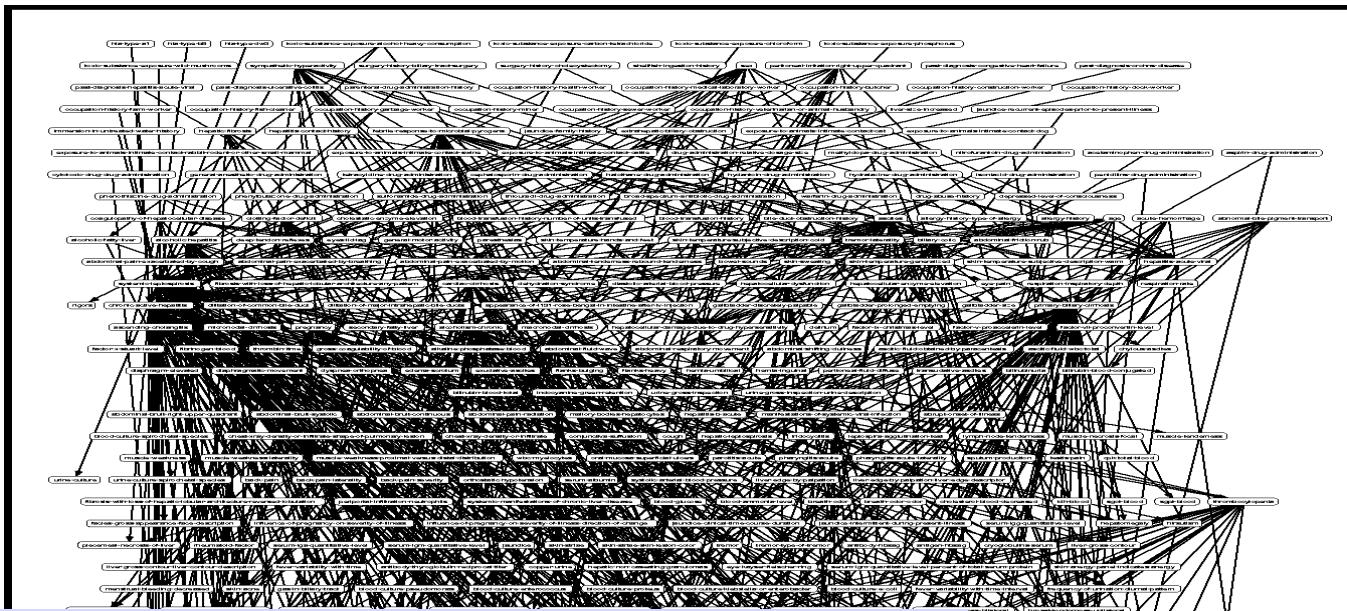
Heckerman et al.

Daphne Koller

CPCS

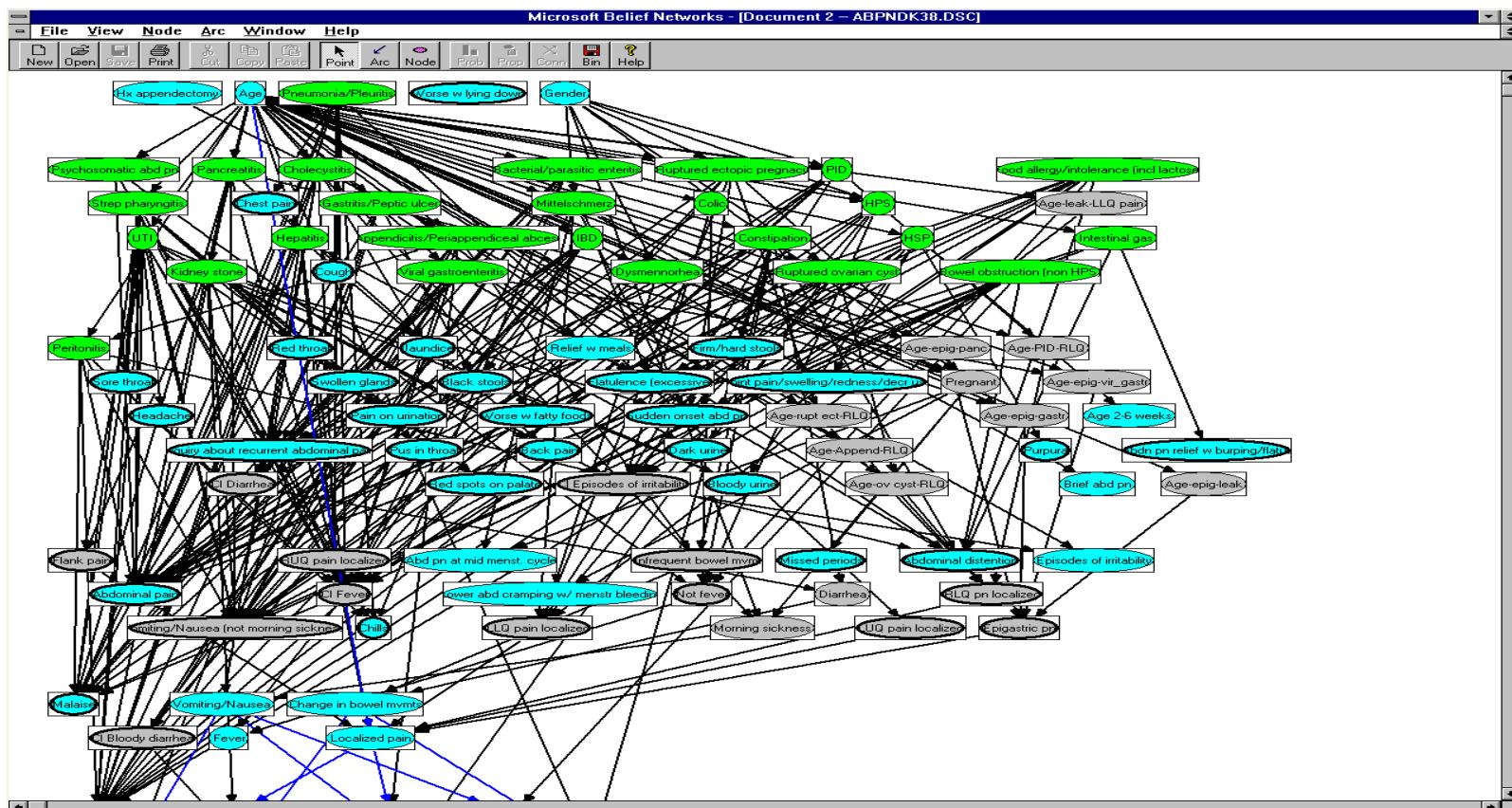
of parameters:
 2^{1000} to 133,931,430 to 8254

M. Pradhan , G. Provan ,
B. Middleton , M. Henrion,
UAI 94



hne Koller

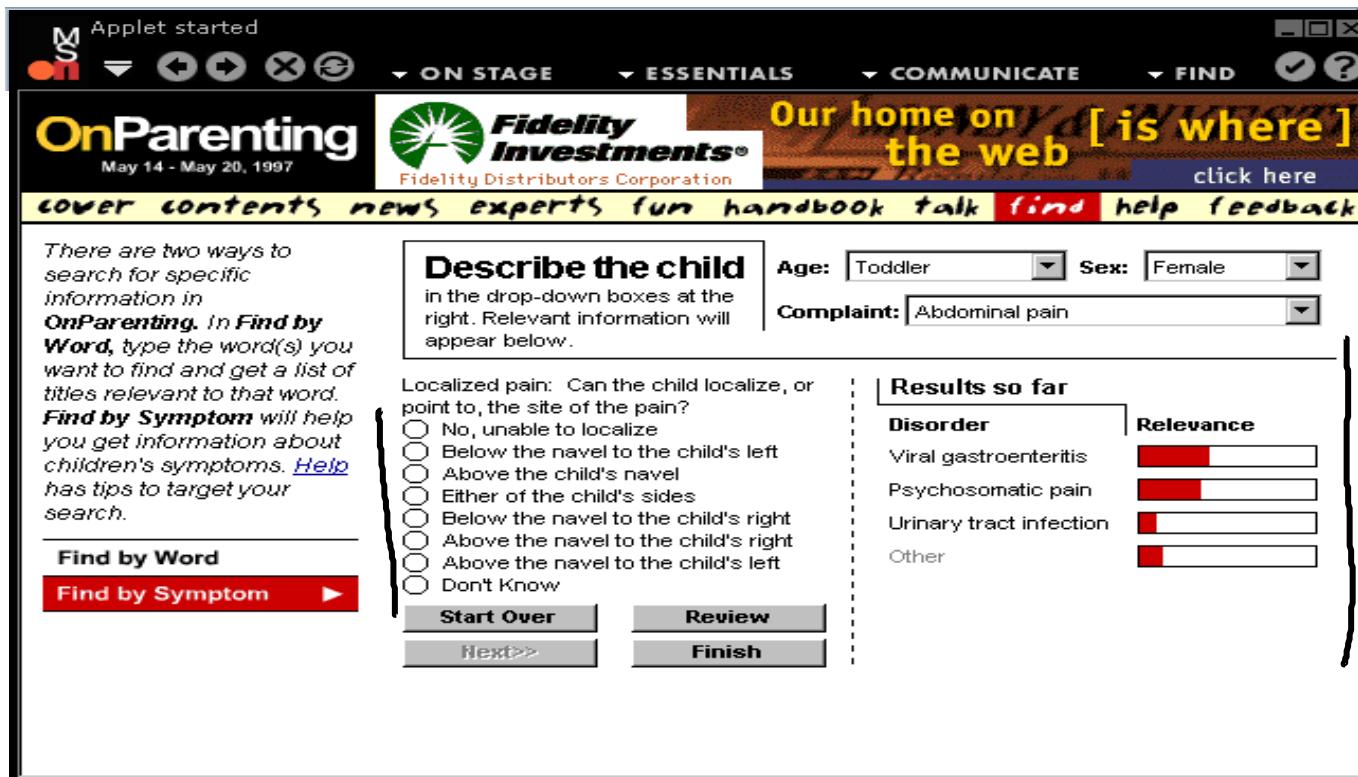
Medical Diagnosis (Microsoft)



Thanks to: Eric Horvitz, Microsoft Research

Daphne Koller

Medical Diagnosis (Microsoft)

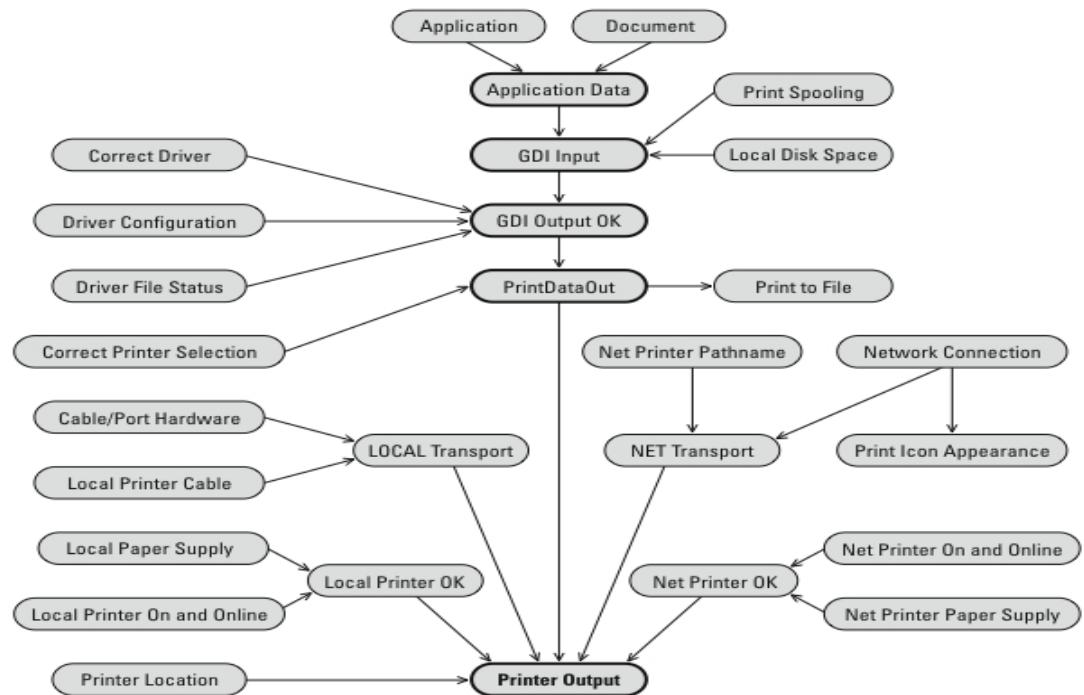


Thanks to: Eric Horvitz, Microsoft Research

Daphne Koller

Fault Diagnosis

- Microsoft troubleshooters

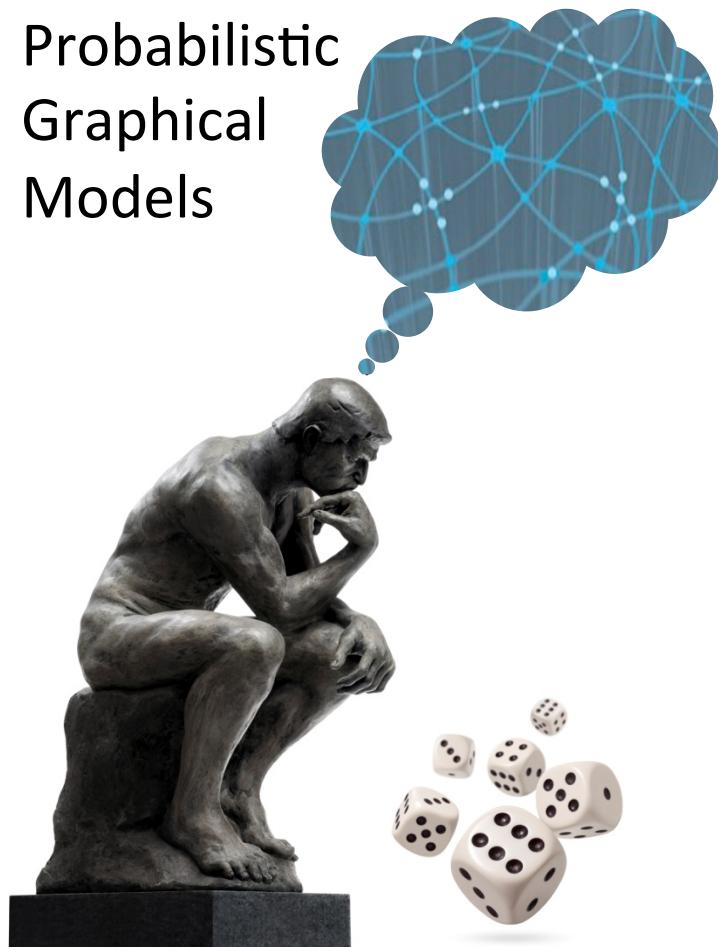


Daphne Koller

Fault Diagnosis

- Many examples:
 - Microsoft troubleshooters
 - Car repair
- Benefits:
 - Flexible user interface
 - Easy to design and maintain ←

Probabilistic
Graphical
Models



Representation

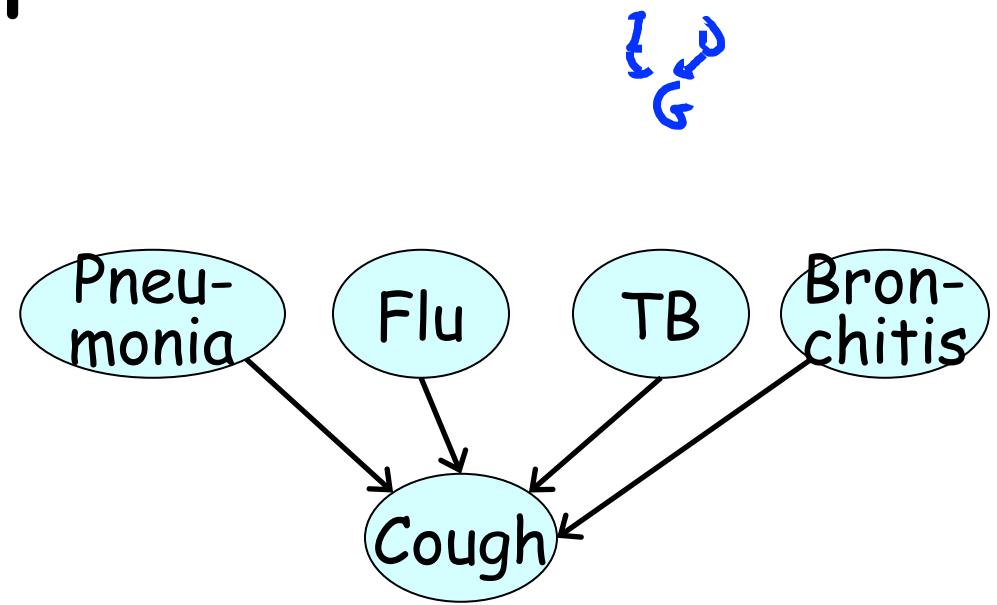
Local Structure

Overview

Tabular Representations

	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

k parents
 $\Theta(2^k)$ entries



General CPD

- CPD $P(X | Y_1, \dots, Y_k)$ specifies distribution over X for each assignment y_1, \dots, y_k
- Can use any function to specify a factor $\phi(X, Y_1, \dots, Y_k)$ such that

$$\sum_x \phi(x, y_1, \dots, y_k) = 1 \text{ for all } y_1, \dots, y_k$$

Many Models

- Deterministic CPDs
- Tree-structured CPDs
- Logistic CPDs & generalizations
- Noisy OR / AND
- Linear Gaussians & generalizations

Context-Specific Independence

$$P \models (\underline{X} \perp_c \underline{Y} \mid \underline{Z}, \underline{c})$$

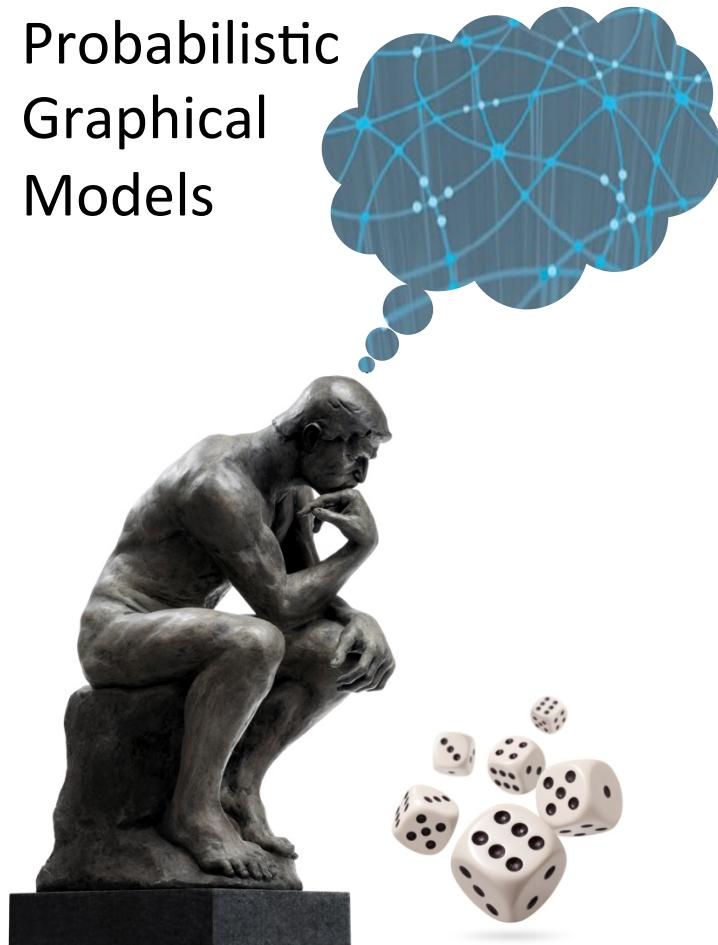
assignment to C

$$P(X, Y \mid \underline{Z}, \underline{c}) = P(X \mid \underline{Z}, \underline{c})P(y \mid \underline{Z}, \underline{c})$$

$$P(X \mid Y, \underline{Z}, \underline{c}) = P(X \mid \underline{Z}, \underline{c})$$

$$P(Y \mid X, \underline{Z}, \underline{c}) = P(Y \mid \underline{Z}, \underline{c})$$

Probabilistic
Graphical
Models

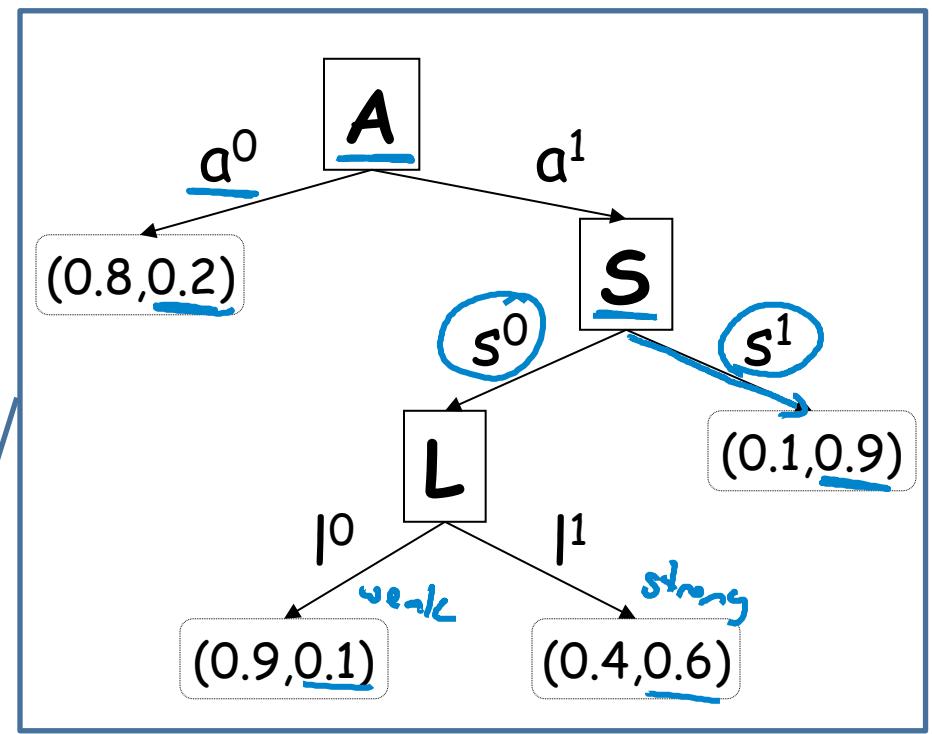
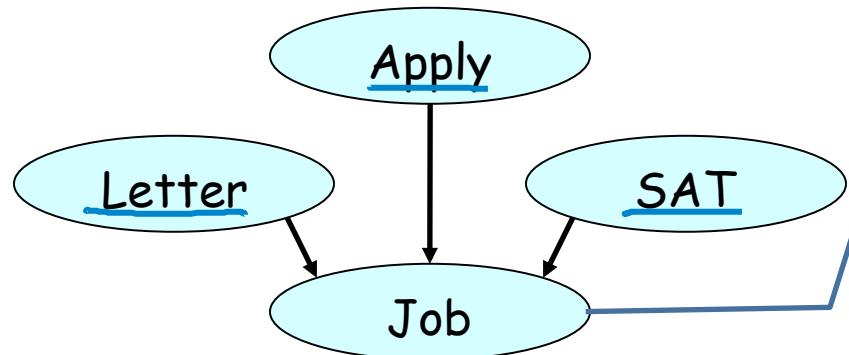


Representation

Local Structure

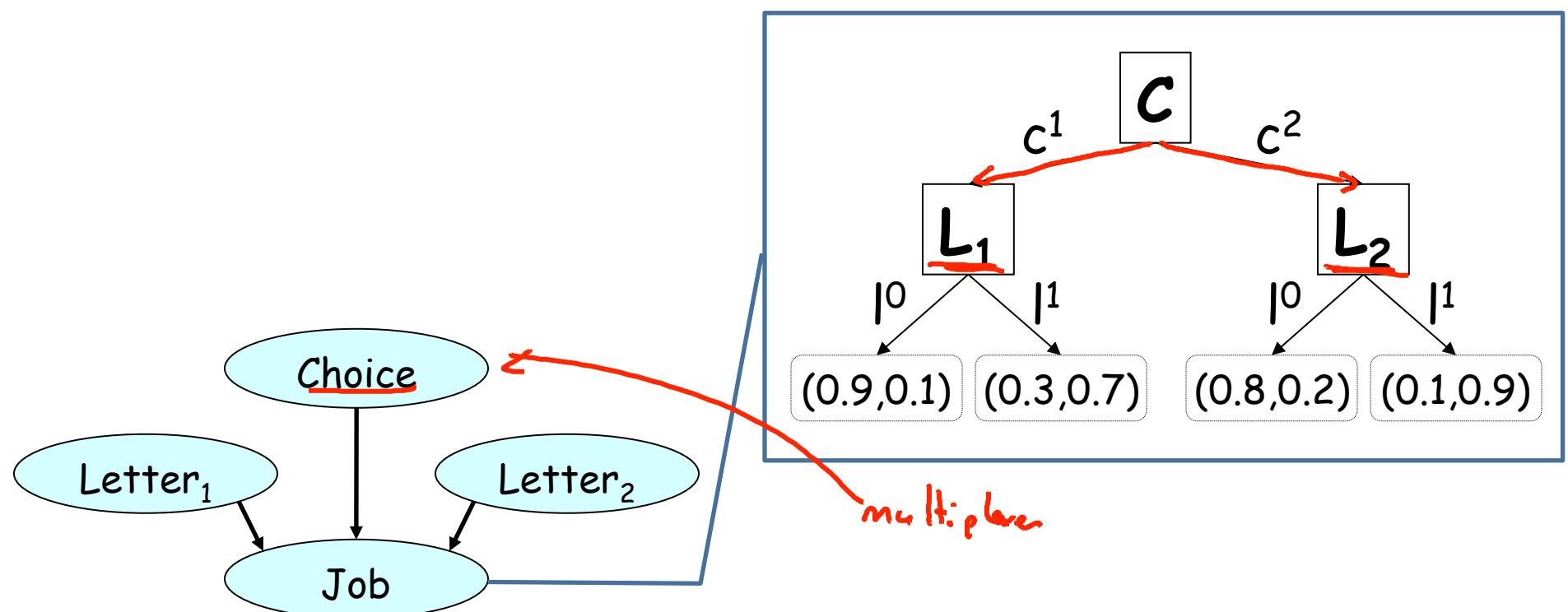
Tree
Structured
CPDs

Tree CPD



Daphne Koller

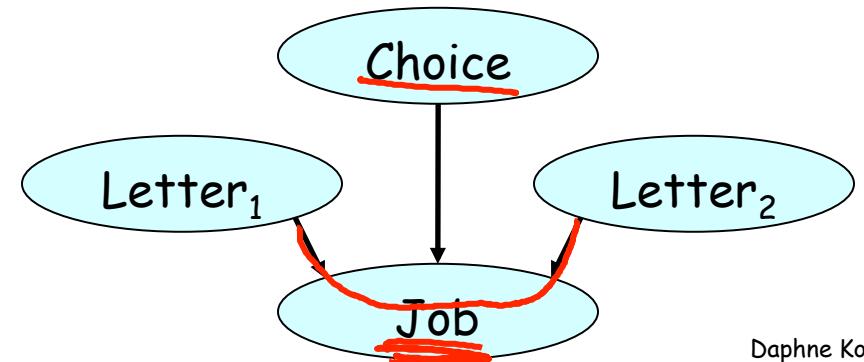
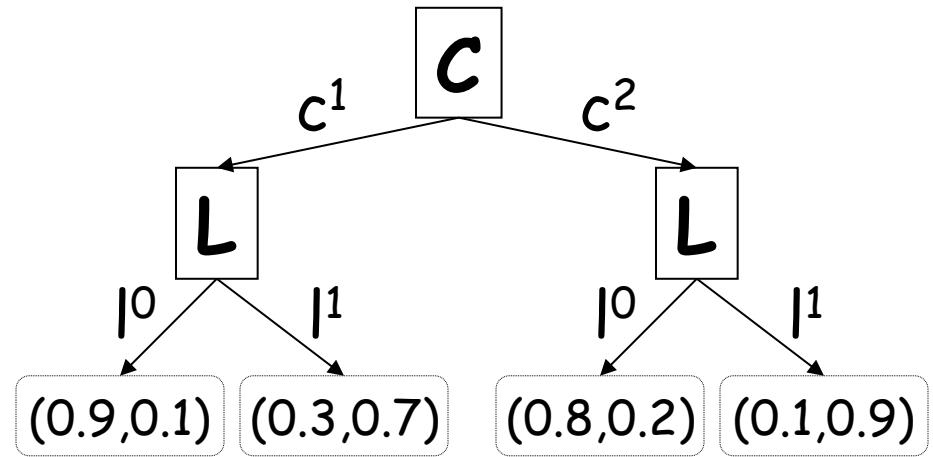
Tree CPD



$$(L_1 \perp L_2 \mid J, C)$$

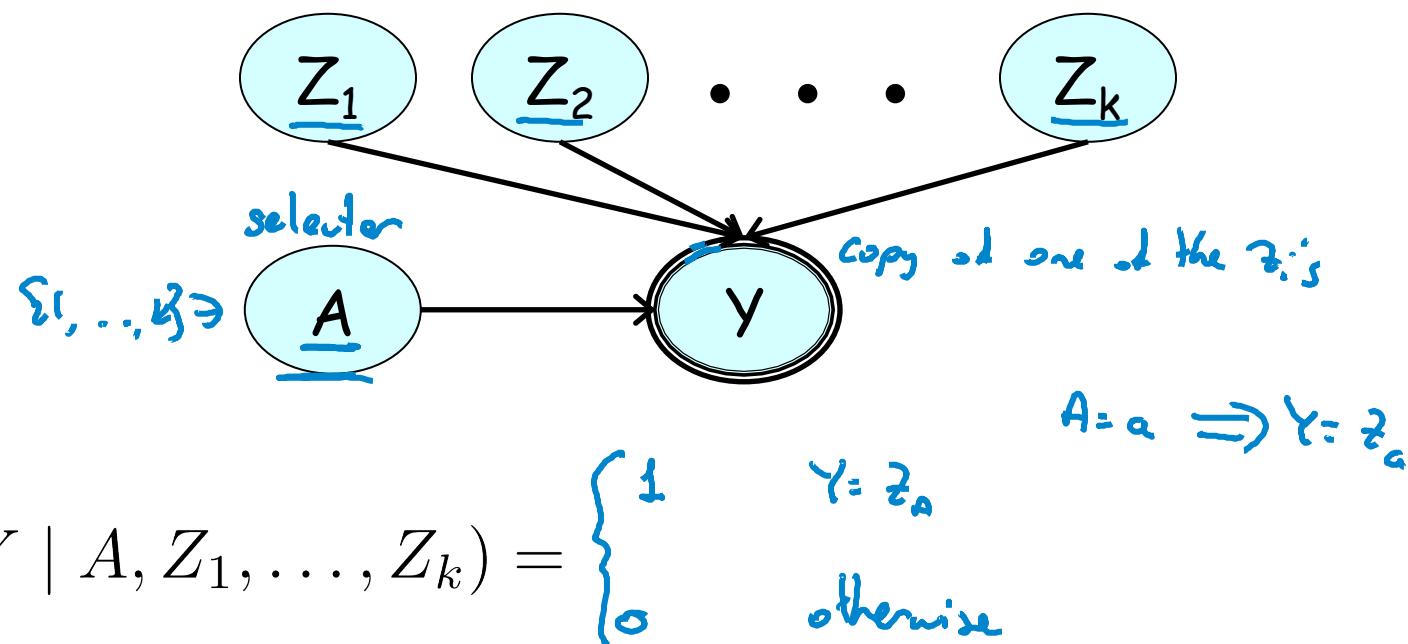
$$(L_1 \perp_c L_2 \mid J, c_1)$$

$$(L_1 \perp_c L_2 \mid J, c_2)$$



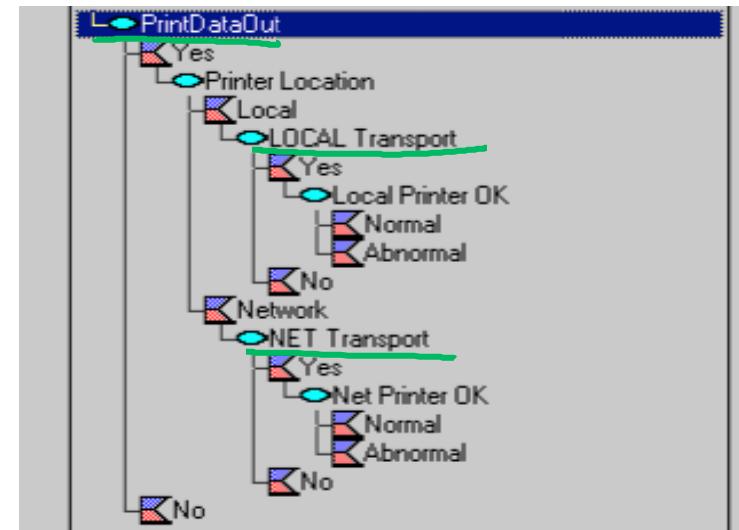
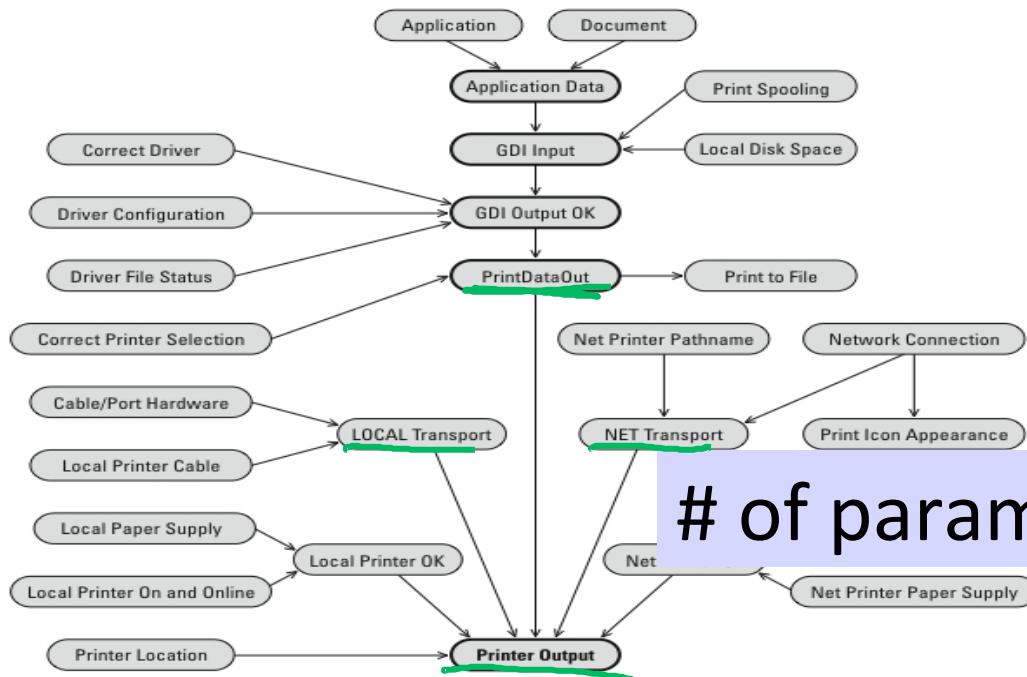
Daphne Koller

Multiplexer CPD



Thanks to: Eric Horvitz, Microsoft Research

Microsoft Troubleshooters



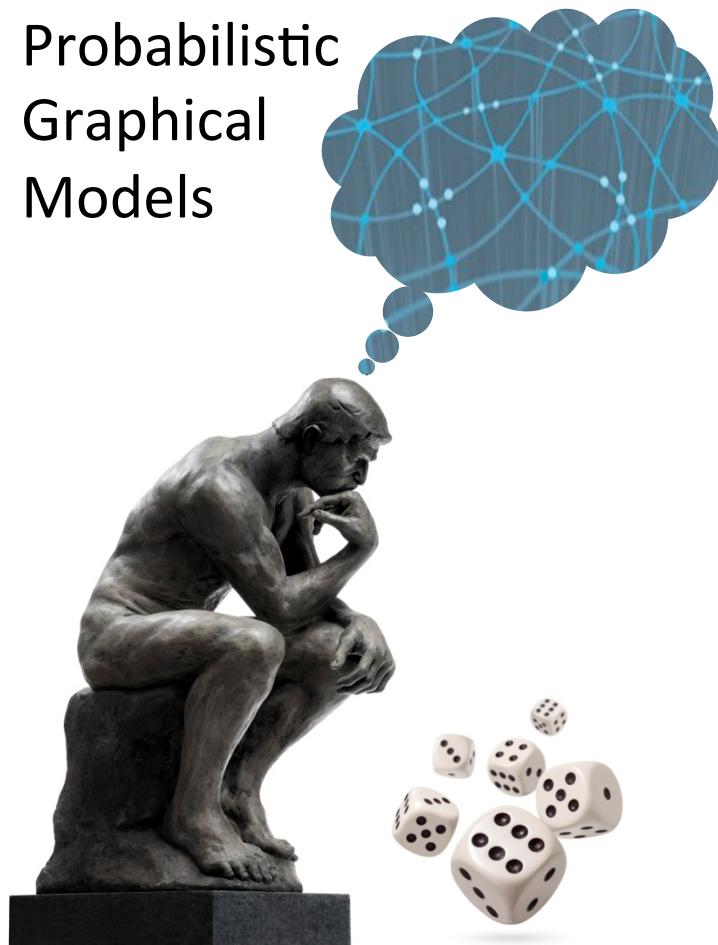
of parameters: 145 to 55

Daphne Koller

Summary

- Compact CPD representation that captures context-specific dependencies
- Relevant in multiple applications:
 - Hardware configuration variables
 - Medical settings
 - Dependence on agent's action
 - Perceptual ambiguity

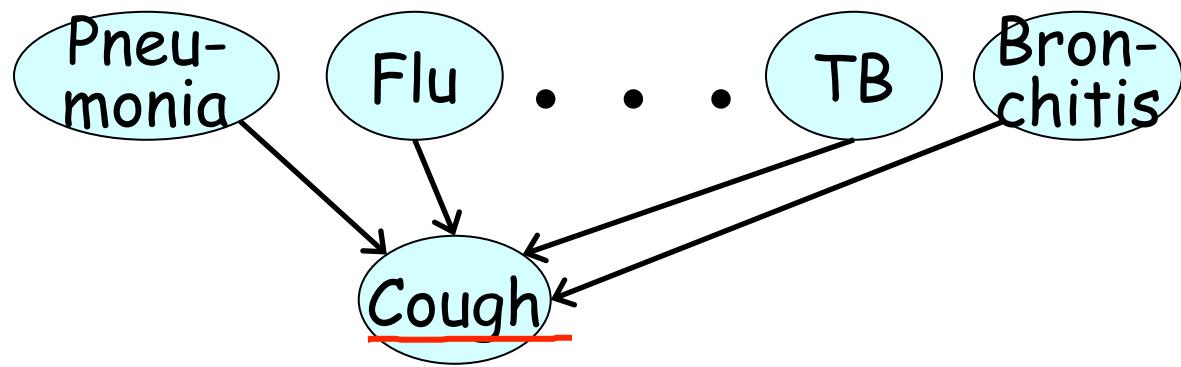
Probabilistic
Graphical
Models



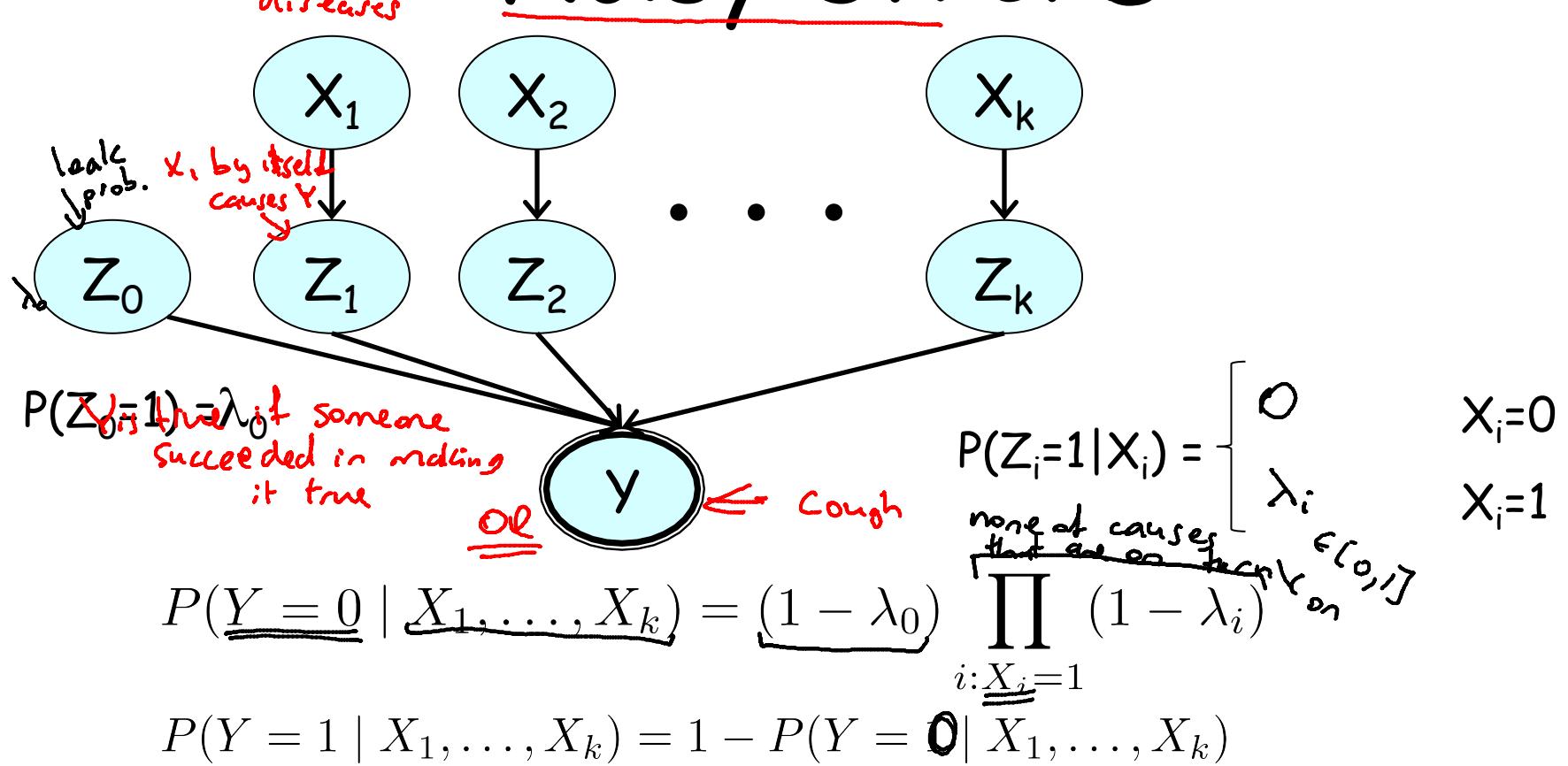
Representation

Local Structure

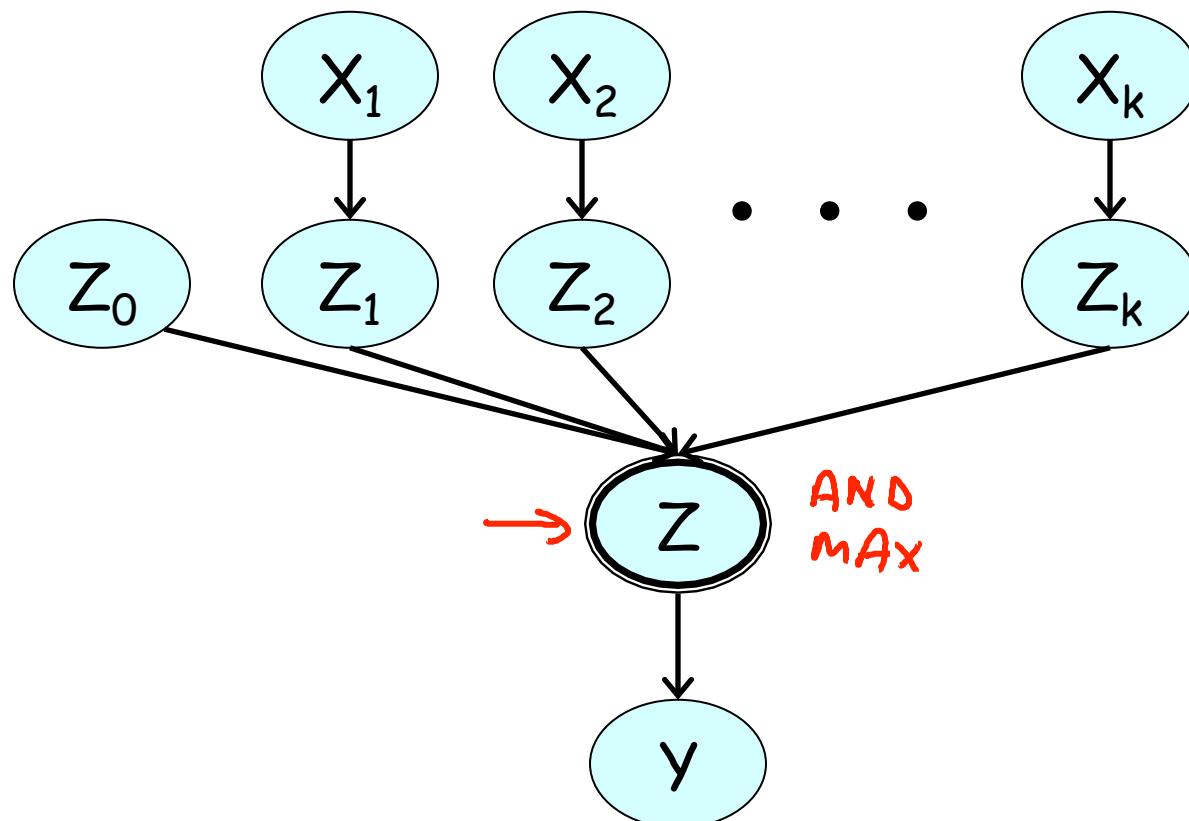
Independence
of Causal
Influence



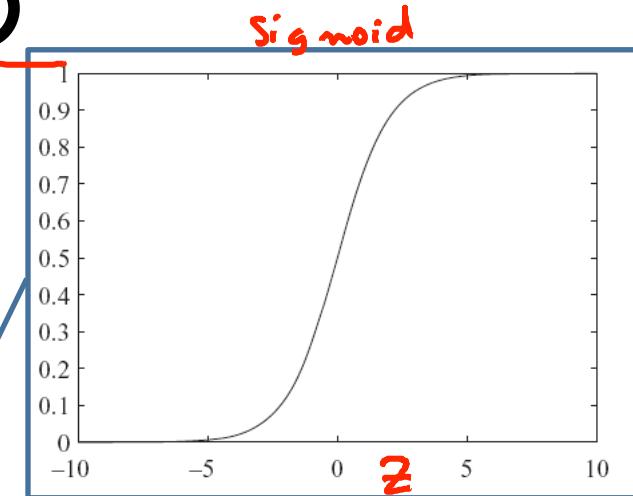
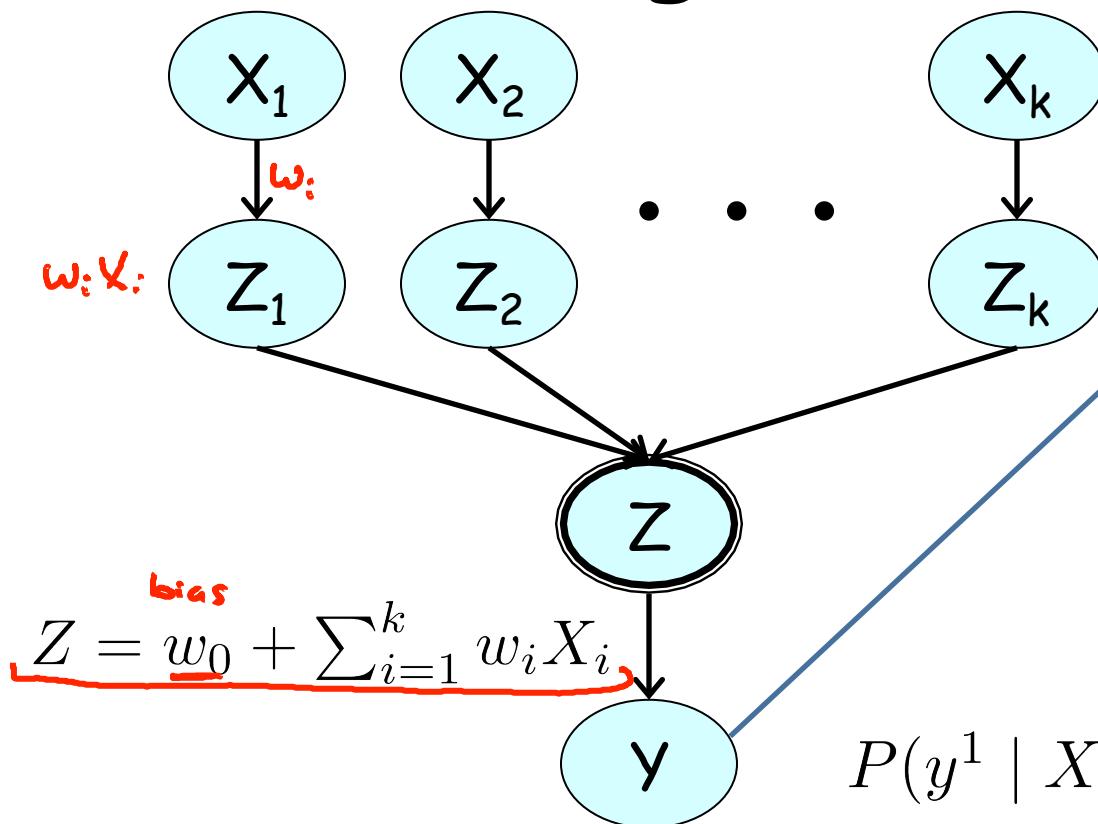
Noisy OR CPD



Independence of Causal Influence



Sigmoid CPD

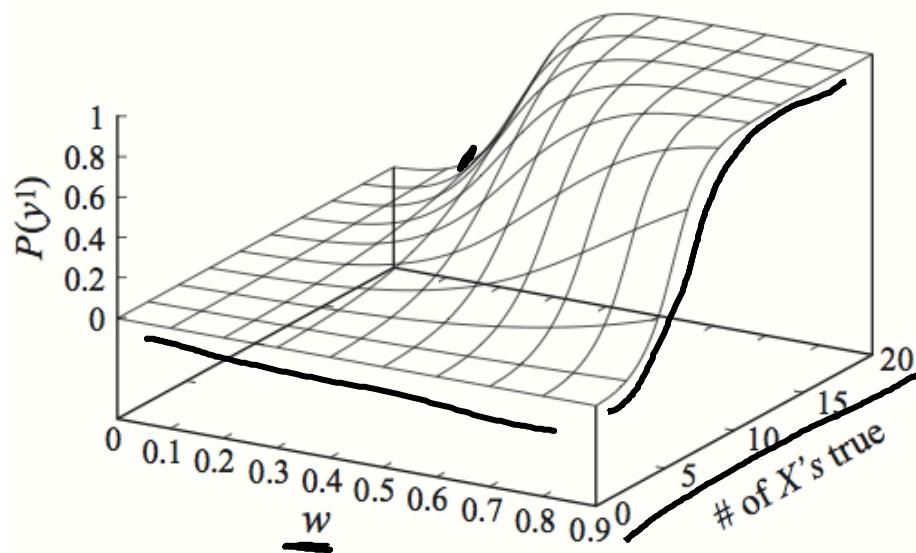


$$\text{sigmoid}(z) = \frac{e^z}{1 + e^z}$$

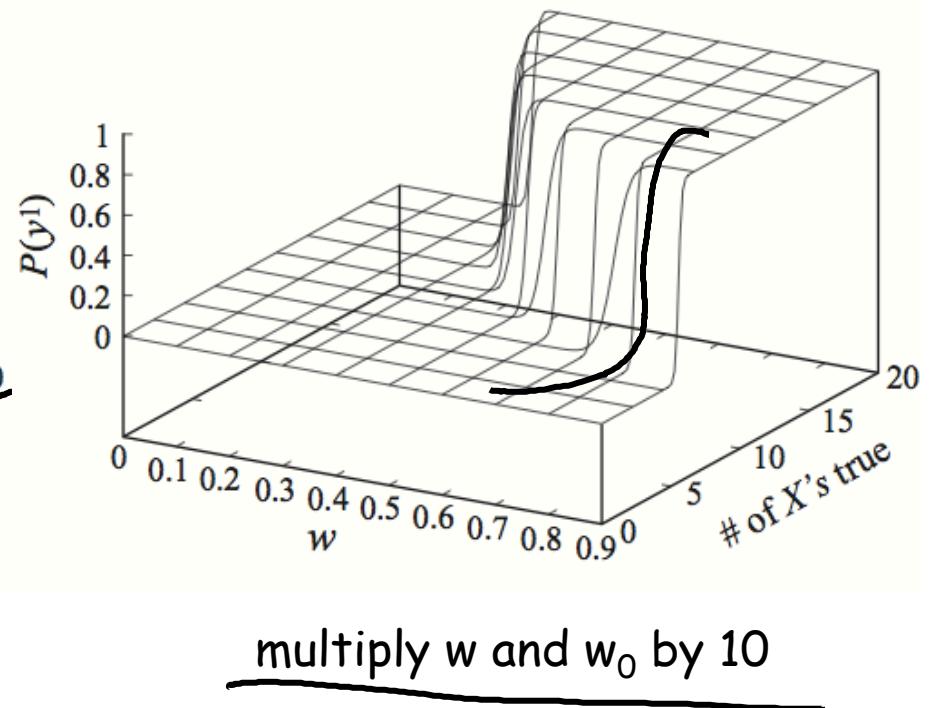
$$P(y^1 | X_1, \dots, X_k) = \text{sigmoid}(Z)$$

Daphne Koller

Behavior of Sigmoid CPD



$$w_0 = -5$$



Daphne Koller

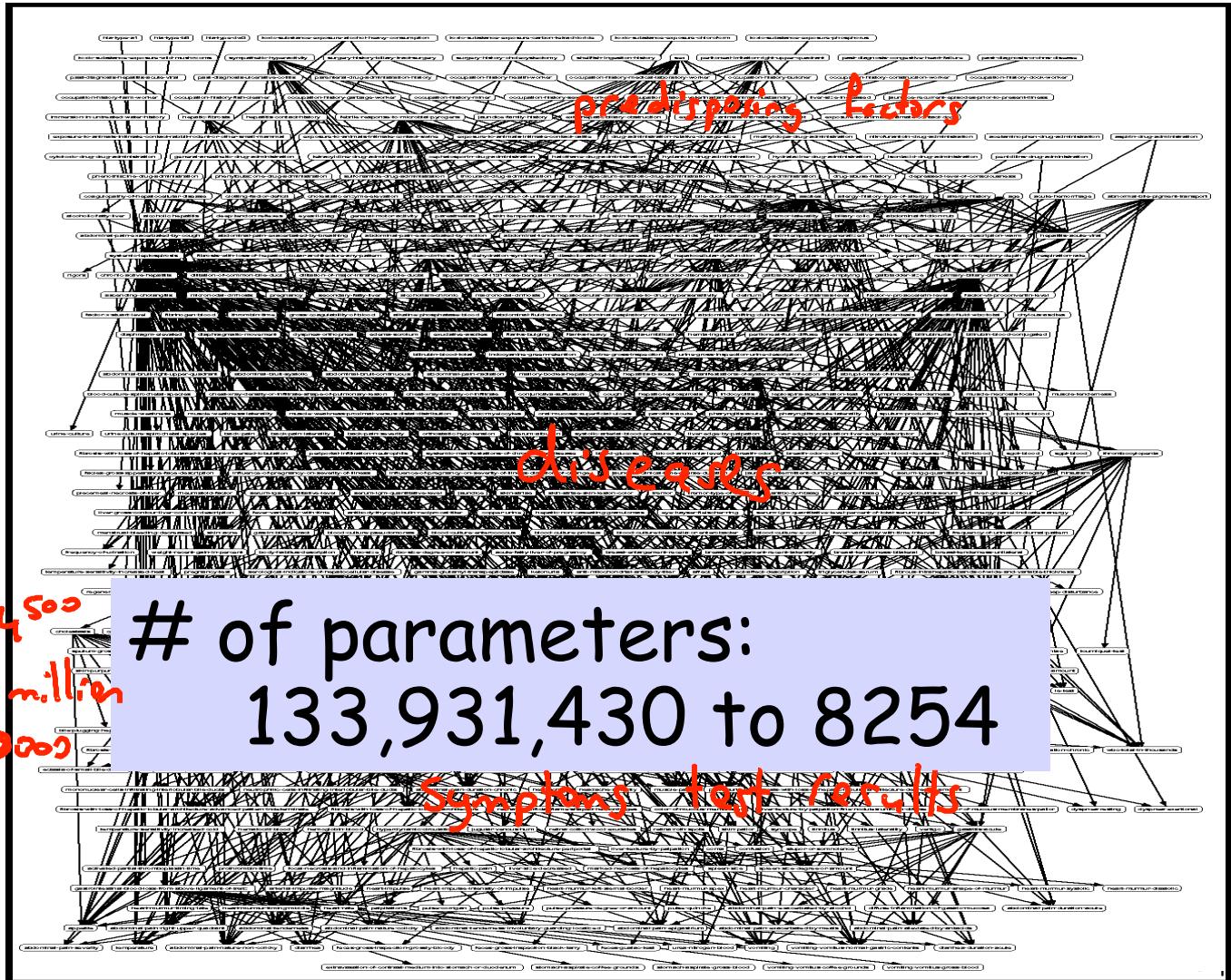
CPCS

M. Pradhan
G. Provan
B. Middleton
M. Henrion
UAI 1994

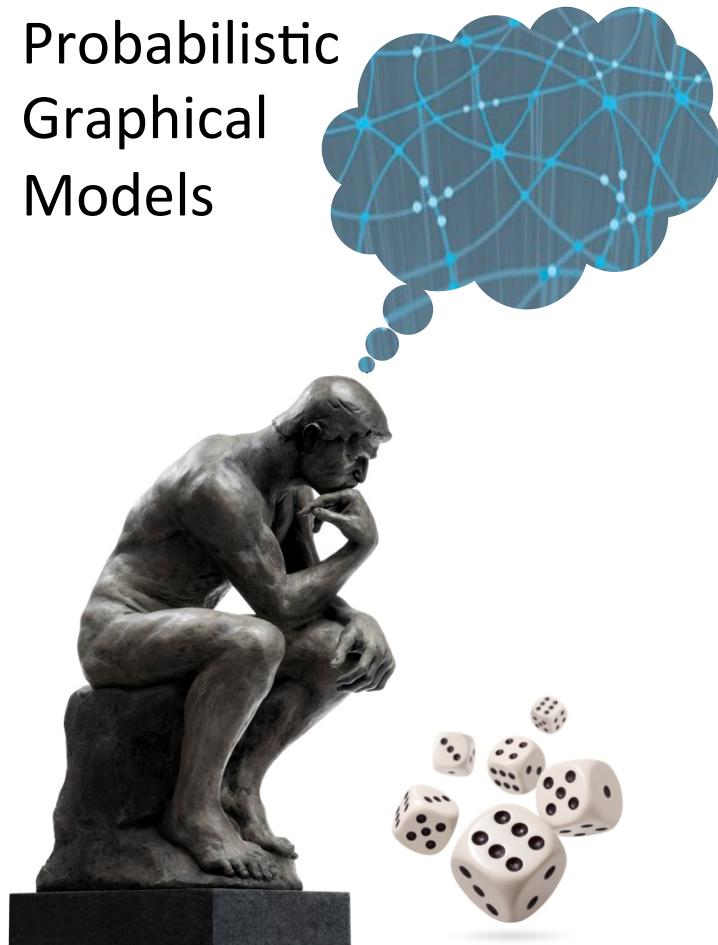
joint dist $\approx 4^{500}$
factorized ≈ 134 million
noisy max CPD ≈ 2000

of parameters:
133,931,430 to 8254

Symptoms test results



Probabilistic
Graphical
Models

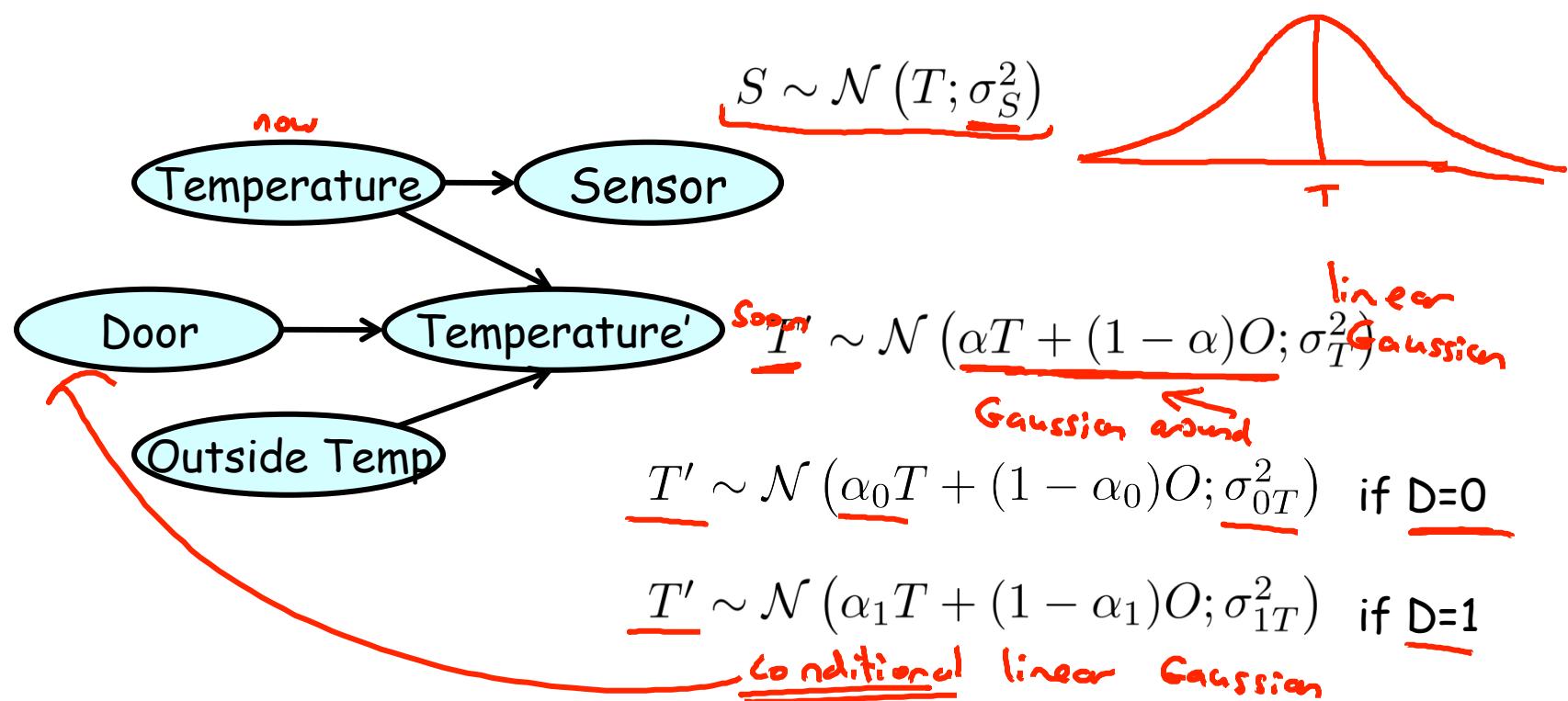


Representation

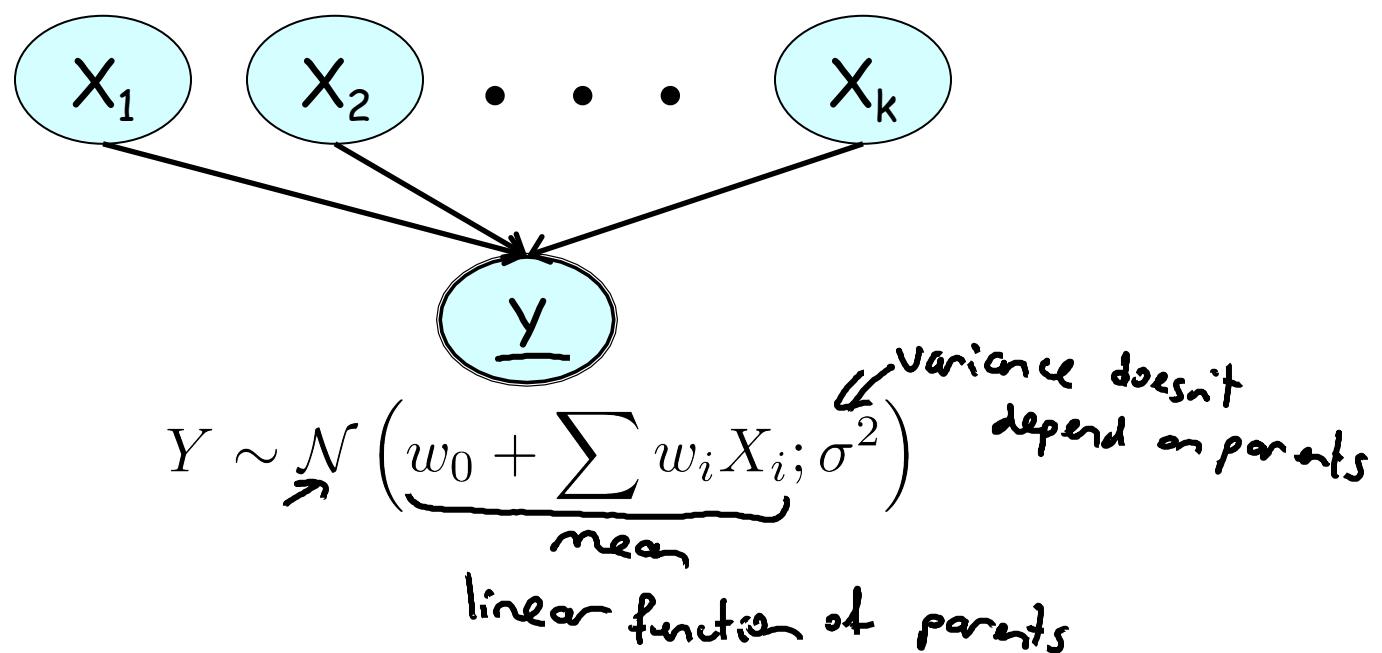
Local Structure

Continuous
Variables

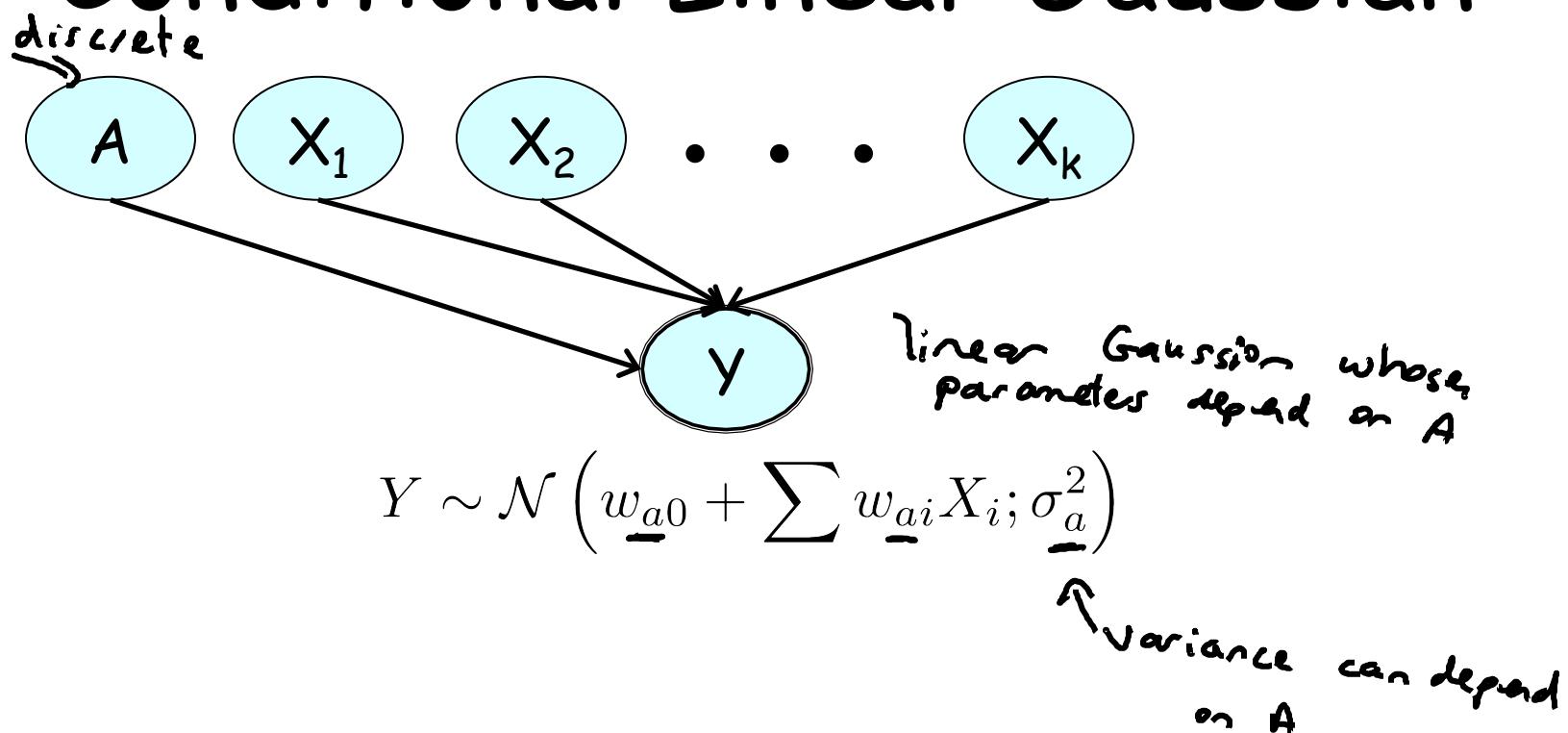
Continuous Variables



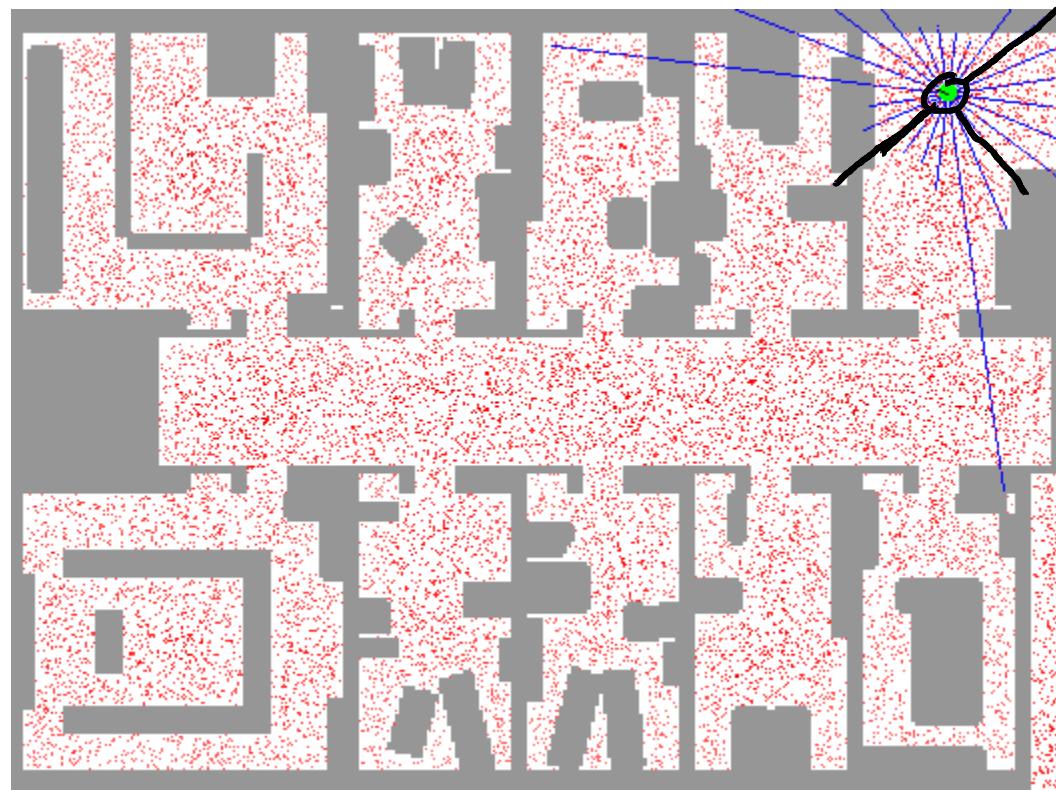
Linear Gaussian



Conditional Linear Gaussian

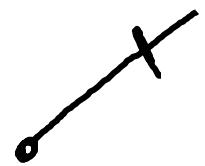


Robot Localization

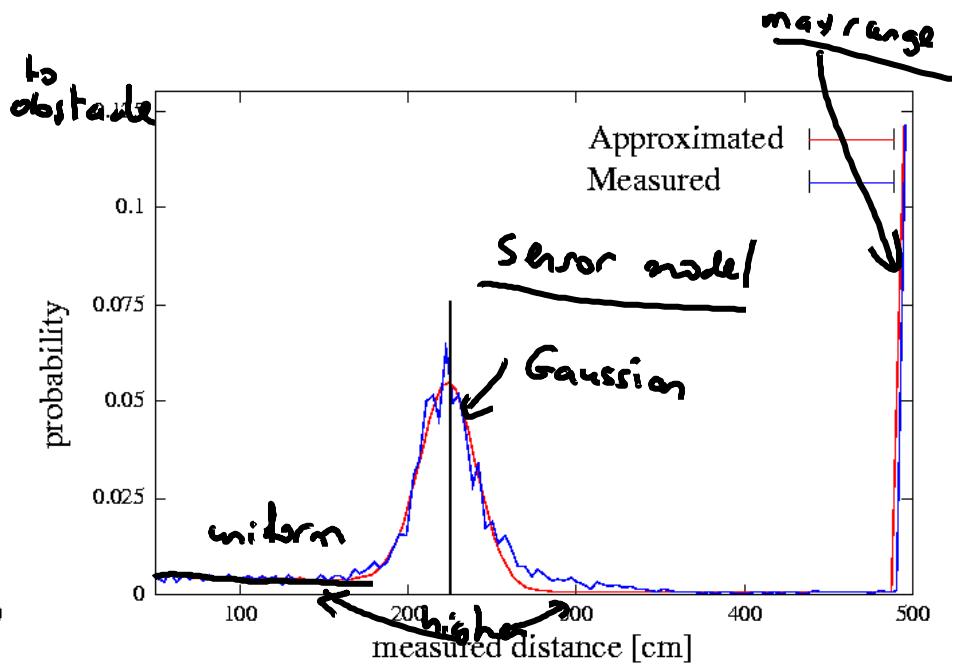
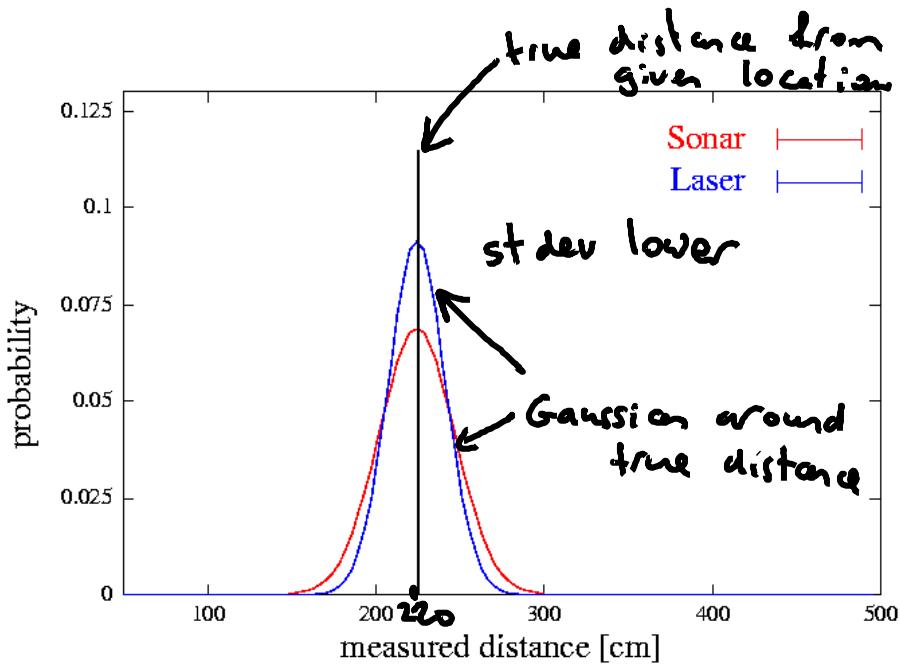


Fox, Burgard, Thrun

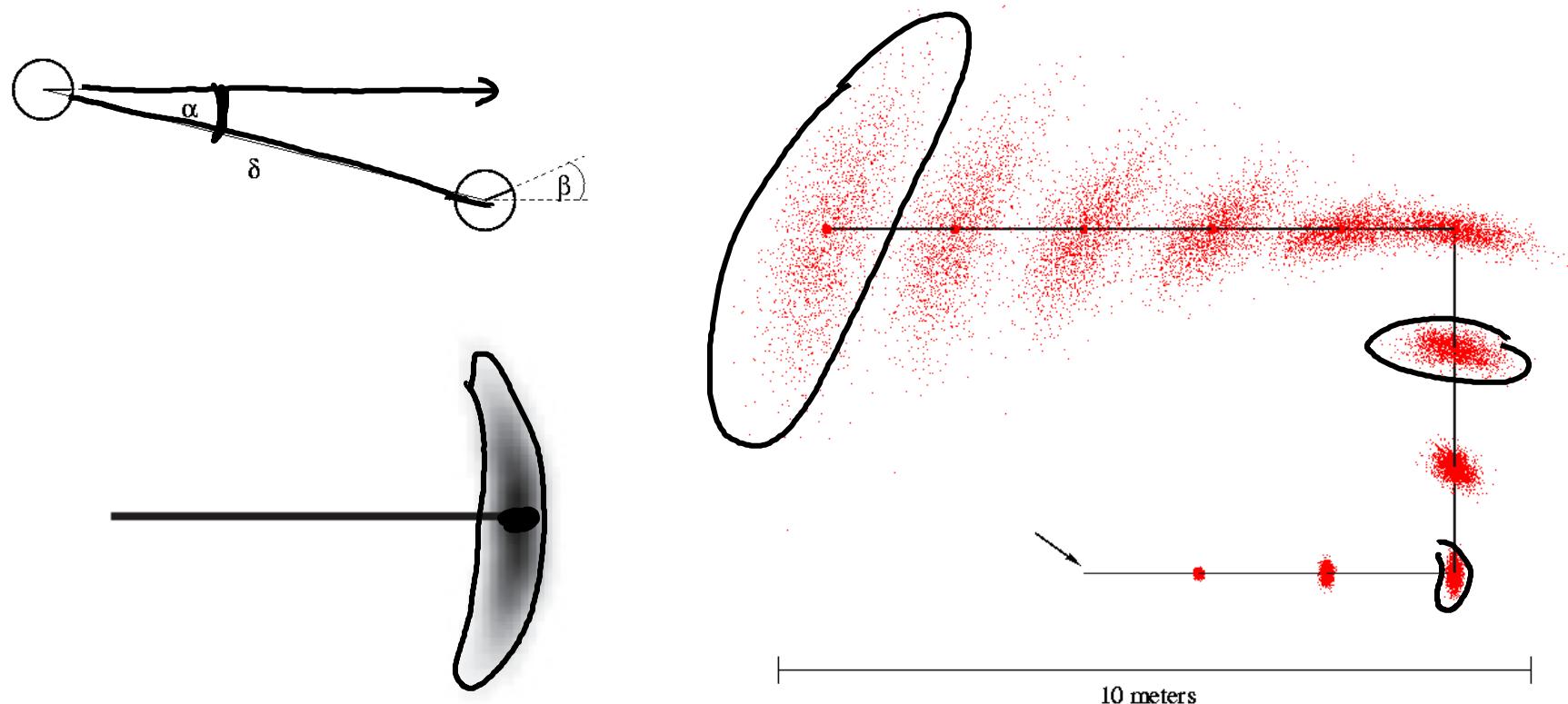
Daphne Koller



Nonlinear Gaussians



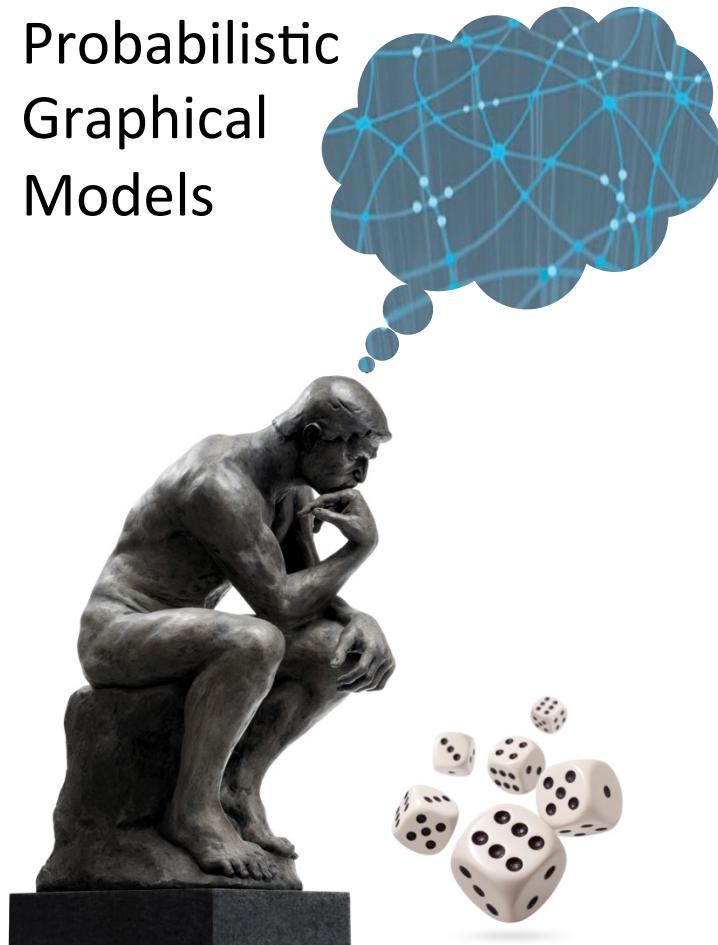
Robot Motion Model



Fox, Burgard, Thrun

Daphne Koller

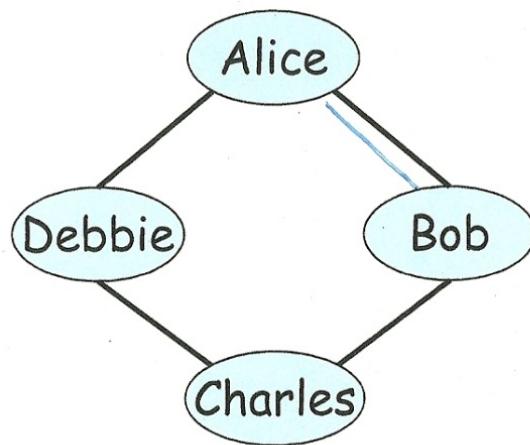
Probabilistic
Graphical
Models

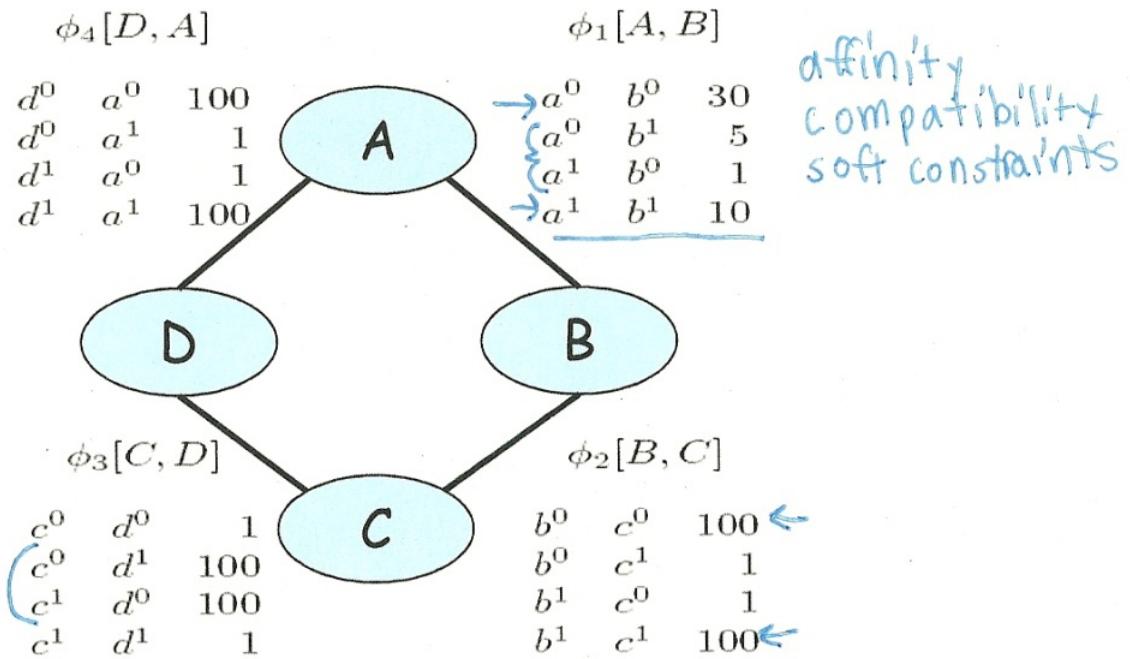


Representation

Markov Networks

Pairwise
Markov
Networks





$$\tilde{P}(A, B, C, D) = \phi_1(A, B) \times \phi_2(B, C) \times \phi_3(C, D) \times \phi_4(A, D)$$

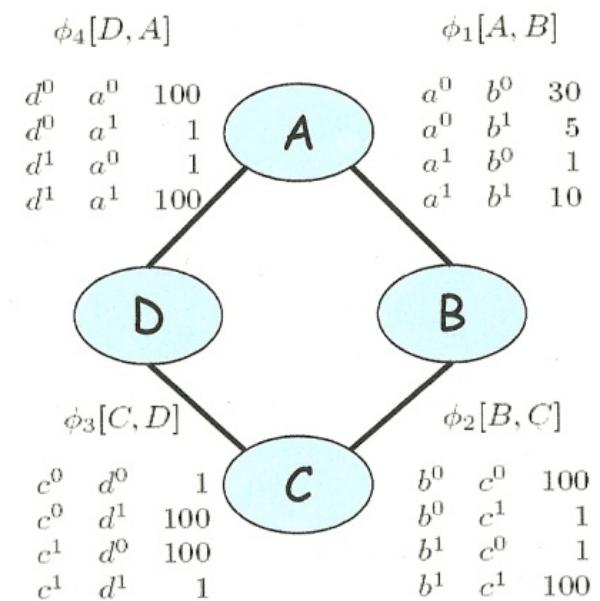
unnormalized measure

$$P(A, B, C, D) = \frac{1}{Z} \tilde{P}(A, B, C, D)$$

partition function

Assignment				Unnormalized
a^0	b^0	c^0	d^0	300000
a^0	b^0	c^0	d^1	300000
a^0	b^0	c^1	d^0	300000
a^0	b^0	c^1	d^1	30
a^0	b^1	c^0	d^0	500
a^0	b^1	c^0	d^1	500
a^0	b^1	c^1	d^0	5000000
a^0	b^1	c^1	d^1	500
a^1	b^0	c^0	d^0	100
a^1	b^0	c^0	d^1	1000000
a^1	b^0	c^1	d^0	100
a^1	b^0	c^1	d^1	100
a^1	b^1	c^0	d^0	10
a^1	b^1	c^0	d^1	100000
a^1	b^1	c^1	d^0	100000
a^1	b^1	c^1	d^1	100000

Z

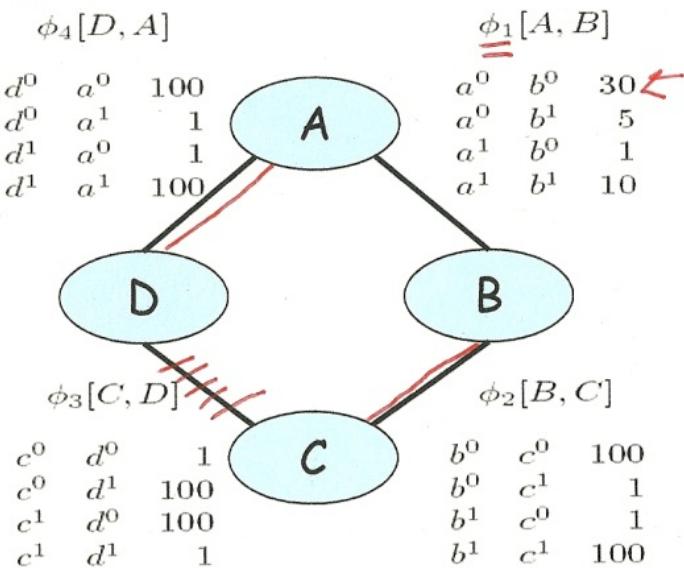


Daphne Koller

$p(A, B)$

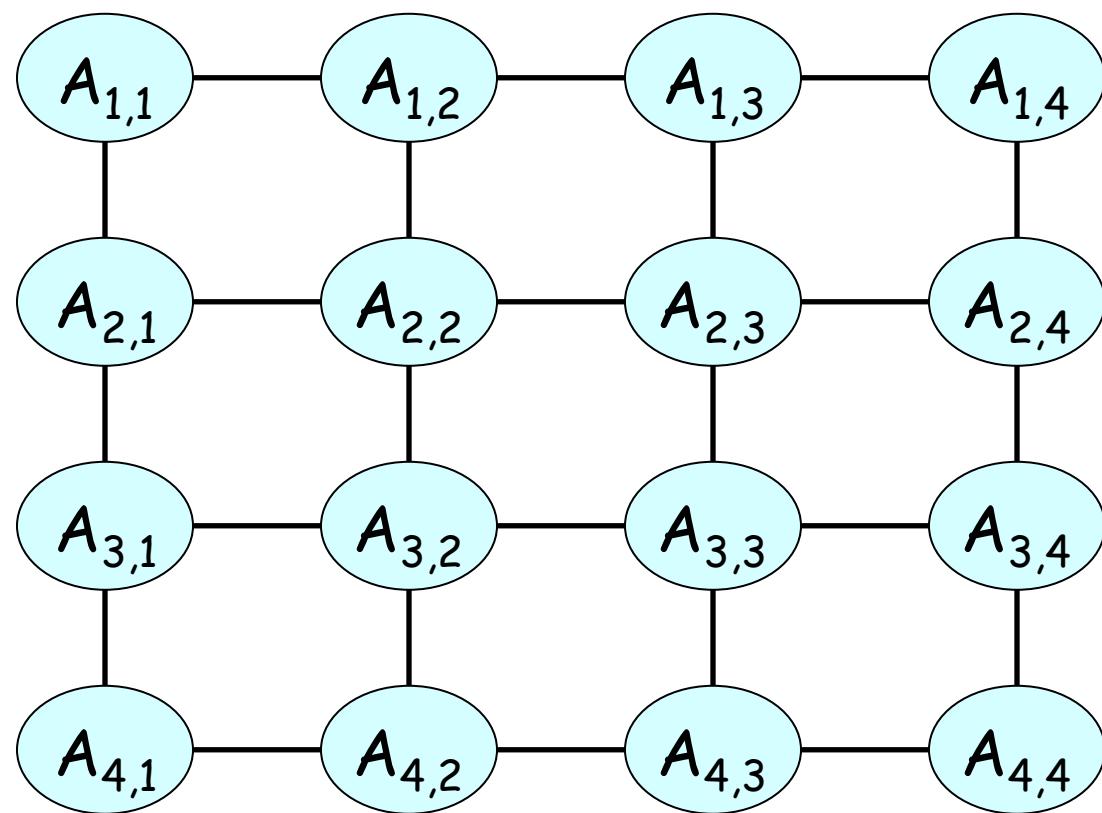
A	B	Prob.
a^0	b^0	0.13
a^0	b^1	0.69
a^1	b^0	0.14
a^1	b^1	0.04

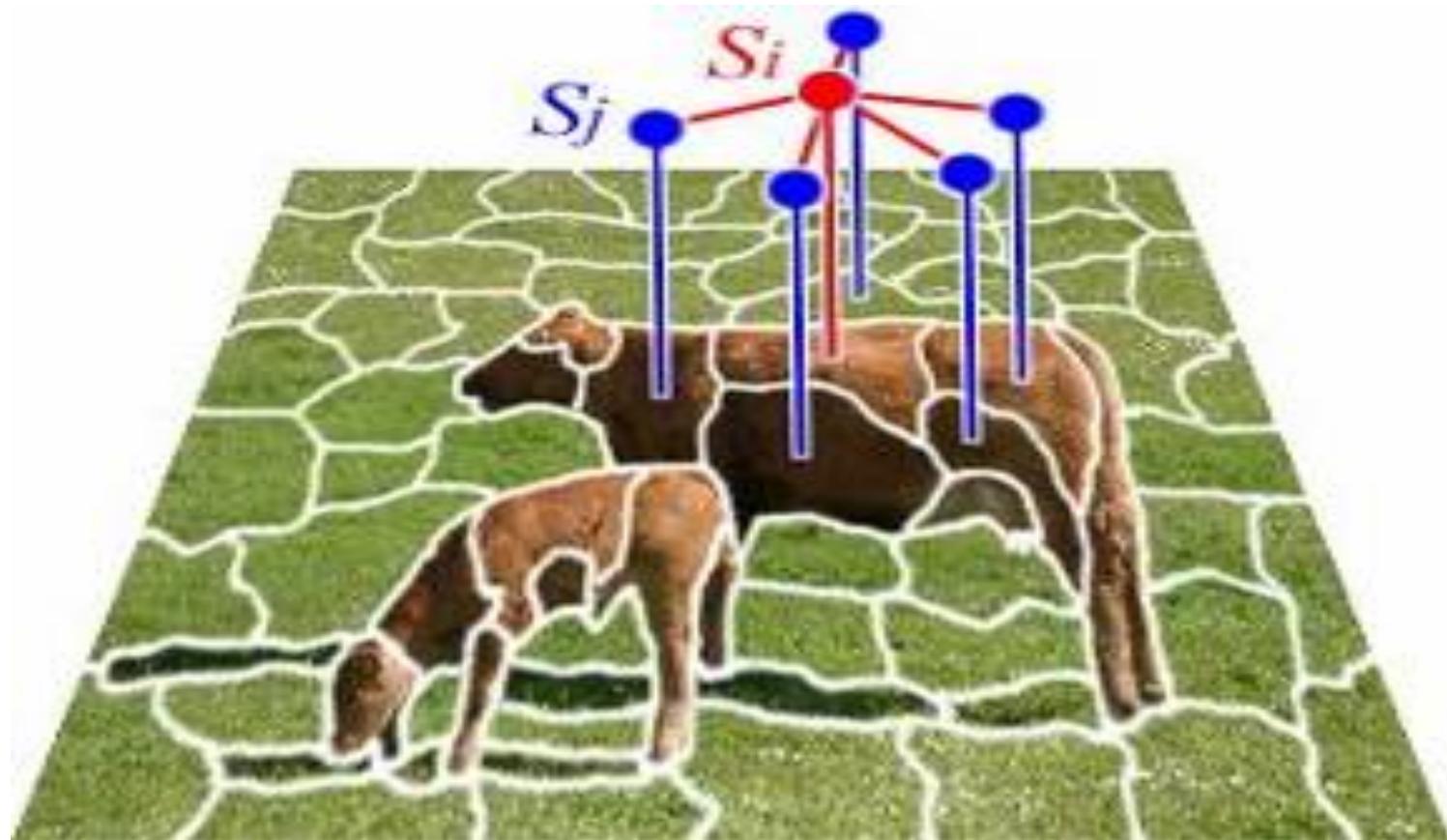
$$\Phi = \{\phi_1, \phi_2, \phi_3, \phi_4\}$$



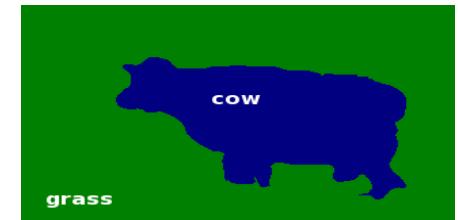
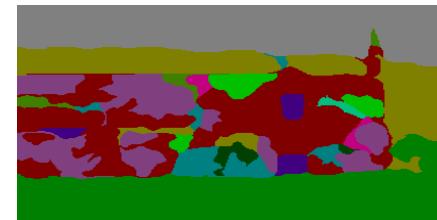
Pairwise Markov Networks

- A pairwise Markov network is an undirected graph whose nodes are X_1, \dots, X_n and each edge $\underline{X_i - X_j}$ is associated with a factor (potential) $\phi_{ij}(X_i, \overset{\text{random variables}}{X_j})$





Daphne Koller



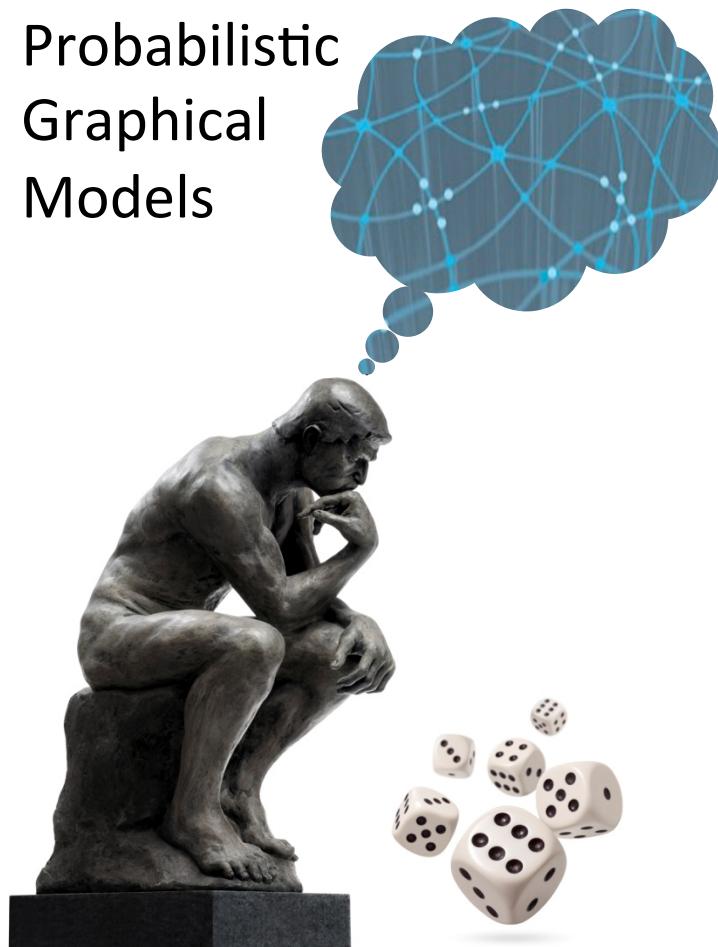
(a)

(b)

(c)

(d)

Probabilistic
Graphical
Models

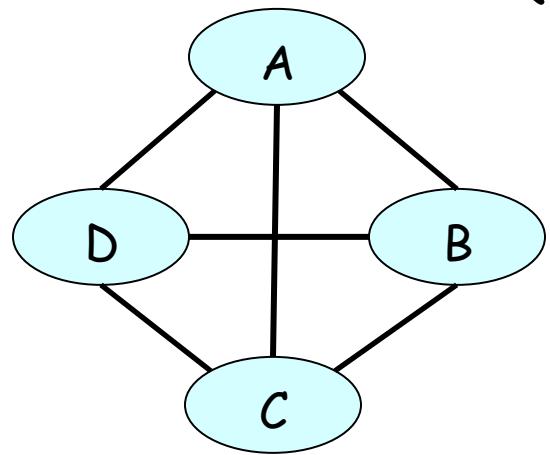


Representation

Markov Networks

General Gibbs
Distribution

$P(A, B, C, D)$



Is this fully expressive?

Gibbs Distribution

- Parameters:

General factors $\phi_i(D_i)$

$$\Phi = \{\phi_i(D_i)\}$$

a ¹	b ¹	c ¹	0.25
a ¹	b ¹	c ²	0.35
a ¹	b ²	c ¹	0.08
a ¹	b ²	c ²	0.16
a ²	b ¹	c ¹	0.05
a ²	b ¹	c ²	0.07
a ²	b ²	c ¹	0
a ²	b ²	c ²	0
a ³	b ¹	c ¹	0.15
a ³	b ¹	c ²	0.21
a ³	b ²	c ¹	0.09
a ³	b ²	c ²	0.18

Gibbs Distribution

Set of factors

$$\underline{\Phi} = \{\phi_1(D_1), \dots, \phi_k(D_k)\}$$

unnormalized measure k

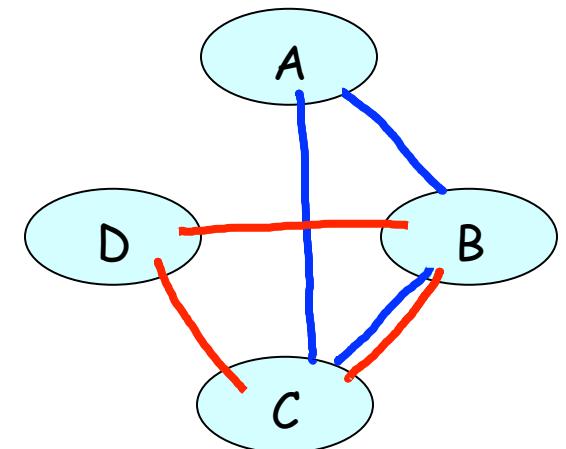
$$\tilde{P}_\Phi(X_1, \dots, X_n) = \prod \phi_i(\underline{D_i}) \quad \text{factor product}$$

$$Z_\Phi = \sum_{\underline{X_1, \dots, X_n}} \tilde{P}_\Phi(\underline{X_1}, \dots, X_n)$$

$$\overline{P}_\Phi(X_1, \dots, X_n) = \frac{1}{\underline{Z_\Phi}} \tilde{P}_\Phi(X_1, \dots, X_n)$$

Induced Markov Network

$\phi_1(\underline{A}, \underline{B}, C)$, $\phi_2(B, \underline{C}, D)$



$\Phi = \{\phi_1(D_1), \dots, \phi_k(D_k)\}$

Induced Markov network H_Φ has an edge $X_i - X_j$ whenever
there exists $\phi_m \in \Phi$ s.t. $x_i, x_j \in D_m$

Factorization

P factorizes over H if

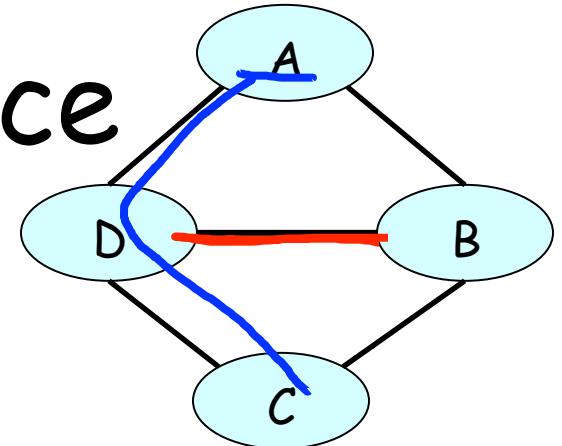
there exist $\underline{\Phi} = \{\phi_1(D_1), \dots, \phi_k(D_k)\}$

such that

$\underline{P} = P_{\underline{\Phi}}$ normalized product of
 \underline{H} is the induced graph for $\underline{\Phi}$ factors in $\underline{\Phi}$

Flow of Influence

$\phi_1(A, B, D)$, $\phi_2(B, C, D)$

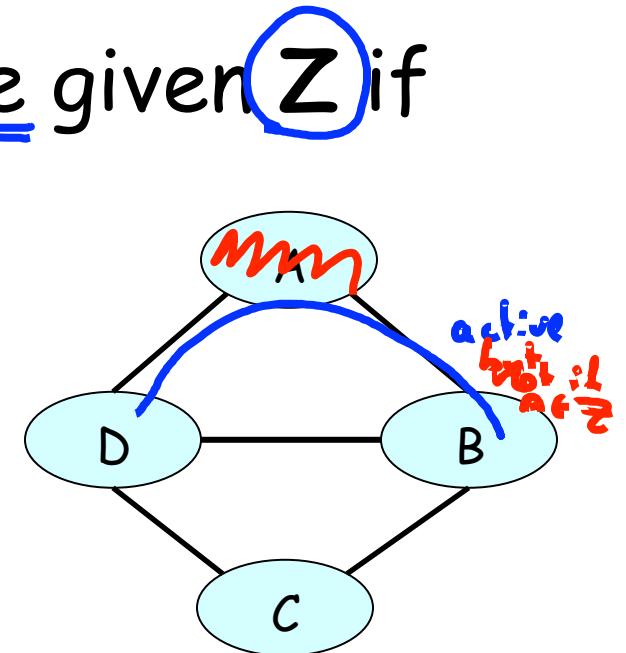


$\phi_1(A, B)$, $\phi_2(B, C)$, $\phi_3(C, D)$, $\phi_4(A, D)$, $\phi_5(B, D)$

- Influence can flow along any trail, regardless of the form of the factors

Active Trails

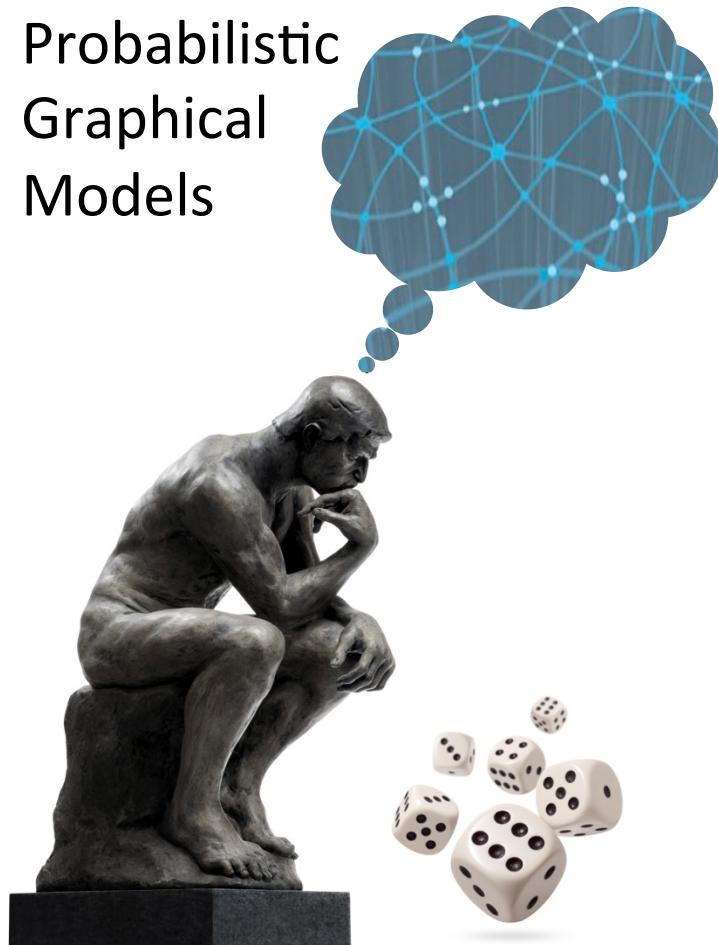
- A trail $X_1 - \dots - X_n$ is active given Z if no X_i is in Z



Summary

- Gibbs distribution represents distribution as a product of factors
- Induced Markov network connects every pair of nodes that are in the same factor
- Markov network structure doesn't fully specify the factorization of P
- But active trails depend only on graph structure

Probabilistic
Graphical
Models



Representation

Markov Networks

Conditional
Random
Fields

Motivation

- Observed variables X
- Target variables Y

$$X \rightarrow Y$$

$$P(X, Y)$$

joint

$$P(Y|X)$$

conditional

X

Y

image
segmentation

pixel values

pixel labels

text processing

words in a
sentence

parts of speech

CRF Representation

$$\phi_1(D_1), \dots, \phi_k(D_k)$$

$$\tilde{P}(\bar{x}, \bar{y}) = \prod_{i=1}^k \phi_i(D_i) \quad \text{unnormalized measure}$$

$$z(\bar{x}) = \sum_{\bar{y}} \tilde{P}(\bar{x}, \bar{y}) \quad \text{different } \tilde{z}(\bar{x}) \text{ for every assignment to the obs. variables } \bar{x}$$

$$P(\bar{y} | \bar{x}) = \frac{1}{z(\bar{x})} \tilde{P}(\bar{x}, \bar{y}) \quad \left(\sum_{\bar{y}} P(\bar{y} | \bar{x}) = 1 \text{ for all } \bar{x} \right)$$

CRFs and Logistic Model

$$\phi_i(X_i, Y) = \exp\{w_i \mathbf{1}\{X_i = 1, Y = 1\}\}$$

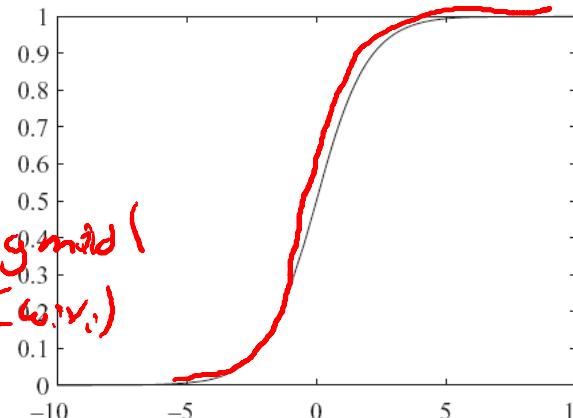
$$\phi_i(x_i, Y=1) = \exp\{\omega_i x_i\}$$

$$\phi_i(x_i, Y=0) = 1$$

$$\tilde{P}(Y=1 | X_1, \dots, X_n) = \exp\left(\sum_i \omega_i x_i\right)$$

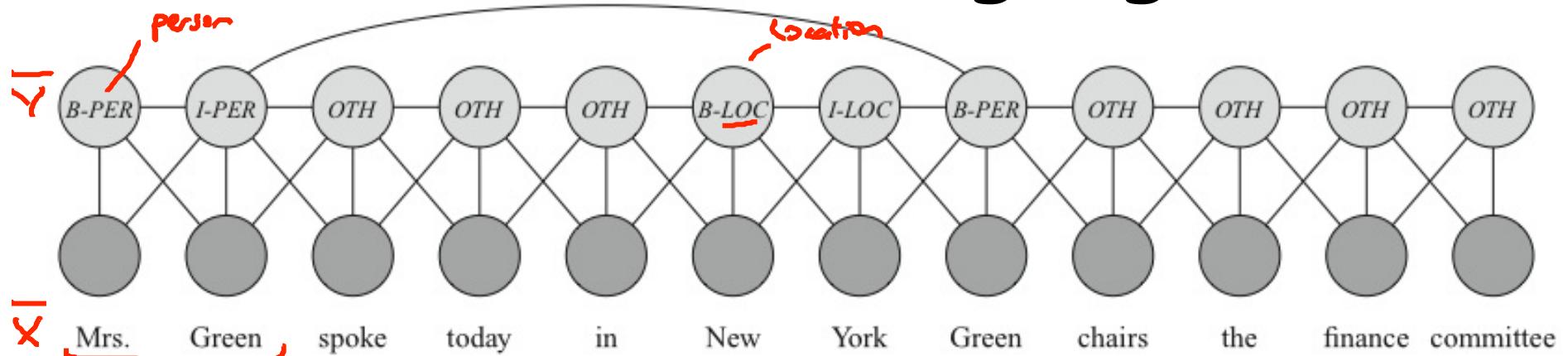
$$\tilde{P}(Y=0 | X_1, \dots, X_n) = 1$$

$$P(Y=1 | X_1, \dots, X_n) = \frac{\exp\left(\sum_i \omega_i x_i\right)}{1 + \exp\left(\sum_i \omega_i x_i\right)} = \text{sigmoid}\left(\sum_i \omega_i x_i\right)$$



Daphne Koller

CRFs for Language

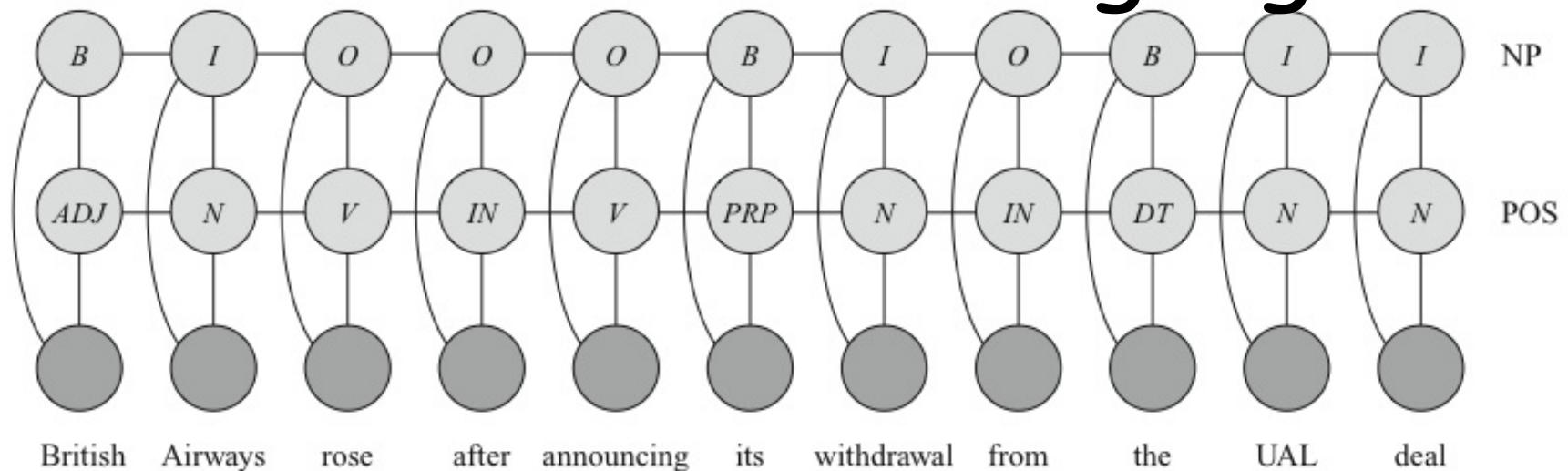


Features: word capitalized, word in atlas or name list, previous word is "Mrs", next word is "Times", ...

$$\tilde{P}(\vec{x}, \vec{y})$$

$$\Rightarrow P(\vec{Y}|\vec{x})$$

More CRFs for Language



KEY

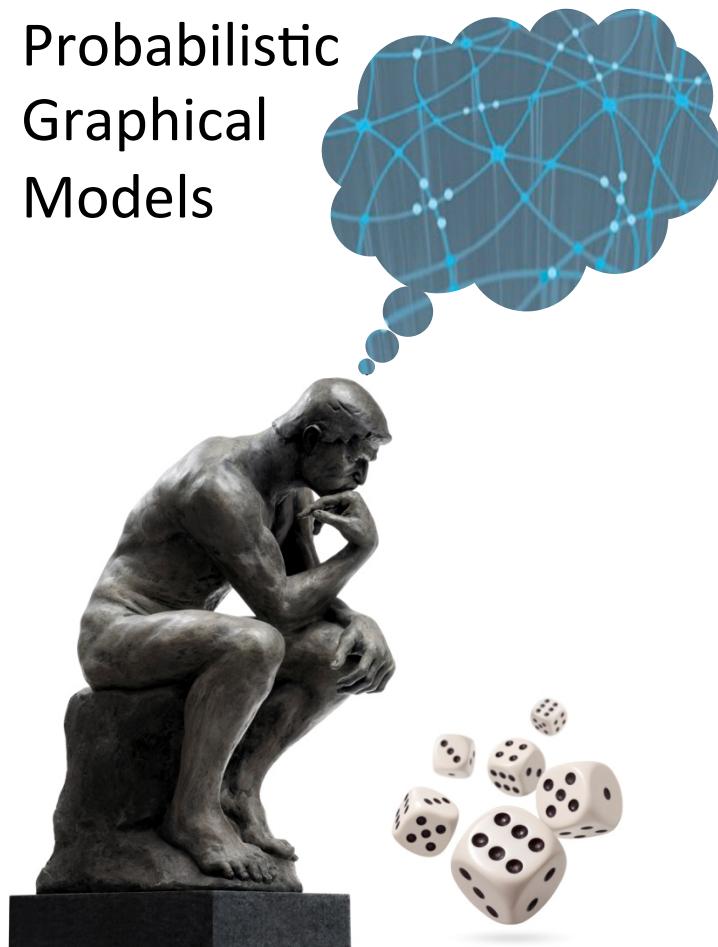
<i>B</i>	Begin noun phrase	<i>V</i>	Verb
<i>I</i>	Within noun phrase	<i>IN</i>	Preposition
<i>O</i>	Not a noun phrase	<i>PRP</i>	Possessive pronoun
<i>N</i>	Noun	<i>DT</i>	Determiner (e.g., a, an, the)
<i>ADJ</i>	Adjective		

Daphne Koller

Summary

- A CRF is parameterized the same as a Gibbs distribution, but normalized differently
- Don't need to model distribution over variables we don't care about
- Allows models with highly expressive features, without worrying about wrong independencies

Probabilistic
Graphical
Models



Representation

Independencies

Markov
Networks

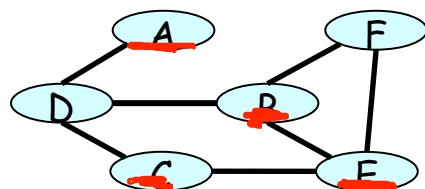
Separation in MNs

Definition:

X and Y are separated in H given Z

if there is no active trail in H

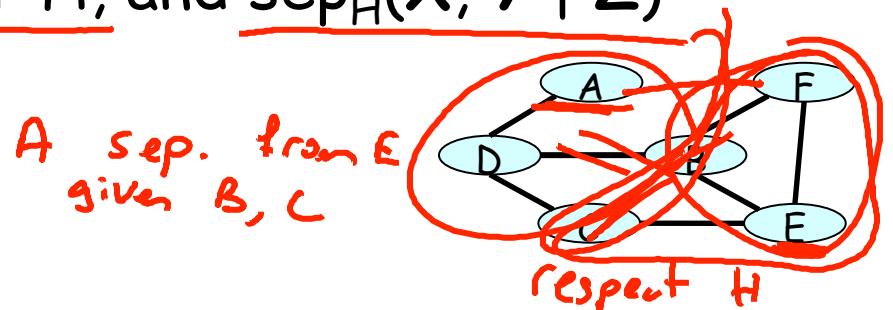
between X and Y given Z *no node along trail is in z*



A,E separated given B,D
given D
given B,C

Factorization \Rightarrow Independence: MNS

Theorem: If P factorizes over H, and $\text{sep}_H(X, Y | Z)$
then P satisfies $(X \perp Y | Z)$



$$\pi \phi_{\text{on } A} \cdot \pi \phi_{\text{on } E} =$$

cannot involve E cannot involve A

Factorization \Rightarrow Independence: MNs

$$\underline{I(H) = \{(X \perp Y \mid Z) : \boxed{\text{sep}_H(X, Y \mid Z)}\}}$$

If P satisfies I(H), we say that H is an I-map
(independency map) of P

Theorem: If P factorizes over H, then H is an I-map of P

Independence \Rightarrow Factorization

- Theorem (Hammersley Clifford):

For a positive distribution P , if H is an I-map for P , then P factorizes over H

$$P(z) \propto \alpha_z$$

Summary

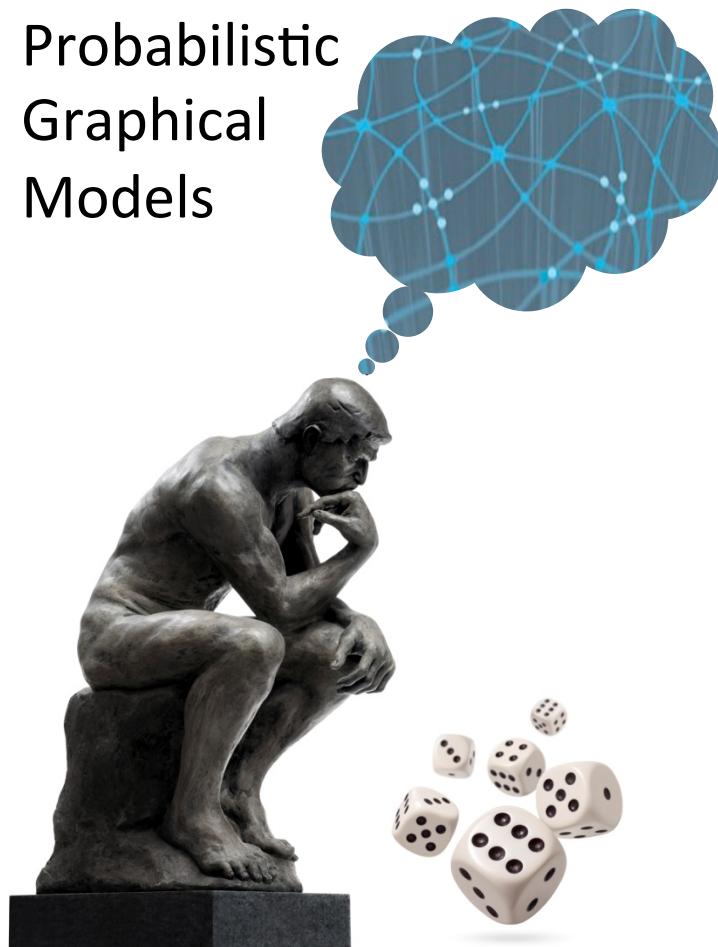
Two equivalent* views of graph structure:

- Factorization: H allows P to be represented
- I-map: Independencies encoded by H hold in P

If P factorizes over a graph H , we can read from the graph independencies that must hold in P (an independency map)

* for positive distributions

Probabilistic
Graphical
Models



Representation

Independencies

I-maps and
Perfect Maps

Capturing Independencies in P

$$\underline{I(P)} = \{\underbrace{(X \perp Y \mid Z)}_{\text{d-separation}} : \underline{P} \models \underbrace{(X \perp Y \mid Z)}_{\substack{\text{independencies} \\ \text{that hold in } P}}\}$$

- P factorizes over G $\Rightarrow G$ is an I-map for P:

$$\text{d-separation} \quad \underline{I(G)} \subseteq I(P)$$

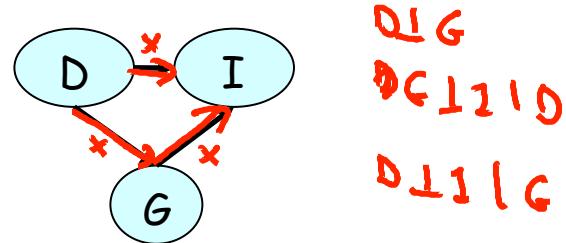
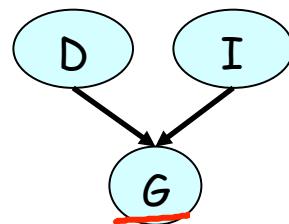
- But not always vice versa: there can be independencies in I(P) that are not in I(G)

Want a Sparse Graph

- If the graph encodes more independencies
 - it is sparser (has fewer parameters)
 - and more informative
- Want a graph that captures as much of the structure in P as possible

Minimal I-map

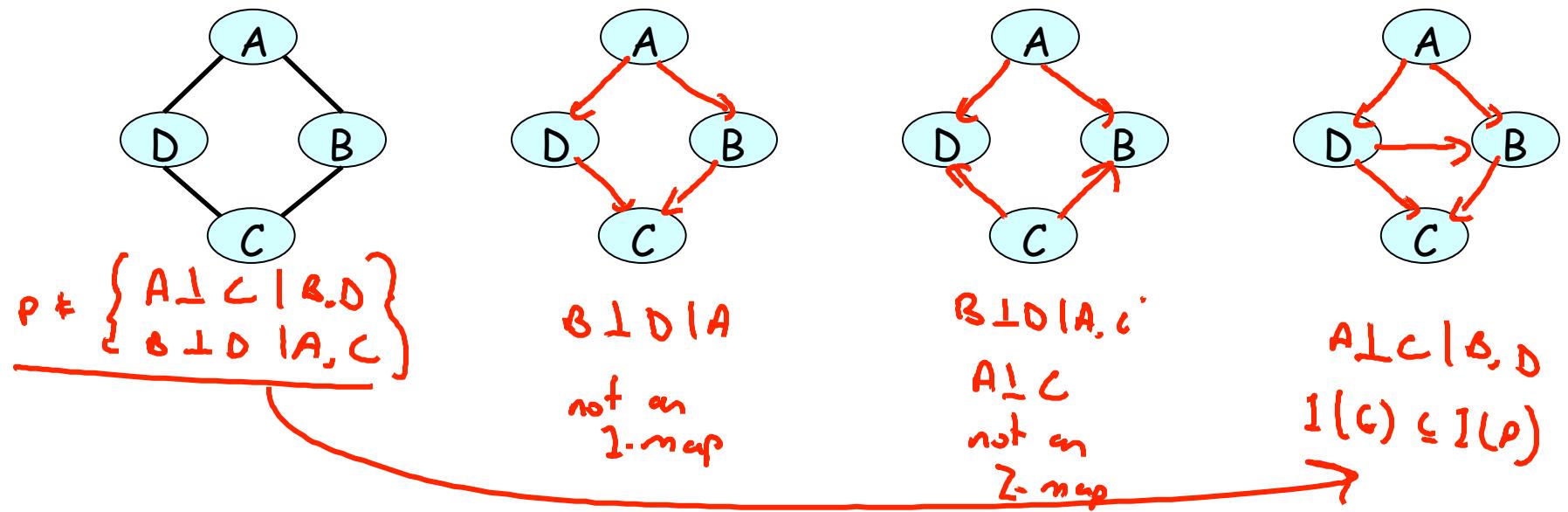
- Minimal I-map: I-map without redundant edges
~~($\phi \rightarrow \psi$) $P(\psi | \phi) = P(\psi | \neg \phi)$~~
- Minimal I-map may still not capture $I(P)$



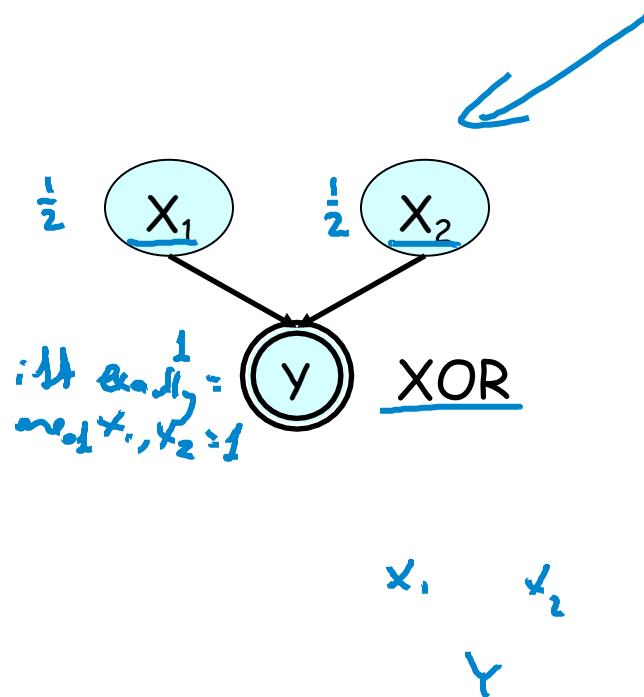
Perfect Map

- Perfect map: $\underline{I(G)} = \underline{I(P)}$
 - G perfectly captures independencies in P

Perfect Map



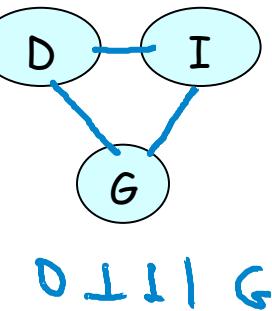
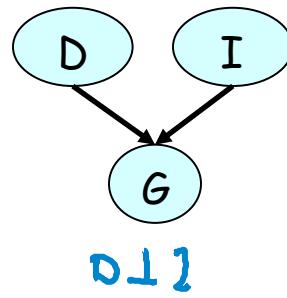
Another imperfect map



x_1	x_2	<u>y</u>	Prob
0	0	0	0.25
0	1	1	0.25
1	0	1	0.25
1	1	0	0.25

MN as a perfect map

- Perfect map: $I(\underline{H}) = I(P)$
 - H perfectly captures independencies in P

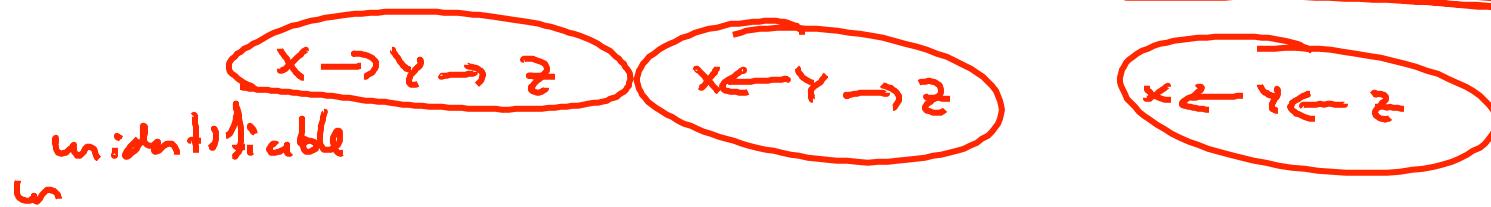


Uniqueness of Perfect Map

$G_1: X \rightarrow Y \quad I(G_1) = \emptyset$
 $G_2: Y \leftarrow X \quad I(G_2) = \emptyset \Rightarrow$ can represent
exactly the same
distribution

I-equivalence

Definition: Two graphs G_1 and G_2 over X_1, \dots, X_n are I-equivalent if $I(G_1) = I(G_2)$



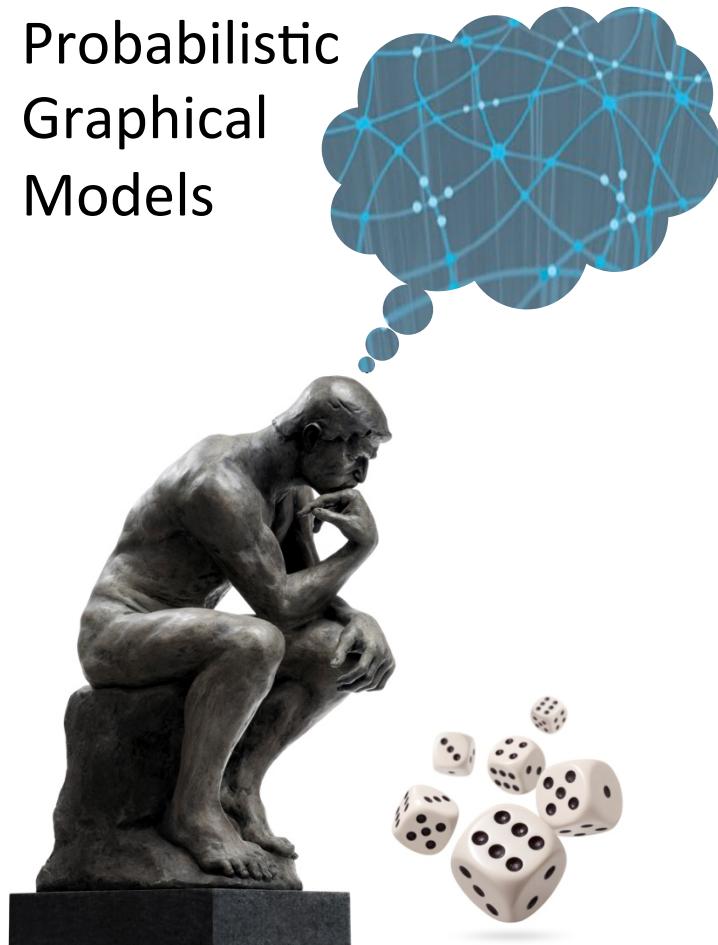
Most G 's have many I-equivalent variants

Summary

- Graphs that capture more of $I(P)$ are more compact and provide more insight
- A minimal I-map may fail to capture a lot of structure even if present *and representable as a PGM*
- A perfect map is great, but may not exist
- Converting BNs \leftrightarrow MNs loses independencies
 - BN to MN: loses independencies in v-structures
 - MN to BN: must add triangulating edges to loops



Probabilistic
Graphical
Models



Representation

Local Structure

Log-Linear
Models

Log-Linear Representation

$$\tilde{P} = \prod_i \phi_i(D_i)$$

$$\tilde{P} = \exp \left(- \sum_j \underbrace{w_j f_j(D_j)}_{\text{features}} \right)$$

$$\tilde{P} = \prod_j \underbrace{\exp (-w_j f_j(D_j))}_{\text{factor}}$$

- Each feature f_j has a scope $\underline{D_j}$
- Different features can have same scope

Representing Table Factors

$$\phi(X_1, X_2) = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}$$

$$f_{12}^{00} = \mathbf{1}\{X_1 = 0, X_2 = 0\}$$
$$f_{12}^{01} = \mathbf{1}\{X_1 = 0, X_2 = 1\}$$
$$f_{12}^{10} = \mathbf{1}\{X_1 = 1, X_2 = 0\}$$
$$f_{12}^{11} = \mathbf{1}\{X_1 = 1, X_2 = 1\}$$

General representation

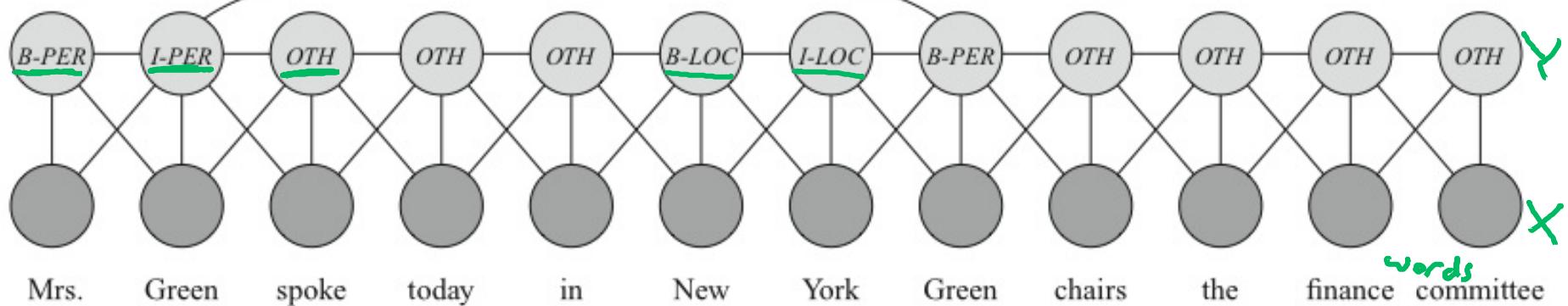
$$\phi(X_1, X_2) = \exp\left(-\sum_{kl} w_{kl} f_{ij}^{kl}(X_1, X_2)\right)$$

$$w_{kl} = -\log a_{kl}$$

$\exp(-w_{00})$ when $X_1=0, X_2=0$
 $\exp(-w_{0,1})$ when $X_1=0, X_2=1$

Daphne Koller

Features for Language



Features: word capitalized, word in atlas or name list, previous word is "Mrs", next word is "Times", ...

$$f(x_i, x_{i+1}) = \begin{cases} 1 & \{x_i = \text{person}, x_{i+1} \text{ is capitalized}\} \\ 1 & \{x_i = \text{B-loc}, x_{i+1} \text{ appears in Atlas}\} \end{cases}$$

Ising Model

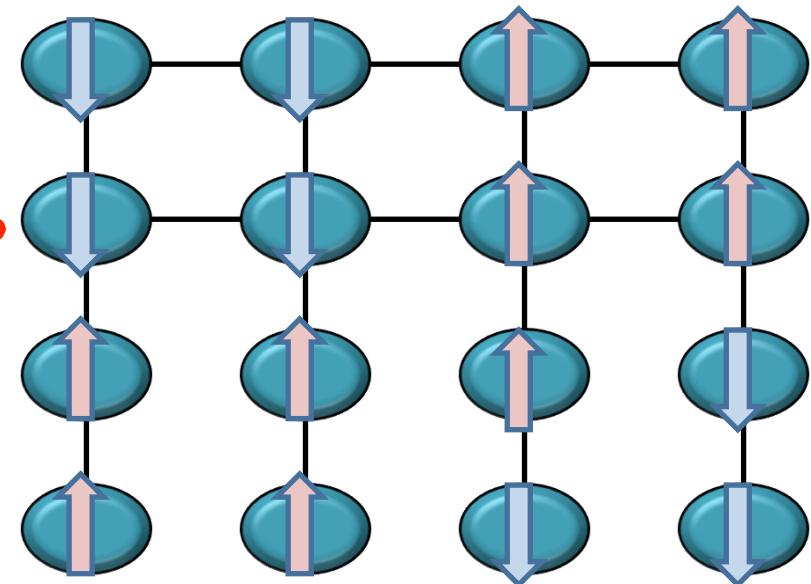
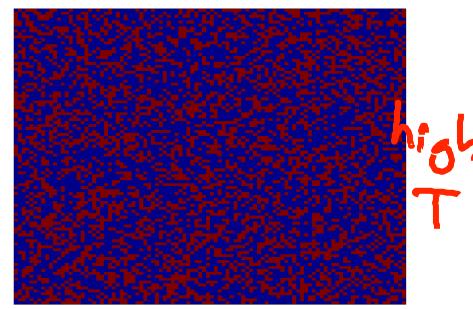
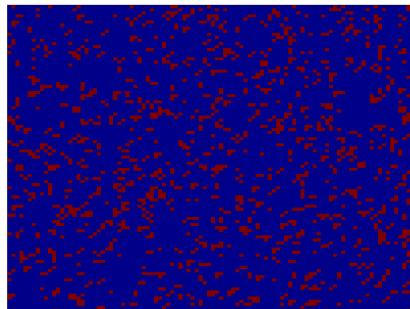
$$E(x_1, \dots, x_n) = - \sum_{i < j} w_{i,j} \cancel{x_i x_j}^{\text{conflict}} - \sum_i u_i x_i$$

pairwise joint spins

$x_i \in \{-1, +1\}$, $f_{i,j}(X_i, X_j) = \underline{\underline{X_i \cdot X_j}}$

law $P(X) \propto e^{-\frac{1}{T} E(X)}$

T groups $\frac{w_{ij}}{T} \rightarrow 0$
 T decreases



Metric MRFs

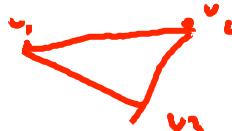
- All X_i take values in label space V



want X_i and X_j to take "similar" values

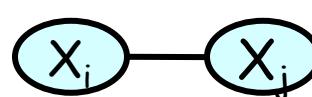
- Distance function $\mu : V \times V \rightarrow \mathbb{R}^+$

- Reflexivity: $\mu(v, v) = 0$ for all v
- Symmetry: $\mu(v_1, v_2) = \mu(v_2, v_1)$ for all v_1, v_2
- Triangle inequality: $\mu(v_1, v_2) \leq \mu(v_1, v_3) + \mu(v_3, v_2)$ for all v_1, v_2, v_3



Metric MRFs

- All X_i take values in label space V



want X_i and X_j to
take "similar" values

- Distance function $\mu : V \times V \rightarrow \mathbb{R}$

$$\underline{f_{i,j}(X_i, X_j)} = \mu(X_i, X_j)$$

$$\exp(-w_{ij} f_{ij}(X_i, X_j))$$

w_{ij}

values of X_i and X_j far in μ



$$w_{ij} > 0$$

lower distance

higher

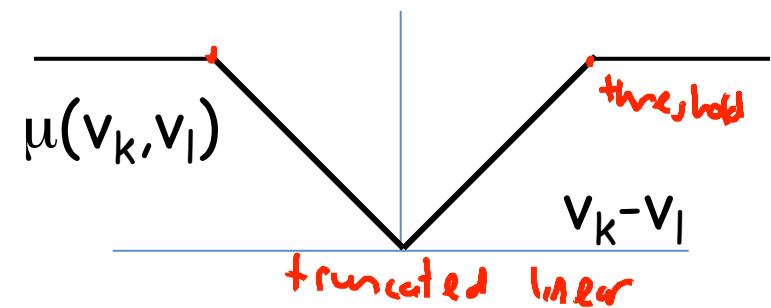
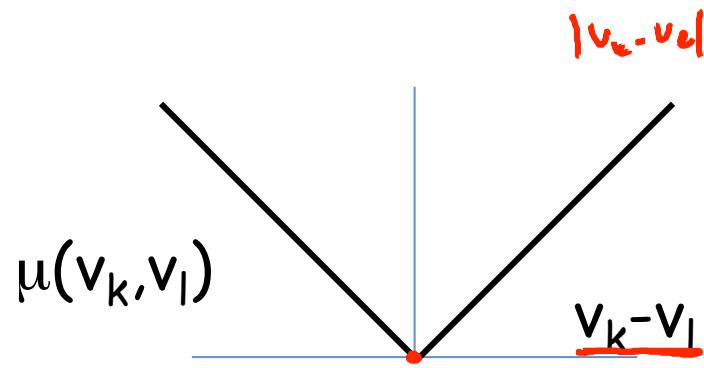
higher probability

lower probability

Metric MRF Examples

$$\mu(v_k, v_l) = \begin{cases} 0 & v_k = v_l \\ 1 & \text{otherwise} \end{cases}$$

0	1	1	1
1	0	1	1
1	1	0	1
1	1	1	0

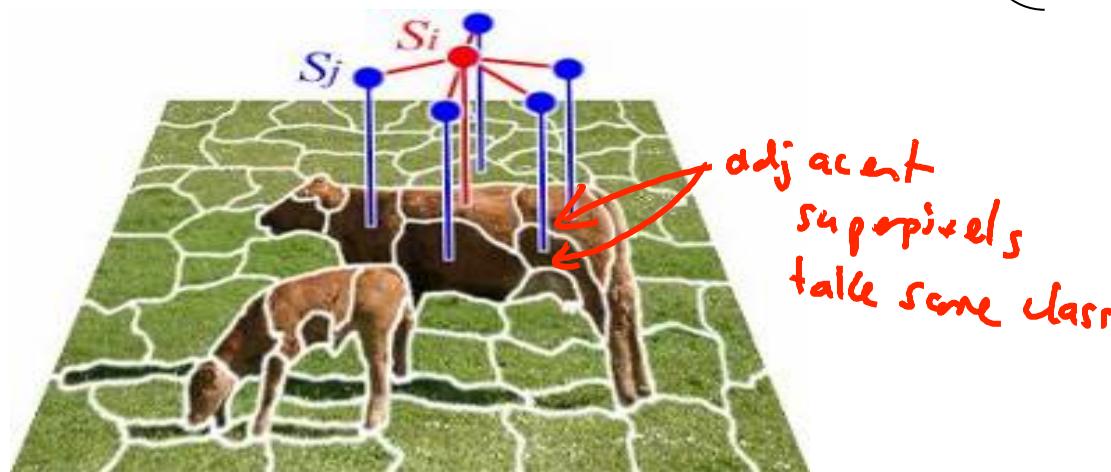


Daphne Koller

Metric MRF: Segmentation

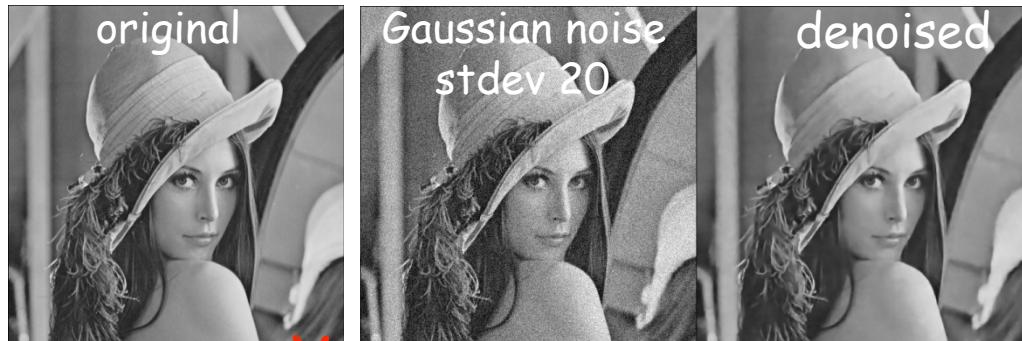
$$\mu(v_k, v_l) = \begin{cases} 0 & v_k = v_l \\ 1 & \text{otherwise} \end{cases}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



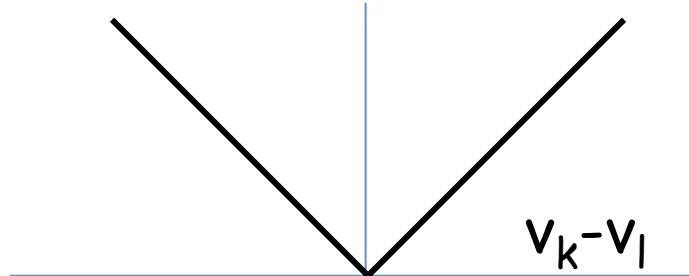
Daphne Koller

Metric MRF: Denoising



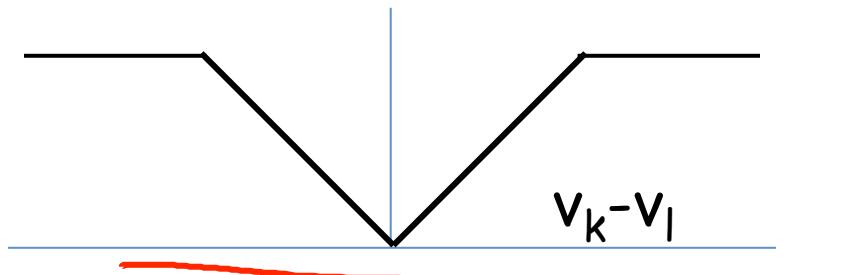
$$\mu(v_k, v_l) = |v_k - v_l|$$

X noisy pixels *Y* clear pixels



Y: close to x ,
Y: close to its neighbors

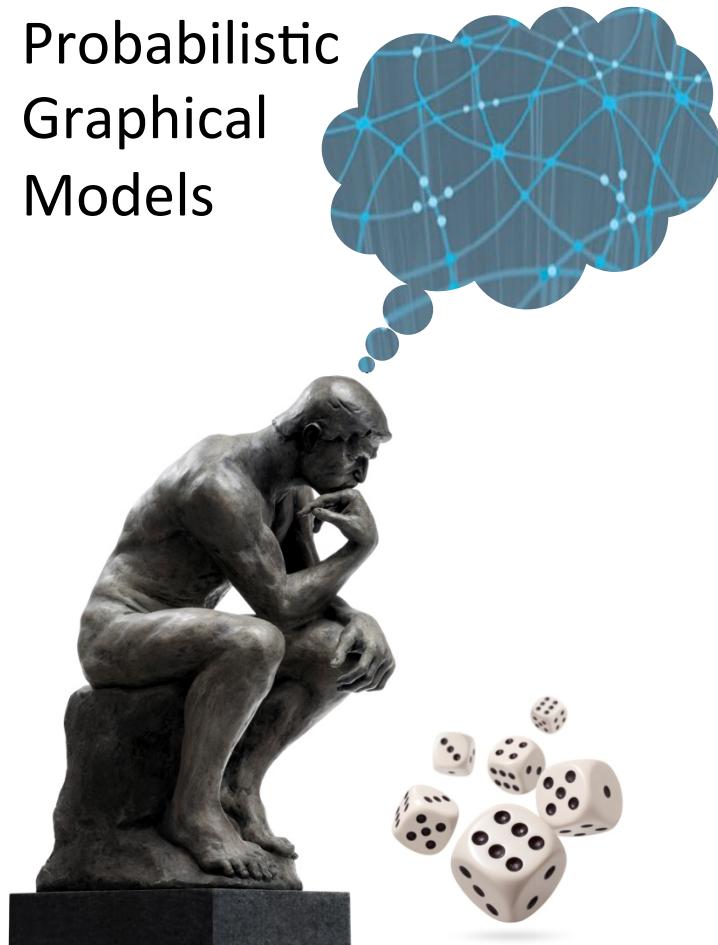
$$\mu(v_k, v_l) = \min(|v_k - v_l|, d)$$



Similar idea for stereo reconstruction

Daphne Koller

Probabilistic
Graphical
Models



Representation

Template Models

Shared
Features in Log-
Linear Models

Ising Models

- In most MRFs, same feature and weight are used over many scopes

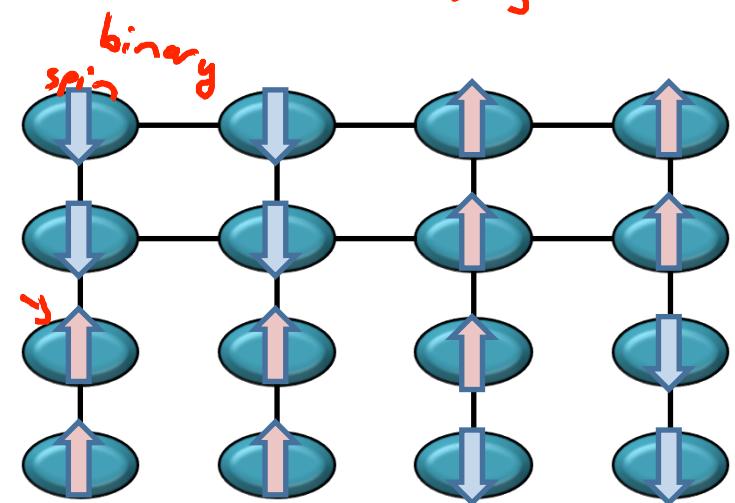
$x_i \in \{-1, +1\}$

Ising Model

$$E(x_1, \dots, x_n) = - \sum_{(i,j) \in \text{Edges}} w_{i,j} x_i x_j - \sum_i u_i x_i$$

w_{i,j} f(x_i, x_j)
weight same feature

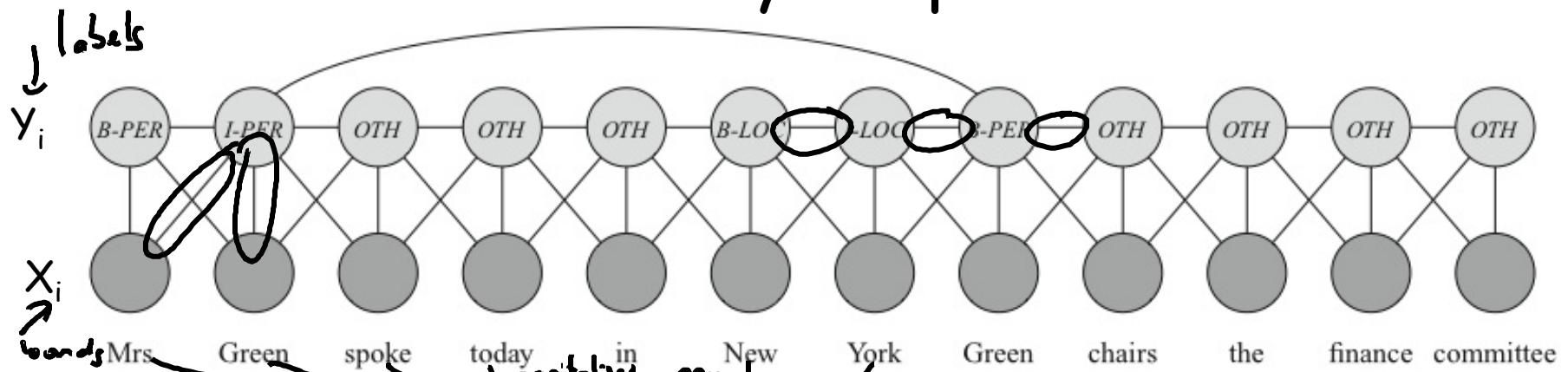
same weight for every adjacent pair



Daphne Koller

Natural Language Processing

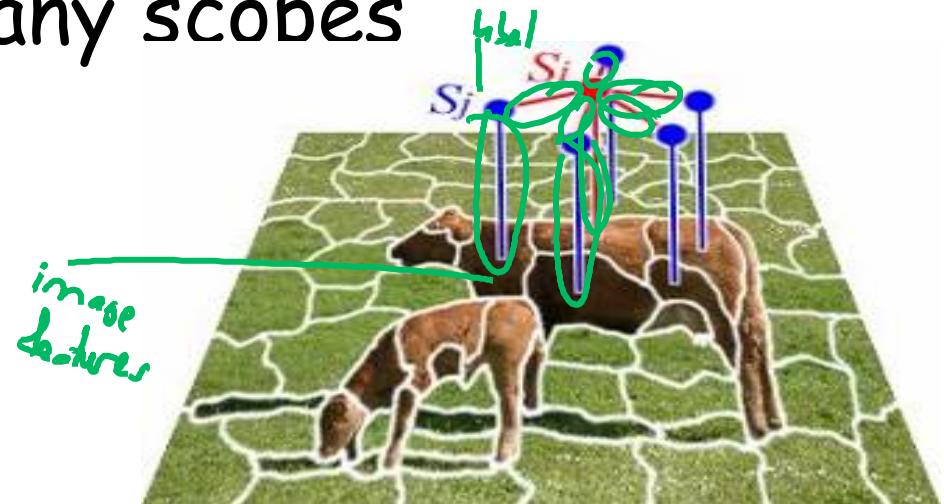
- In most MRFs, same feature and weight are used over many scopes



Same energy terms $w_k f_k(x_i, y_i)$ repeat for all positions i in the sequence
Same energy terms $w_m f_m(y_i, y_{i+1})$ asp repeat for all positions i

Image Segmentation

- In most MRFs, same feature and weight are used over many scopes



Same features and weights for all superpixels in the image

Repeated Features

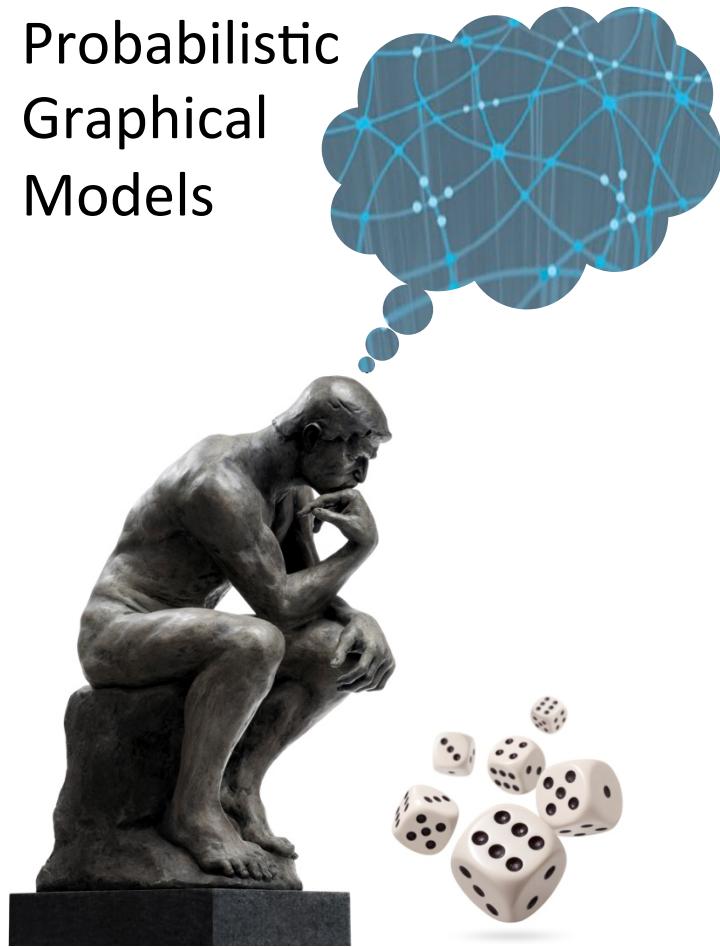
- Need to specify for each feature f_k a set of scopes Scopes[f_k]
- For each $D_k \in \text{Scopes}[f_k]$ we have a term $w_k f_k(D_k)$ in the energy function

$$w_k \sum_{D_k \in \text{Scopes}(f_k)} f_k(D_k)$$

Summary

- Same feature & weight can be used for multiple subsets of variables
 - Pairs of adjacent pixels/atoms/words
 - Occurrences of same word in document
- Can provide a single template for multiple MNs
 - Different images
 - Different sentences
- Parameters and structure are reused within an MN and across different MNs
- Need to specify set of scopes for each feature

Probabilistic
Graphical
Models

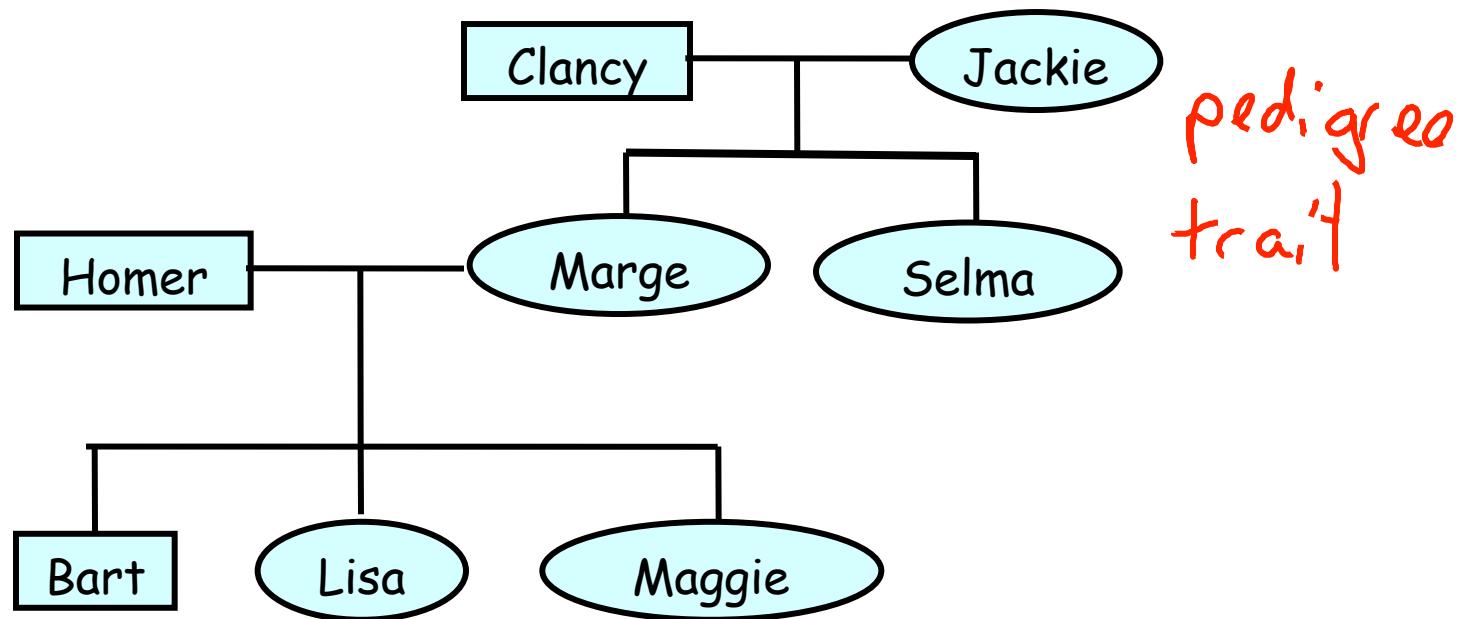


Representation

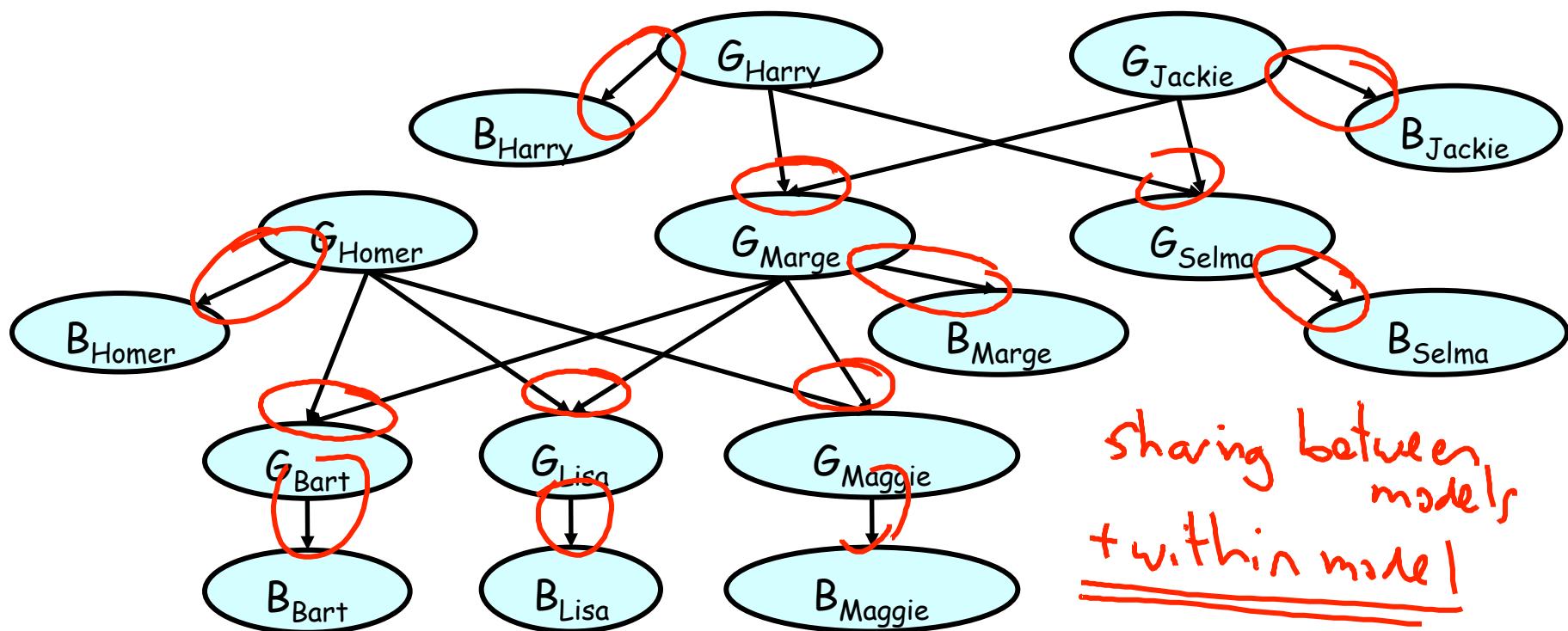
Template Models

Overview

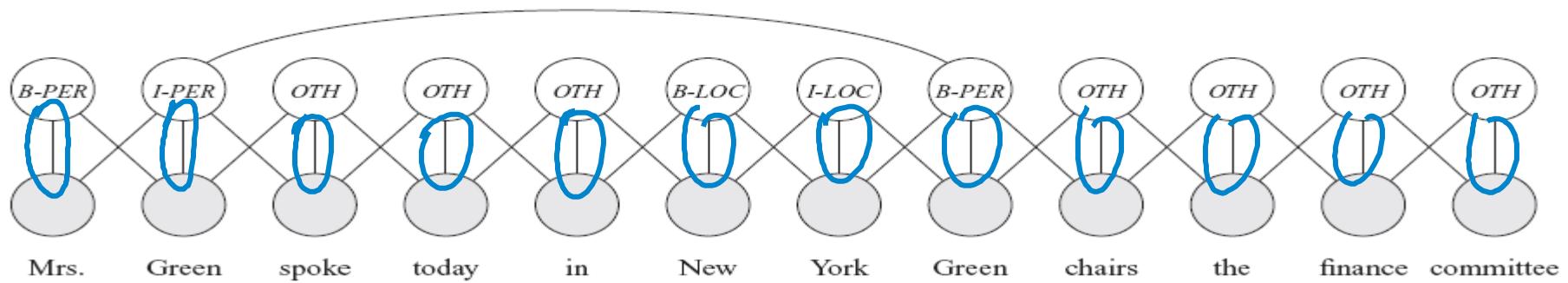
Genetic Inheritance



Genetic Inheritance



NLP Sequence Models



KEY

<i>B-PER</i>	Begin person name	<i>I-LOC</i>	Within location name
<i>I-PER</i>	Within person name	<i>OTH</i>	Not an entity
<i>B-LOC</i>	Begin location name		

Named entity recognition

Image Segmentation

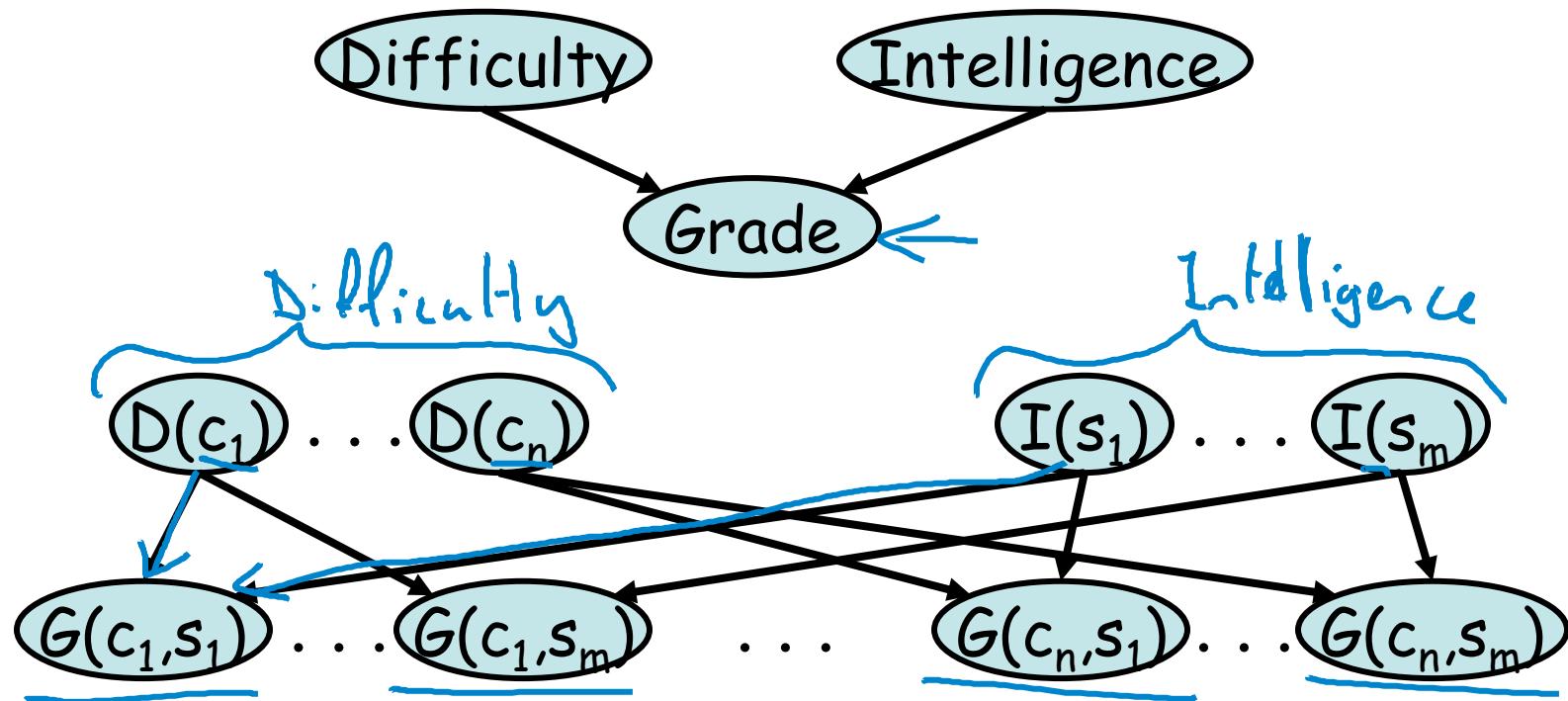
sharing "across
pixel
and pairs
of superpixels

between
and within



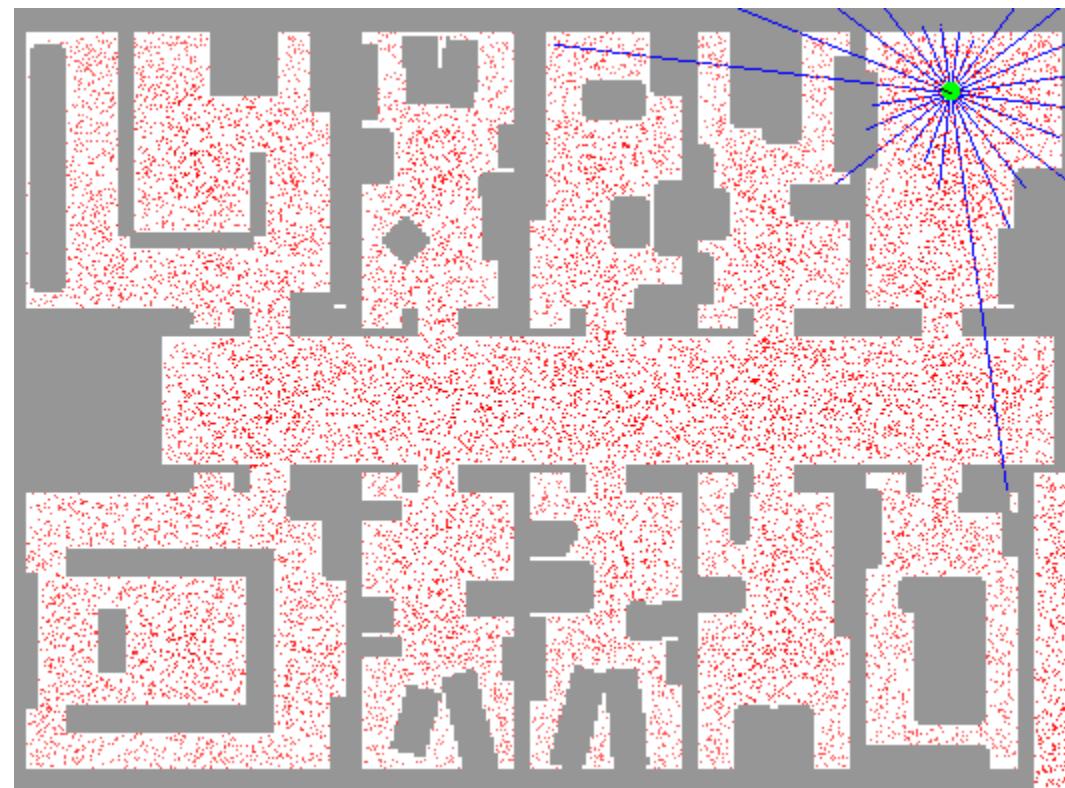
Daphne Koller

The University Example



Robot Localization

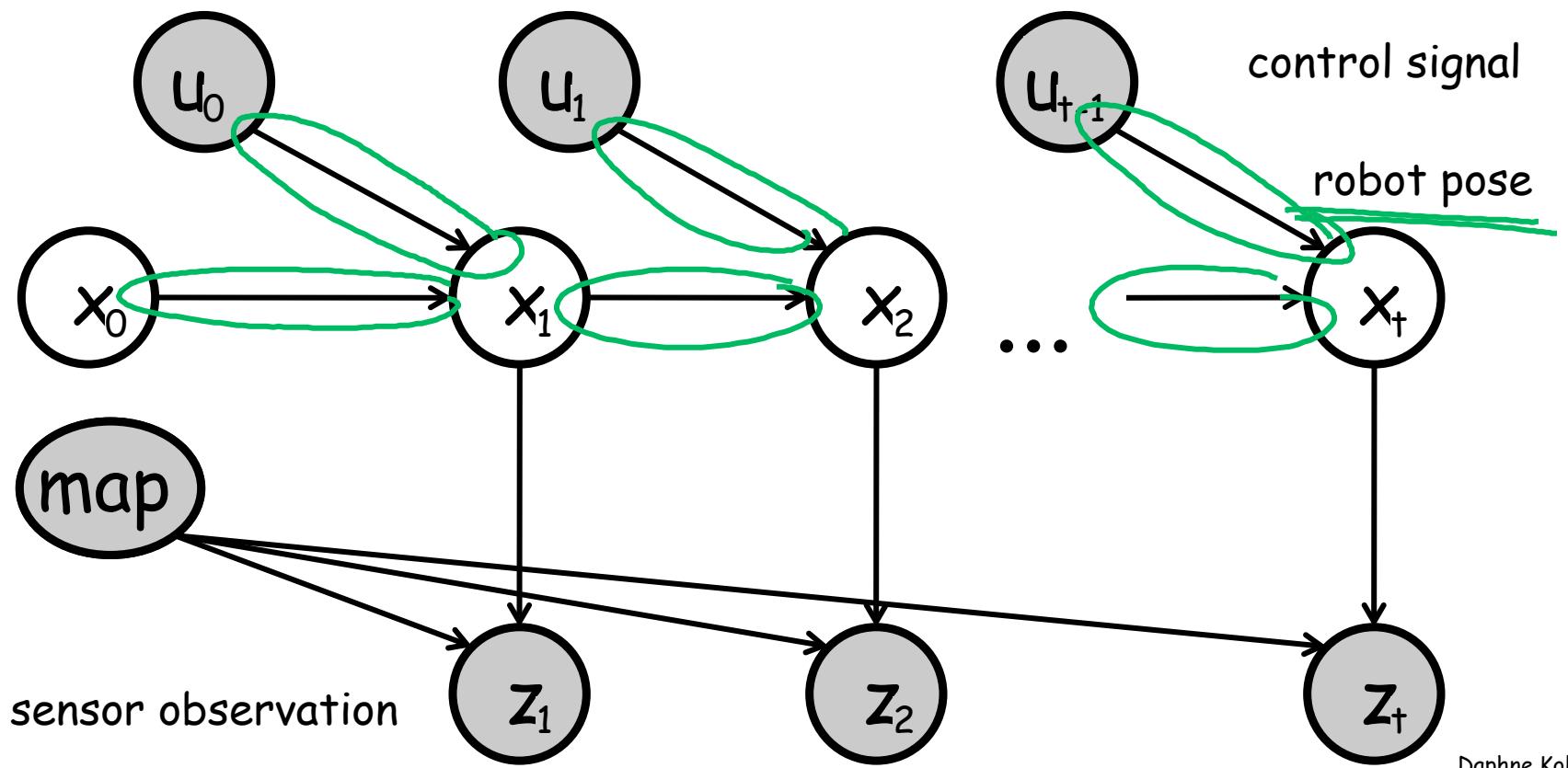
time series
position at
time t
changes over
time
robot dynamics
are fixed



Fox, Burgard, Thrun

Daphne Koller

Robot Localization



Daphne Koller

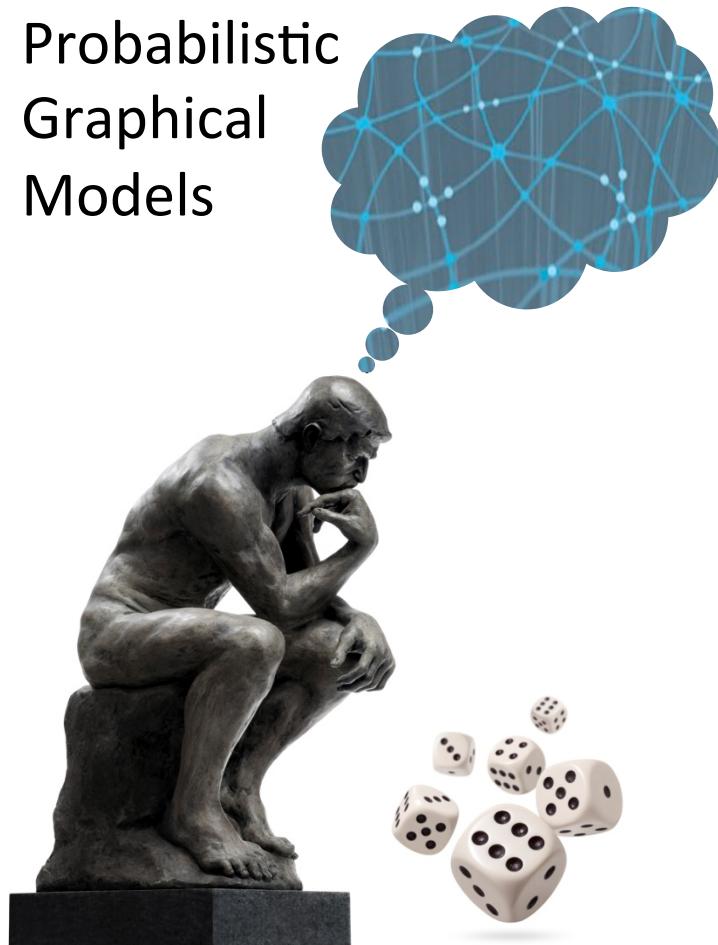
Template Variables

- Template variable $X(U_1, \dots, U_k)$ is instantiated (duplicated) multiple times
 - Location(t), Sonar(t)
 - Genotype(person), Phenotype(person)
 - Label(pixel)
 - Difficulty(course), Intelligence(student), Grade(course, student)

Template Models

- Languages that specify how variables inherit dependency model from template
- Dynamic Bayesian networks ← temporal
- Object-relational models *people, courses, pixels,..*
 - Directed
 - Plate models
 - Undirected

Probabilistic
Graphical
Models



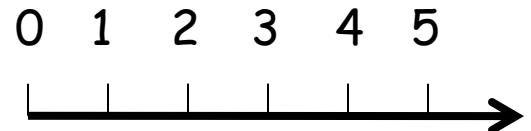
Representation

Template Models

Temporal
Models

Distributions over Trajectories

discretize time



- Pick time granularity $\underline{\Delta}$
- $\underline{X(t)}$ - variable X at time $\underline{t\Delta}$
- $\underline{X(t:t')} = \{X^{(t)}, \dots, X^{(t')}\}$ ($t \leq t'$)
- Want to represent $P(X^{(t:t')})$ for any t, t'

Markov Assumption

$$P(X^{(0:T)}) = P(X^{(0)}) \prod_{t=0}^{T-1} P(\underline{X^{(t+1)}} \mid \underline{X^{(0:t)}}) \quad \begin{matrix} \text{chain rule for} \\ \text{probabilities} \end{matrix}$$

time flows forward

$$(\underline{\underline{X^{(t+1)}}} \perp \boxed{X^{(0:t-1)}} \mid \underline{X^{(t)}}) \quad \text{forgetting}$$

next step *past* *present*

$$P(X^{(0:T)}) = P(X^{(0)}) \prod_{t=0}^{T-1} P(\underline{X^{(t+1)}} \mid \underline{X^{(t)}})$$

Is this true?

$X = \text{Location of robot}$ probably not
 $L^{t+1} \perp L^{t+1} \mid L^t ?$ velocity
enrich state by adding v_t and other variables
(adding dependencies between time - semi-Markov)

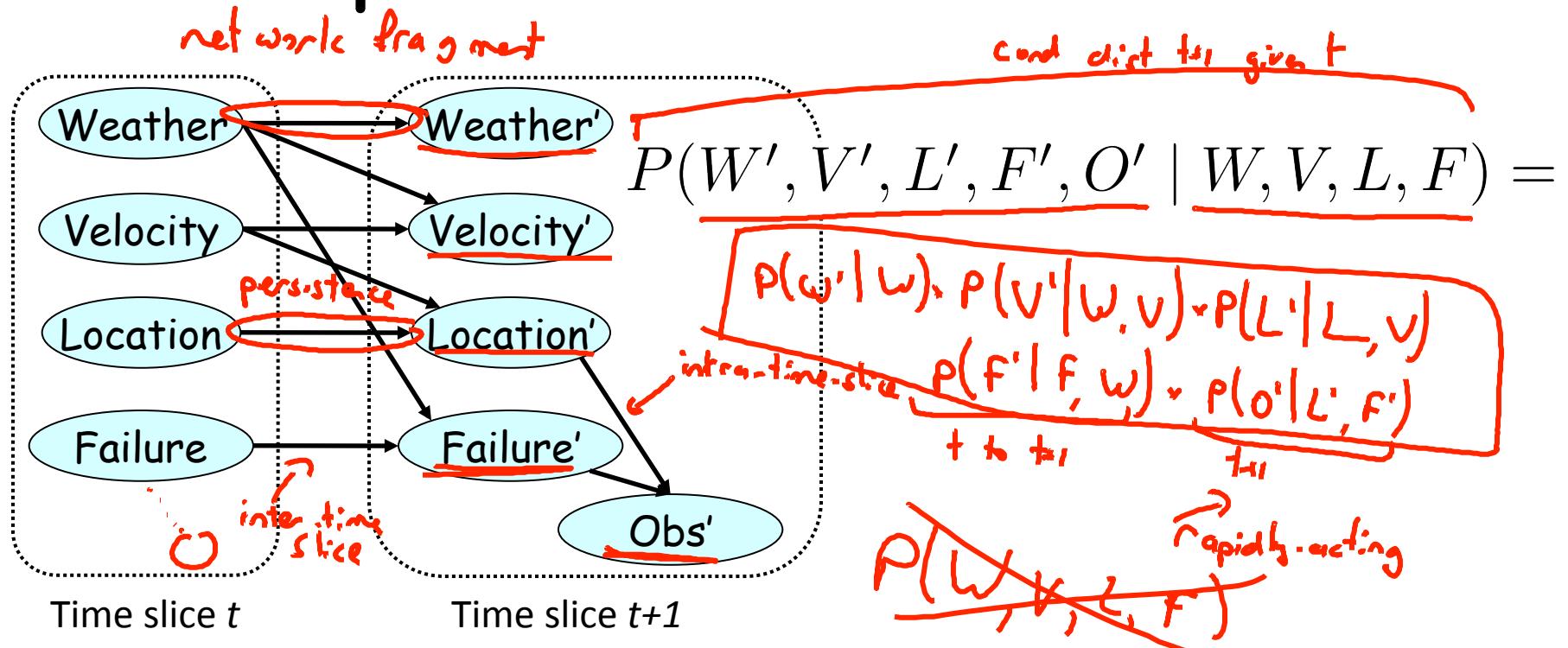
Time Invariance

- Template probability model $P(X' | X)$
- For all t :

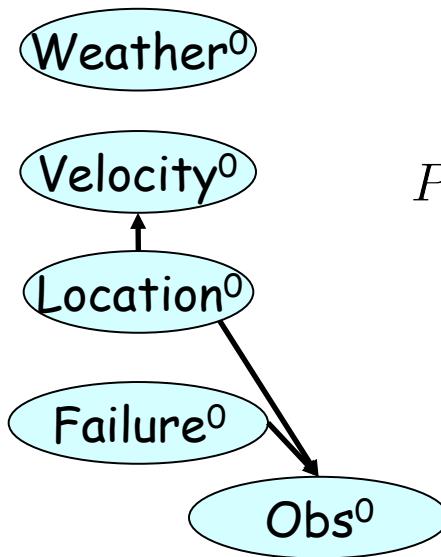
$$P(X^{(t+1)} | X^{(t)}) = P(X' | X)$$

traffic time of day, day of week, football,
enrich model by including

Template Transition Model



Initial State Distribution



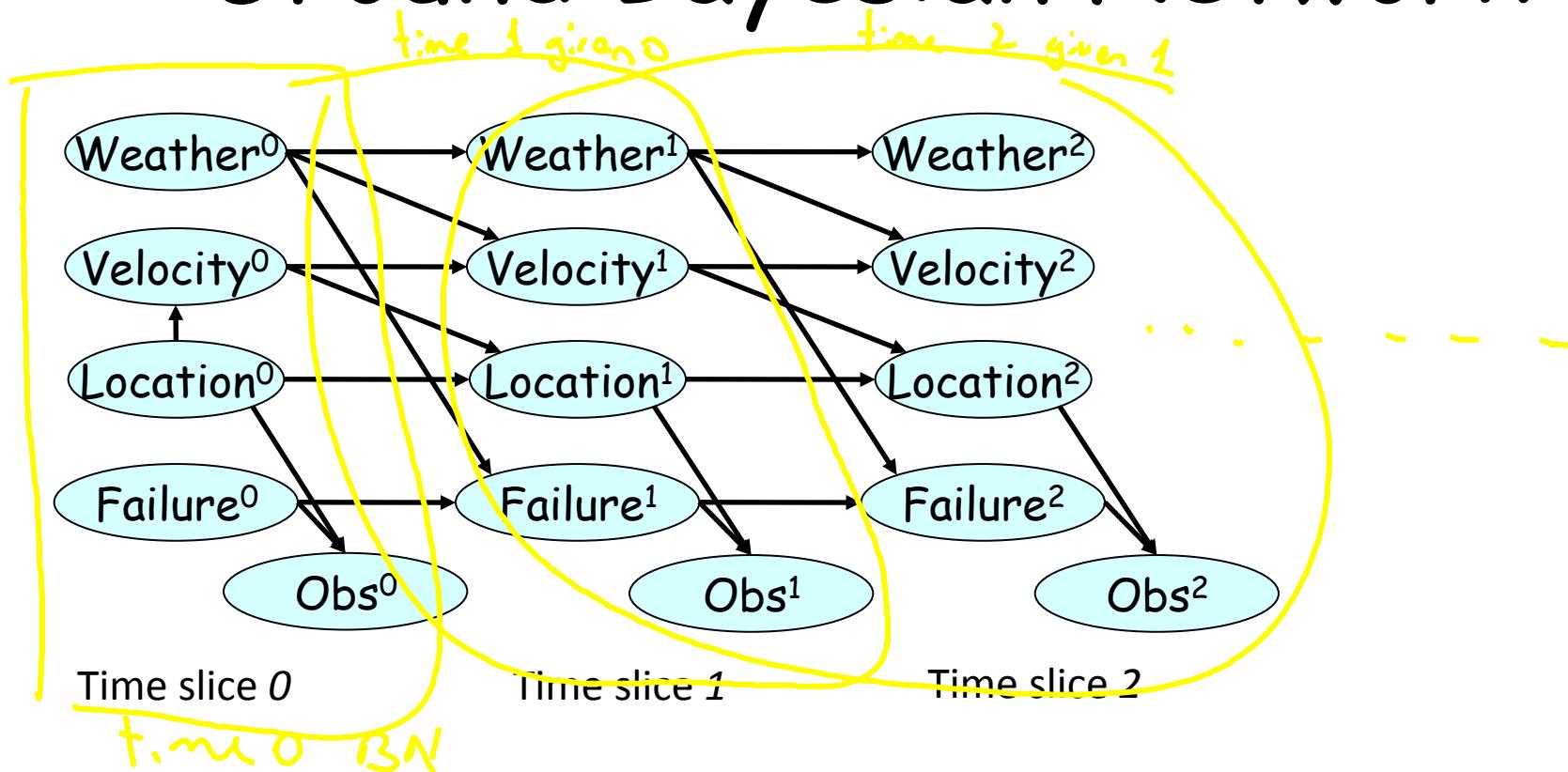
Time slice 0

$$P(W^{(0)}, V^{(0)}, L^{(0)}, F^{(0)}, O^{(0)}) =$$

$$P(W^{(0)})P(V^{(0)} \mid L^{(0)})P(L^{(0)})P(F^{(0)})P(O^{(0)} \mid F^{(0)}, L^{(0)})$$

chain rule

Ground Bayesian Network



2-time-slice Bayesian Network

- A transition model (2TBN) over X_1, \dots, X_n is specified as a BN fragment such that:
 - The nodes include X'_1, \dots, X'_n and a subset of X_1, \dots, X_n
 - Only the nodes X'_1, \dots, X'_n have parents and a CPD the time + vars that directly affect state at t+1
- The 2TBN defines a conditional distribution

$$\underline{P(X' | X)} = \prod_{i=1}^n P(\underline{X'_i} | \text{Pa}_{X'_i})$$

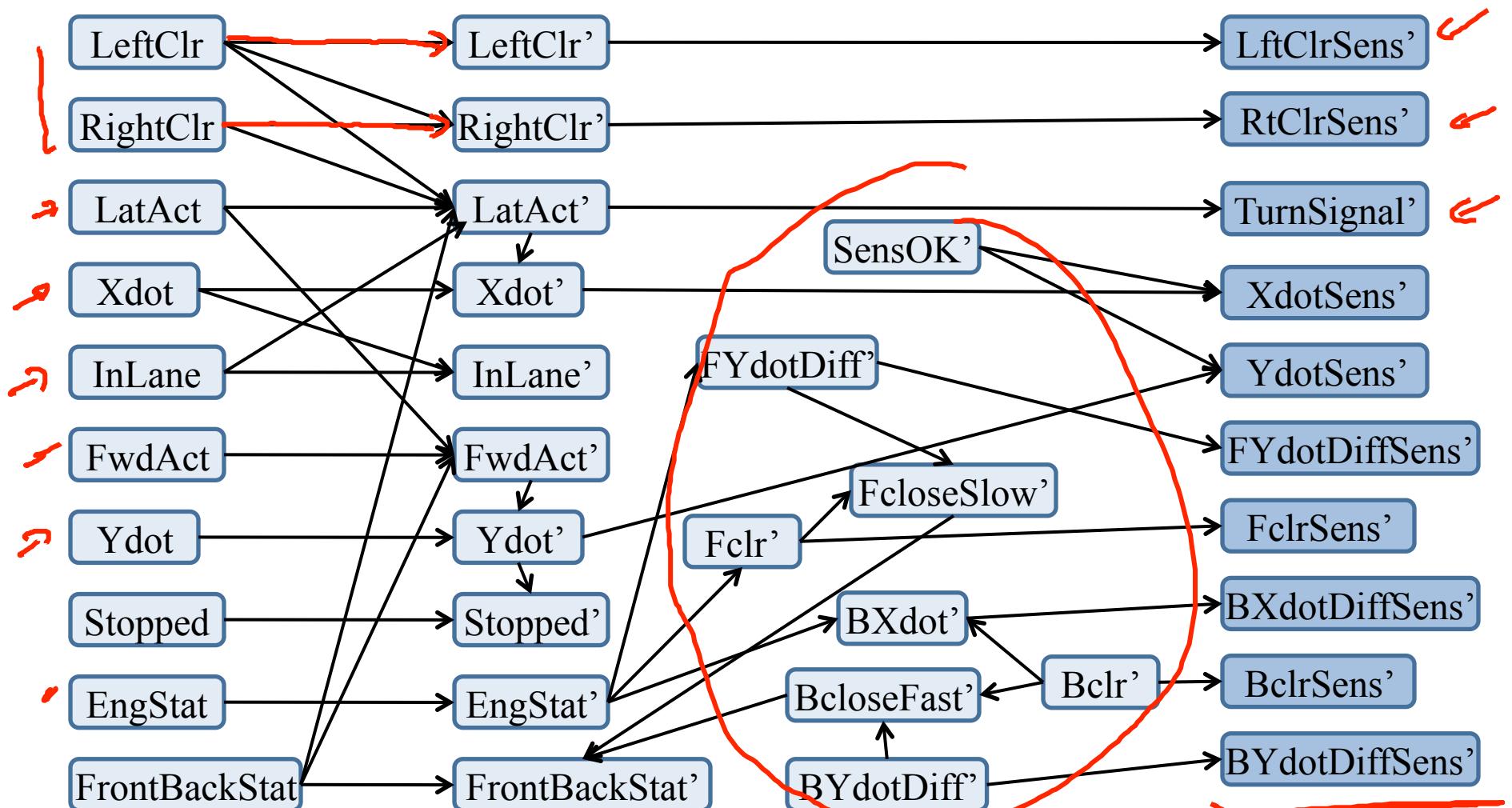
chain rule

Dynamic Bayesian Network

- A dynamic Bayesian network (DBN) over X_1, \dots, X_n is defined by a
 - 2 TBN BN over X_1, \dots, X_n *dynamics*
 - a Bayesian network BN⁽⁰⁾ over $X_1^{(0)}, \dots, X_n^{(0)}$

Ground Network

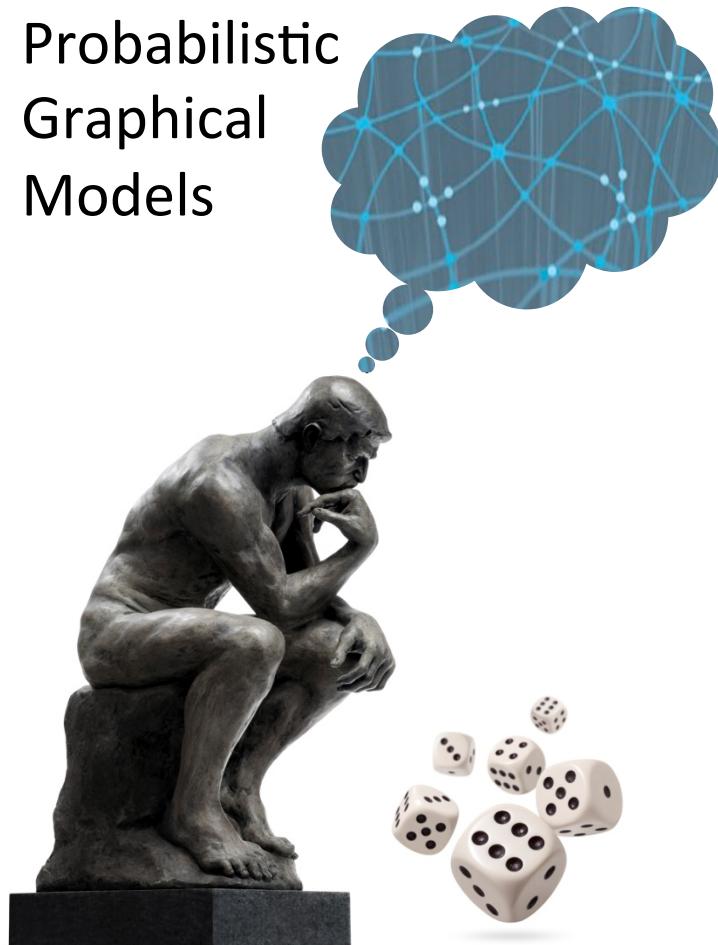
- For a trajectory over $0, \dots, T$ we define a ground (unrolled network) such that
 - The dependency model for $X_1^{(0)}, \dots, X_n^{(0)}$ is copied from $BN^{(0)}$
 - The dependency model for $X_1^{(t)}, \dots, X_n^{(t)}$ for all $t > 0$ is copied from BN_{\rightarrow}



Summary

- DBNS are a compact representation for encoding structured distributions over arbitrarily long temporal trajectories
- They make assumptions that may require appropriate model (re)design:
 - Markov assumption
 - Time invariance

Probabilistic
Graphical
Models

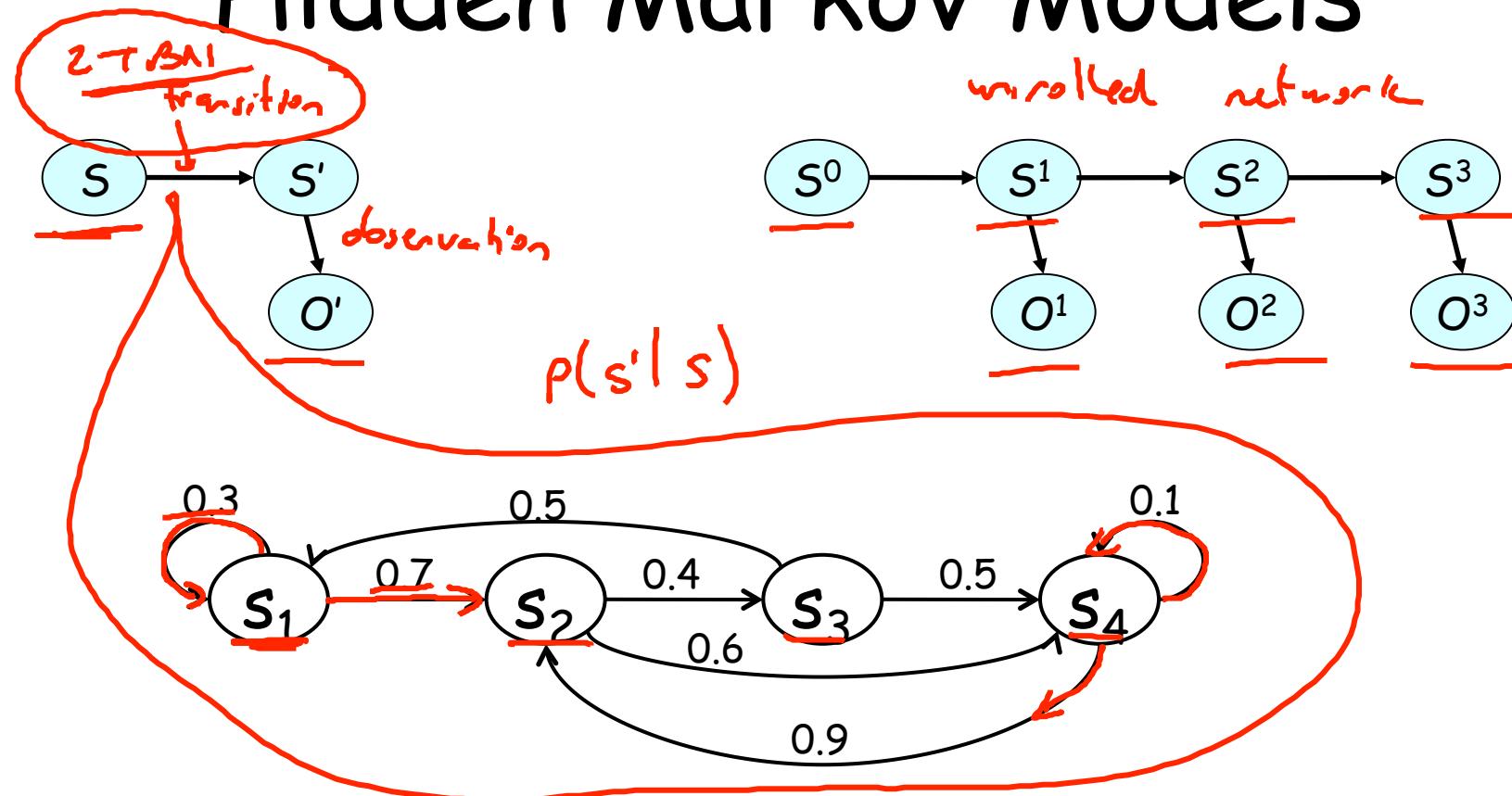


Representation

Template Models

Hidden
Markov
Models

Hidden Markov Models

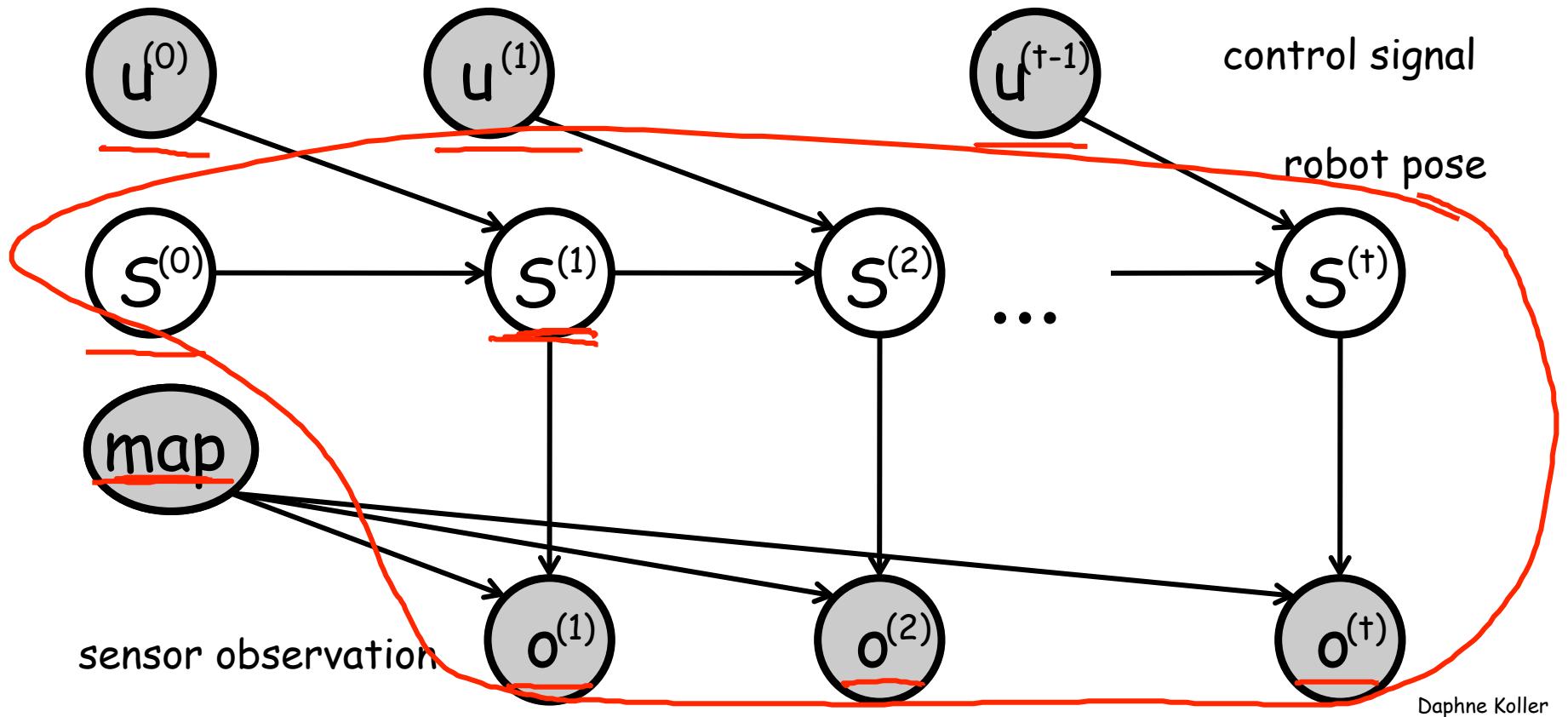


Daphne Koller

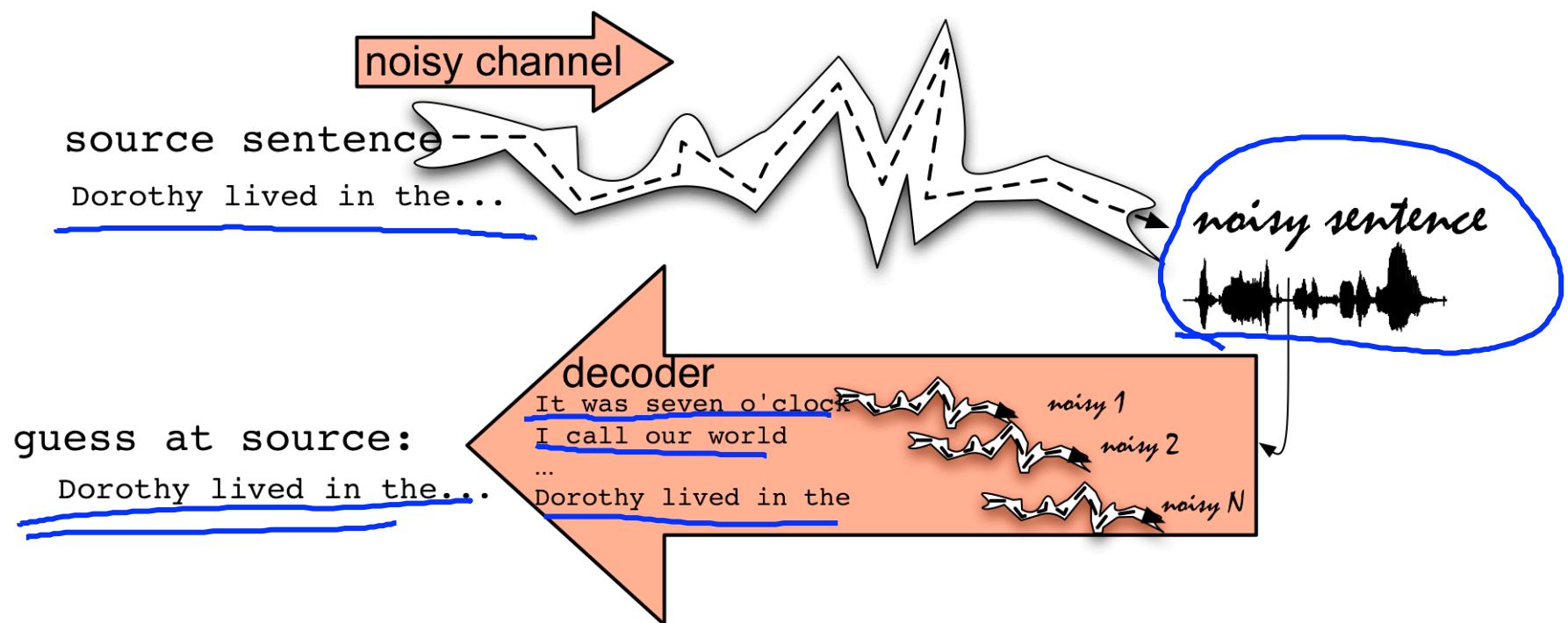
Numerous Applications

- Robot localization
- Speech recognition
- Biological sequence analysis
- Text annotation

Robot Localization



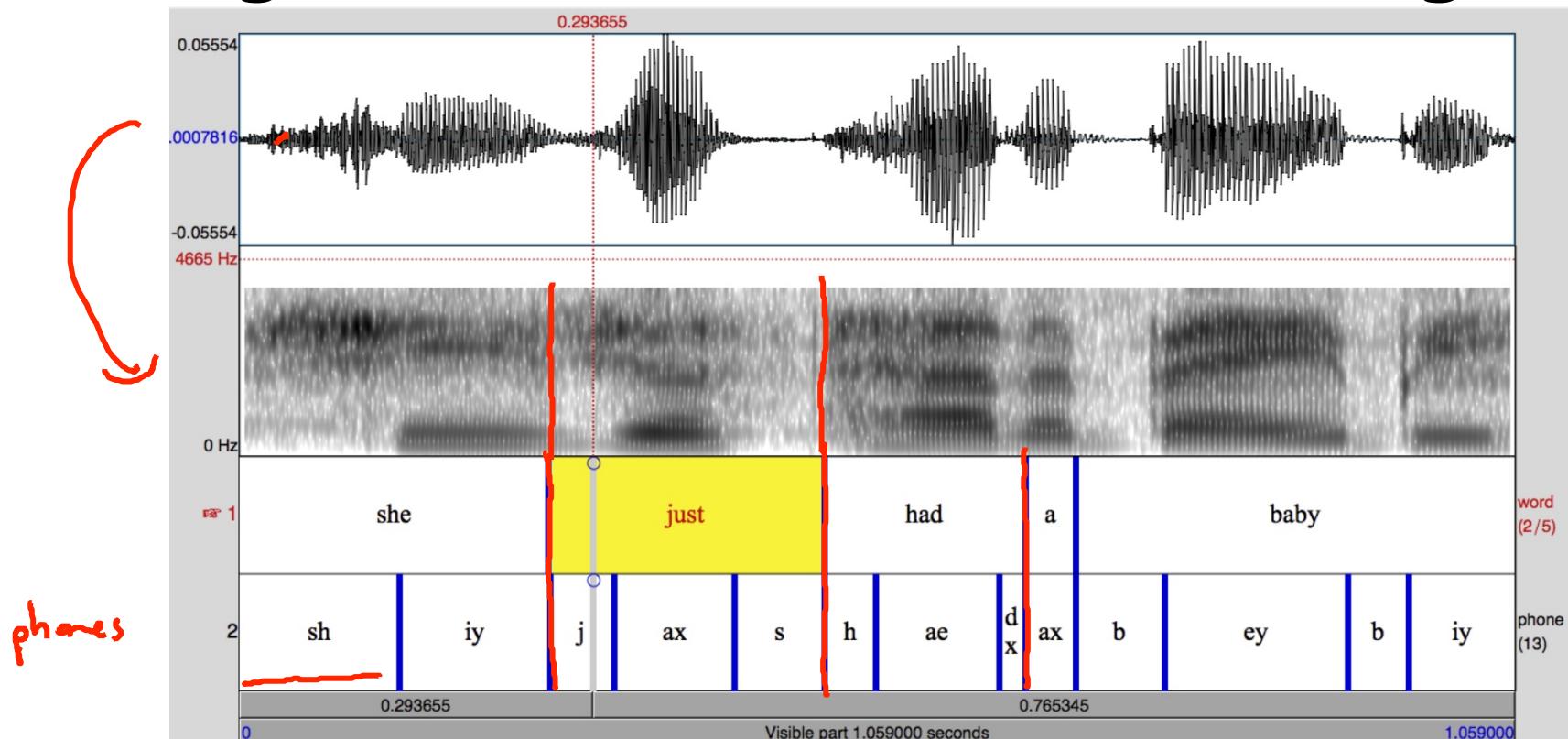
Speech Recognition



Dan Jurafsky, Stanford

Daphne Koller

Segmentation of Acoustic Signal



Dan Jurafsky, Stanford

Daphne Koller

Phonetic Alphabet

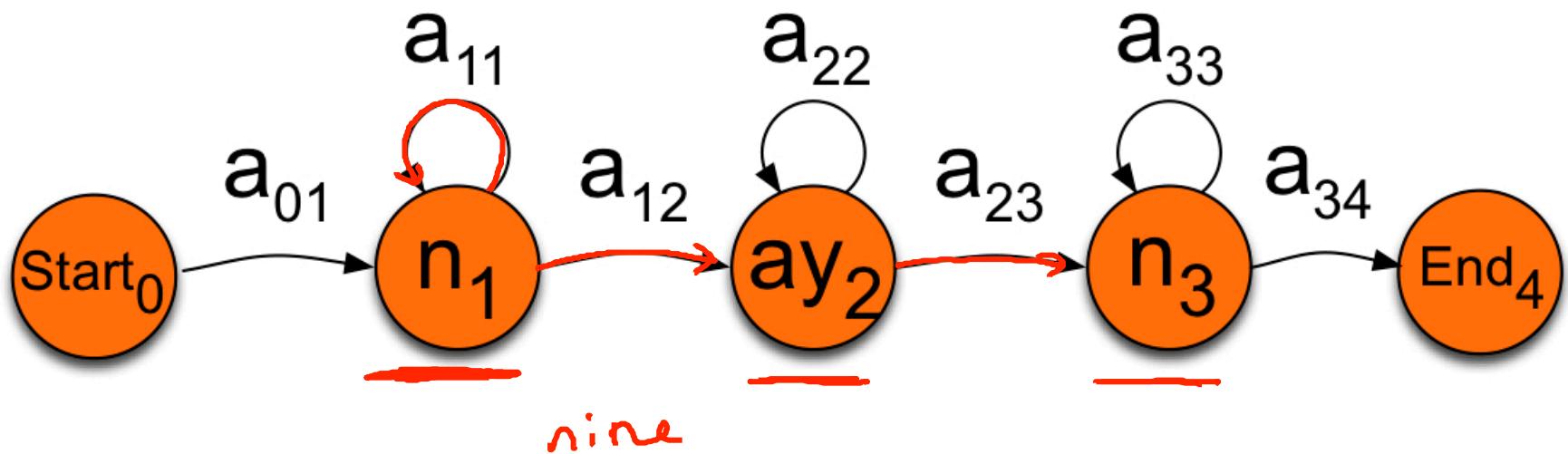
• AA	odd	AA D	• G	green	G R I Y N	• R	read	R I Y D
• AE	at	AE T	• HH	he	HH I Y	• S	sea	S I Y
• AH	hut	HH AH T	• IH	it	I H T	• SH	she	SH I Y
• AO	ought	A O T	• IY	eat	I Y T	• T	tea	T I Y
• AW	cow	K A W	• JH	gee	J H I Y	• TH	theta	TH E Y T A H
• AY	hide	HH A Y D	• K	key	K I Y	• UH	hood	HH U H D
• B	be	B I Y	• L	lee	L I Y	• UW	two	T U W
• CH	cheese	CH I Y Z	• M	me	M I Y	• V	vee	V I Y
• D	dee	D I Y	• N	knee	N I Y	• W	we	W I Y
• DH	thee	D H I Y	• NG	ping	P I H N G	• Y	yield	Y I Y L D
• EH	Ed	E H D	• OW	oat	O W T	• Z	zee	Z I Y
• ER	hurt	HH E R T	• OY	toy	T O Y	• ZH	seizure	S I Y Z H E R
• EY	ate	E Y T	• P	pee	P I Y			
• F	fee	F I Y						

<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

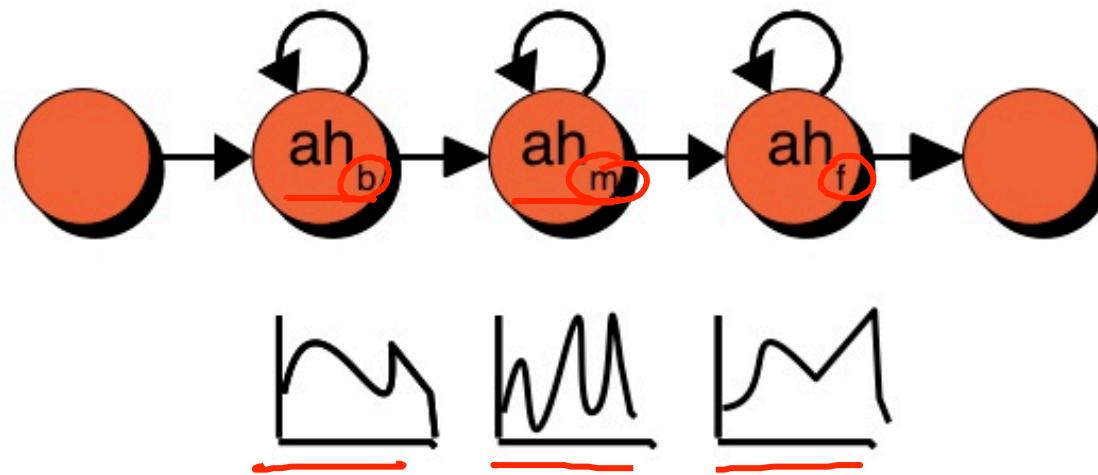


The CMU Pronouncing Dictionary

Word HMM



Phone HMM



Dan Jurafsky, Stanford

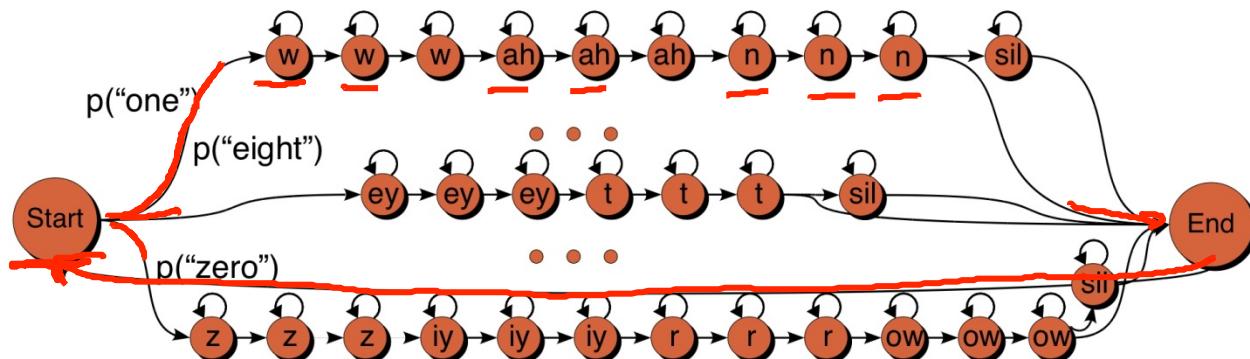
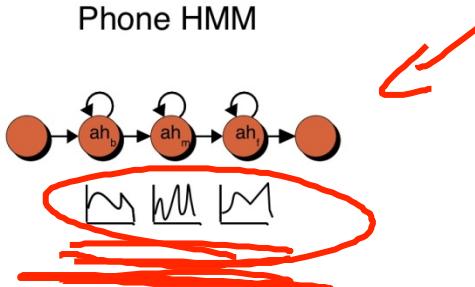
Daphne Koller

Recognition HMM

Lexicon

one	w ah n
two	t uw
three	th r iy
four	f ao r
five	f ay v
six	s ih k s
seven	s eh v ax n
eight	ey t
nine	n ay n
zero	z iy r ow

Phone HMM



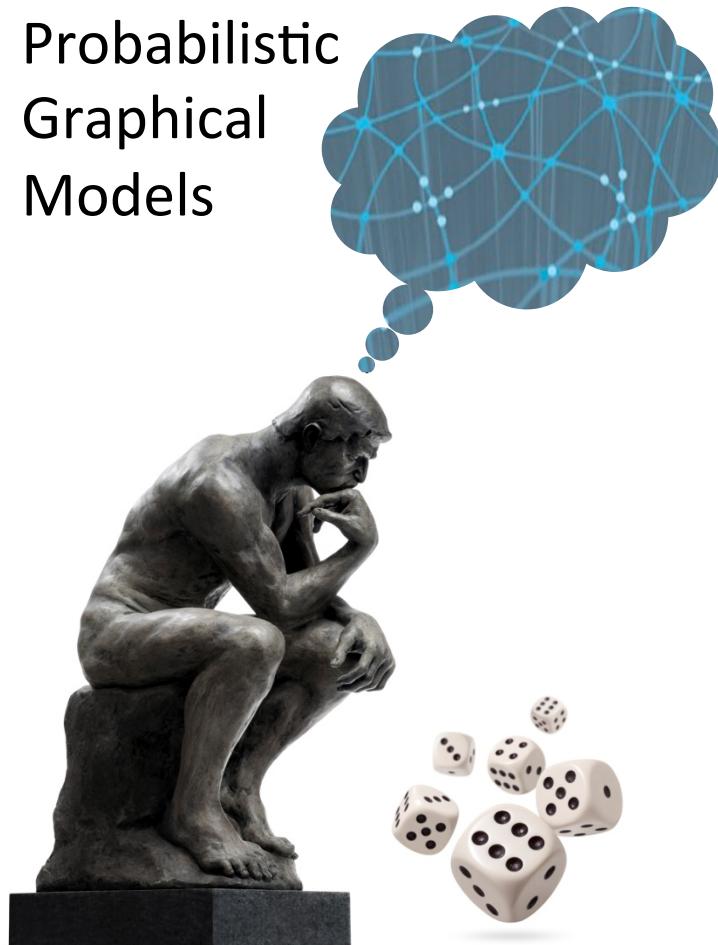
Dan Jurafsky, Stanford

Daphne Koller

Summary

- HMMs can be viewed as a subclass of DBNs
- HMMs seem unstructured at the level of random variables
- HMM structure typically manifests in sparsity and repeated elements within the transition matrix
- HMMs are used in a wide variety of applications for modeling sequences

Probabilistic
Graphical
Models

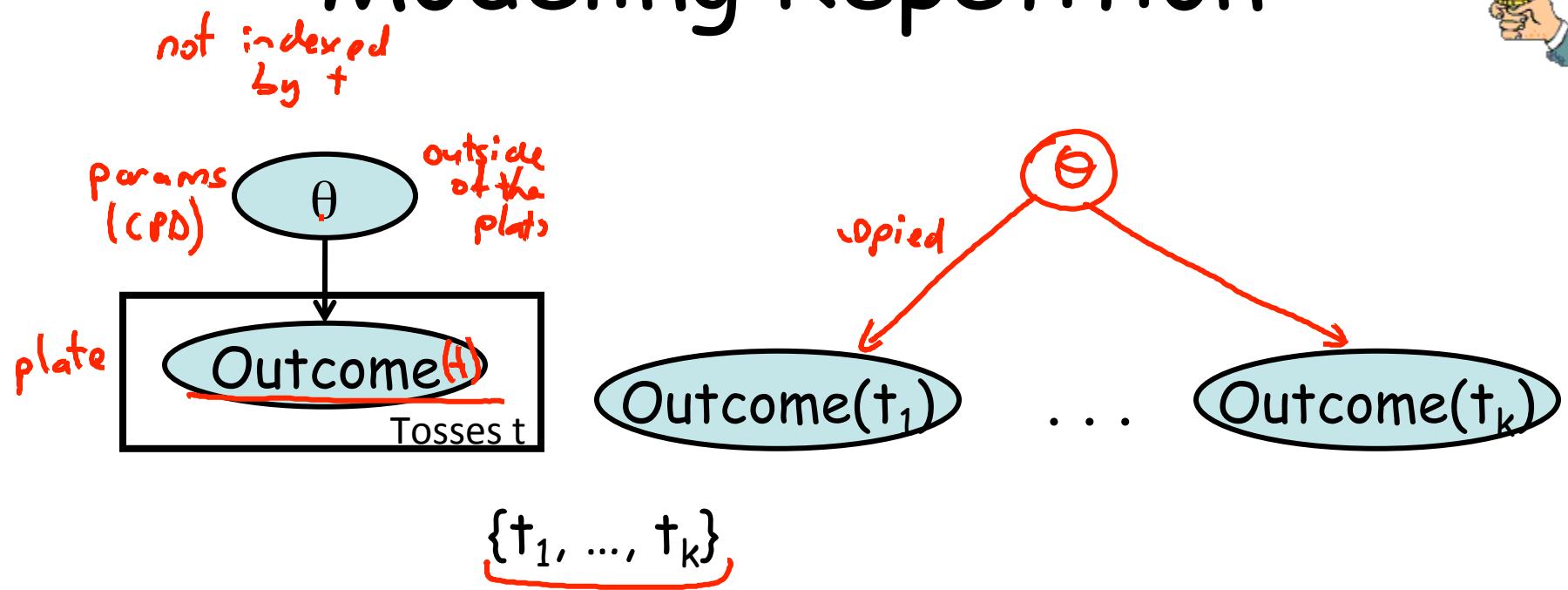


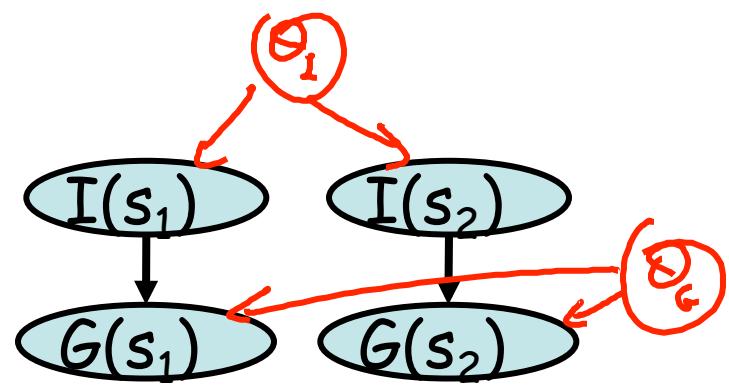
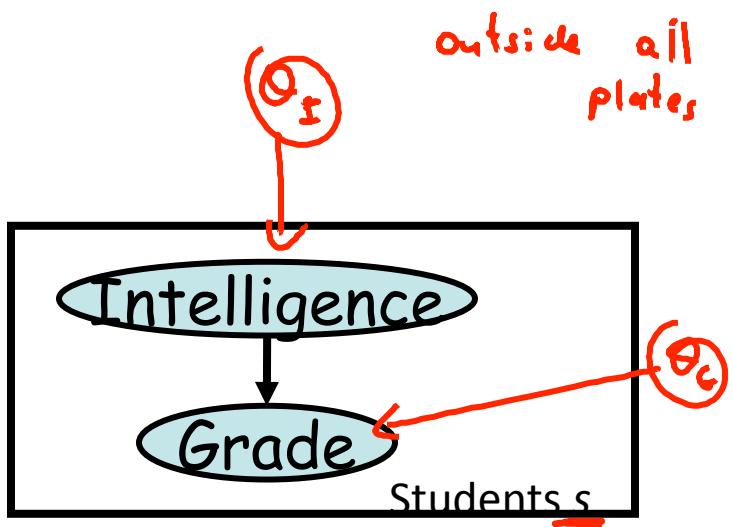
Representation

Template Models

Plate Models

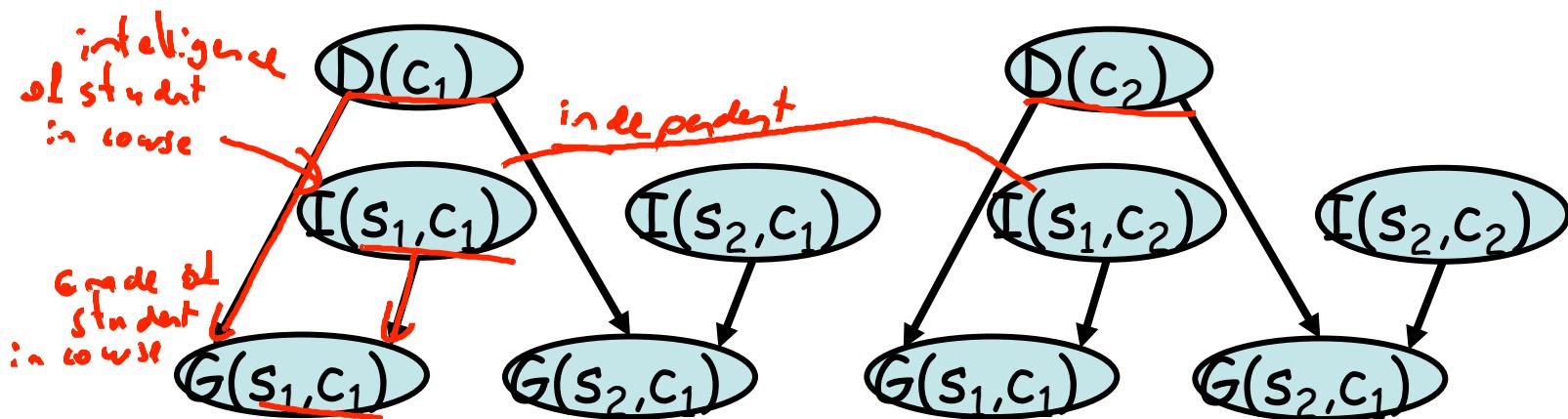
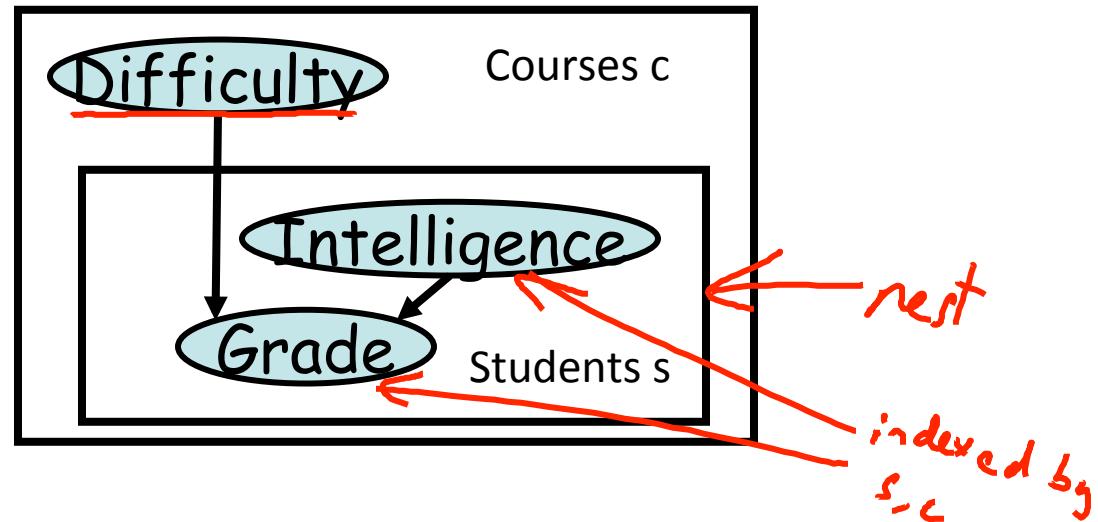
Modeling Repetition





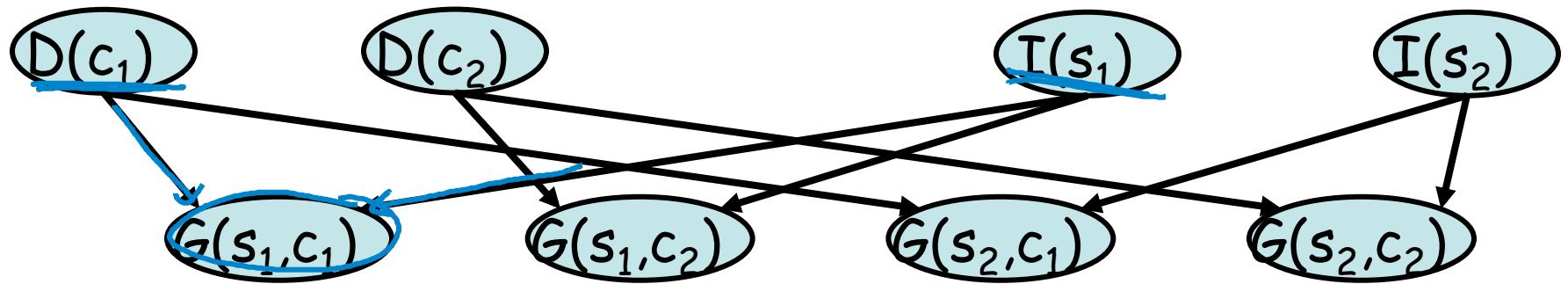
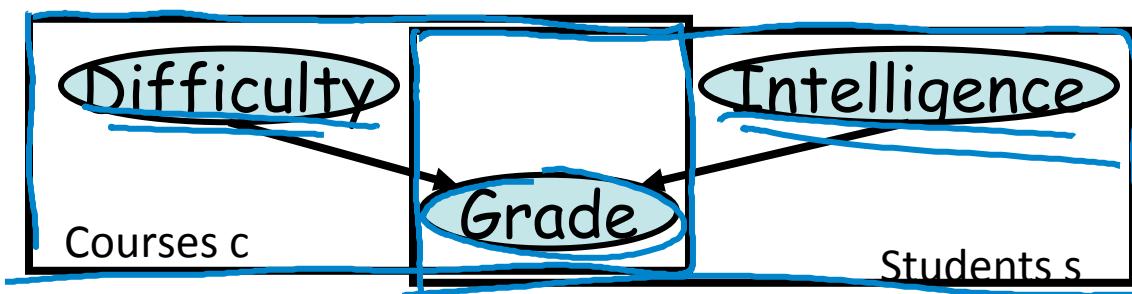
Nested Plates

courses c
students s



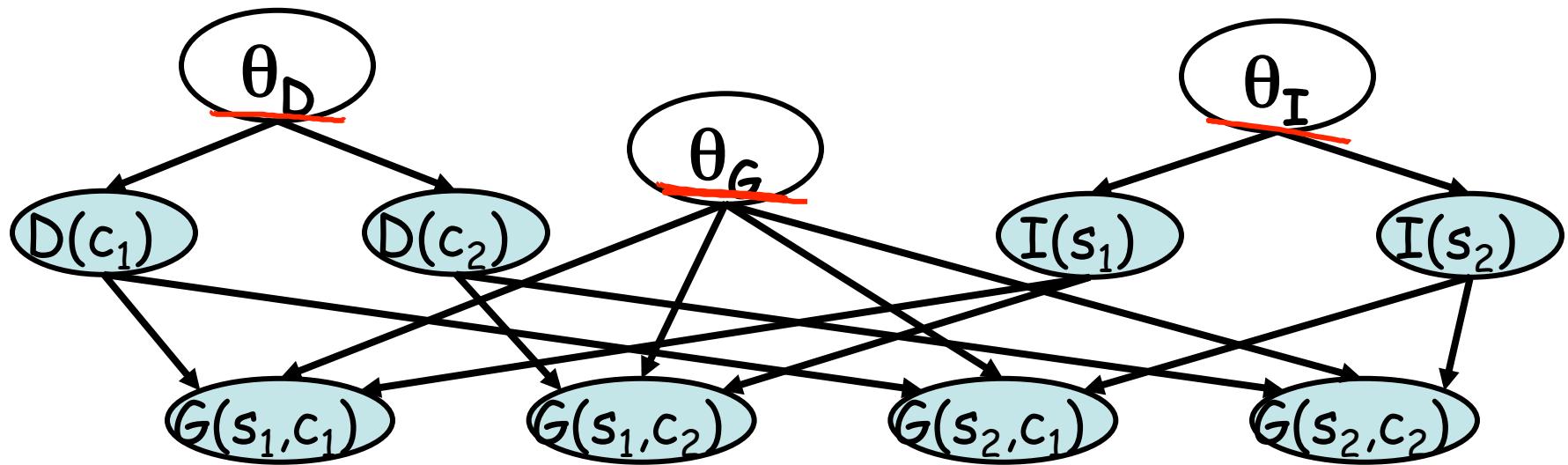
Daphne Koller

Overlapping Plates



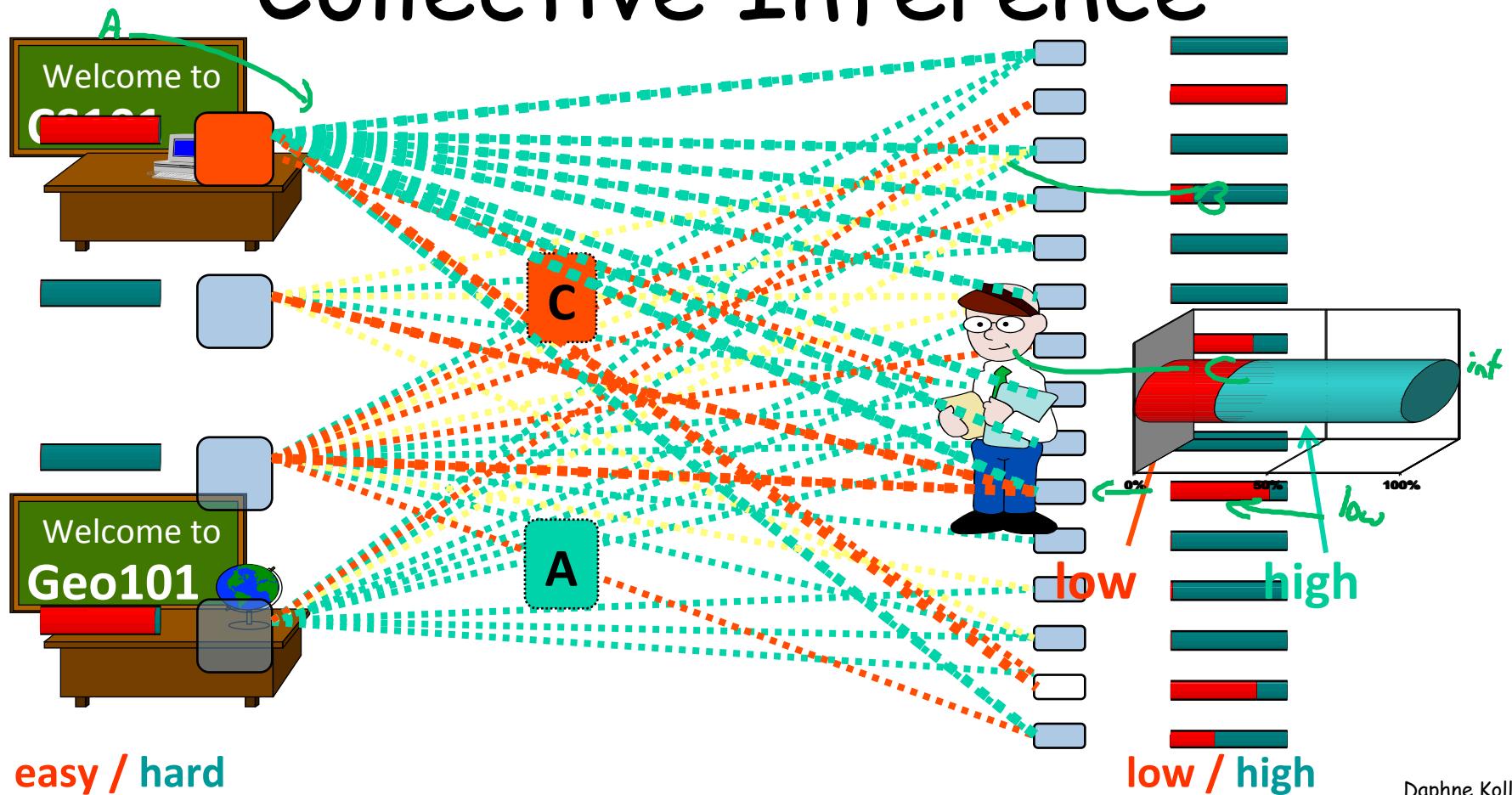
Daphne Koller

Explicit Parameter Sharing



Daphne Koller

Collective Inference



Daphne Koller

Plate Dependency Model

- For a template variable $A(U_1, \dots, U_k)$:

– Template parents $B_1(U_1), \dots, B_m(U_m)$

$$\begin{array}{ccc} I(s) & & D(c) \\ \downarrow & & \downarrow \\ G(s, c) & & \end{array}$$

template

$$\begin{array}{ccc} G(s, c) & & U_i \subseteq \{u_1, \dots, u_n\} \\ \downarrow & & \\ \text{Honors}_3(s) & & \begin{array}{l} \text{unbounded} \\ \# \text{ of parents} \end{array} \end{array}$$

aggregator CPD

– CPD $P(A | B_1, \dots, B_m)$

Ground Network

Let $A(U_1, \dots, U_k)$ with parents $B_1(U_1), \dots, B_m(U_m)$

- for any instantiation u_1, \dots, u_k to U_1, \dots, U_k we would have:

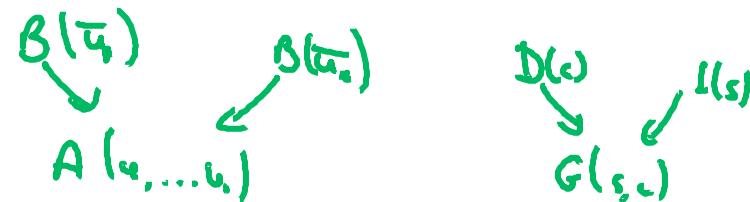


Plate Dependency Model

Let $A(U_1, \dots, U_k)$ with parents $B_1(U_1), \dots, B_m(U_m)$

- For each i , we must have $U_i \subseteq U_1, \dots, U_k$
 - No indices in parent that are not in child

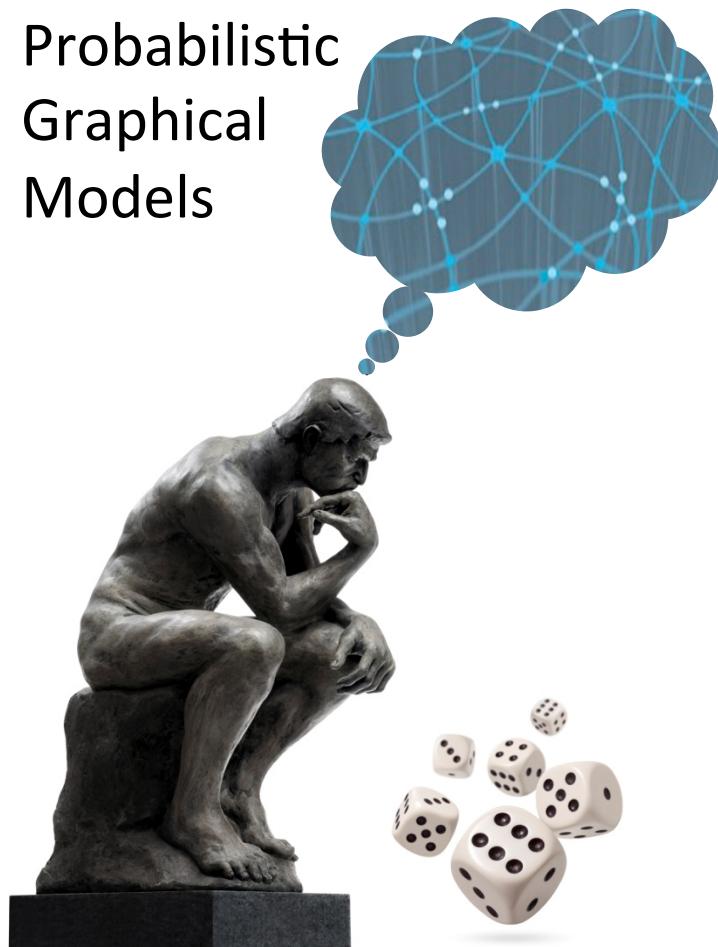
$$\begin{matrix} G(n) \\ \searrow & \swarrow \\ G(p) & \\ \downarrow & \\ G(c) \end{matrix}$$

Summary

$$x^{+,-} \rightarrow x^+$$

- Template for an infinite set of BNs, each induced by a different set of domain objects
- Parameters and structure are reused within a BN and across different BNs
- Models encode correlations across multiple objects, allowing collective inference
- Multiple "languages", each with different tradeoffs in expressive power

Probabilistic
Graphical
Models

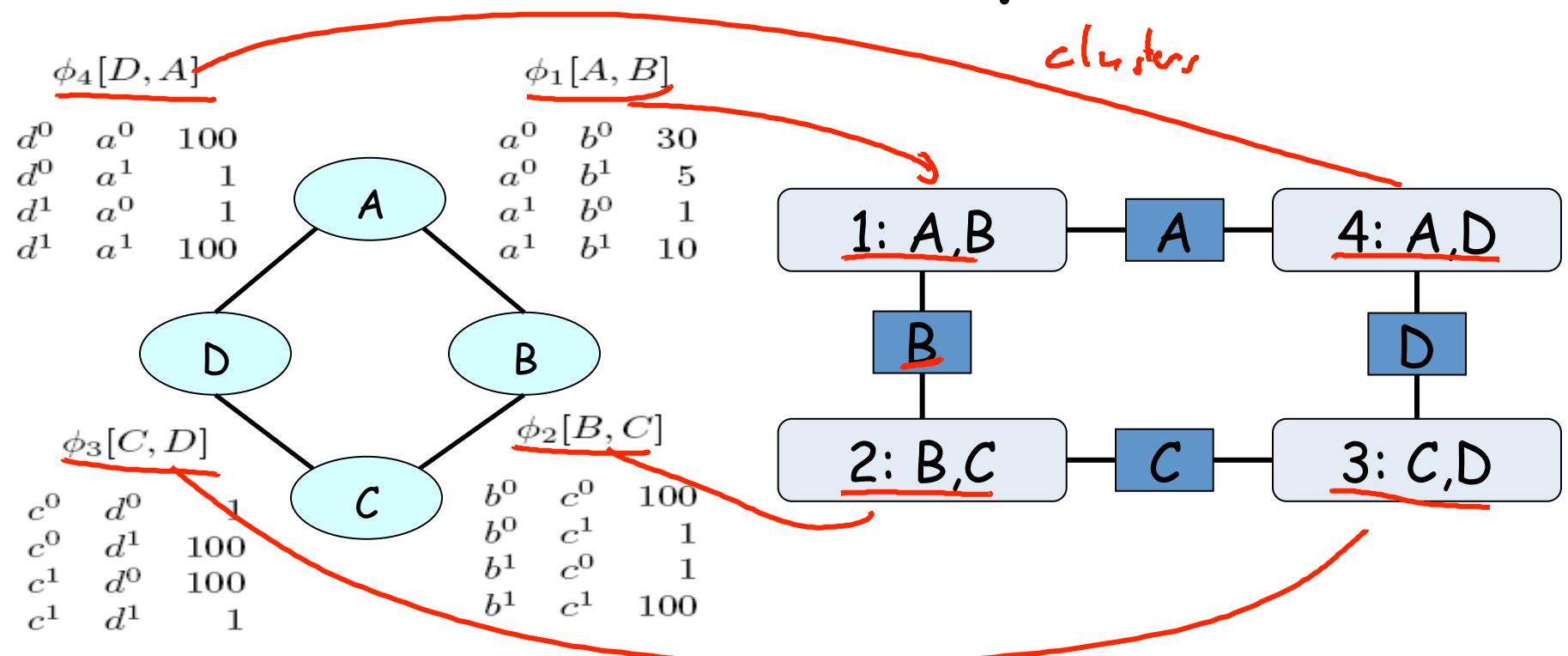


Inference

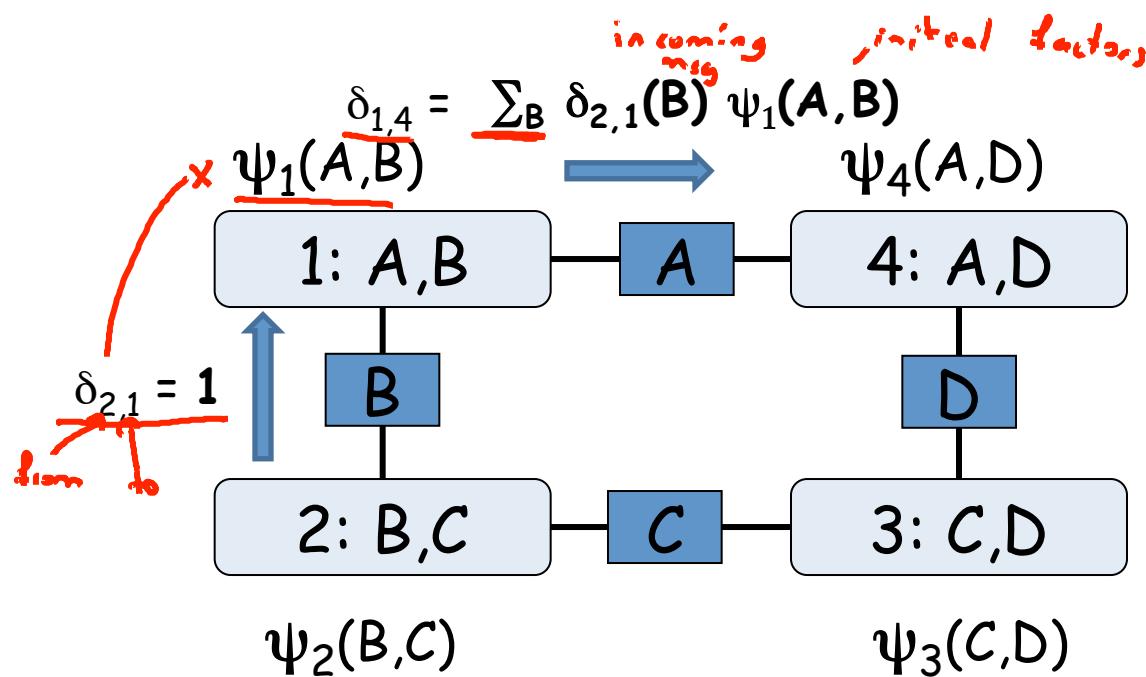
Message Passing

Belief
Propagation
Algorithm

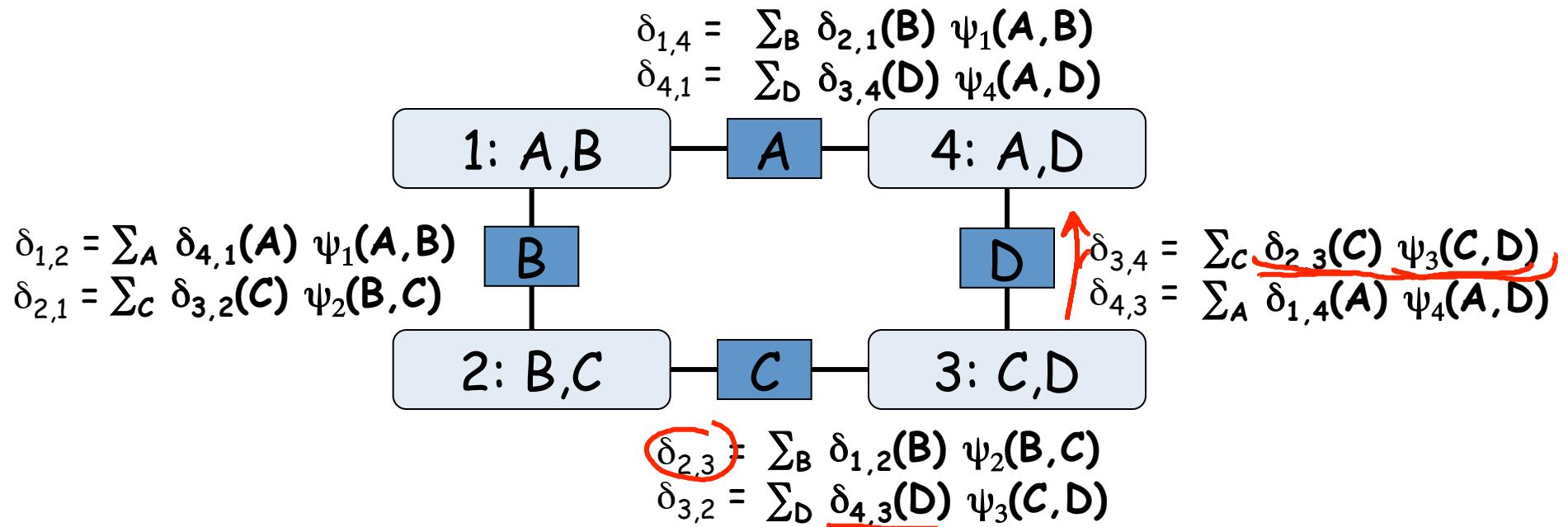
Cluster Graph



Passing Messages



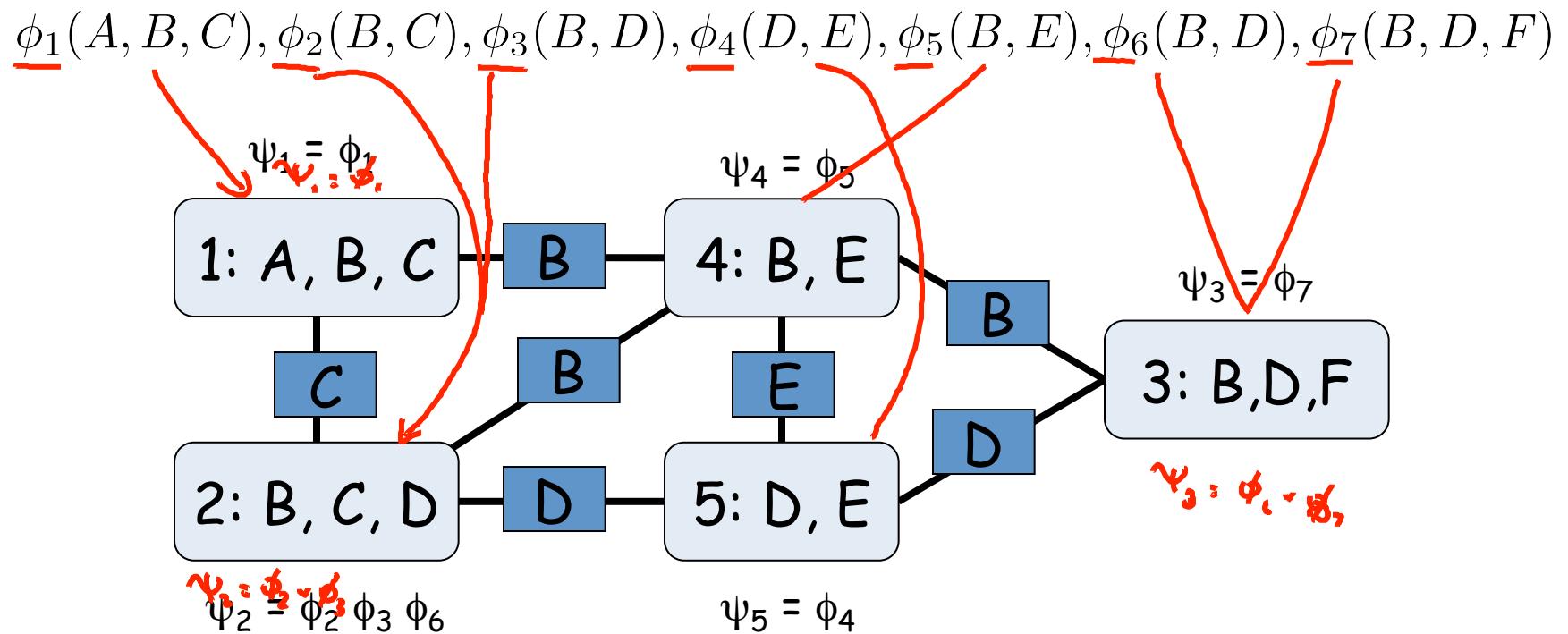
Passing Messages



Cluster Graphs

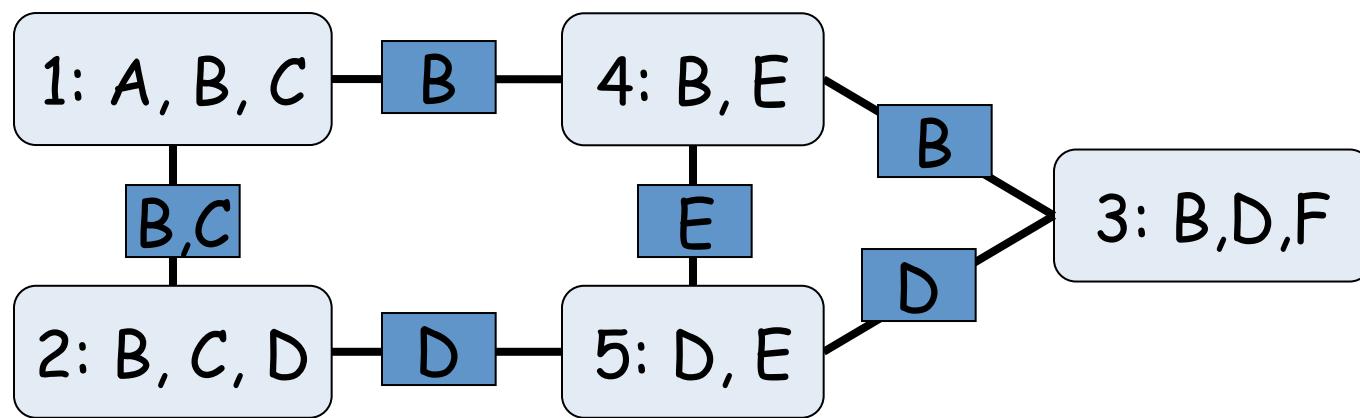
- Undirected graph such that:
 - nodes are clusters $C_i \subseteq \{X_1, \dots, X_n\}$ *Subsets of variables*
 - edge between C_i and C_j associated with sepset $S_{i,j} \subseteq C_i \cap C_j$ *Variables that they talk about*
- Given set of factors Φ , we assign each ϕ_k to a cluster $C_{\alpha(k)}$ s.t. Scope [ϕ_k] $\subseteq C_{\alpha(k)}$
- Define $\psi_i(C_i) = \prod_{k: \alpha(k)=i} \phi_k$ *all factors assigned to it*

Example Cluster Graph



Different Cluster Graph

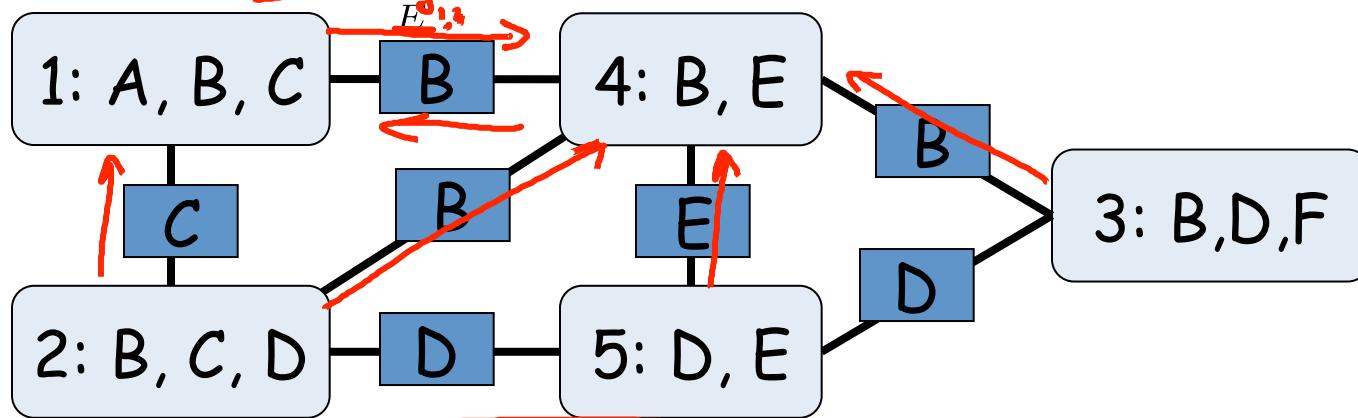
$\phi_1(A, B, C), \phi_2(B, C), \phi_3(B, D), \phi_4(D, E), \phi_5(B, E), \phi_6(B, D), \phi_7(B, D, F)$



Message Passing

$$\delta_{1 \rightarrow 4}(B) = \sum_{A,C} \psi_1(A, B, C) \delta_{2 \rightarrow 1}(C)$$

$$\delta_{4 \rightarrow 1}(B) = \sum_{E} \psi_4(B, E) \times \delta_{2 \rightarrow 4}(B) \times \delta_{5 \rightarrow 4}(E) \times \delta_{3 \rightarrow 4}(B)$$



$$\delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi_i \times \prod_{k \in (\mathcal{N}_i - \{j\})} \delta_{k \rightarrow i}$$

incoming msgs
only from;
from;

Daphne Koller

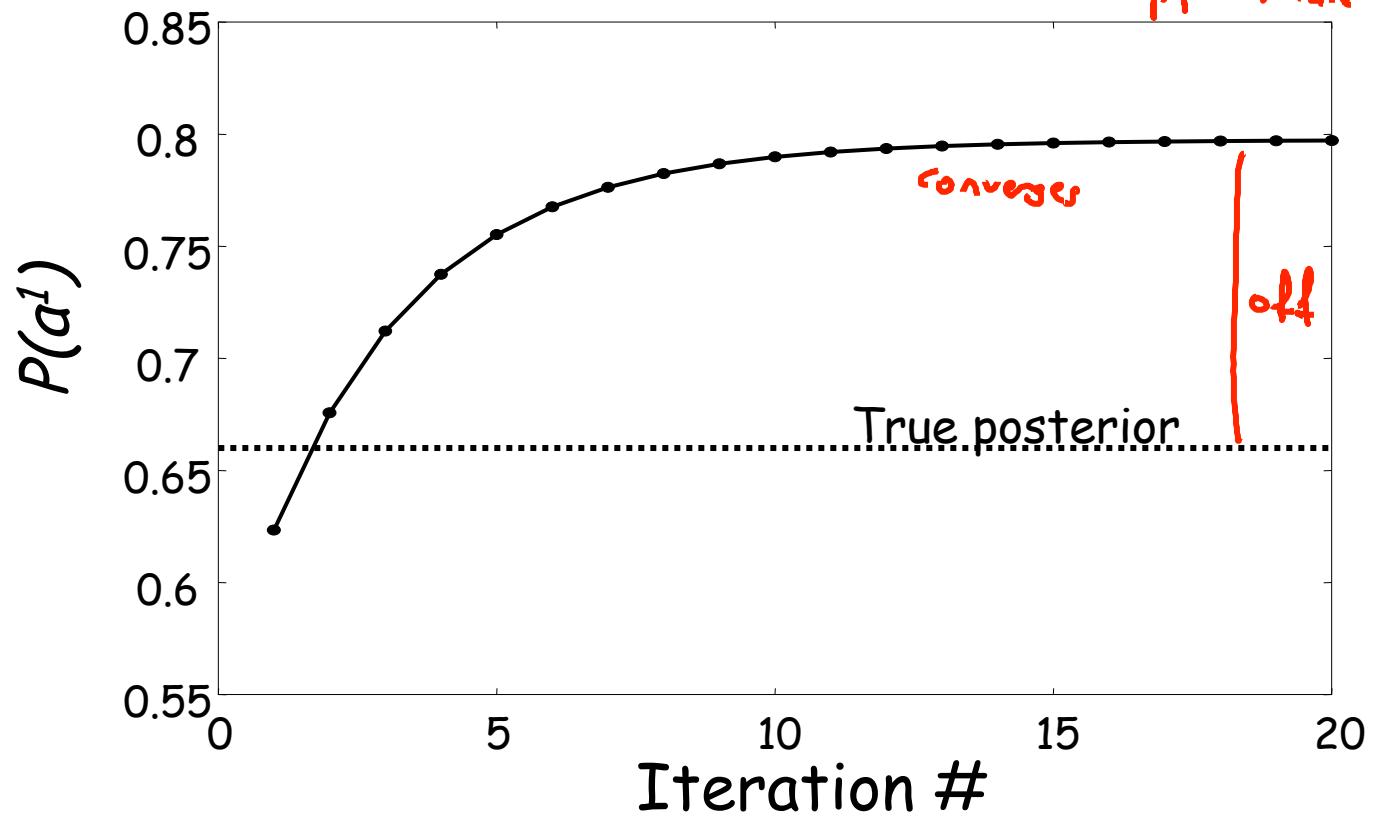
Belief Propagation Algorithm

- Assign each factor $\phi_k \in \Phi$ to a cluster $C_{\alpha(k)}$
- Construct initial potentials $\psi_i(C_i) = \prod_{k:\alpha(k)=i} \phi_k$
- Initialize all messages to be 1
- Repeat until when?
 - Select edge (i,j) and pass message

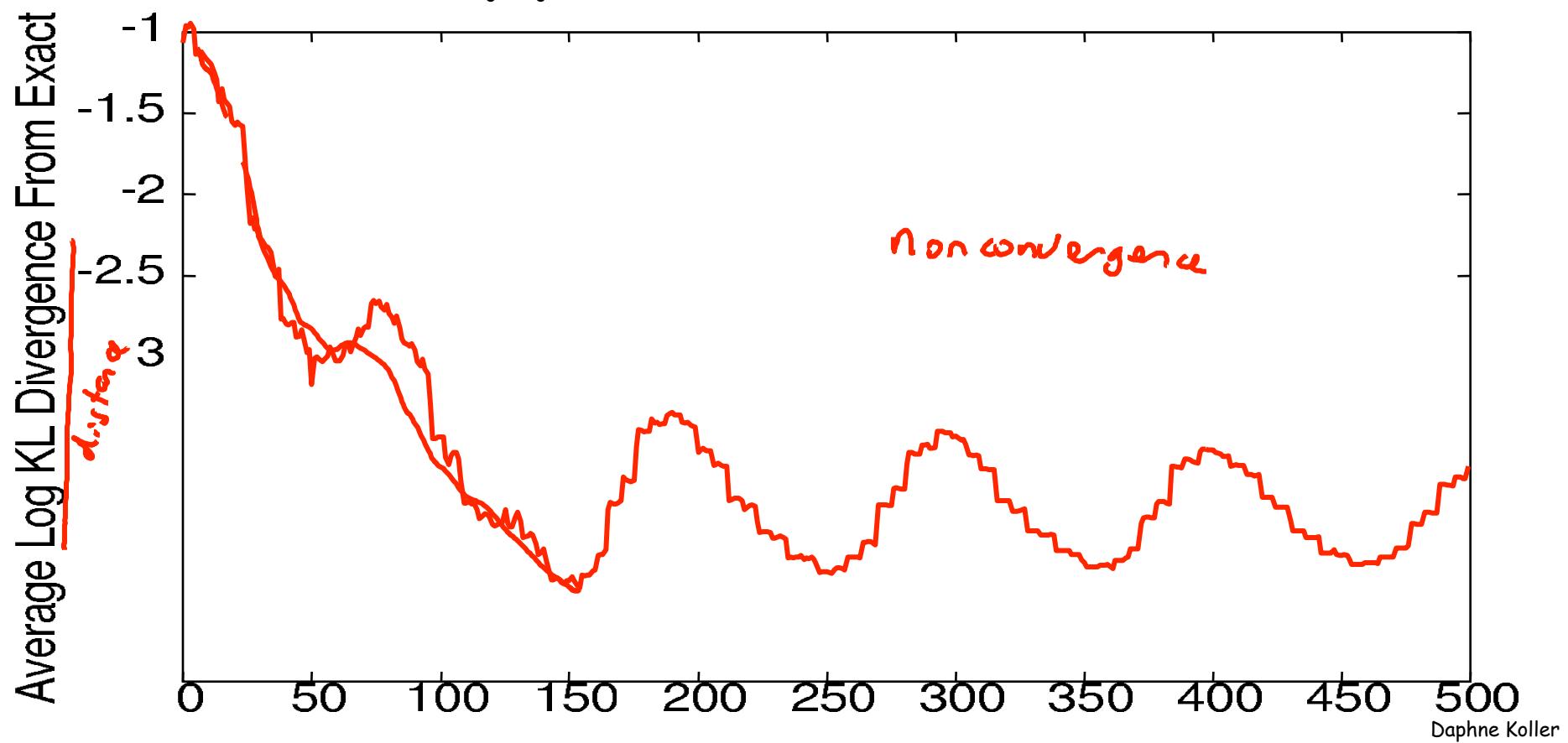
$$\delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \psi_i \times \prod_{k \in (\mathcal{N}_i - \{j\})} \delta_{k \rightarrow i}$$

- Compute $\beta_i(C_i) = \psi_i \times \prod_{k \in \mathcal{N}_i} \delta_{k \rightarrow i}$ — all neighbors

Belief Propagation Run



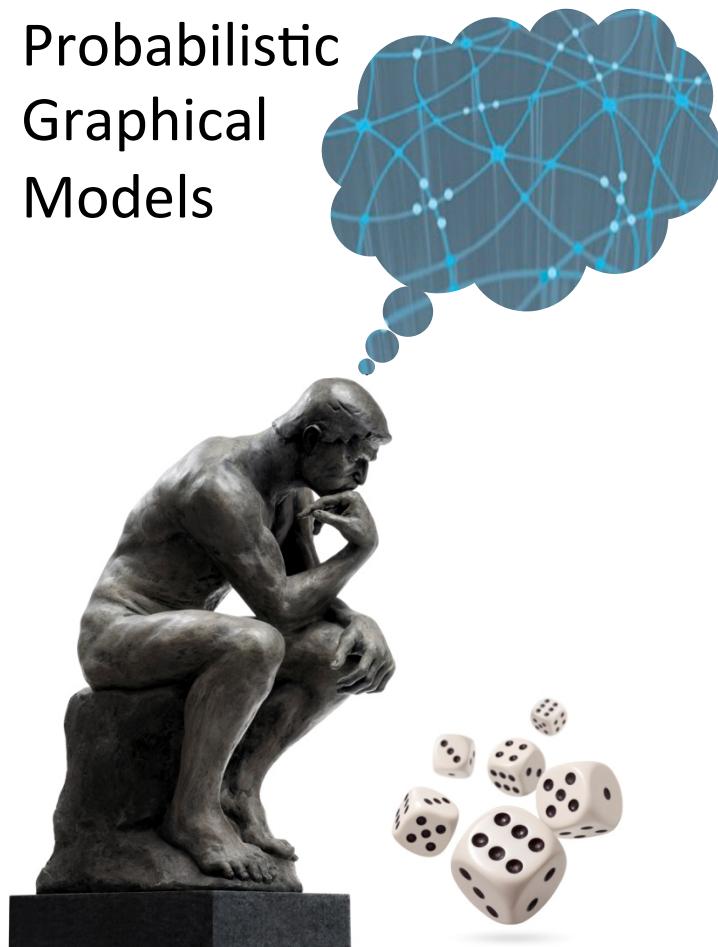
Different BP Run



Summary

- Graph of clusters connected by sepsets
- Adjacent clusters pass information to each other about variables in sepset
 - Message from i to j summarizes everything i knows, except information obtained from j
- Algorithm may not converge *not marginals + P_j*
- The resulting beliefs are pseudo-marginals
- Nevertheless, very useful in practice

Probabilistic
Graphical
Models



Inference

Message Passing

Cluster Graph Properties

Cluster Graphs

- Undirected graph such that:
 - nodes are clusters $C_i \subseteq \{X_1, \dots, X_n\}$
 - edge between C_i and C_j associated with sepset $S_{i,j} \subseteq C_i \cap C_j$

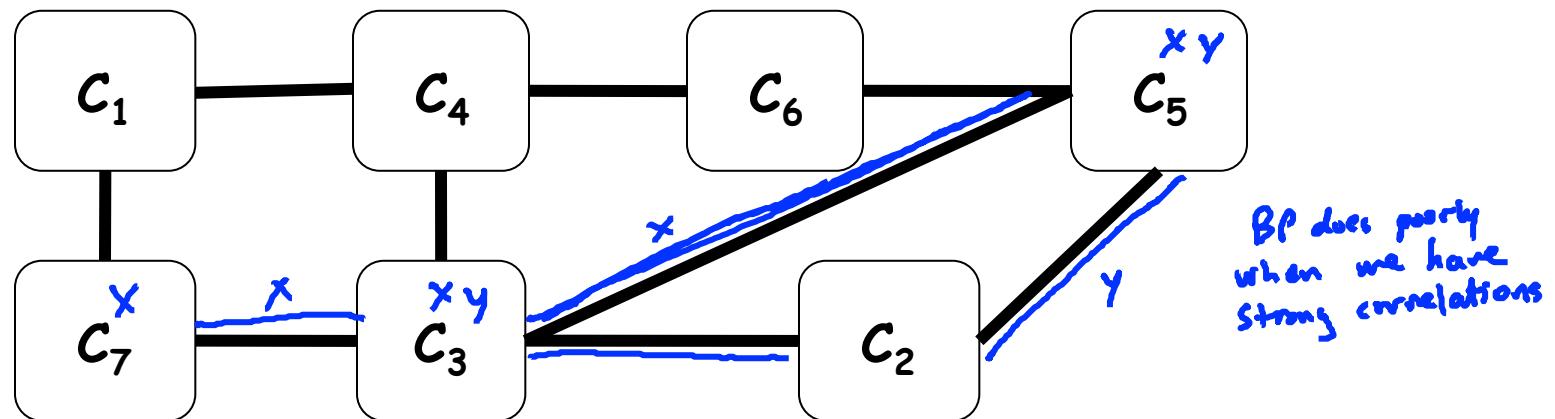
Family Preservation

- Given set of factors Φ , we assign each ϕ_k to a cluster $C_{\alpha(k)}$ s.t. $\text{Scope}[\phi_k] \subseteq C_{\alpha(k)}$
- For each factor $\phi_k \in \Phi$, there exists a cluster C_i s.t. $\text{Scope}[\phi_k] \subseteq C_i$ \leftarrow accommodates ϕ_k

Running Intersection Property

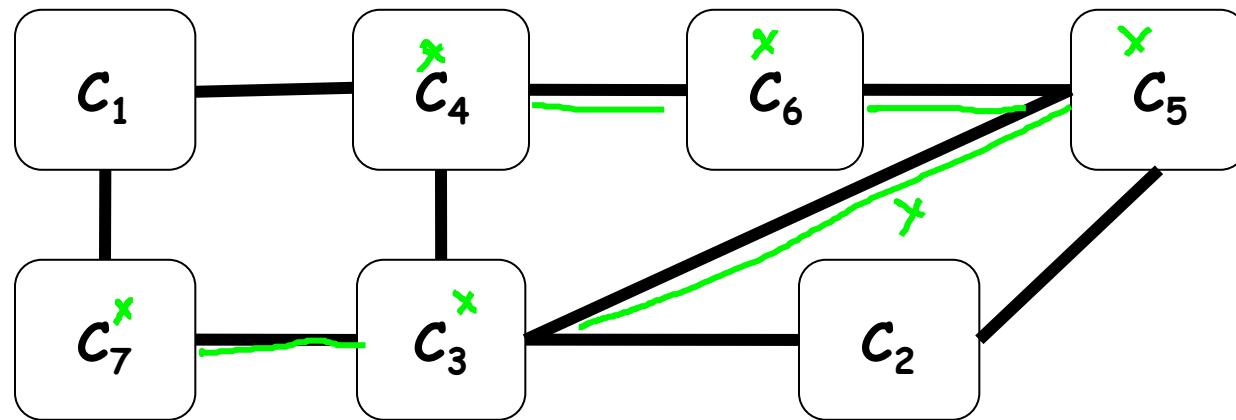
- For each pair of clusters C_i, C_j and variable $X \in C_i \cap C_j$ there exists a unique path between C_i and C_j for which all clusters and sepsets contain X

x and Y that are very strongly correlated

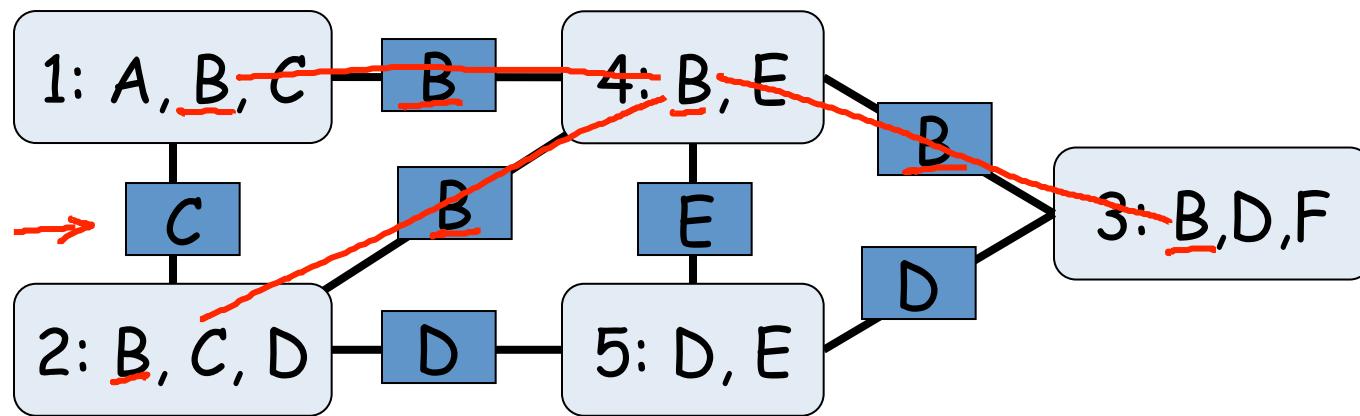


Running Intersection Property

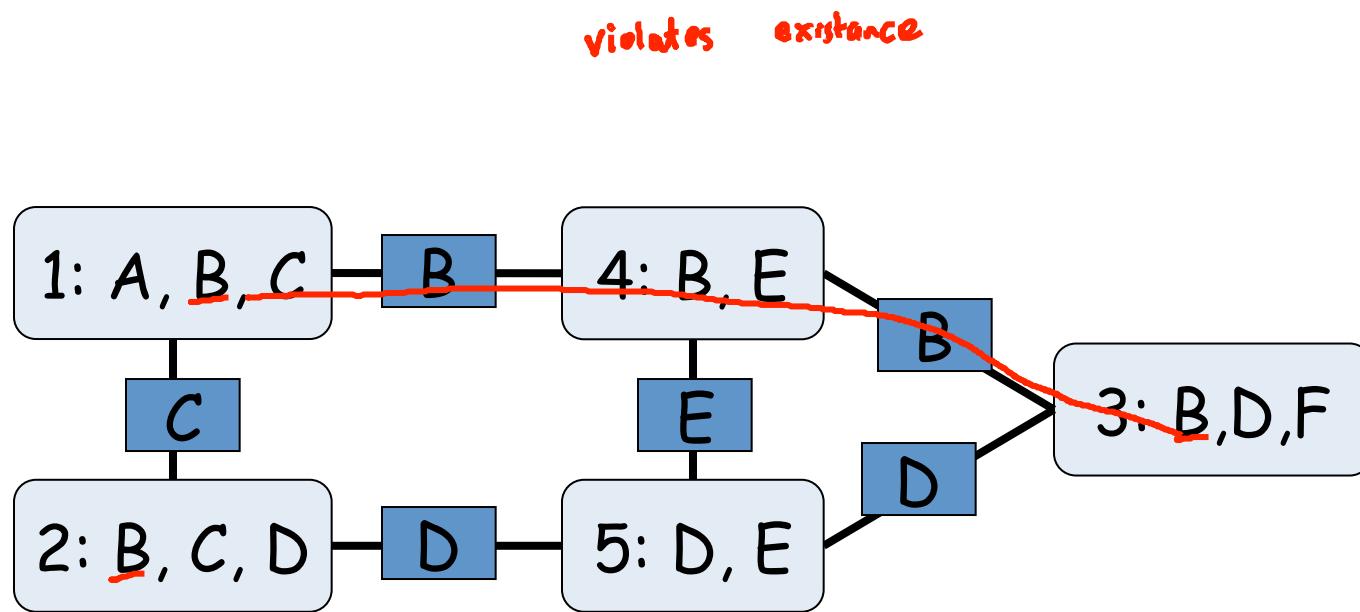
- Equivalently: For any X , the set of clusters and sepsets containing X forms a tree



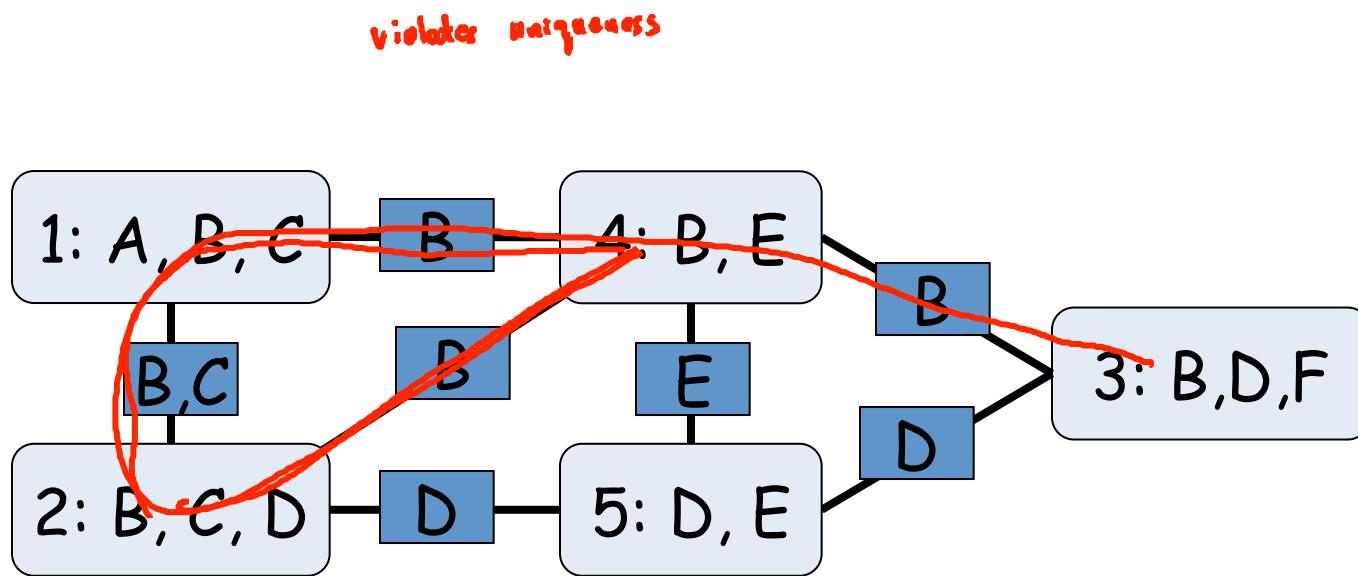
Example Cluster Graph



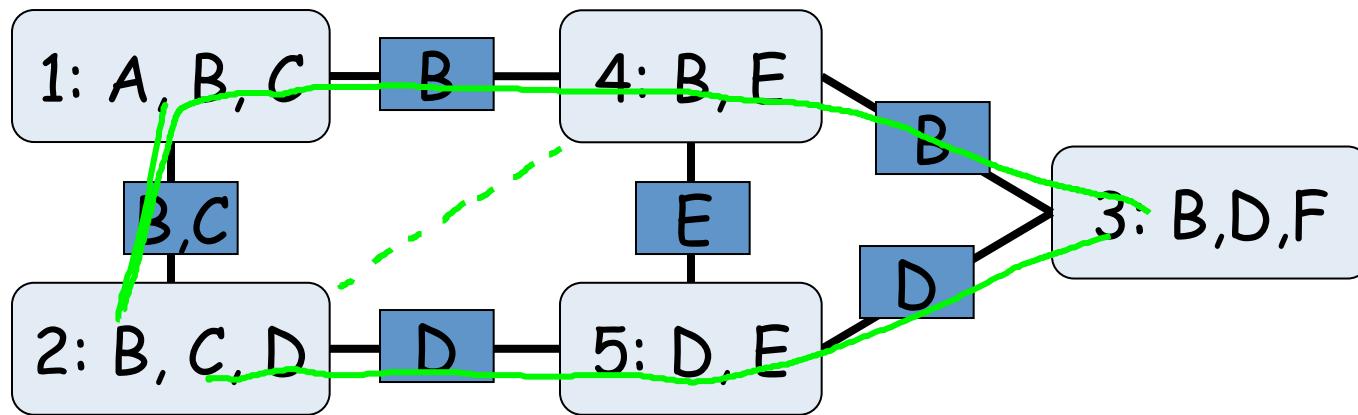
Illegal Cluster Graph I



Illegal Cluster Graph II



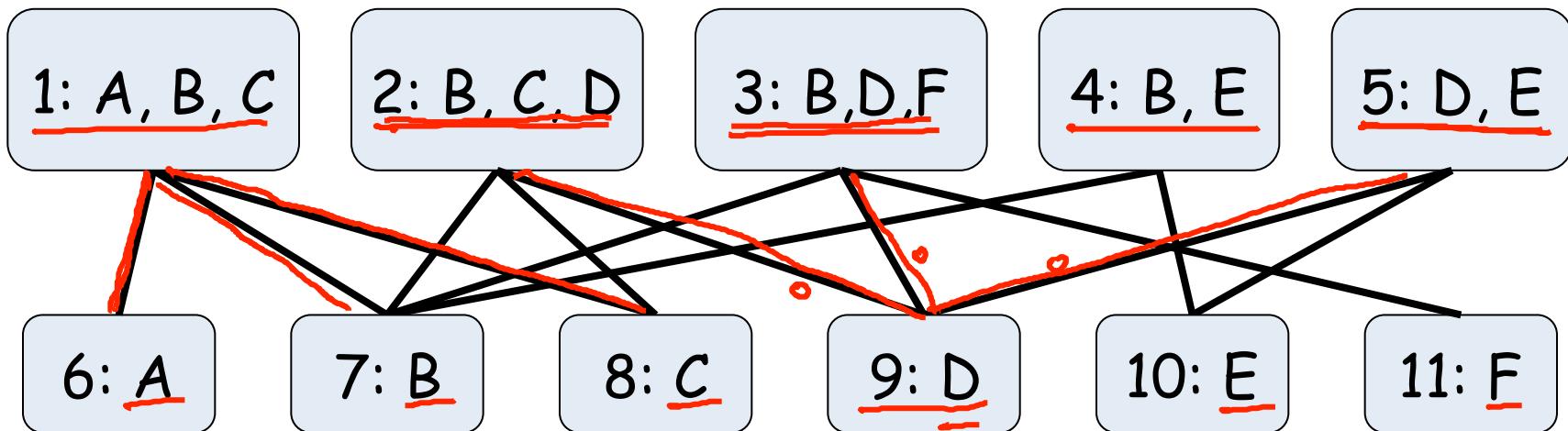
Alternative Legal Cluster Graph



Bethe Cluster Graph

big clusters = factor in Φ

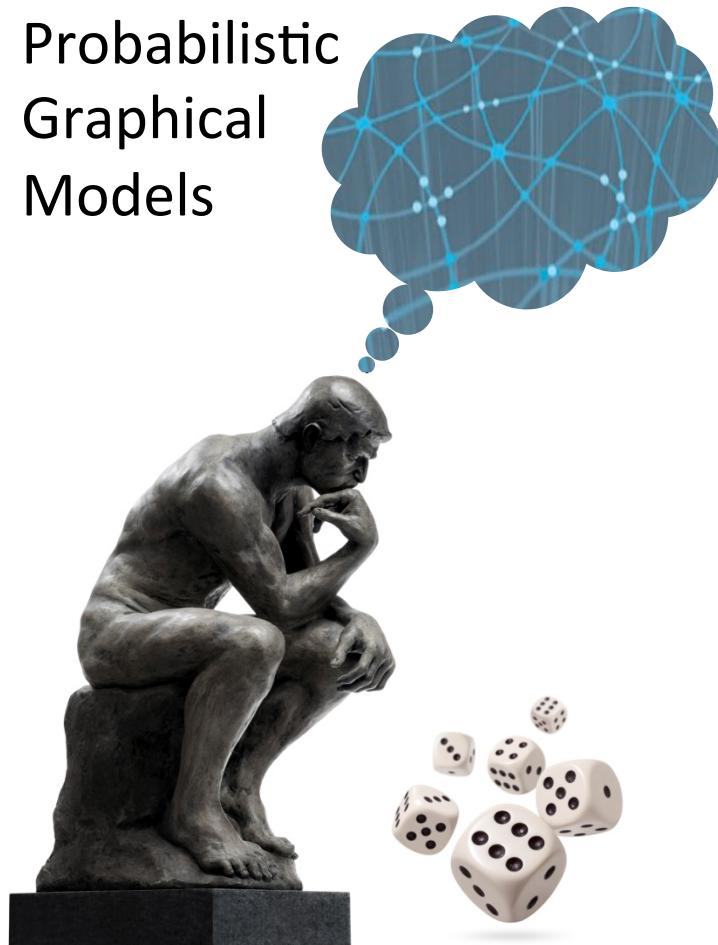
- For each $\phi_k \in \Phi$, a factor cluster $C_k = \text{Scope}[\phi_k]$
- For each X_i a singleton cluster $\{X_i\}$
- Edge $C_k — X_i$ if $X_i \in C_k$



Summary

- Cluster graph must satisfy two properties
 - family preservation: allows Φ to be encoded
 - running intersection: connects all information about any variable, but without feedback loops
- Bethe cluster graph is often first default
- Richer cluster graph structures can offer different tradeoffs wrt computational cost and preservation of dependencies

Probabilistic
Graphical
Models



Inference

Message Passing

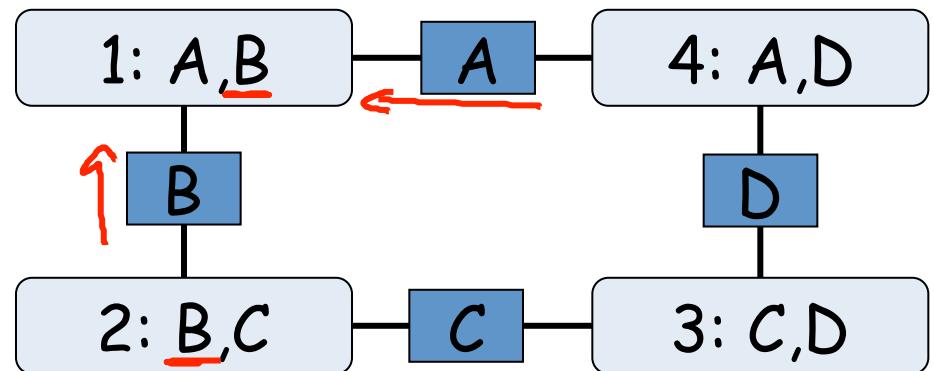
Properties of BP Algorithm

Calibration

$$\beta_1(A, B) = \underline{\psi_1(A, B)} \times \delta_{4 \rightarrow 1}(A) \times \delta_{2 \rightarrow 1}(B)$$

- Cluster beliefs:

$$\underline{\beta_i(C_i)} = \underline{\psi_i} \times \prod_{k \in \mathcal{N}_i} \underline{\delta_{k \rightarrow i}}$$



- A cluster graph is calibrated if every pair of adjacent clusters C_i, C_j agree on their sepset $S_{i,j}$

$$\sum_{C_i - S_{i,j}} \underline{\beta_i(C_i)} = \sum_{C_j - S_{i,j}} \underline{\beta_j(C_j)}$$

sepset $S_{i,j}$

Convergence \Rightarrow Calibration

- Convergence:

$$\delta_{i \rightarrow j}(S_{i,j}) = \delta'_{i \rightarrow j}(S_{i,j})$$

$$\beta_i(C_i) = \psi_i \times \prod_{k \in \mathcal{N}_i} \delta_{k \rightarrow i}$$

$$\delta'_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \left(\psi_i \times \prod_{k \in (\mathcal{N}_i - \{j\})} \delta_{k \rightarrow i} \right) = \sum_{C_i - S_{i,j}} \frac{\beta_i(C_i)}{\delta_{j \rightarrow i}(S_{i,j})} =$$

all msgs

$$\delta_{j \rightarrow i}(S_{i,j}) \delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \underline{\beta_i(C_i)}$$

calibration

$$\delta_{j \rightarrow i}(S_{i,j}) \delta_{i \rightarrow j}(S_{i,j}) = \sum_{C_j - S_{i,j}} \underline{\beta_j(C_j)}$$

subset beliefs

$$\mu_{i,j}(S_{i,j}) = \delta_{j \rightarrow i} \delta_{i \rightarrow j} = \sum_{C_j - S_{i,j}} \beta_j(C_j)$$

Daphne Koller

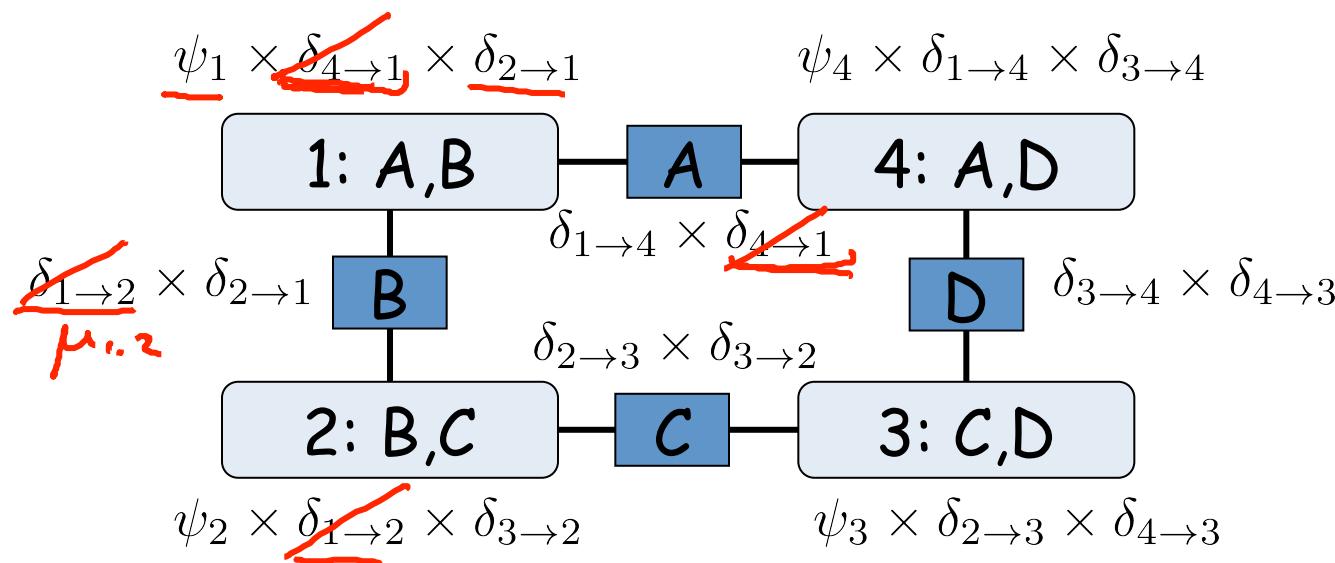
Reparameterization

$$\beta_i(C_i) = \psi_i \times \prod_{k \in \mathcal{N}_i} \delta_{k \rightarrow i}$$

$$\mu_{i,j}(S_{i,j}) = \delta_{j \rightarrow i} \delta_{i \rightarrow j}$$

separat beliefs

$$\frac{\prod_i \beta_i}{\prod_{i,j} \mu_{i,j}}$$



Reparameterization

no information loss

$$\beta_i(C_i) = \psi_i \times \prod_{k \in \mathcal{N}_i} \delta_{k \rightarrow i} \quad \mu_{i,j}(S_{i,j}) = \delta_{j \rightarrow i} \delta_{i \rightarrow j}$$

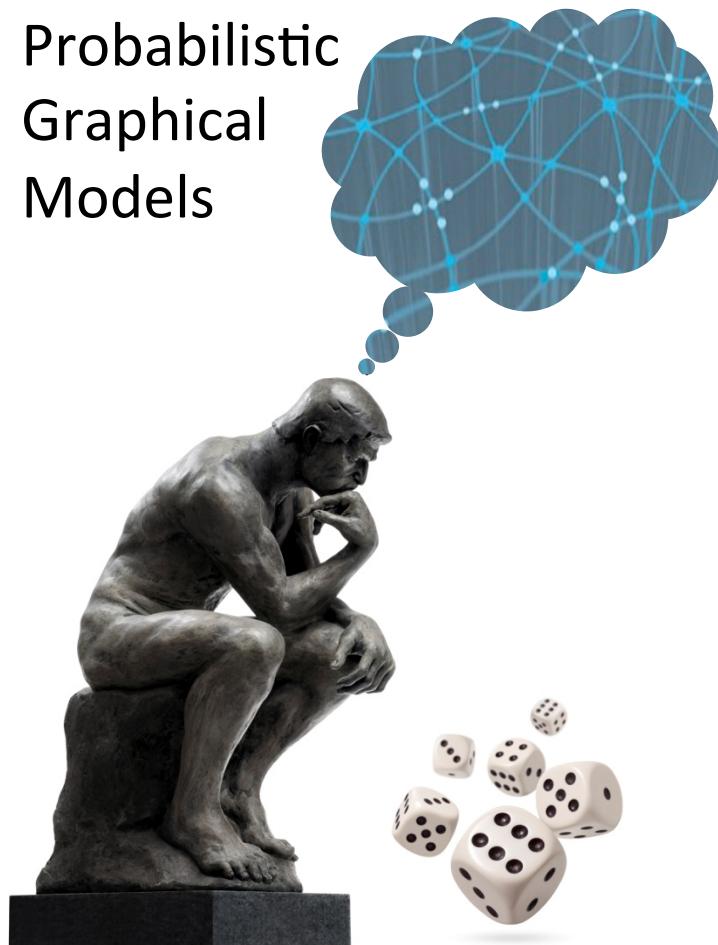
$$\begin{aligned} \frac{\prod_i \beta_i}{\prod_{i,j} \mu_{i,j}} &= \frac{\prod_i (\psi_i \prod_{j \in \mathcal{N}_i} \delta_{j \rightarrow i})}{\prod_{i,j} \delta_{i \rightarrow j}} \\ &= \prod_i \psi_i = \tilde{P}_{\Phi}(X_1, \dots, X_n) \end{aligned}$$

unnormalized measure

Summary

- At convergence of BP, cluster graph beliefs are calibrated:
 - beliefs at adjacent clusters agree on sepsets
- Cluster graph beliefs are an alternative, calibrated parameterization of the original unnormalized density
 - No information is lost by message passing

Probabilistic
Graphical
Models

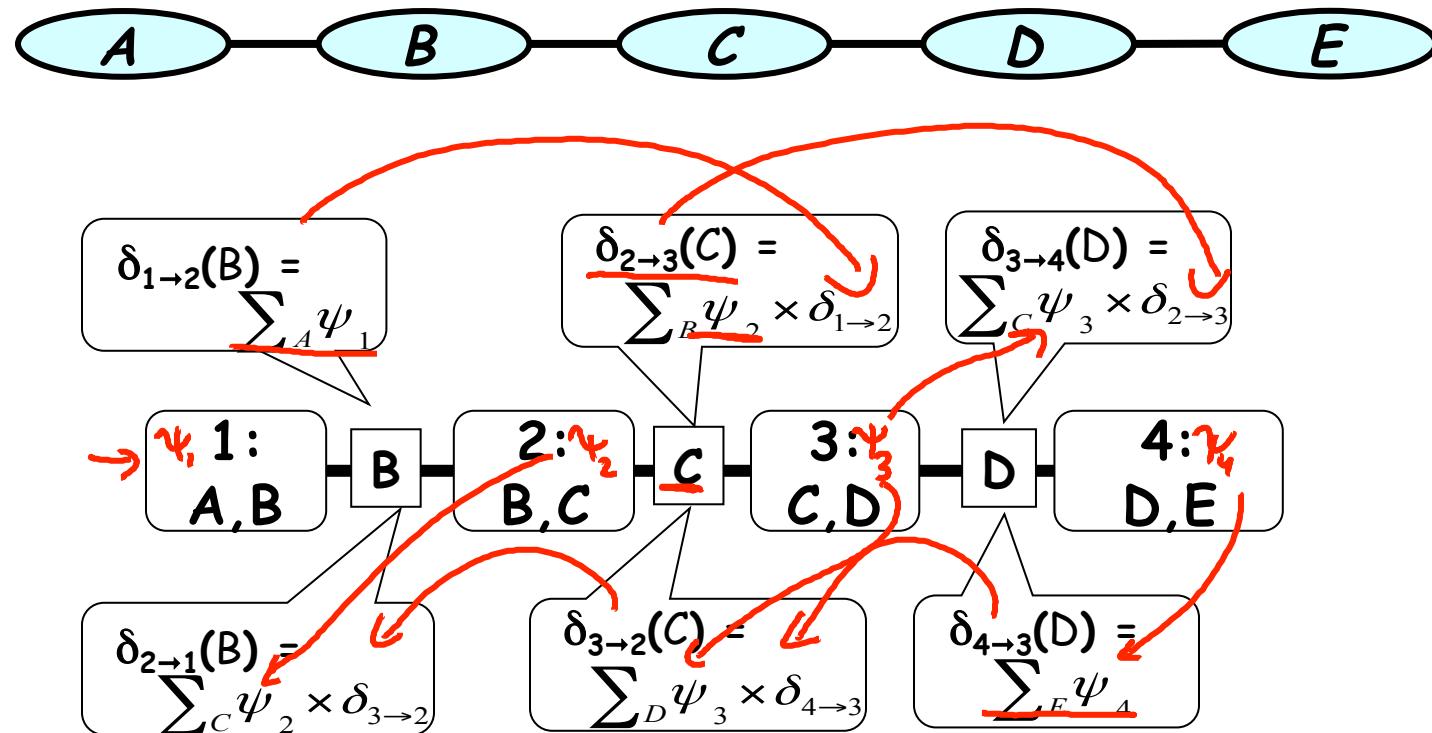


Inference

Message Passing

Clique Tree Algorithm & Correctness

Message Passing in Trees



Correctness

$$\begin{aligned}
 \delta_{1 \rightarrow 2}(B) &= \sum_A \psi_1 \\
 \delta_{2 \rightarrow 3}(C) &= \sum_B \psi_2 \times \delta_{1 \rightarrow 2} \\
 1: A, B &\quad B \quad 2: B, C \quad C \quad 3: C, D \quad D \quad 4: D, E \\
 \beta_3(C, D) &= \psi_3 \times \delta_{2 \rightarrow 3} \times \delta_{4 \rightarrow 3} \\
 &= \psi_3 \times \left(\sum_B (\psi_2 \times \delta_{1 \rightarrow 2}) \right) \times \sum_E \psi_4 \\
 &= \psi_3 \times \left(\sum_B \psi_2 \times \left(\sum_A \psi_1 \right) \right) \times \sum_E \psi_4
 \end{aligned}$$

legal order of operations
 product of factors
 marginalized out unnecessary variables

Daphne Koller

Clique Tree

- Undirected tree such that:
 - nodes are clusters $C_i \subseteq \{X_1, \dots, X_n\}$
 - edge between C_i and C_j associated with sepset $S_{i,j} = C_i \cap C_j$

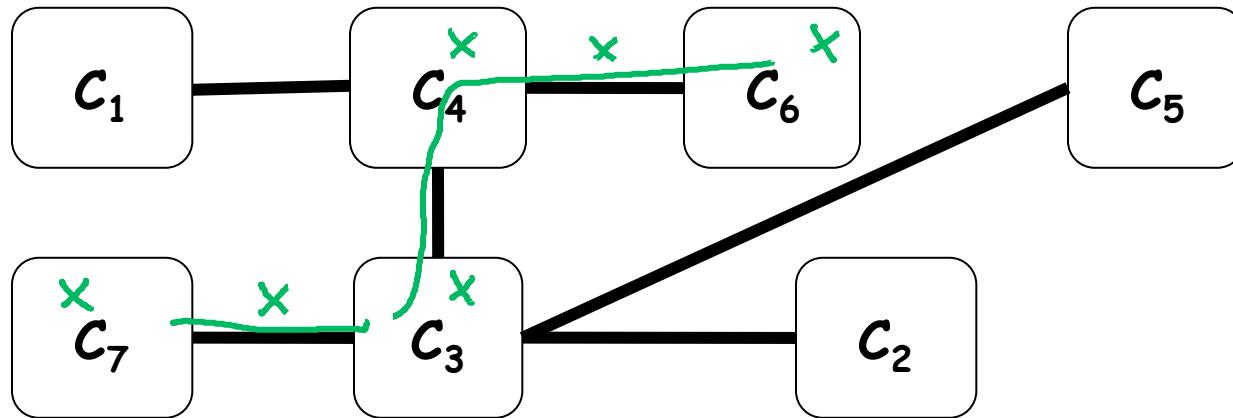
Family Preservation

- Given set of factors Φ , we assign each $\phi_k \in \Phi$ to a cluster $C_{\alpha(k)}$ s.t. $\text{Scope}[\phi_k] \subseteq C_{\alpha(k)}$
- For each factor $\phi_k \in \Phi$, there exists a cluster C_i s.t. $\text{Scope}[\phi_k] \subseteq C_i$

Running Intersection Property

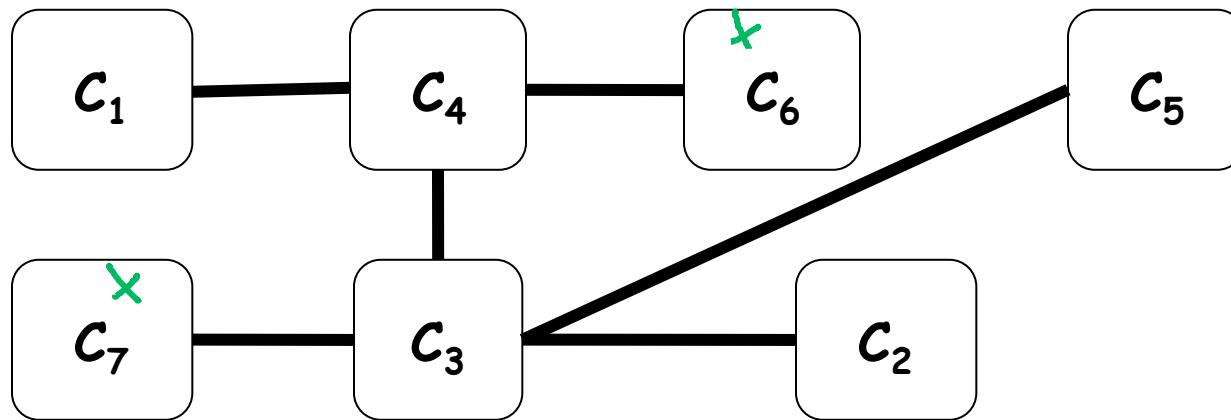
Cluster graph variant

- For each pair of clusters C_i, C_j and variable $X \in C_i \cap C_j$ there exists a unique path between C_i and C_j for which all clusters and sepsets contain X

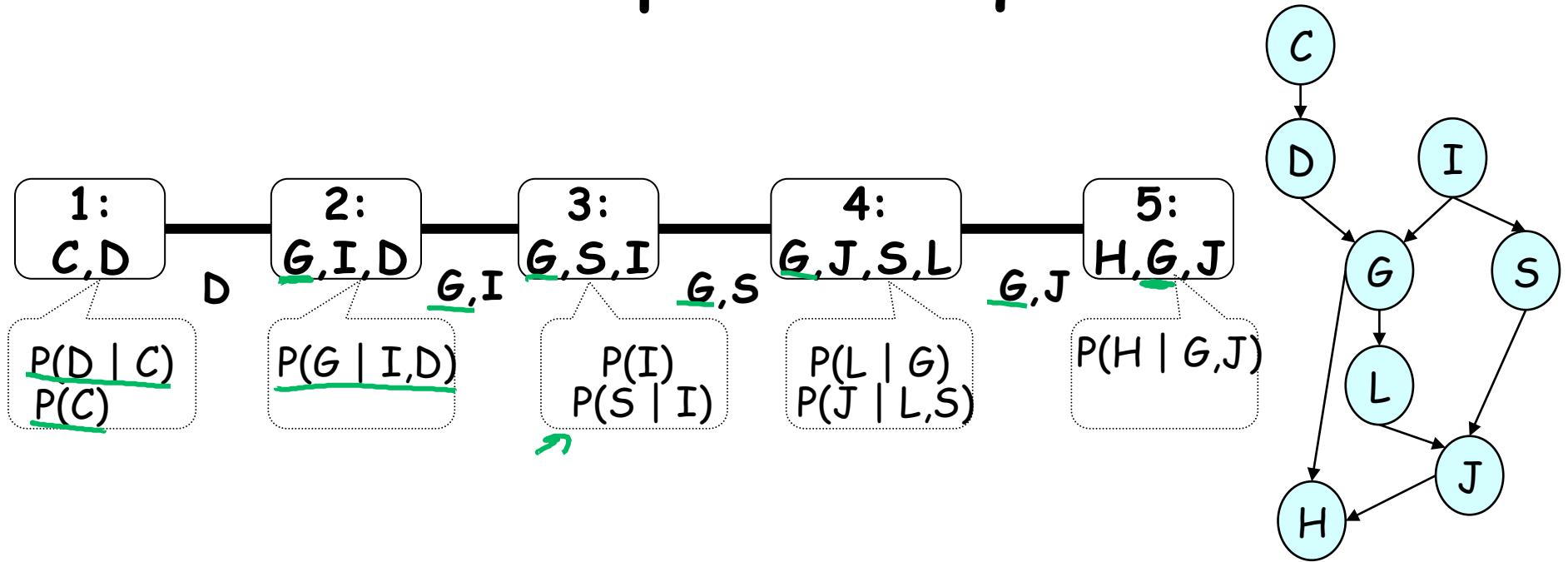


Running Intersection Property

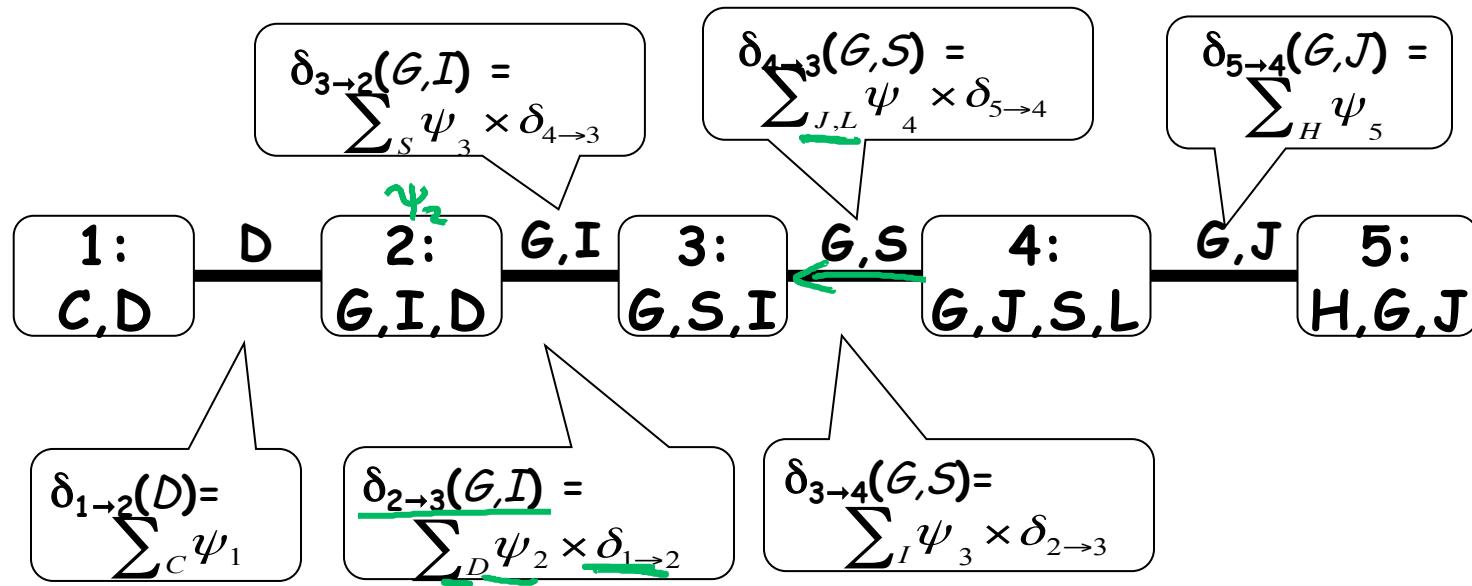
- For each pair of clusters C_i, C_j and variable $X \in C_i \cap C_j$, in the unique path between C_i and C_j , all clusters and sepsets contain X



More Complex Clique Tree

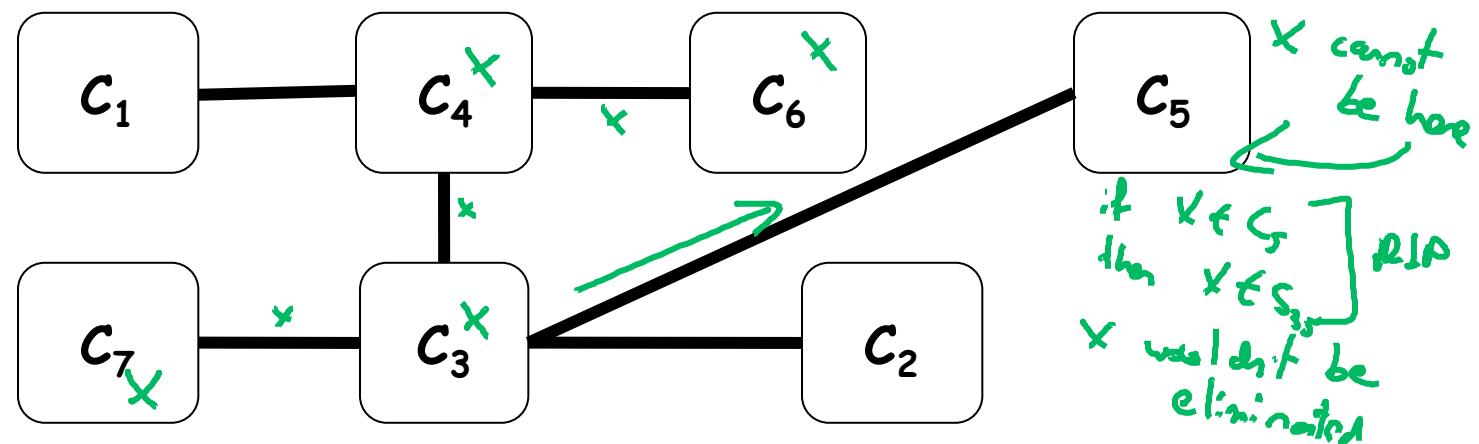


Clique Tree Message Passing

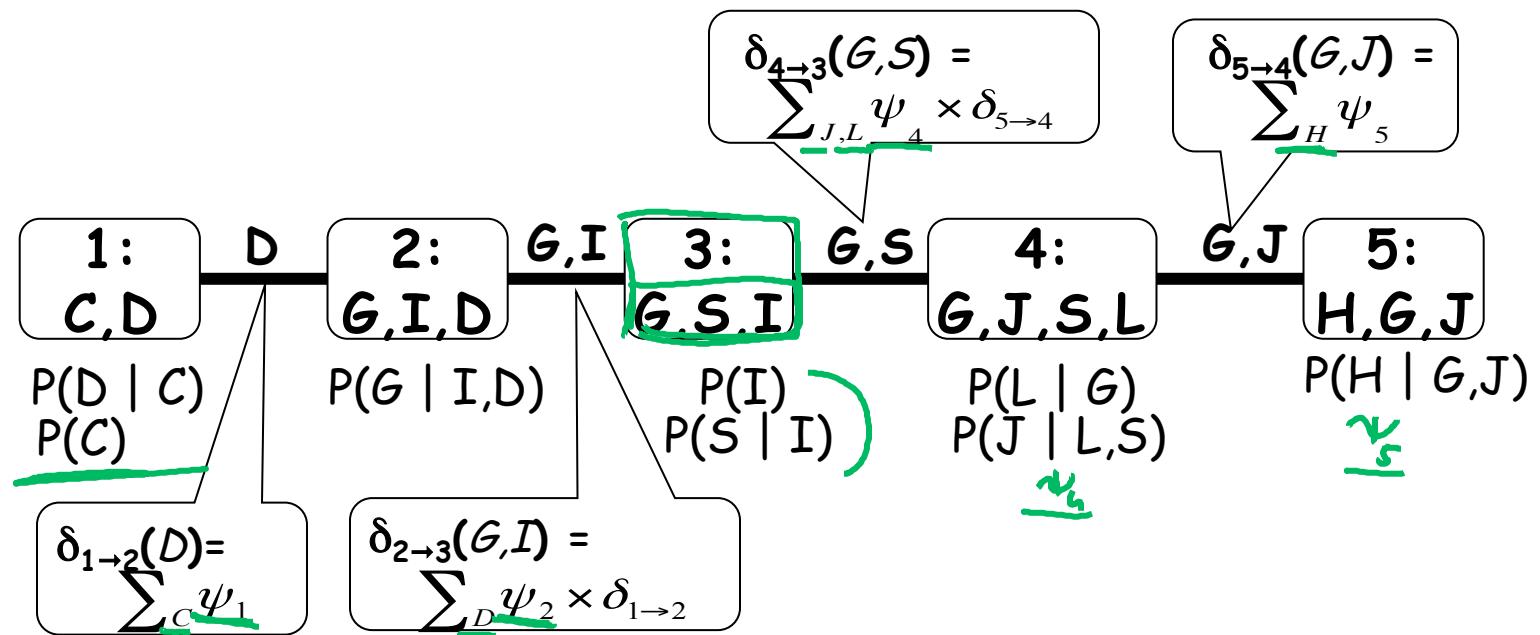


RIP \Rightarrow Clique Tree Correctness

- If X is eliminated when we pass the message $C_i \rightarrow C_j$
- Then X does not appear in the C_j side of the tree



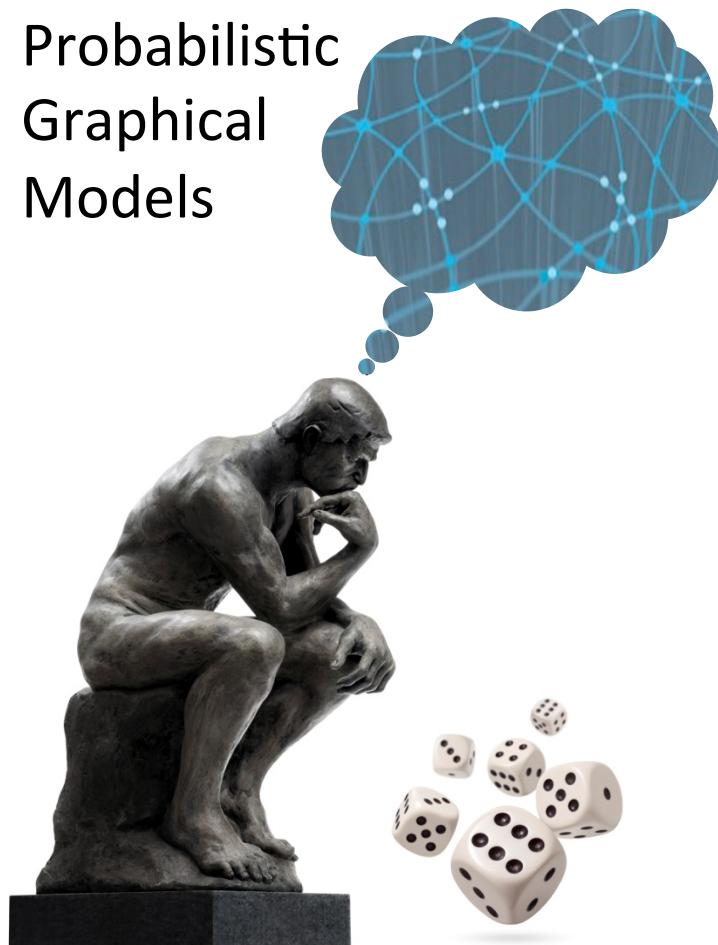
Clique Tree Correctness



Summary

- Belief propagation can be run over a tree-structured cluster graph
- In this case, computation is a variant of variable elimination
- Resulting beliefs are guaranteed to be correct marginals $\pi_1 \dots \pi_5$

Probabilistic
Graphical
Models



Inference

Message Passing

Clique Tree Algorithm: Computation

Message Passing in Trees



*once computed
never changes*

$$\delta_{1 \rightarrow 2}(B) = \sum_A \psi_1$$

1:
A, B

wait for $\delta_{1 \rightarrow 2}$

$$\delta_{2 \rightarrow 3}(C) = \sum_B \psi_2 \times \delta_{1 \rightarrow 2}$$

2:
B, C

wait for $\delta_{2 \rightarrow 3}$

$$\delta_{3 \rightarrow 4}(D) = \sum_C \psi_3 \times \delta_{2 \rightarrow 3}$$

3:
C, D

*Beliefs that
are $\pi_D(C_i)$*

$$\delta_{4 \rightarrow 3}(D) = \sum_E \psi_4$$

4:
D, E

$$\delta_{2 \rightarrow 1}(B) = \sum_C \psi_2 \times \delta_{3 \rightarrow 2}$$

wait for $\delta_{3 \rightarrow 2}$

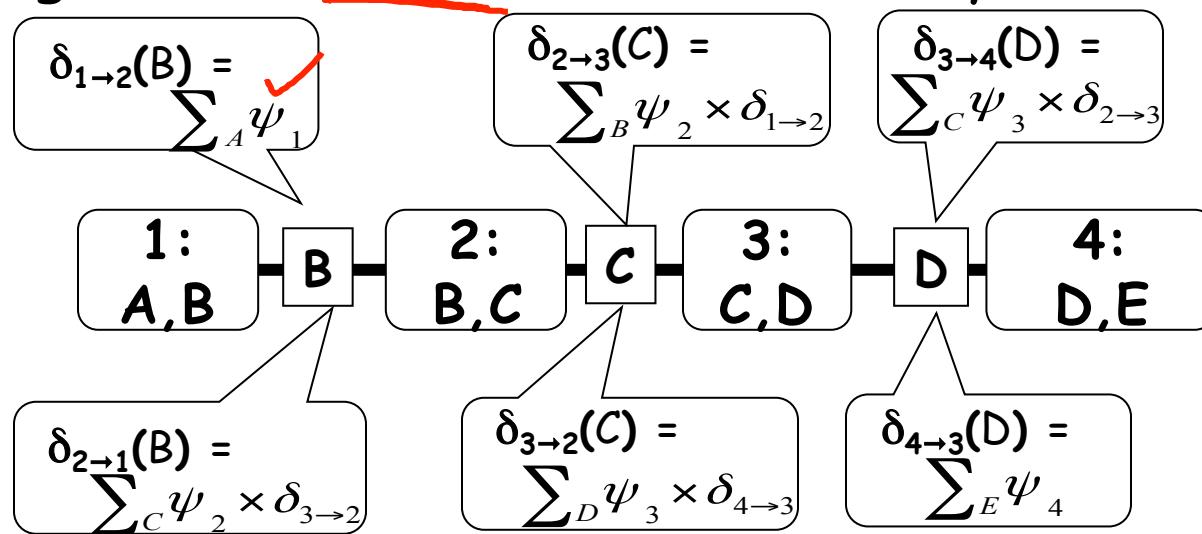
$$\delta_{3 \rightarrow 2}(C) = \sum_D \psi_3 \times \delta_{4 \rightarrow 3}$$

wait for $\delta_{4 \rightarrow 3}$

*converges
instantly*

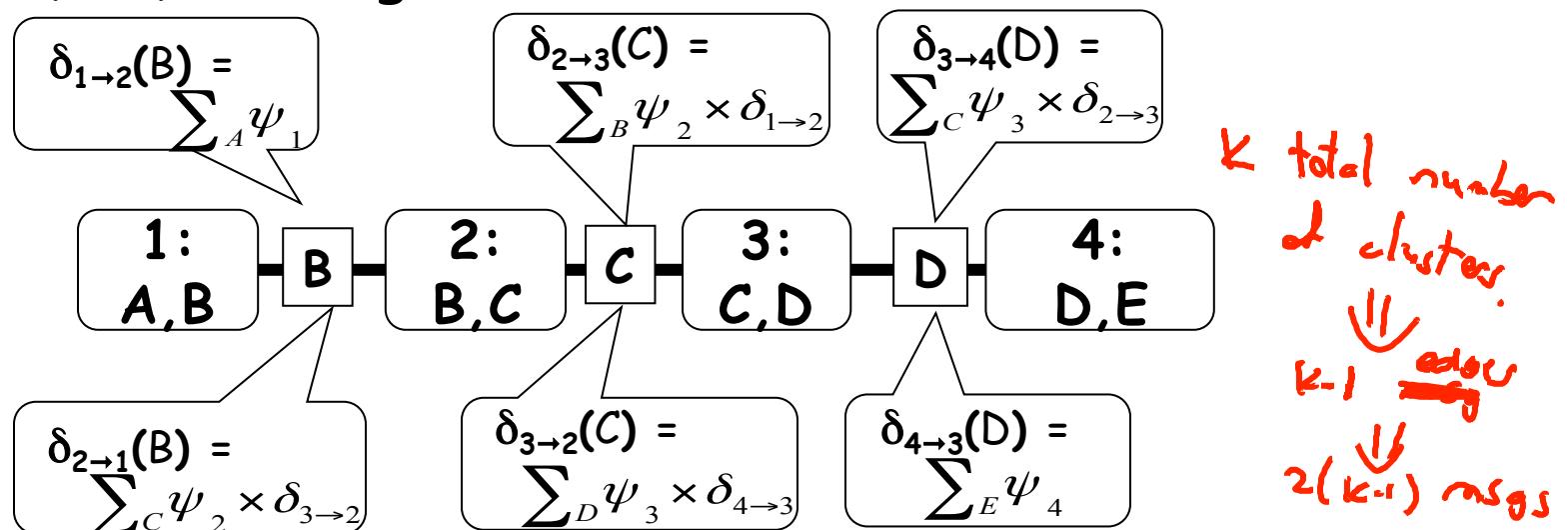
Convergence of Message Passing

- Once C_i receives a final message from all neighbors except C_j , then $\delta_{i \rightarrow j}$ is also final (will never change)
- Messages from leaves are immediately final

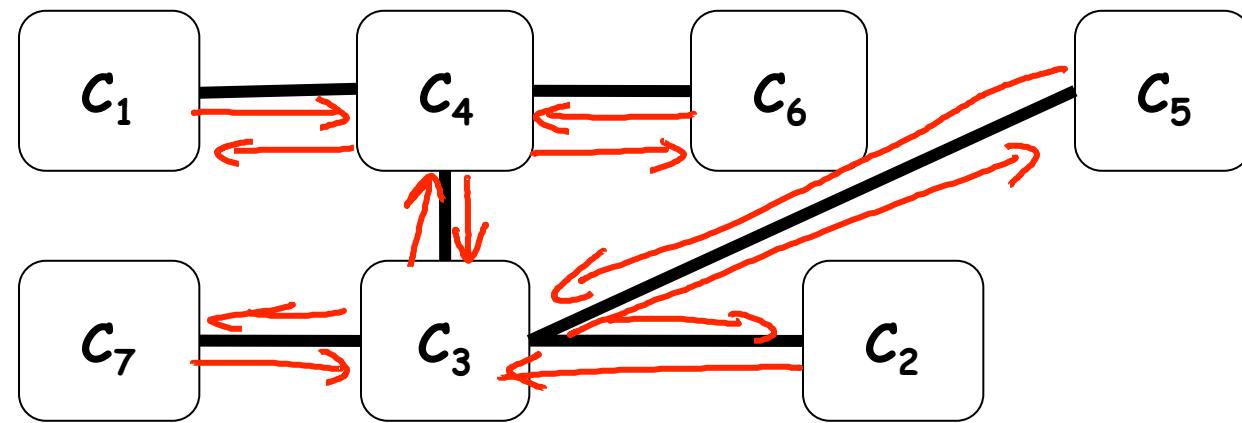


Convergence of Message Passing

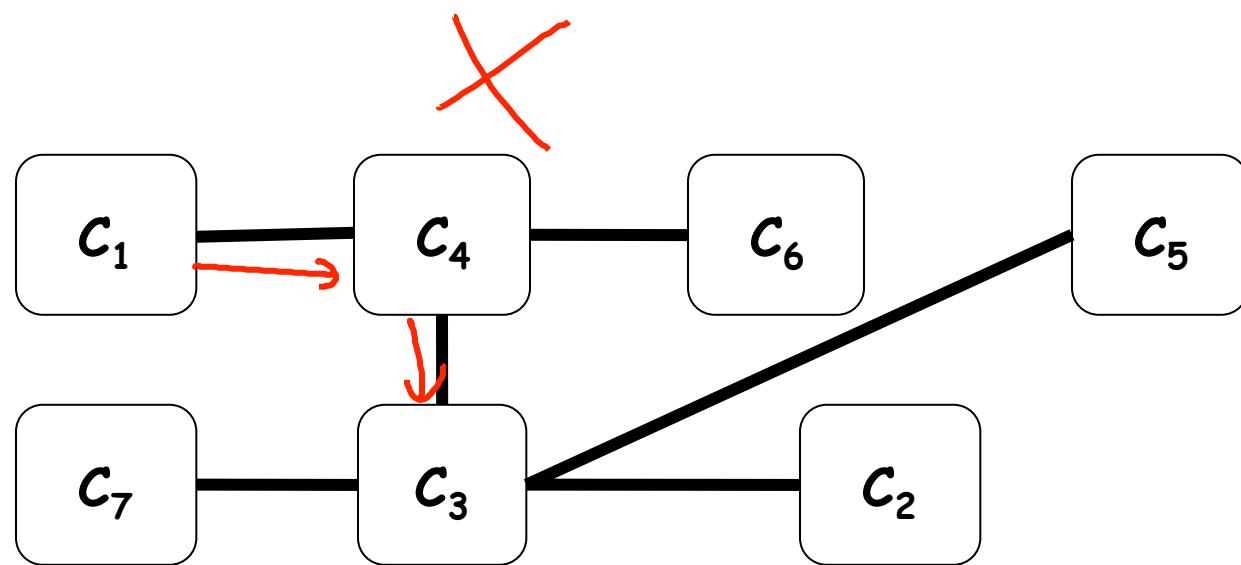
- Can pass messages from leaves inward
- If messages are passed in the right order, only need to pass $2(K-1)$ messages



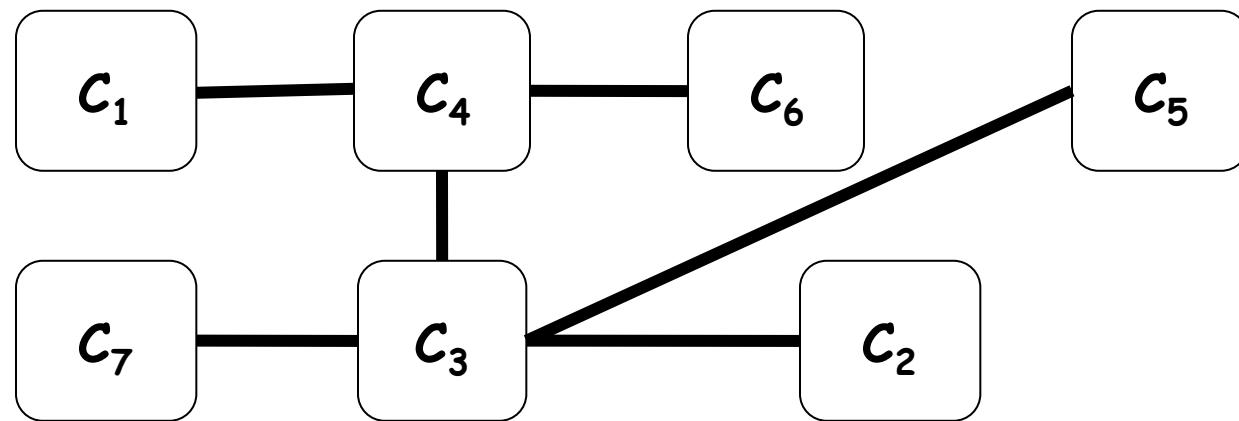
Message Passing Order I



Message Passing Order II



Message Passing Order III

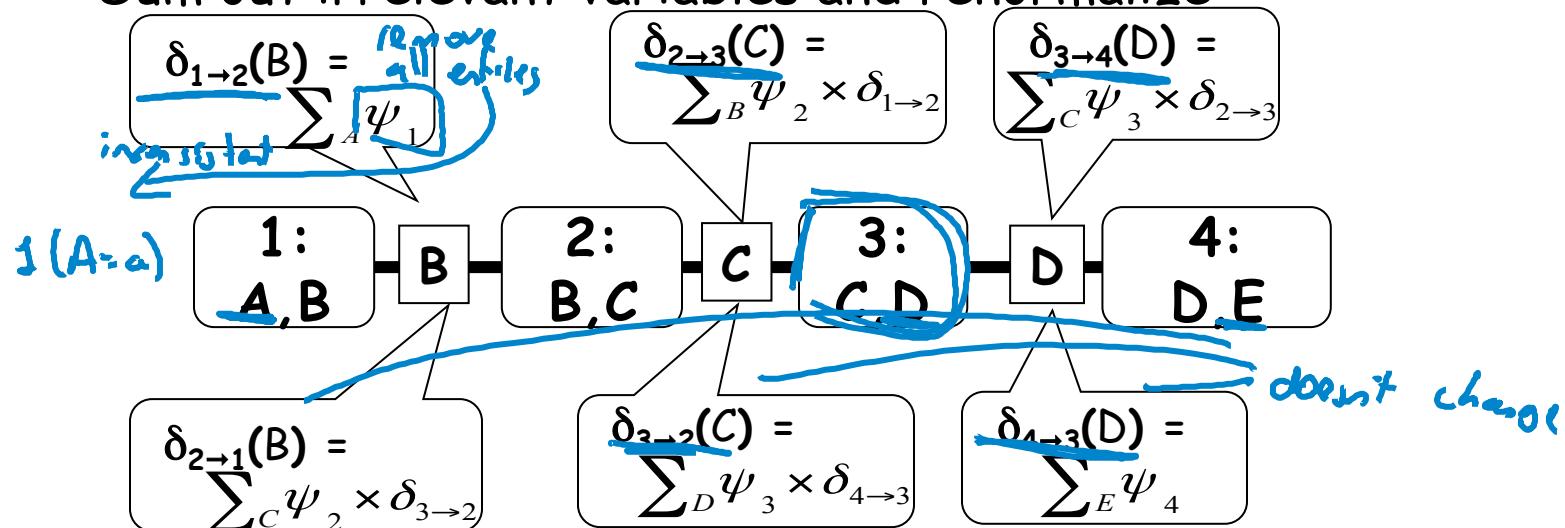


Answering Queries

- Posterior distribution queries on variables that appear together in clique
 - Sum out irrelevant variables from any clique containing those variables p_{ϕ} renormalize
- Introducing new evidence $Z=z$ and querying X
 - If X appears in clique with Z incremental inference
 - Multiply clique that contains X and Z with indicator function $1(Z=z)$ reduce clique $p_{\phi}(z, X)$
 - Sum out irrelevant variables and renormalize

And More Queries

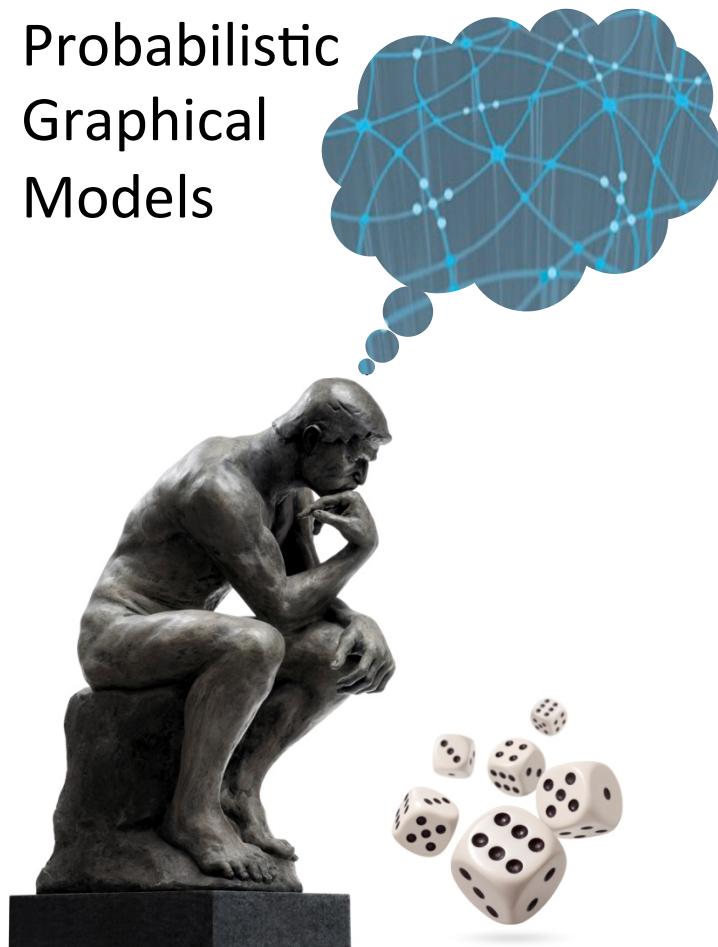
- Introducing new evidence $Z=z$ and querying X if X does not share a clique with Z
 - Multiply $\mathbf{1}(Z=z)$ into some clique containing Z *reduction of factor*
 - Propagate messages along path to clique containing X
 - Sum out irrelevant variables and renormalize



Summary

- In clique tree with K cliques, if messages are passed starting at leaves, $2(K-1)$ messages suffice to compute all beliefs
- Can compute marginals over all variables at only twice the cost of variable elimination
- By storing messages, inference can be reused in incremental queries

Probabilistic
Graphical
Models



Inference

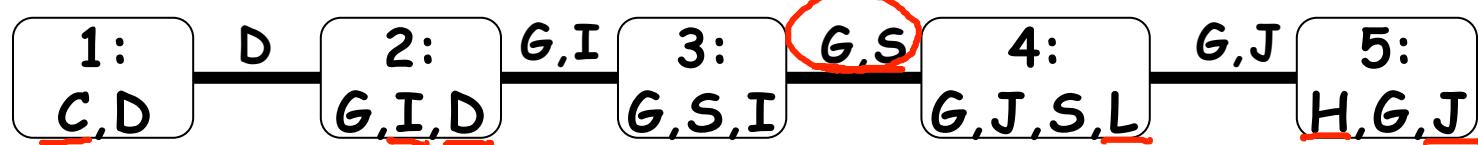
Message Passing

Clique Tree &
Independence

RIP and Independence

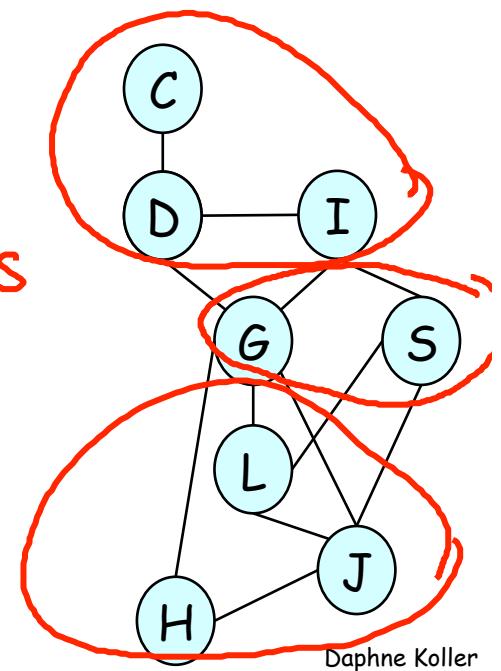
- For an edge (i,j) in T , let:
 - $W_{<(i,j)}$ = all variables that appear only on C_i side of T
 - $W_{<(j,i)}$ = all variables that appear only on C_j side
 - Variables on both sides are in the sepset $S_{i,j}$
- Theorem: T satisfies RIP if and only if, for every $\underline{(i,j)}$ $P_\Phi \models (W_{<(i,j)} \perp W_{<(j,i)} \mid S_{i,j})$

RIP and Independence



$$P_\Phi \models (\{C, I, D\} \perp \{J, L, H\} \mid \{G, S\})$$

C, I, D separated from H, L, J given G, S

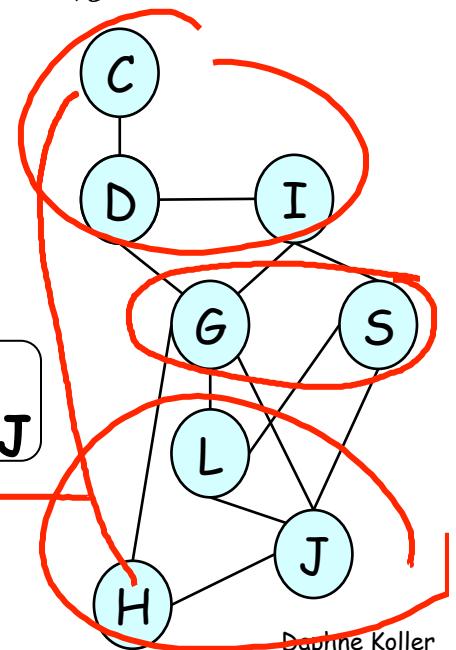


RIP and Independence

- Theorem: T satisfies RIP if and only if, for every edge (i,j) $P_\Phi \models (W_{<(i,j)} \perp W_{<(j,i)} \mid S_{i,j})$

Assume otherwise $\Rightarrow \exists$ path in induced Markov network between $W_{<(i,j)}$ $W_{<(j,i)}$ that doesn't go through $S_{i,j}$

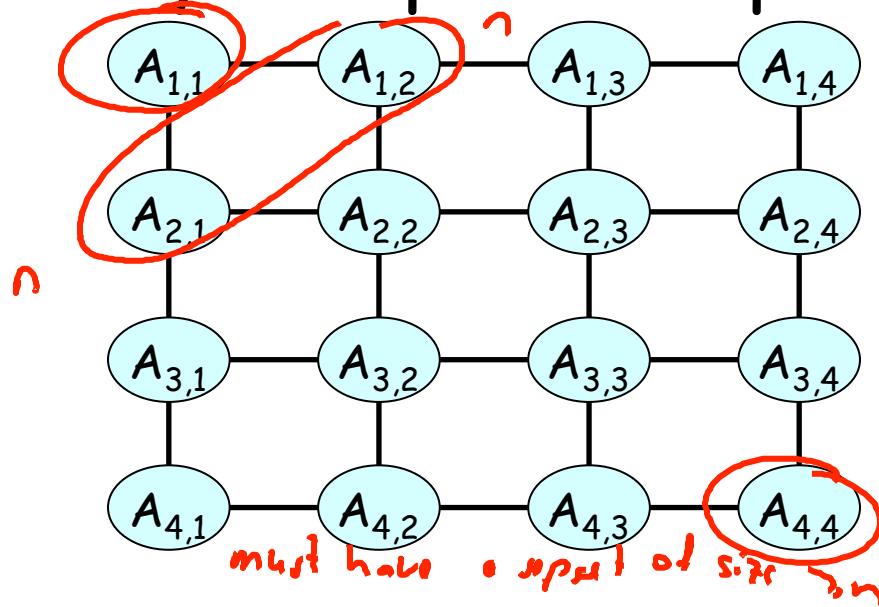
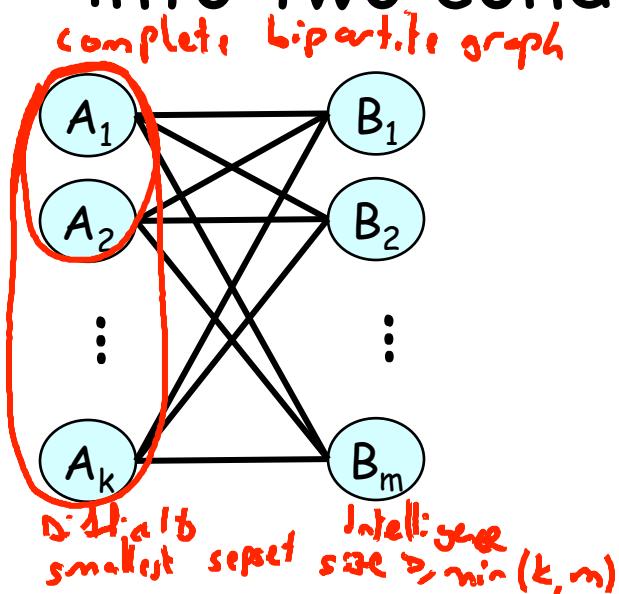
Factor $\phi(c, h)$



Daphne Koller

Implications

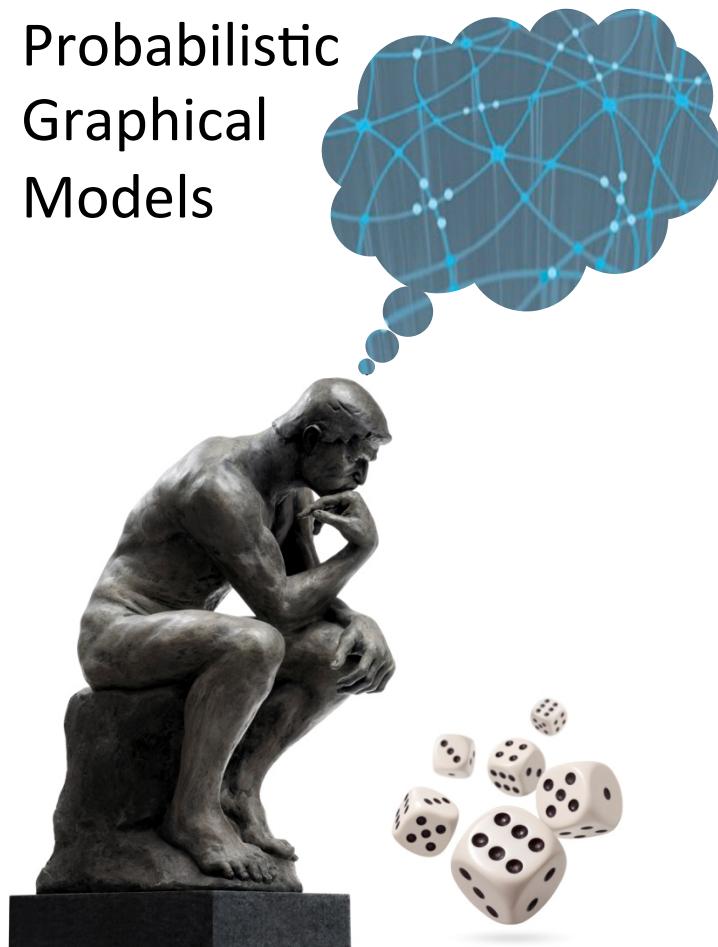
- Each sepset needs to separate graph into two conditionally independent parts



Summary

- Correctness of clique tree inference relies on running intersection property
- Running intersection property implies separation in original distribution
- Implies minimal complexity incurred by any clique tree:
 - Related to minimal induced width of graph

Probabilistic
Graphical
Models



Inference

Message Passing

Clique Tree
and VE

Variable Elimination & Clique Trees

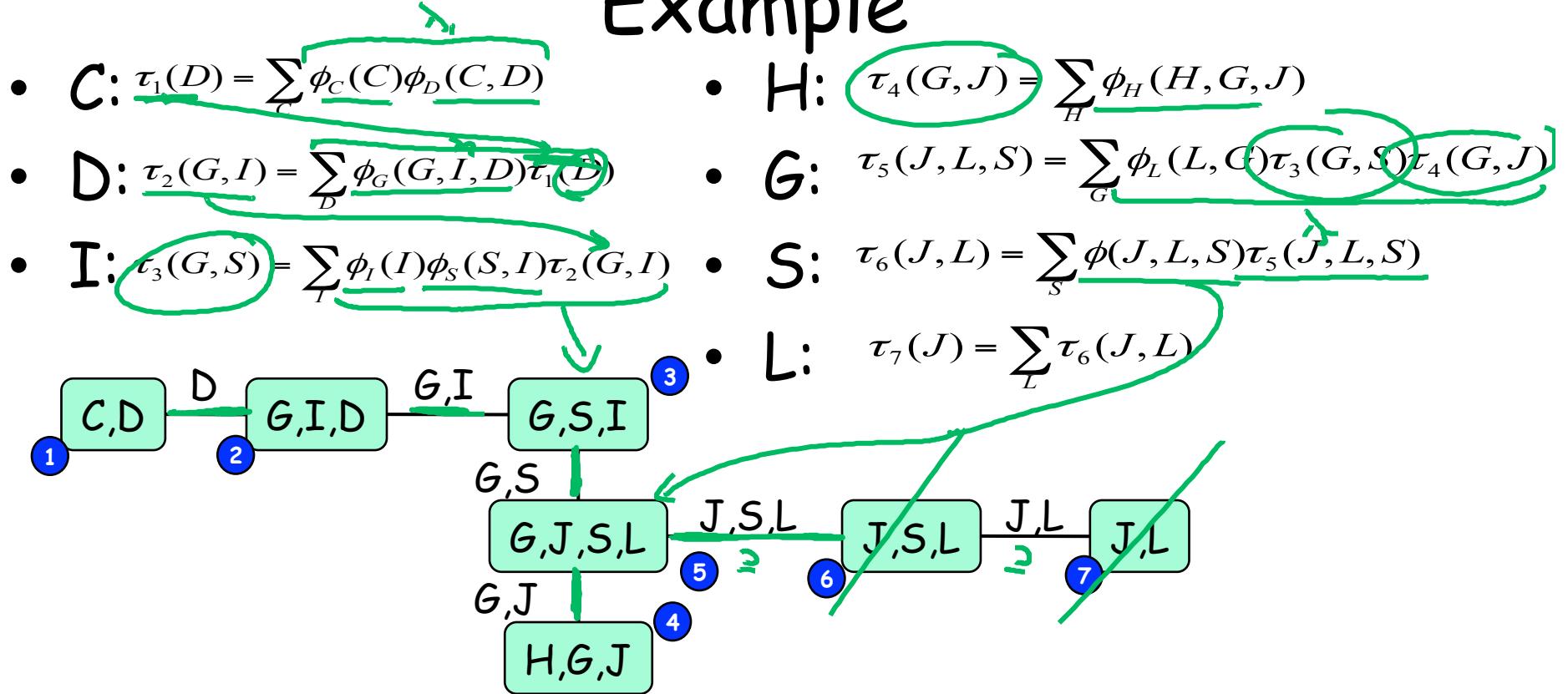
- Variable elimination
 - Each step creates a factor λ_i through factor product
 - A variable is eliminated in λ_i to generate new factor τ_i
 - τ_i is used in computing other factors λ_j
- Clique tree view
 - Intermediate factors λ_i are cliques
 - τ_i are "messages" generated by clique λ_i and transmitted to another clique λ_j

Clique Tree from VE

- VE defines a graph
 - Cluster C_i for each factor λ_i used in the computation
 - Draw edge $C_i - C_j$ if the factor generated from λ_i is used in the computation of λ_j



Example



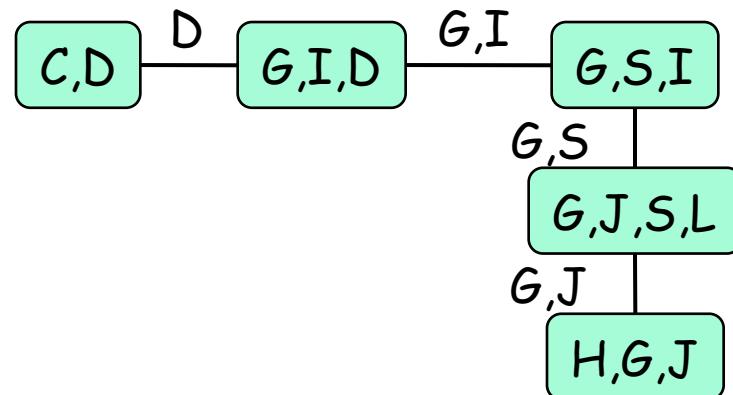
Remove redundant cliques:
those whose scope is a subset of adjacent clique's scope

Properties of Tree

- VE process induces a tree T_i
 - In VE, each intermediate factor is used only once
 - Hence, each cluster “passes” a factor (message) to exactly one other cluster *(every cluster has at most one parent)*
- Tree is family preserving: $\phi \in \Phi$
 - Each of the original factors must be used in some elimination step
 - And therefore contained in scope of associated ψ_i
Scope that contains $\text{Scop}(\phi)$

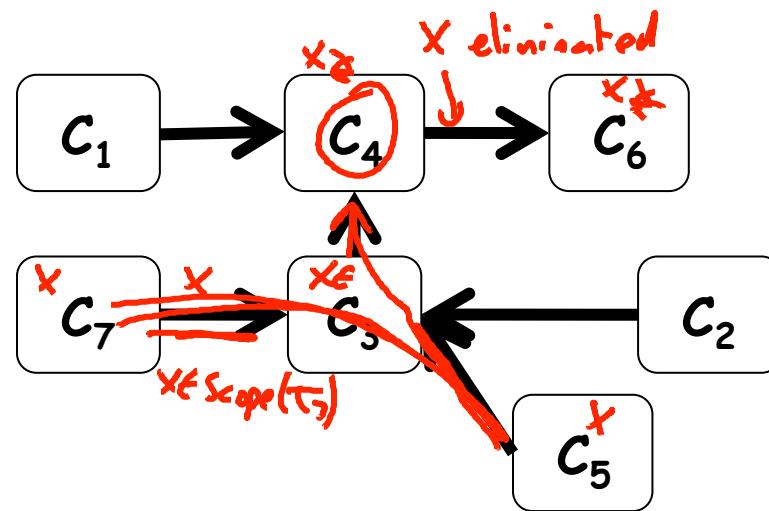
Properties of Tree

- Tree obeys running intersection property
 - If $\underline{X} \in C_i$ and $\underline{X} \in C_j$ then X is in each cluster in the (unique) path between C_i and C_j



Running Intersection Property

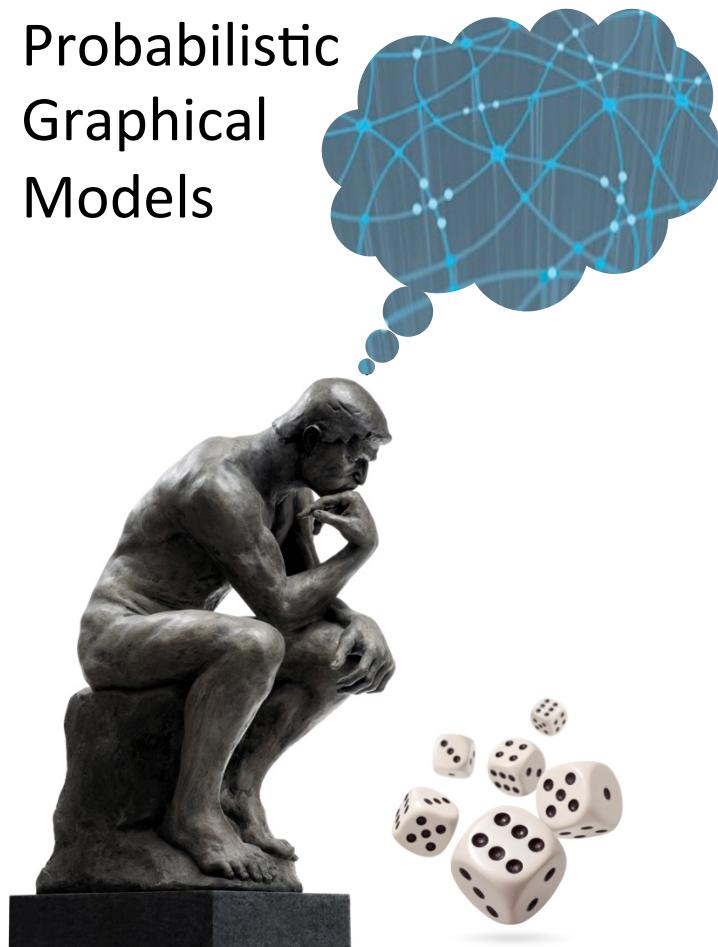
- **Theorem:** If T is a tree of clusters induced by VE, then T obeys RIP



Summary

- A run of variable elimination implicitly defines a correct clique tree
 - We can "simulate" a run of VE to define cliques and connections between them
- Cost of variable elimination is ~ the same as passing messages in one direction in tree
- Clique trees use dynamic programming (storing messages) to compute marginals over all variables at only twice the cost of VE

Probabilistic
Graphical
Models

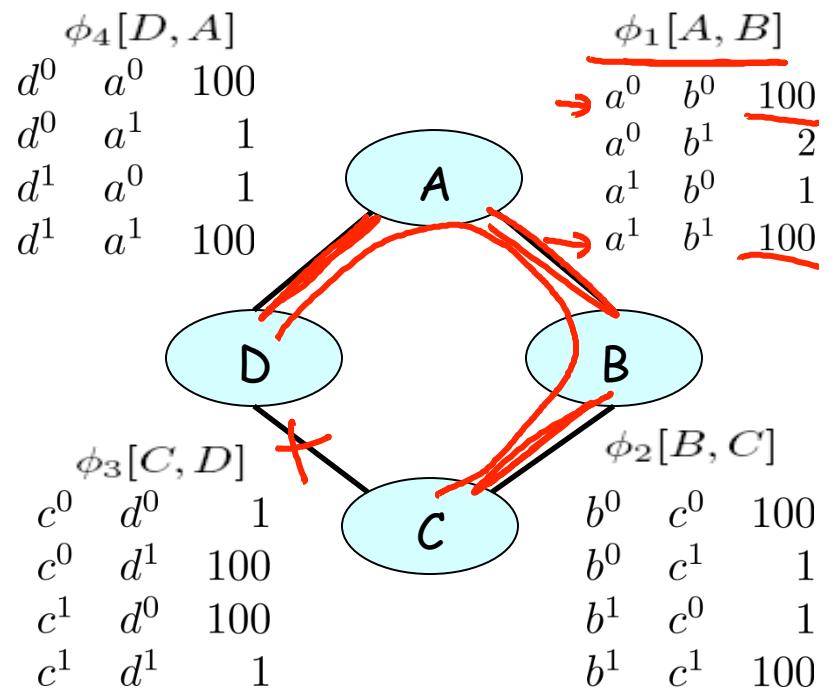
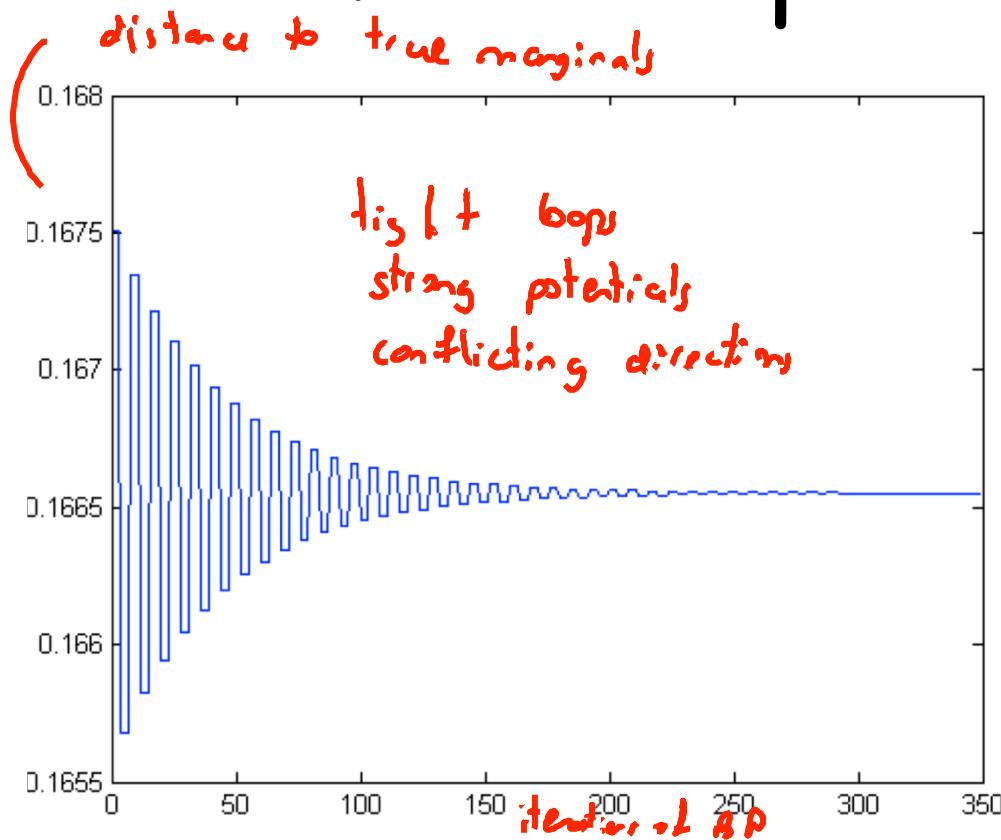


Inference

Message Passing

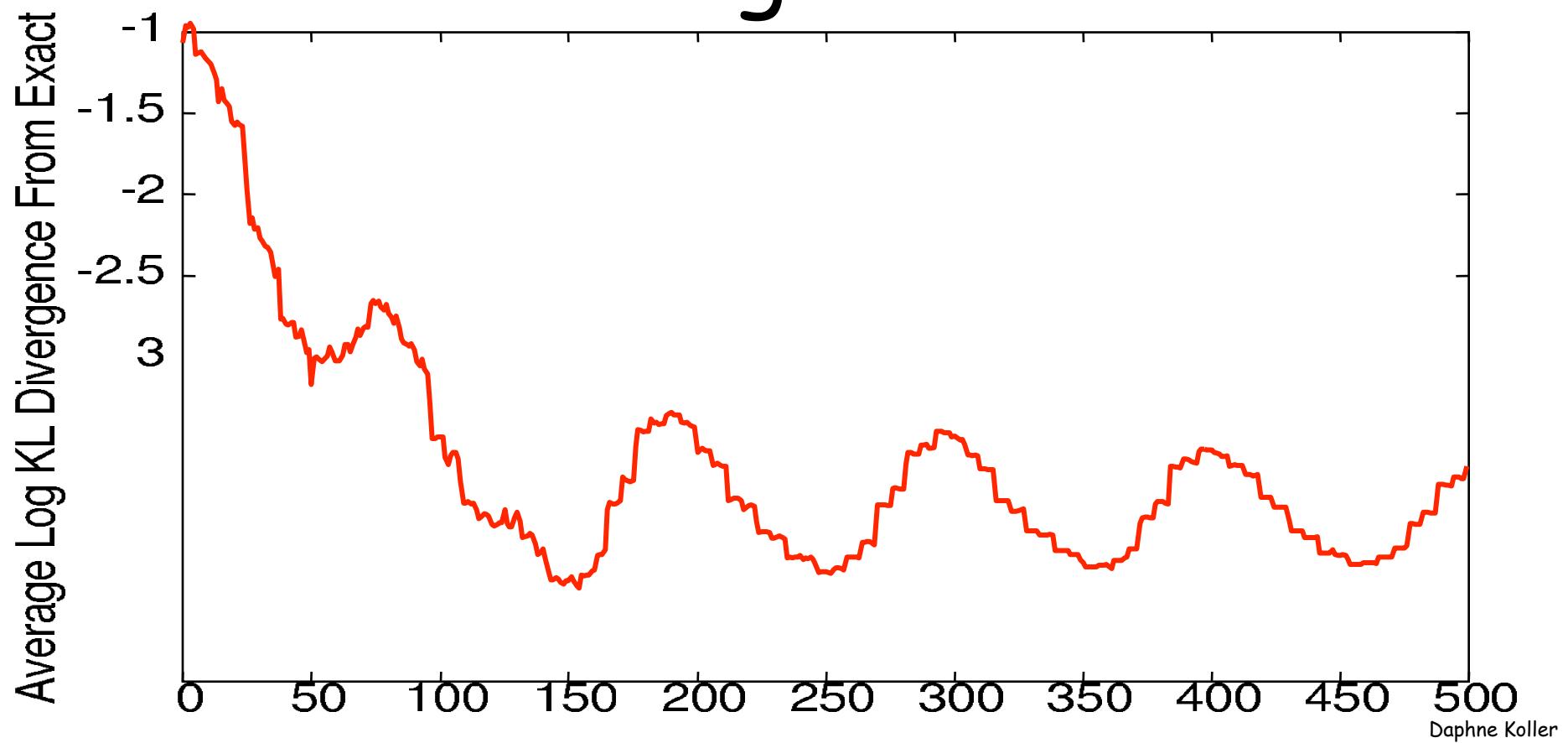
BP In Practice

Misconception Revisited



Daphne Koller

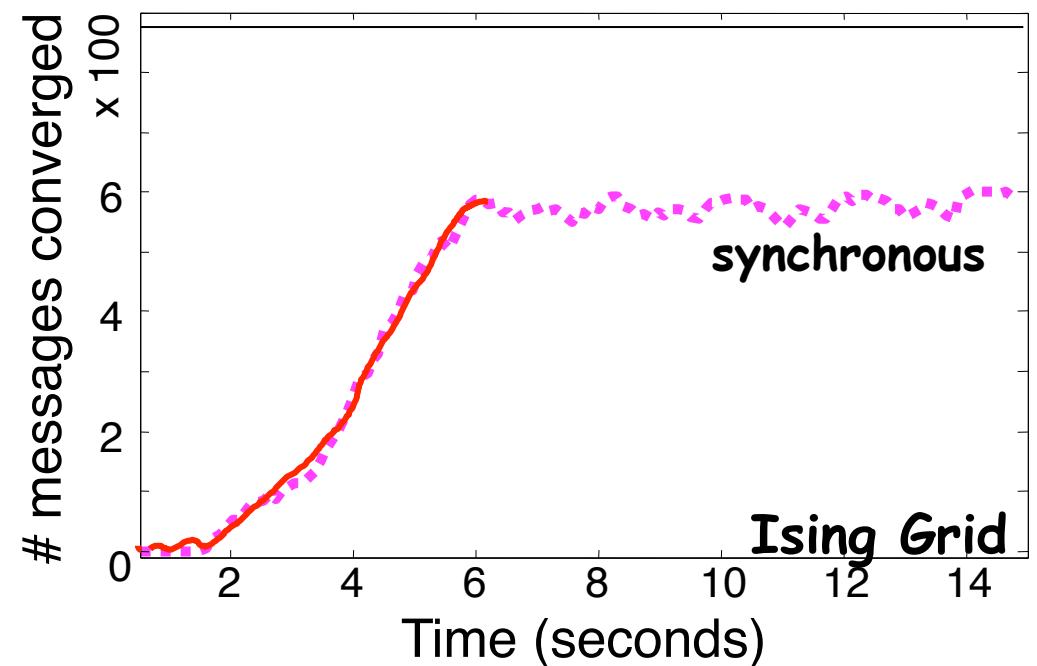
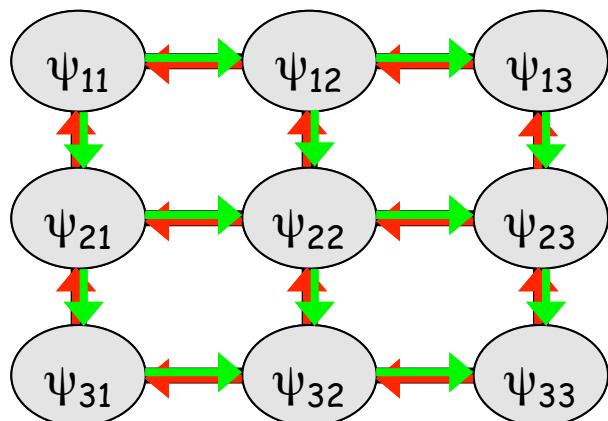
Nonconvergent BP Run



Daphne Koller

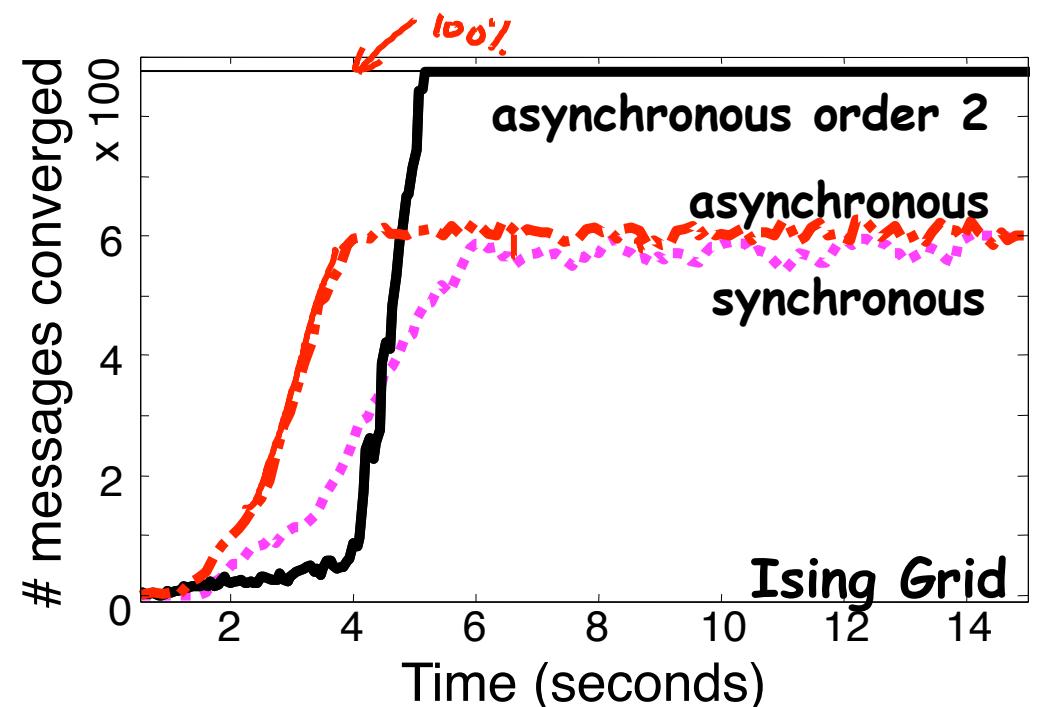
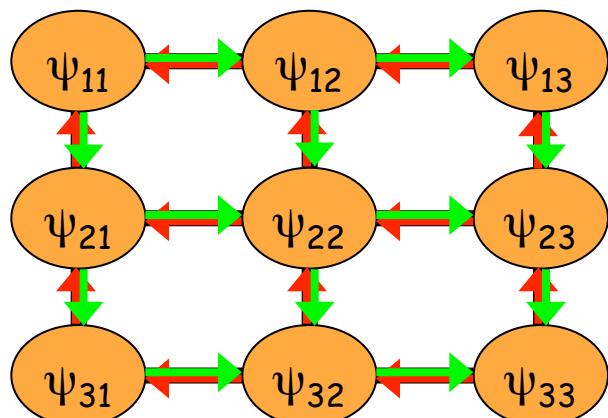
Different Variants of BP

Synchronous BP:
all messages are
updated in parallel



Different Variants of BP

Asynchronous BP:
Messages are updated
one at a time

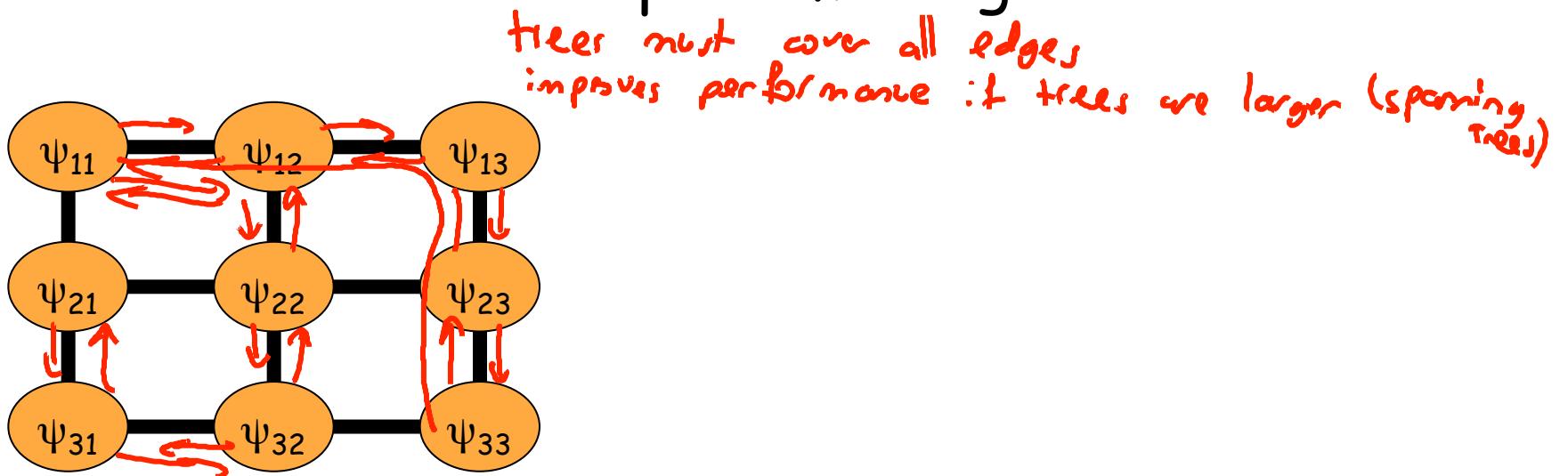


Observations

- Convergence is a local property:
 - some messages converge soon
 - others may never converge
- Synchronous BP converges considerably worse than asynchronous
- Message passing order makes a difference to extent and rate of convergence

Informed Message Scheduling

- Tree reparameterization (TRP)
 - Pick a tree and pass messages to calibrate



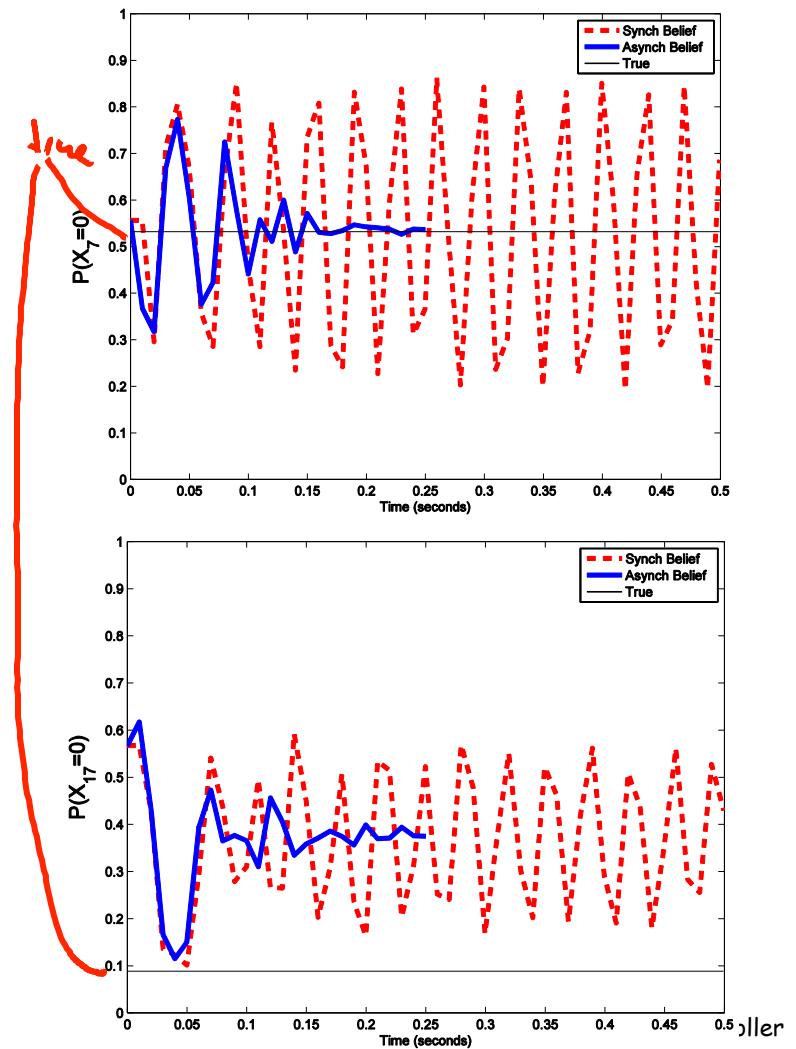
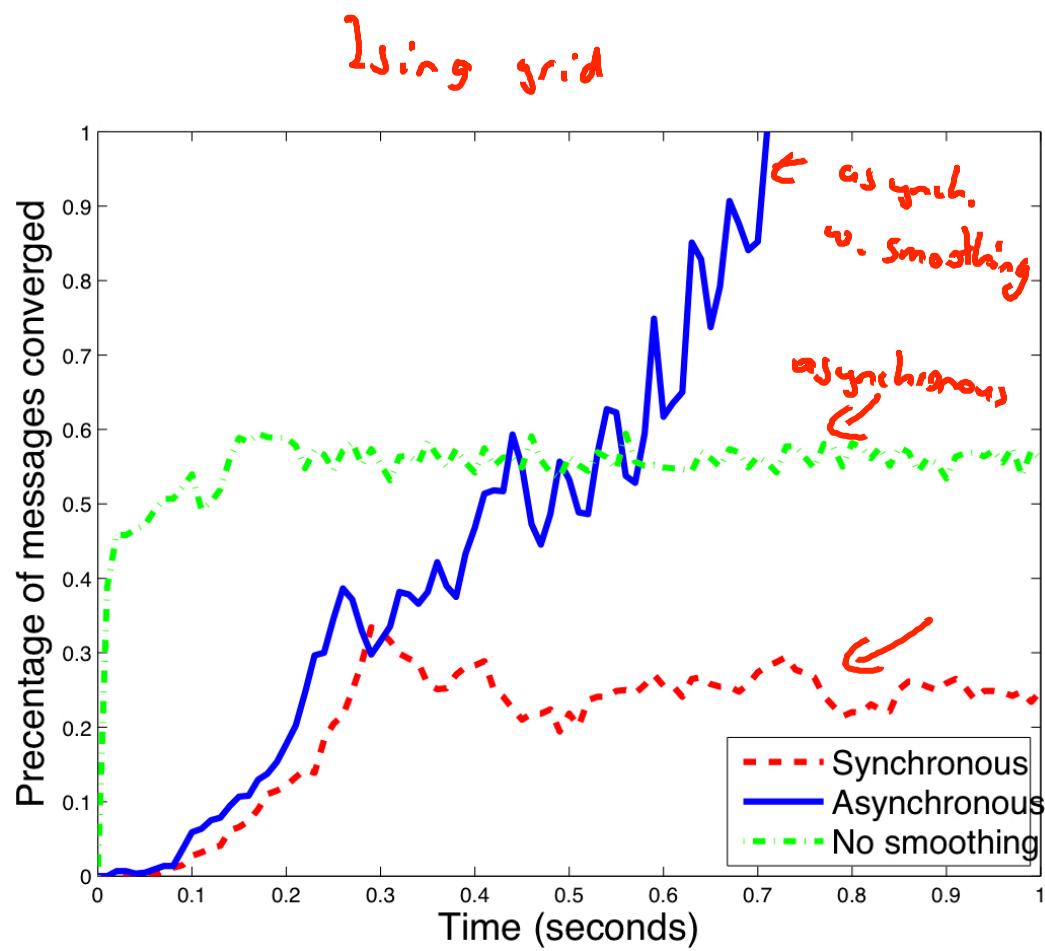
Informed Message Scheduling

- Tree reparameterization (TRP)
 - Pick a tree and pass messages to calibrate
- Residual belief propagation (RBP)
 - Pass messages between two clusters whose beliefs over the sepset disagree the most
priorly in w of edges

Smoothing (Damping) Messages

$$\delta_{i \rightarrow j} \leftarrow \underbrace{\sum_{C_i - S_{i,j}} \psi_i \prod_{k \neq j} \delta_{k \rightarrow i}}_{\text{new msg}}$$
$$\delta_{i \rightarrow j} \leftarrow \underbrace{\lambda}_{\text{new msg}} \left(\sum_{C_i - S_{i,j}} \psi_i \prod_{k \neq j} \delta_{k \rightarrow i} \right) + (1 - \lambda) \underbrace{\delta_{i \rightarrow j}^{\text{old}}}_{\text{old msg}}$$

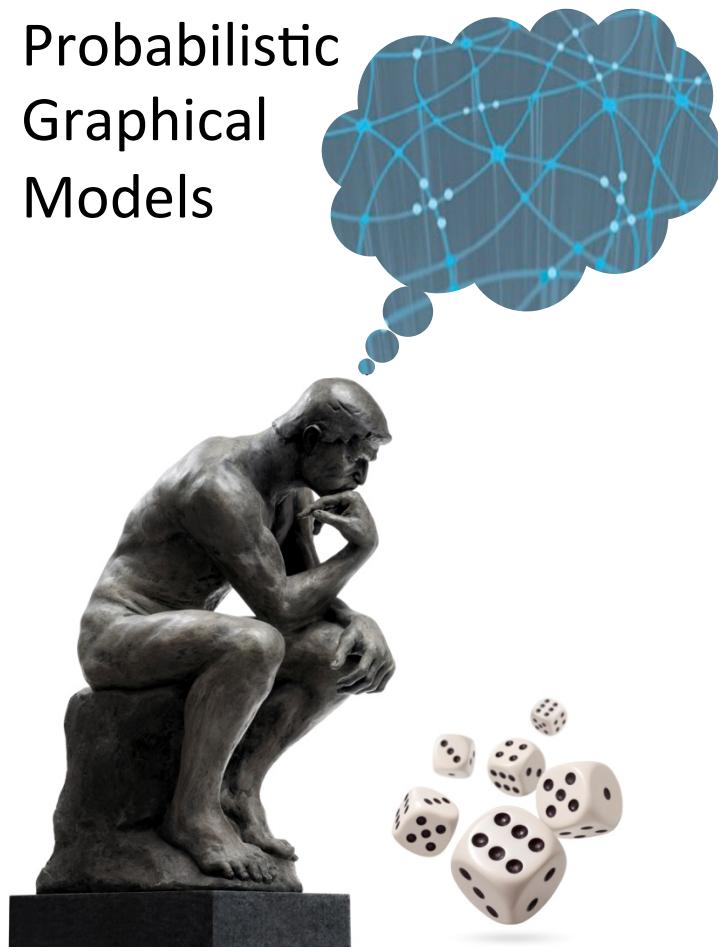
- Dampens oscillations in messages



Summary

- To achieve BP convergence, two main tricks
 - Damping
 - Intelligent message ordering
- Convergence doesn't guarantee correctness
- Bad cases for BP – both convergence & accuracy:
 - Strong potentials pulling in different directions
 - Tight loops
- Some new algorithms have better convergence:
 - Optimization-based view to inference

Probabilistic
Graphical
Models

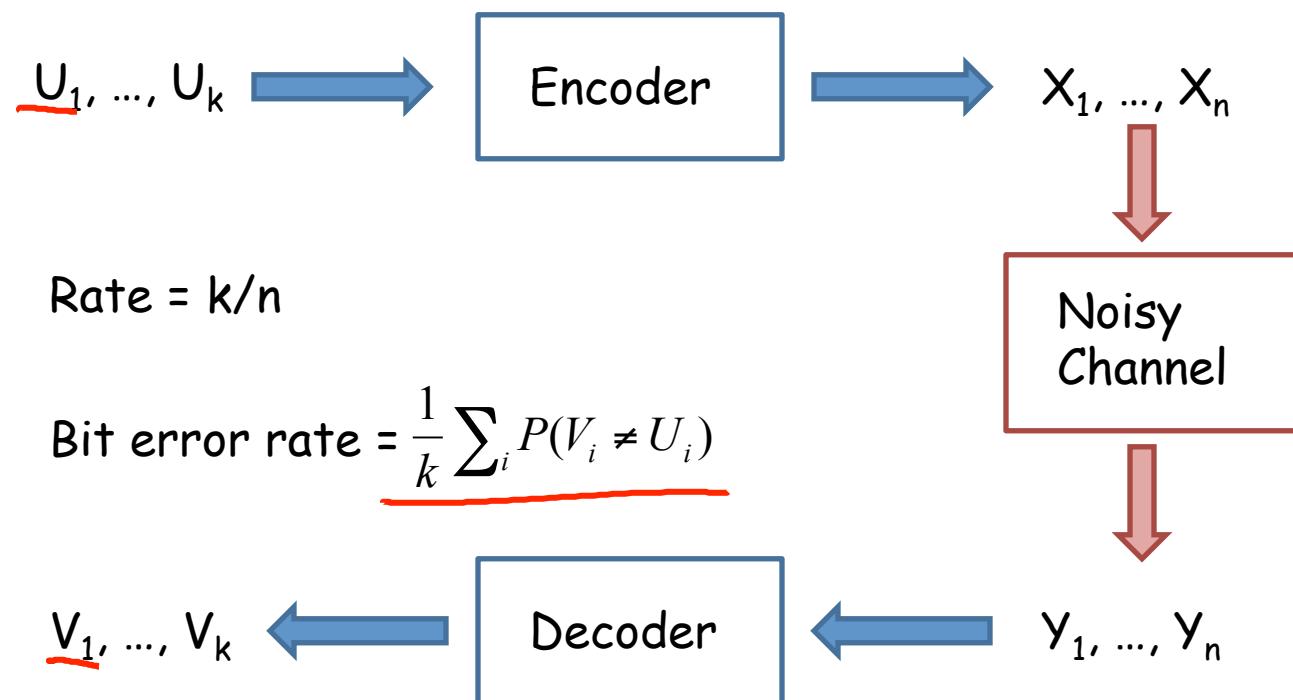


Inference

Message Passing

Loopy BP and
Message
Decoding

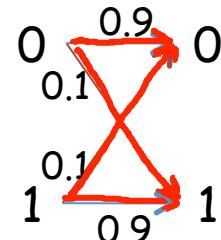
Message Coding & Decoding



Noisy
Channel

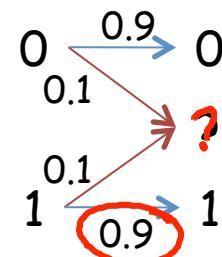
Channel Capacity

Binary
symmetric
channel

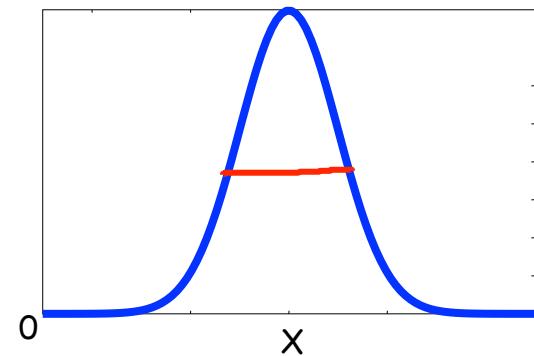


$$\text{capacity} = \underline{0.531}$$

Binary
erasure
channel

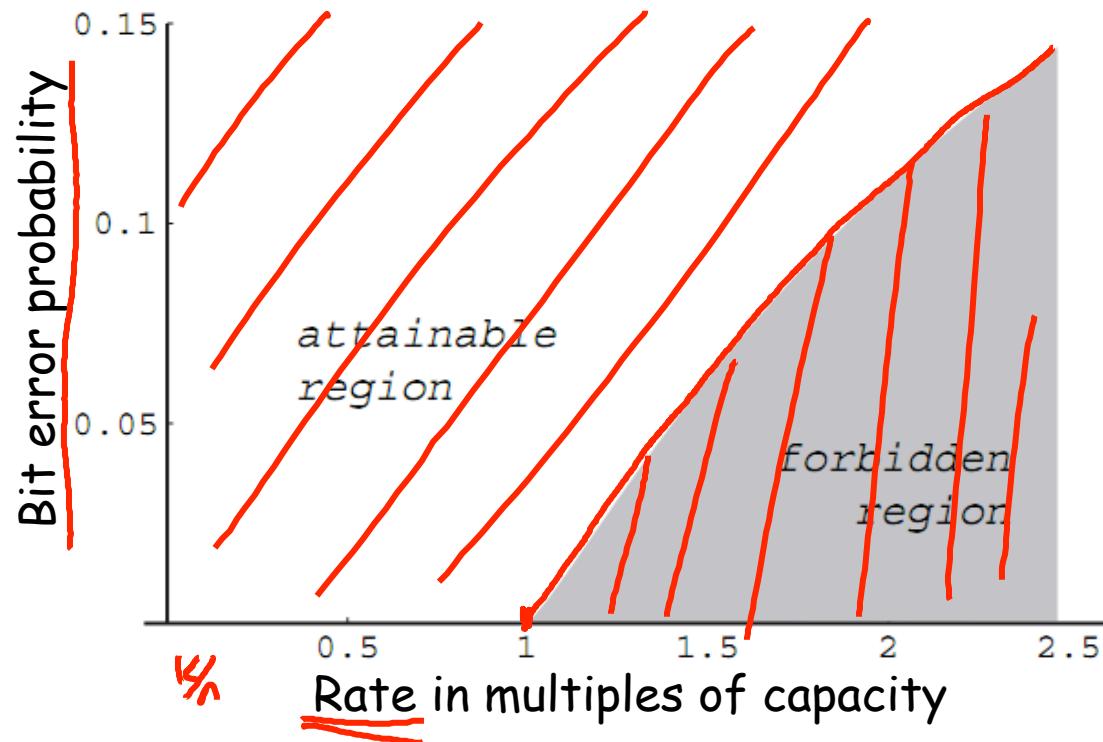


$$\text{capacity} = \underline{0.9}$$



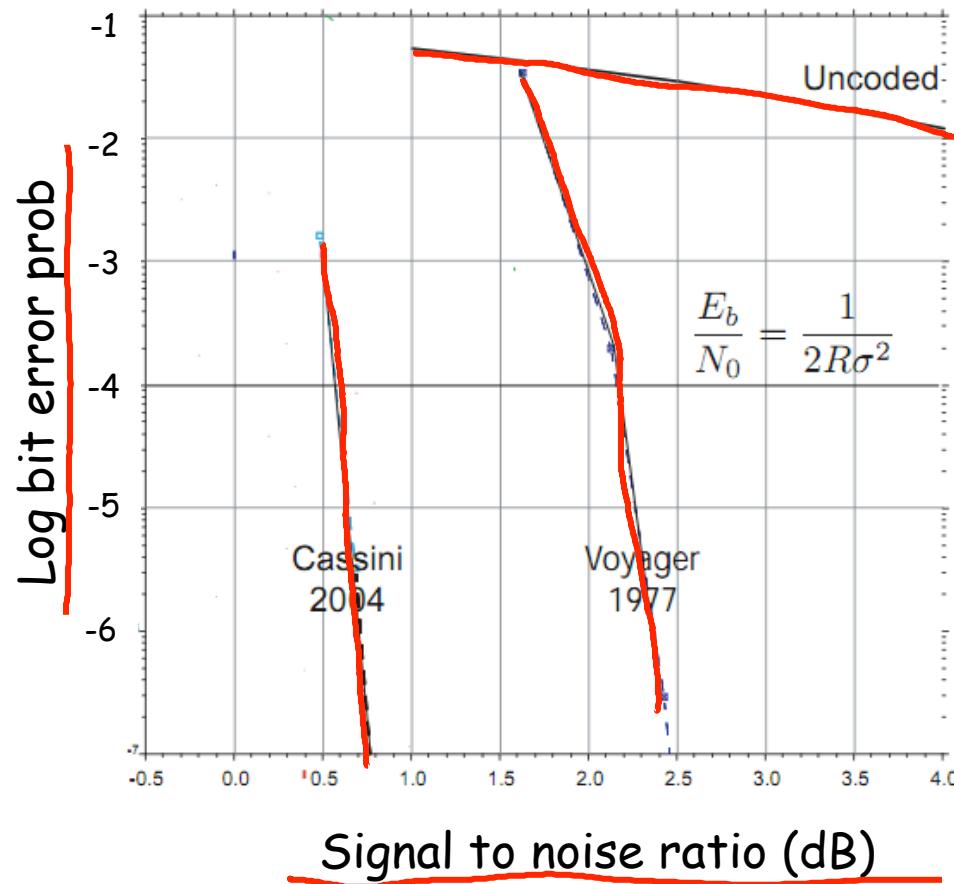
$$\text{capacity} = \underline{\frac{1}{2} \log \left(1 + \frac{E(X^2)}{\sigma^2} \right)}$$

Shannon's Theorem



McEliece

How close to C can we get?



Daphne Koller

Turbocodes (May 1993)

NEAR SHANNON LIMIT ERROR - CORRECTING
CODING AND DECODING : TURBO-CODES (1)

Claude Berrou, Alain Glavieux and Punya Thitimajshima

Claude Berrou, Integrated Circuits for Telecommunication Laboratory

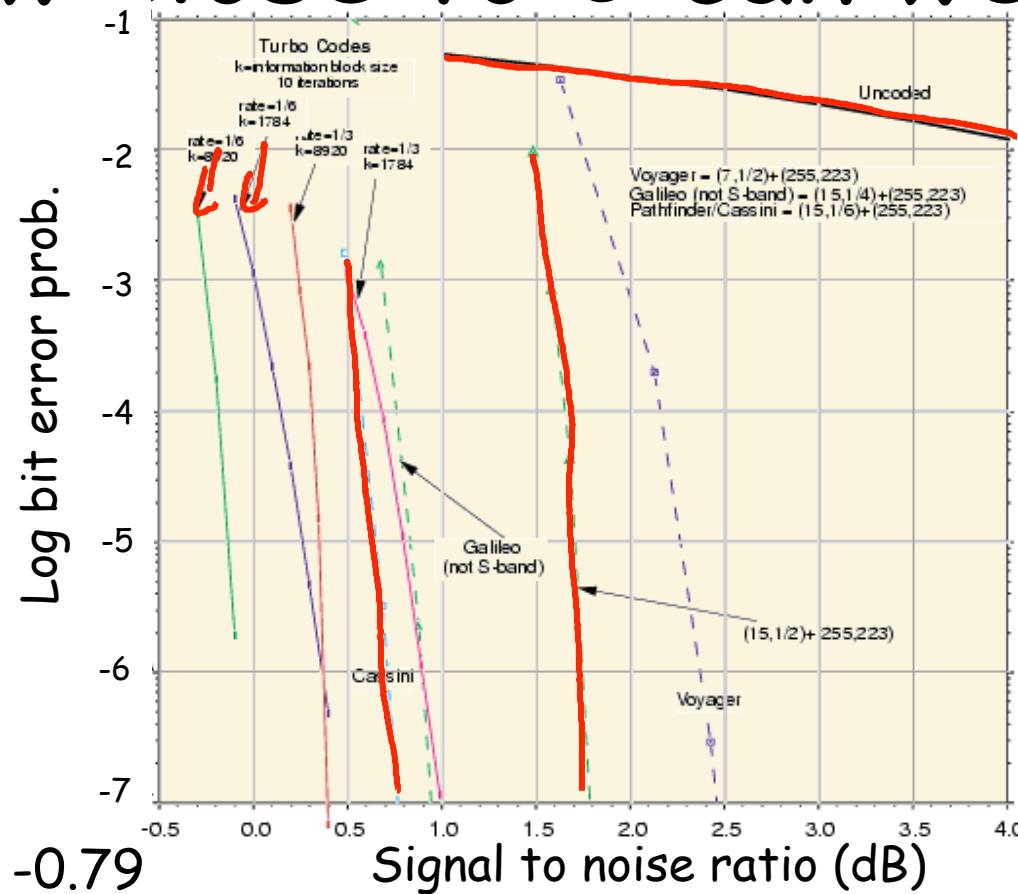
Alain Glavieux and Punya Thitimajshima, Digital Communication Laboratory

Ecole Nationale Supérieure des Télécommunications de Bretagne, France

(1) Patents N° 9105279 (France), N° 92460011.7 (Europe), N° 07/870,483 (USA)

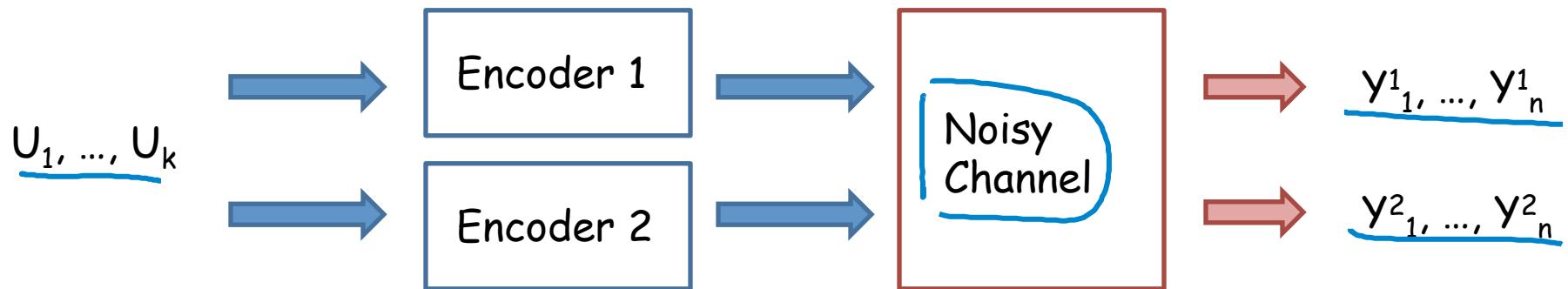
McEliece

How close to C can we get?

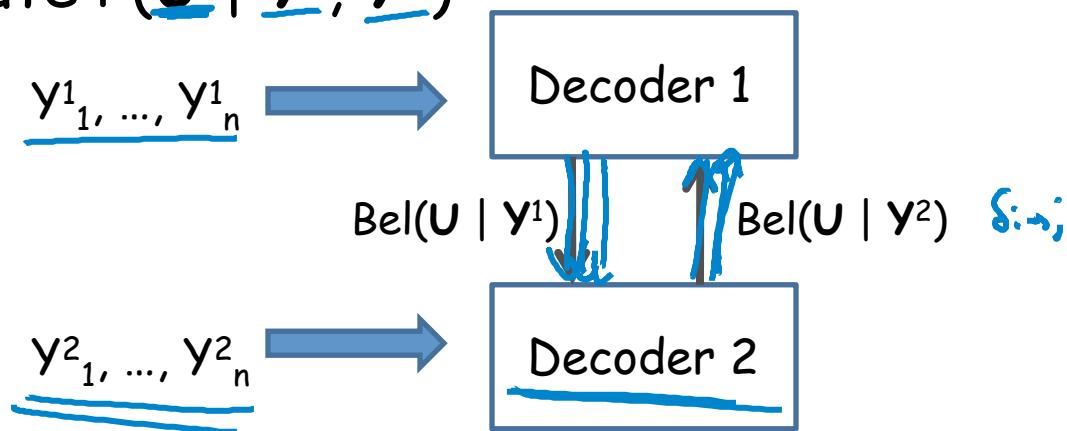


Daphne Koller

Turbocodes: The Idea

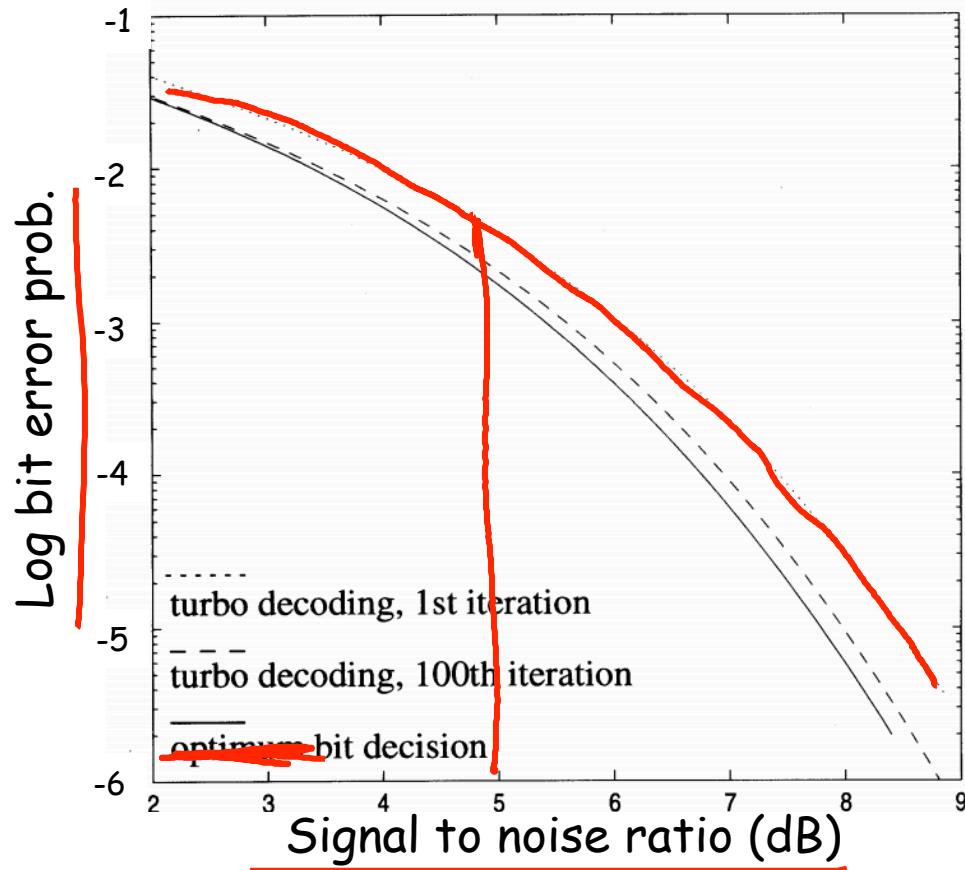


Compute $P(\underline{U} \mid \underline{y^1}, \underline{y^2})$



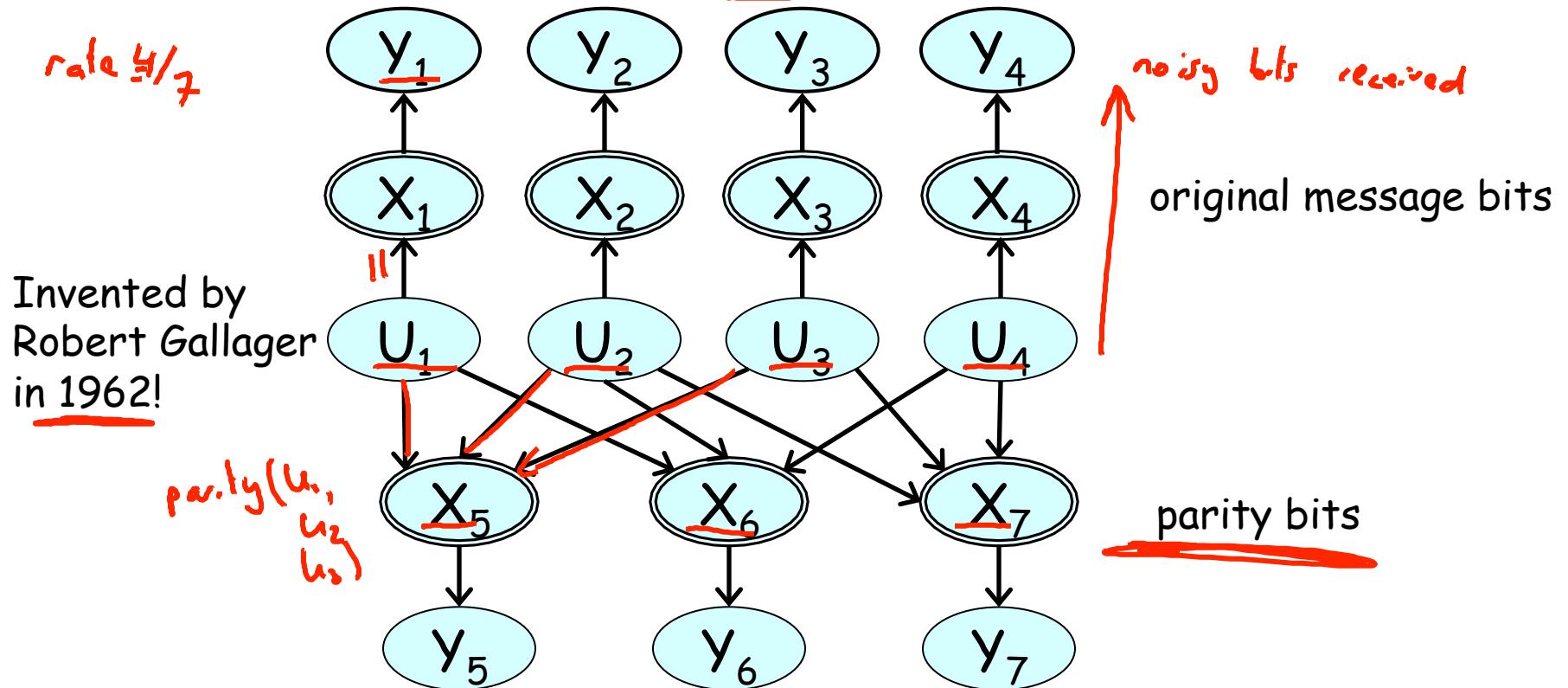
McEliece

Iterations of Turbo Decoding

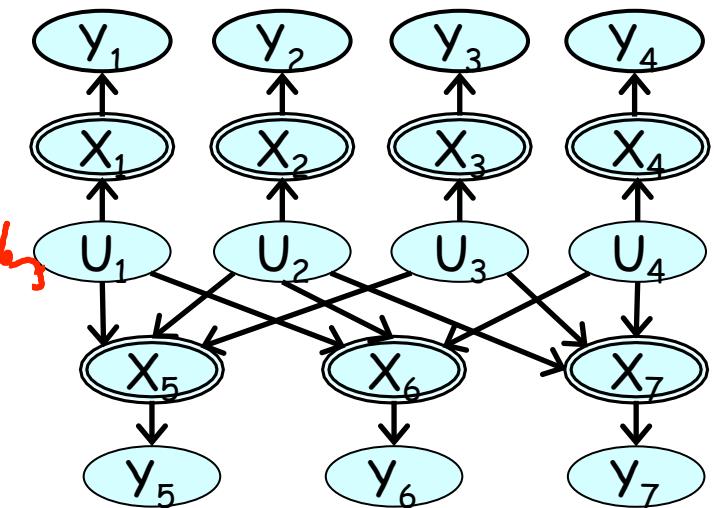
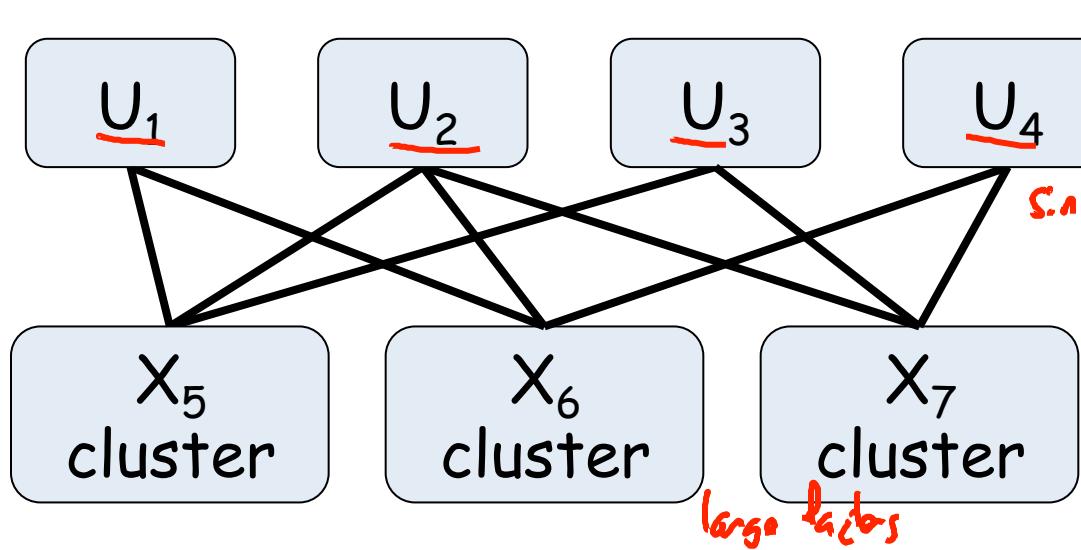


Daphne Koller

Low-Density Parity Checking Codes



Decoding as Loopy BP



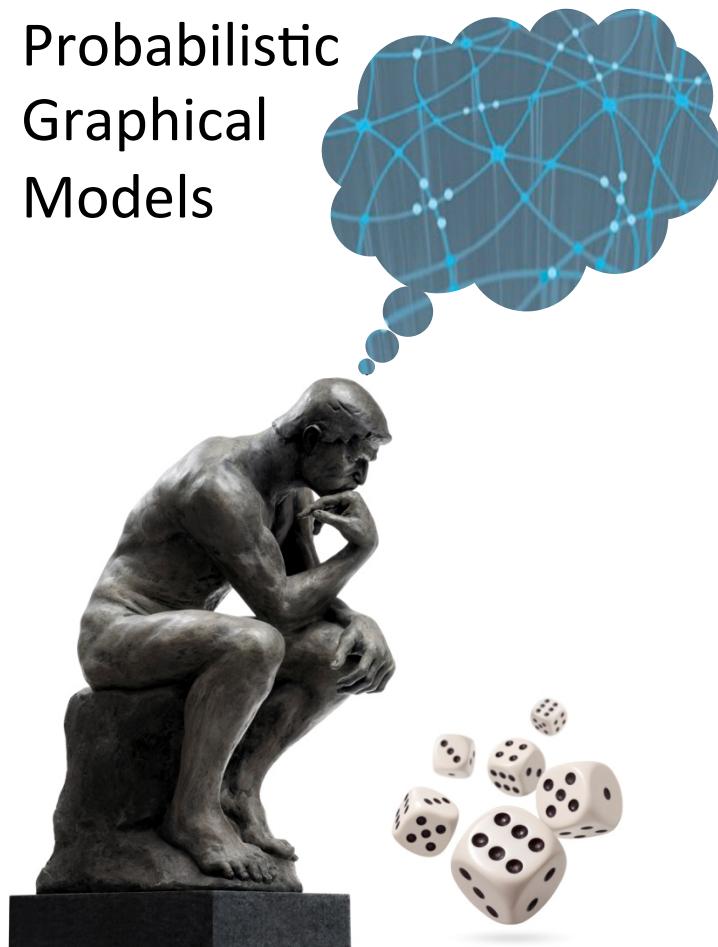
Turbo-Codes & LDPCs

- 3G and 4G mobile telephony standards
- Mobile television system from Qualcomm
- Digital video broadcasting
- Satellite communication systems
- New NASA missions (e.g., Mars Orbiter)
- Wireless metropolitan network standard

Summary

- Loopy BP rediscovered by coding practitioners
- Understanding turbocodes as loopy BP led to development of many new and better codes
 - Current codes coming closer and closer to Shannon limit
- Resurgence of interest in BP led to much deeper understanding of approximate inference in graphical models
 - Many new algorithms

Probabilistic
Graphical
Models



Inference

MAP

Max-Sum
Exact Inference

Product \Rightarrow Summation

$$P_{\Phi}(x) \propto \prod_k \phi_k(D_k)$$

$$\operatorname{argmax} \prod_k \phi_k(D_k)$$

$\log \phi_k(D_k)$

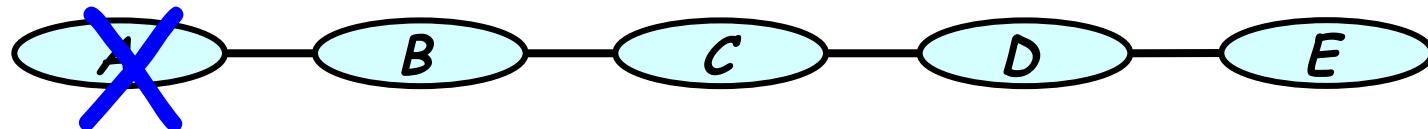
$\operatorname{argmax} \sum_k \theta_k(D_k)$
 $\theta(X_1, \dots, X_n)$

a ¹	b ¹	8
a ¹	b ²	1
a ²	b ¹	0.5
a ²	b ²	2

↓ log₂

a ¹	b ¹	3
a ¹	b ²	0
a ²	b ¹	-1
a ²	b ²	1

Max-Sum Elimination in Chains



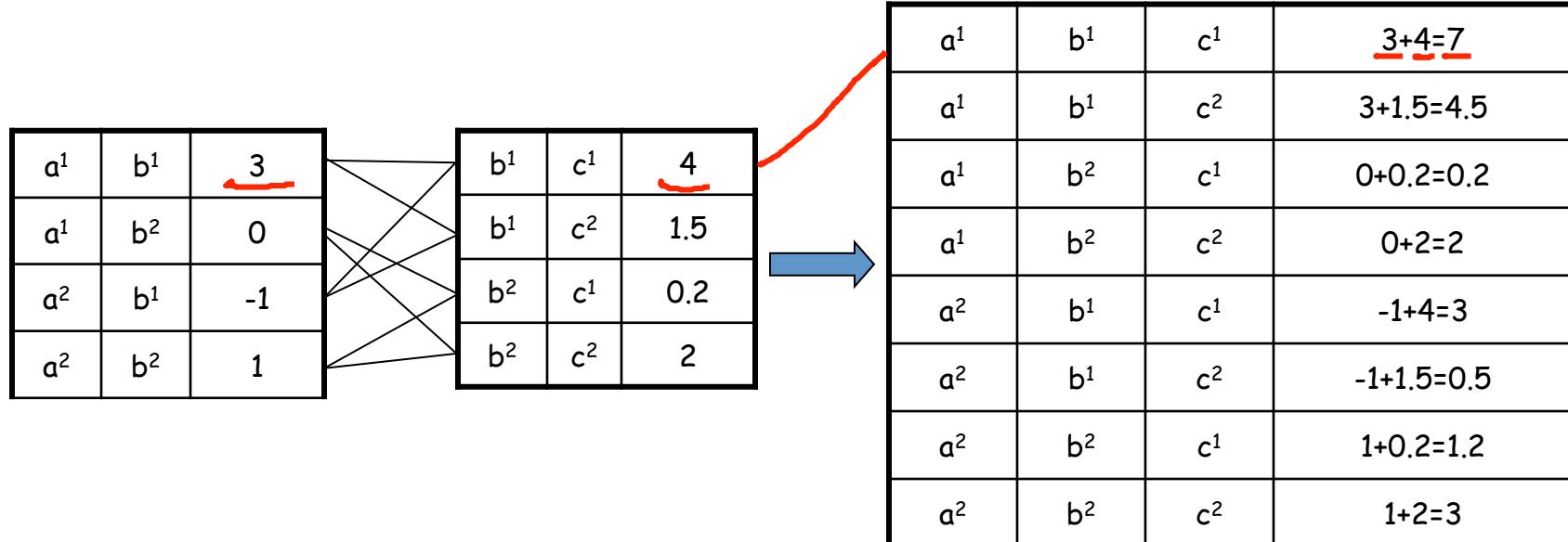
$\theta(A, B, C, D, E)$

$$\max_D \max_C \max_B \max_A (\theta_1(A, B) + \theta_2(B, C) + \theta_3(C, D) + \theta_4(D, E))$$

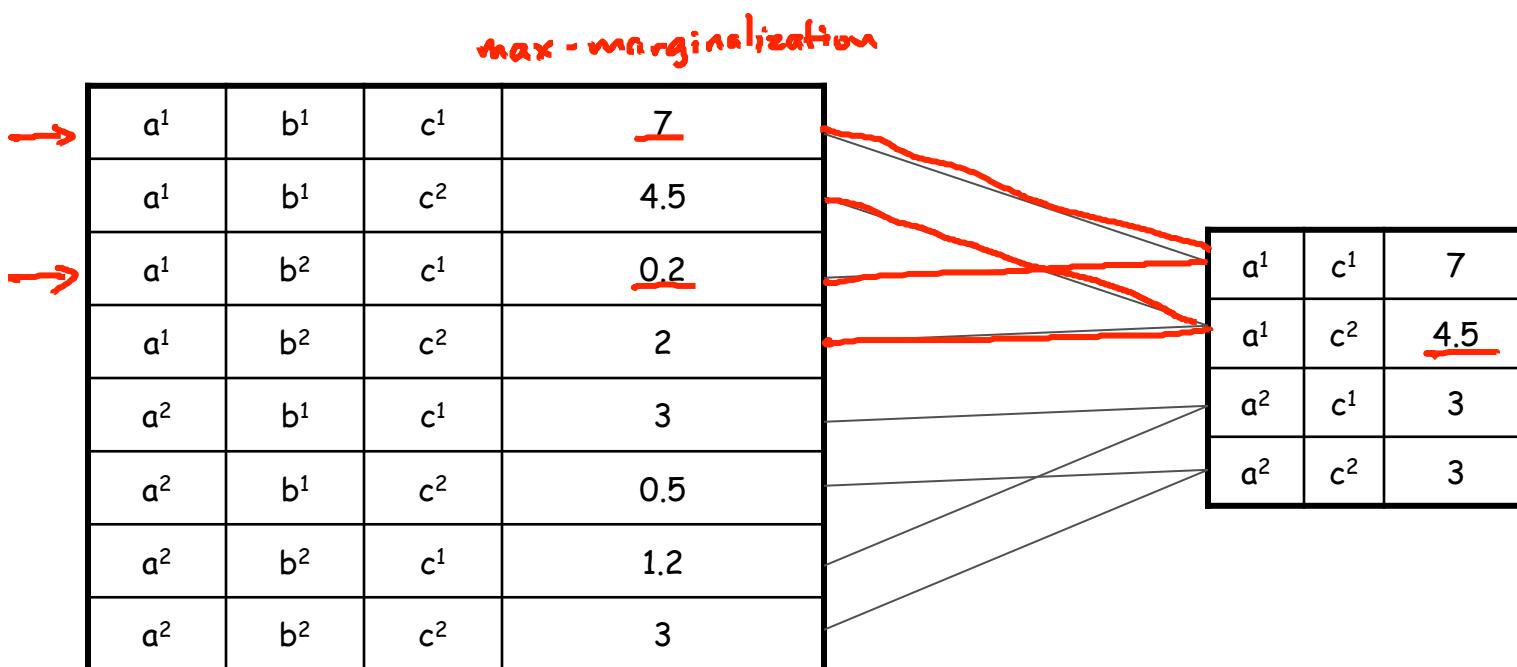
$$\max_D \max_C \max_B (\theta_2(B, C) + \theta_3(C, D) + \theta_4(D, E) + \underline{\max_A \theta_1(A, B)})$$

$$\max_D \max_C \max_B (\theta_2(B, C) + \theta_3(C, D) + \theta_4(D, E) + \underbrace{\lambda_1(B)}_{//})$$

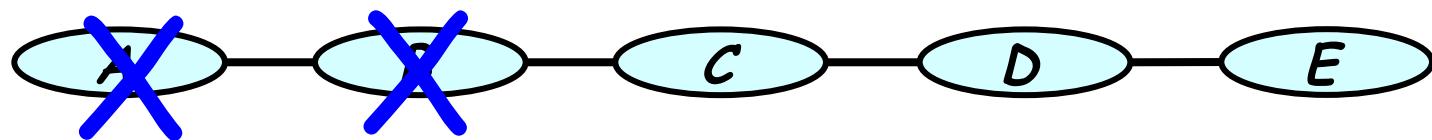
Factor Summation



Factor Maximization



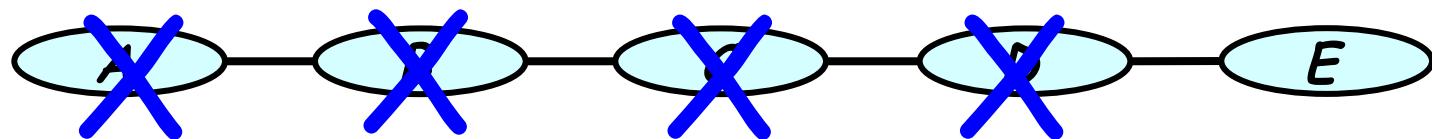
Max-Sum Elimination in Chains



$$\max_D \max_C \max_B (\theta_2(B, C) + \theta_3(C, D) + \theta_4(D, E) + \lambda_l(B))$$
$$\max_D \max_C (\theta_3(C, D) + \theta_4(D, E) + \max_B (\theta_2(B, C) + \lambda_l(B)))$$

$$\max_D \max_C (\theta_3(C, D) + \theta_4(D, E) + \lambda_2(C))$$

Max-Sum Elimination in Chains



$$\max_D \max_C (\theta_3(C, D) + \theta_4(D, E) + \lambda_2(C))$$

$$\max_D (\theta_4(D, E) + \lambda_3(D))$$

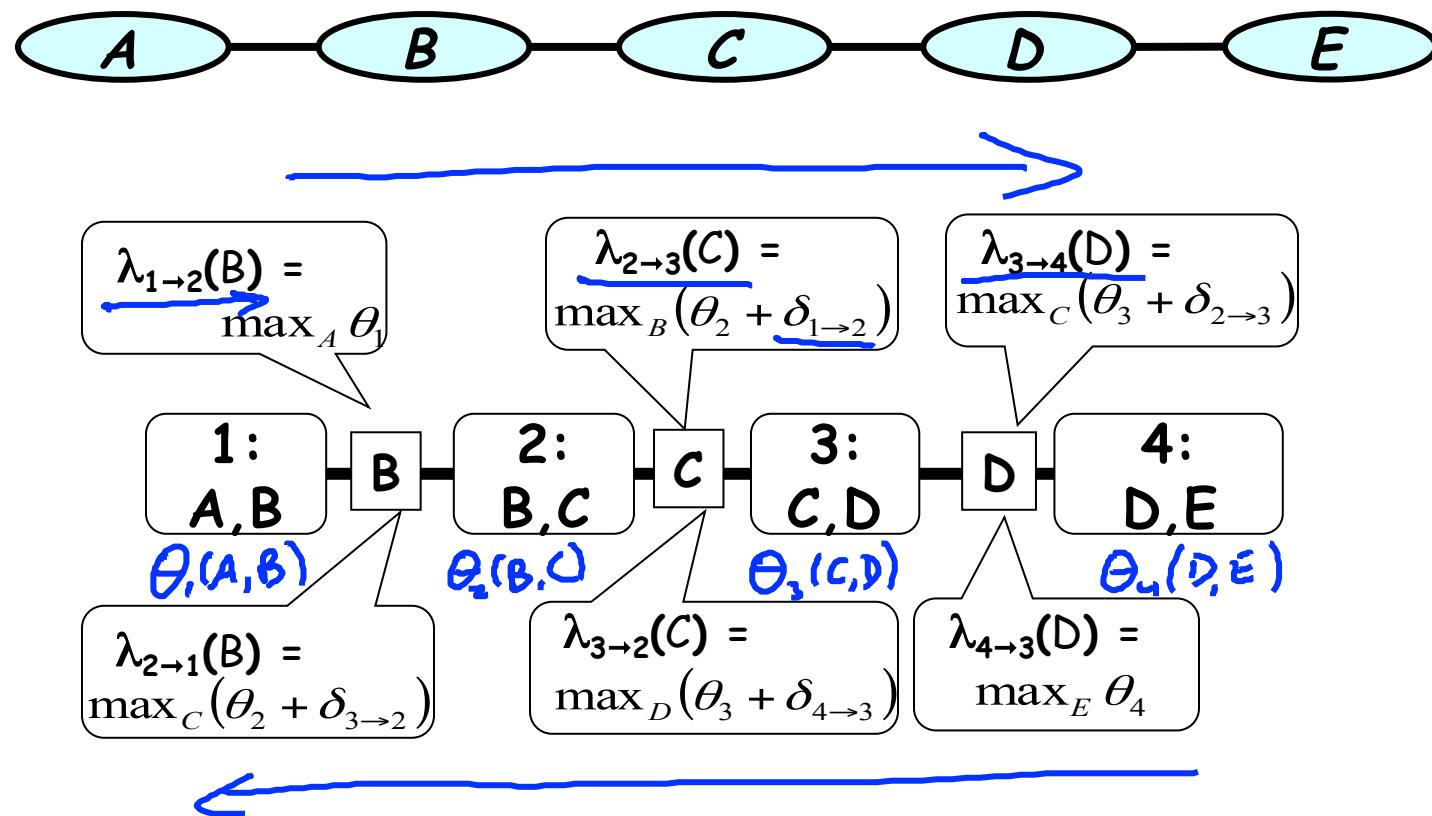
$$\lambda_4(E)$$

$$\lambda_4(e) = \max_{a,b,c,d} \Theta(a,b,c,d,e)$$

max-marginal

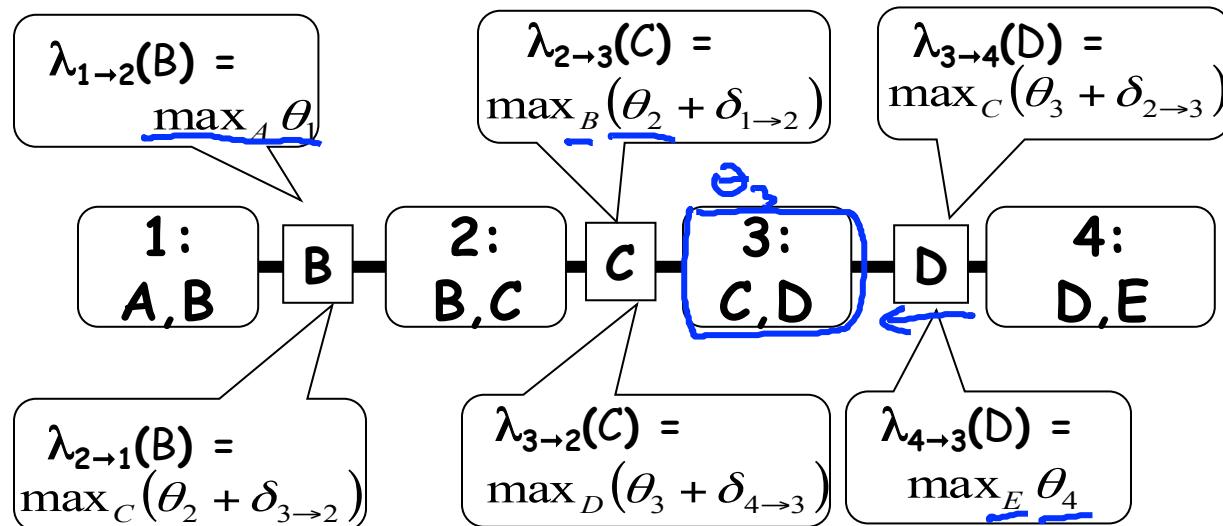
best value that
I can get if we
mandate $E=e$

Max-Sum in Clique Trees



Convergence of Message Passing

- Once C_i receives a final message from all neighbors except C_j , then $\lambda_{i \rightarrow j}$ is also final (will never change)
- Messages from leaves are immediately final



Simple Example



$\Theta_{\text{1}}(A, B)$

a ¹	b ¹	3
a ¹	b ²	0
a ²	b ¹	-1
a ²	b ²	1

$\Theta_{\text{2}}(B, C)$

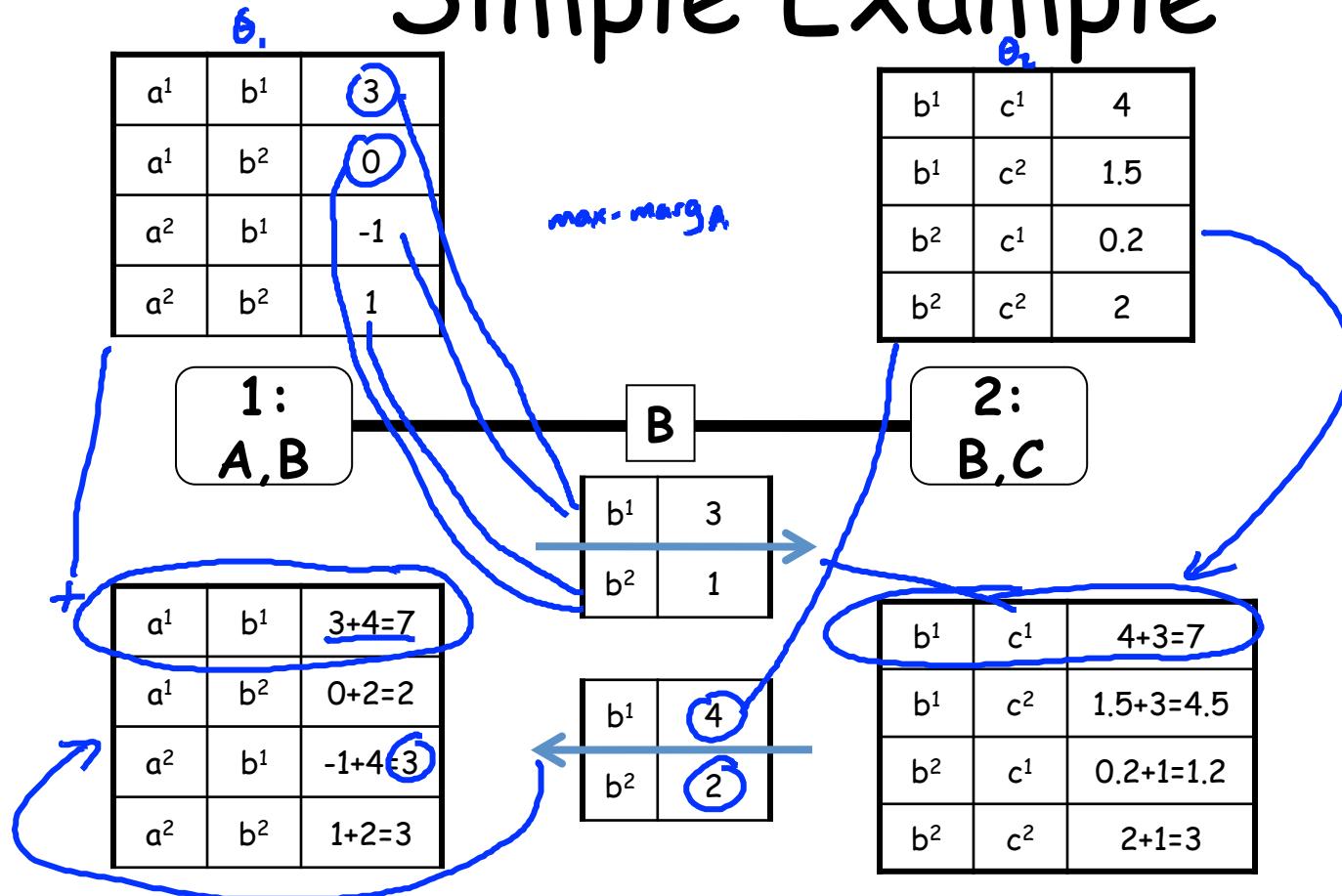
b ¹	c ¹	4
b ¹	c ²	1.5
b ²	c ¹	0.2
b ²	c ²	2

$$\Theta = \Theta_1 + \Theta_2$$



a ¹	b ¹	c ¹	3+4=7
a ¹	b ¹	c ²	3+1.5=4.5
a ¹	b ²	c ¹	0+0.2=0.2
a ¹	b ²	c ²	0+2=2
a ²	b ¹	c ¹	-1+4=3
a ²	b ¹	c ²	-1+1.5=0.5
a ²	b ²	c ¹	1+0.2=1.2
a ²	b ²	c ²	1+2=3

Simple Example



Max-Sum BP at Convergence

- Beliefs at each clique are max-marginals

$$\beta_i(C_i) = \underbrace{\theta_i(C_i)}_{k} + \sum_{k \rightarrow i} \lambda_{k \rightarrow i}$$

incoming
msgs

$$\beta_i(\underline{C}_i) = \max_{W_i} \theta(\underline{C}_i, W_i)$$

$$W_i = \{X_1, \dots, X_n\} - C_i$$

- Calibration: cliques agree on shared variables

			$\max_{C_i - S_{i,j}} \beta_i(C_i) = \max_{C_j - S_{i,j}} \beta_j(C_j)$
a^1	b^1	$3+4=7$	$b^1 7$
a^1	b^2	$0+2=2$	$b^2 3$
a^2	b^1	$-1+4=3$	$b^1 .7$
a^2	b^2	$1+2=3$	$b^2 .3$

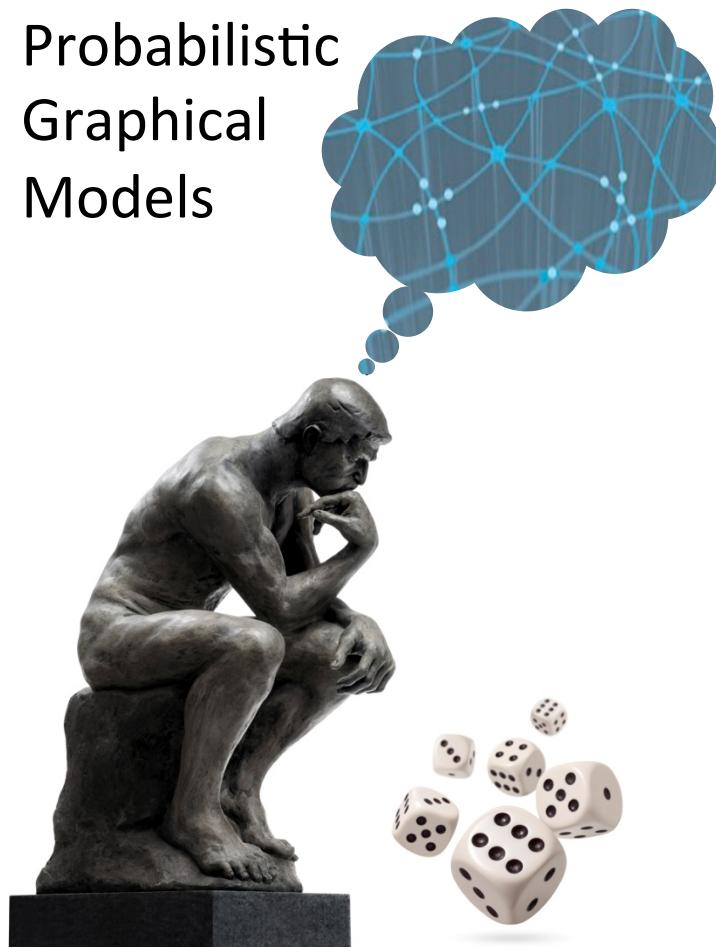
$\beta_i = \theta_i + \lambda_{i \rightarrow j} \rightarrow 1$
 $\beta_j = \theta_j + \lambda_{j \rightarrow i} \rightarrow 2$

b^1	c^1	$4+3=7$
b^1	c^2	$1.5+3=4$
b^2	c^1	$0.2+1=1$
b^2	c^2	$2+1=3$

Summary

- The same clique tree algorithm used for sum-product can be used for max-sum
- As in sum-product, convergence is achieved after a single up-down pass
- Result is a max-marginal at each clique C :
 - For each assignment c to C , what is the score of the best completion to c

Probabilistic
Graphical
Models



Inference

MAP

Finding a MAP Assignment

Decoding a MAP Assignment

- Easy if MAP assignment is unique
 - Single maximizing assignment at each clique
 - Whose value is the θ value of the MAP assignment
 - Due to calibration, choices at all cliques must agree

a^1	b^1	c^1	7
a^1	b^1	c^2	4.5
a^1	b^2	c^1	0.2
a^1	b^2	c^2	2
a^2	b^1	c^1	3
a^2	b^1	c^2	0.5
a^2	b^2	c^1	1.2
a^2	b^2	c^2	3

a^1	b^1	$3+4=7$
a^1	b^2	$0+2=2$
a^2	b^1	$-1+4=3$
a^2	b^2	$1+2=3$

b^1	c^1	$4+3=7$
b^1	c^2	$1.5+3=4.5$
b^2	c^1	$0.2+1=1.2$
b^2	c^2	$2+1=3$

Decoding a MAP assignment

- If MAP assignment is not unique, we may have multiple choices at some cliques
- Arbitrary tie-breaking may not produce a MAP assignment

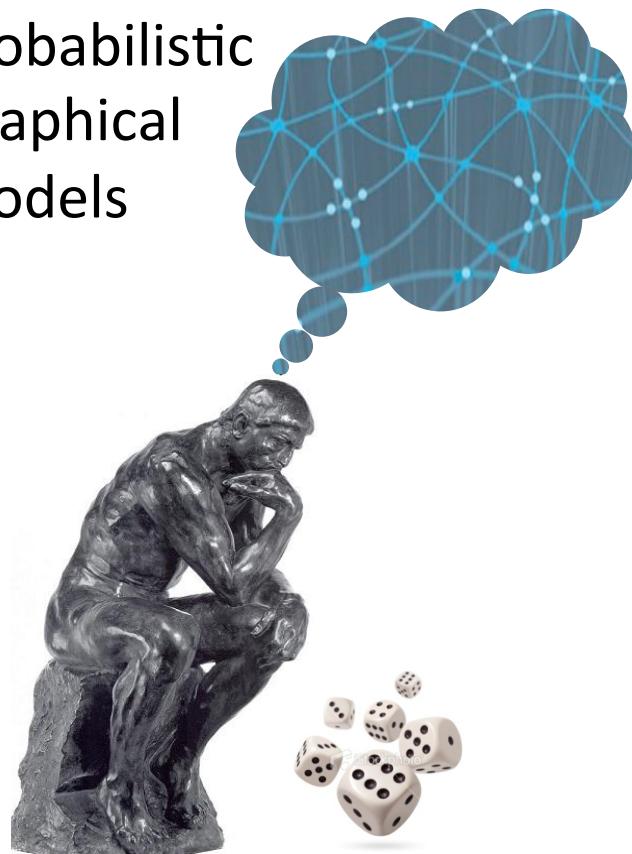
a^1	b^1	2
a^1	b^2	1
a^2	b^1	1
a^2	b^2	2

b^1	c^1	2
b^1	c^2	1
b^2	c^1	1
b^2	c^2	2

Decoding a MAP assignment

- If MAP assignment is not unique, we may have multiple choices at some cliques
- Arbitrary tie-breaking may not produce a MAP assignment
- Two options:
 - Slightly perturb parameters to make MAP unique
 - Use traceback procedure that incrementally builds a MAP assignment, one variable at a time

Probabilistic
Graphical
Models



Inference

MAP

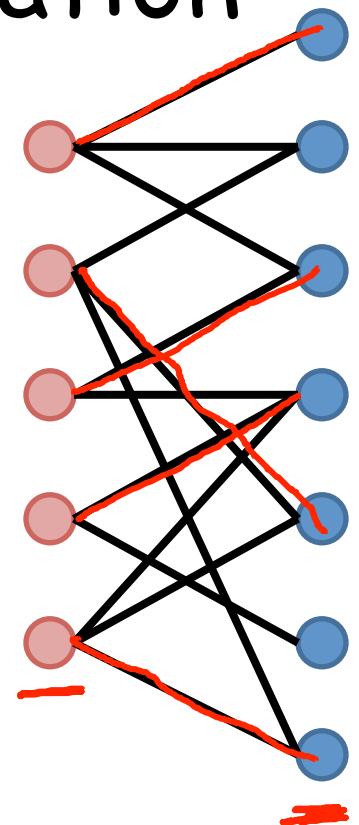
Tractable
MAP
Problems

Correspondence / data association

$$X_{ij} = \begin{cases} 1 & \text{if } i \text{ matched to } j \\ 0 & \text{otherwise} \end{cases}$$

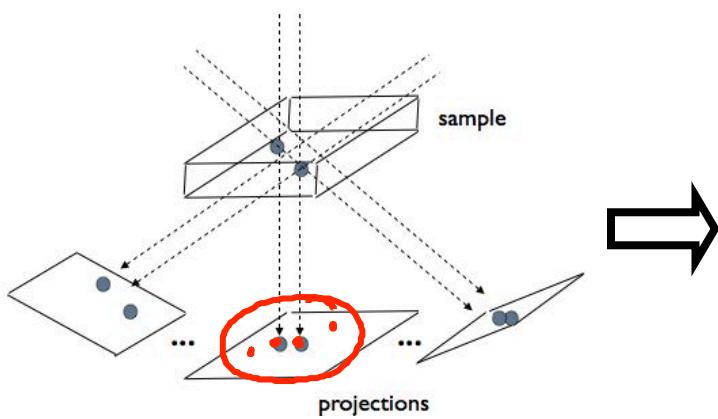
θ_{ij} = quality of "match" between i and j

- Find highest scoring matching
 - maximize $\sum_{ij} \theta_{ij} X_{ij}$
 - subject to mutual exclusion constraint
- Easily solved using matching algorithms
- Many applications
 - matching sensor readings to objects
 - matching features in two related images ←
 - matching mentions in text to entities

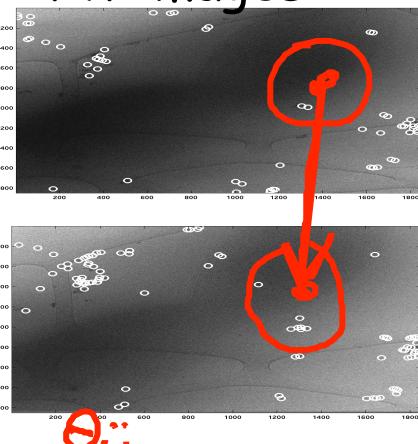


Daphne Koller

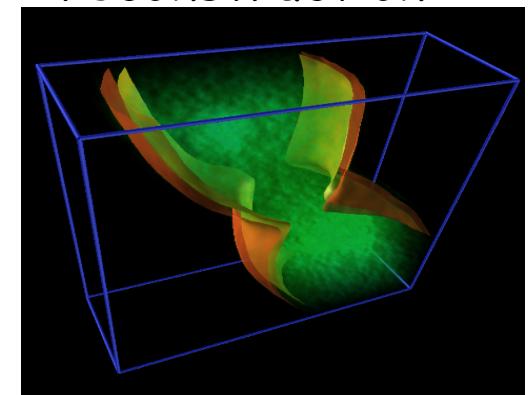
3D Cell Reconstruction



correspond
tilt images



compute 3D
reconstruction

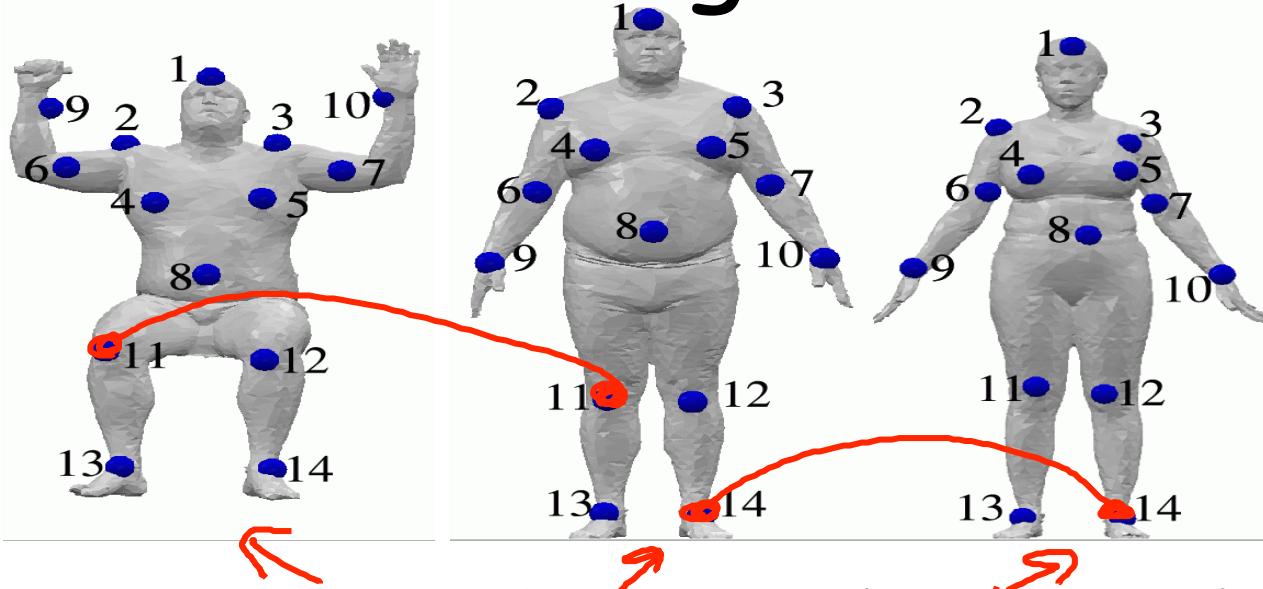


- Matching weights: similarity of location and local neighborhood appearance

Duchi, Tarlow, Elidan, and Koller, NIPS 2006. Amat, Moussavi, Comolli, Elidan, Downing, Horowitz, Journal of Structural Biology, 2006.

Daphne Koller

Mesh Registration



- Matching ~~pose~~ weights; similarity ~~of gap~~ Θ_{ij} of location and local neighborhood appearance

[Anguelov, Koller, Srinivasan, Thrun, Pang, Davis, NIPS 2004]

Daphne Koller

Associative potentials

- Arbitrary network over binary variables using only singleton θ_i and supermodular pairwise potentials θ_{ij}
 - Exact solution using algorithms for finding minimum cuts in graphs
- Many related variants admit efficient exact or approximate solutions
 - Metric MRFs ^{vision}

	0	1
0	a	b
1	c	d

$$a+d \geq b+c$$

Example: Depth Reconstruction



view 1

view 2

depth
reconstruction

denoising , infilling , FG/BG segmentation

Scharstein & Szeliski, "High-accuracy stereo depth maps using structured light"
Proc. IEEE CVPR 2003

Daphne Koller

Cardinality Factors

- A factor over arbitrarily many binary variables X_1, \dots, X_k
- Score(X_1, \dots, X_k) = $f(\sum_i X_i)$
- Example applications:
 - soft parity constraints
 - prior on # pixels in a given category
 - prior on # of instances assigned to a given cluster

A	B	C	D	score
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	1
0	1	0	1	2
0	1	1	0	3
0	1	1	1	4
1	0	0	0	0
1	0	0	1	1
1	0	1	0	2
1	0	1	1	3
1	1	0	0	1
1	1	0	1	2
1	1	1	0	3
1	1	1	1	4

Daphne Koller

Sparse Pattern Factors

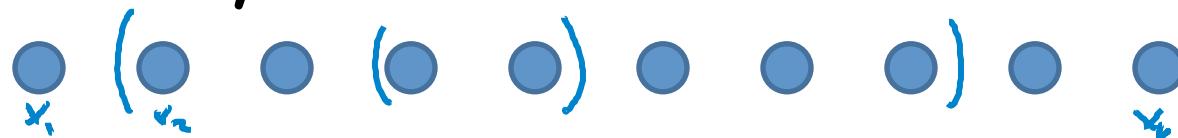
- A factor over variables X_1, \dots, X_k
 - $\text{Score}(X_1, \dots, X_k)$ specified for some small # of assignments x_1, \dots, x_k
 - Constant for all other assignments
- Examples: give higher score to combinations that occur in real data
 - In spelling, letter combinations that occur in dictionary
 - 5x5 image patches that appear in natural images

A	B	C	D	score
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Daphne Koller

Convexity Factors

- Ordered binary variables X_1, \dots, X_k
- Convexity constraints

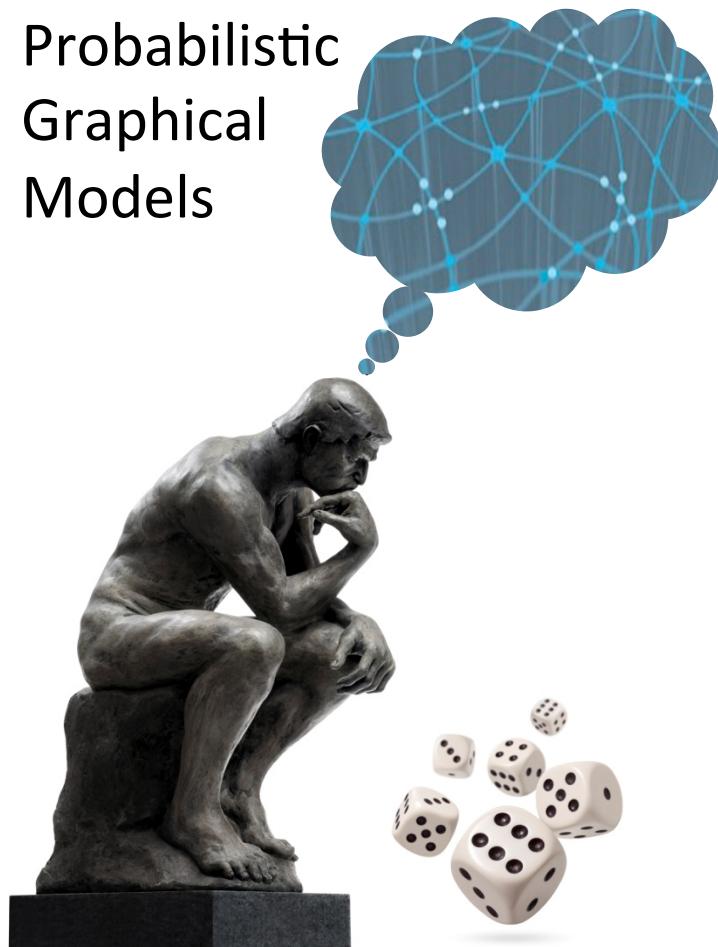


- Examples:
 - Convexity of “parts” in image segmentation
 - Contiguity of word labeling in text
 - Temporal contiguity of subactivities

Summary

- Many specialized models admit tractable MAP solution
 - Many do not have tractable algorithms for computing marginals
- These specialized models are useful
 - On their own
 - As a component in a larger model with other types of factors

Probabilistic
Graphical
Models



Inference

MAP

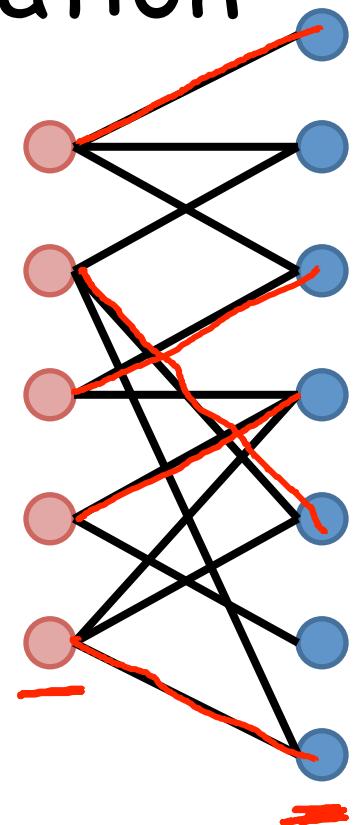
Tractable
MAP
Problems

Correspondence / data association

$$X_{ij} = \begin{cases} 1 & \text{if } i \text{ matched to } j \\ 0 & \text{otherwise} \end{cases}$$

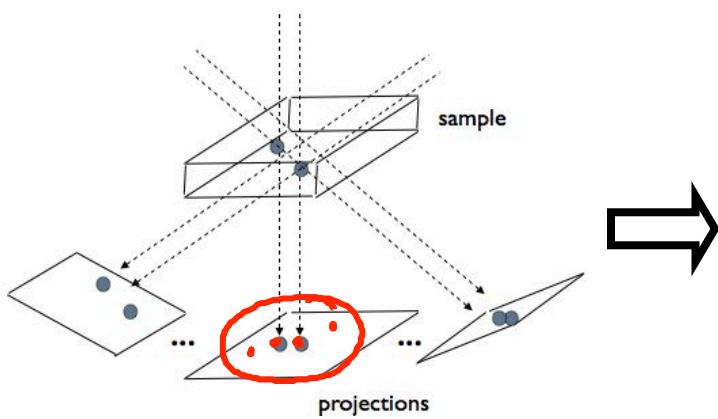
θ_{ij} = quality of "match" between i and j

- Find highest scoring matching
 - maximize $\sum_{ij} \theta_{ij} X_{ij}$
 - subject to mutual exclusion constraint
- Easily solved using matching algorithms
- Many applications
 - matching sensor readings to objects
 - matching features in two related images ←
 - matching mentions in text to entities

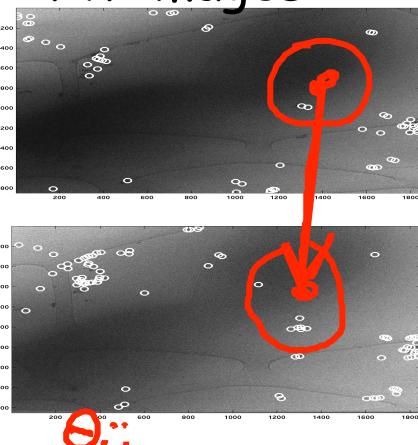


Daphne Koller

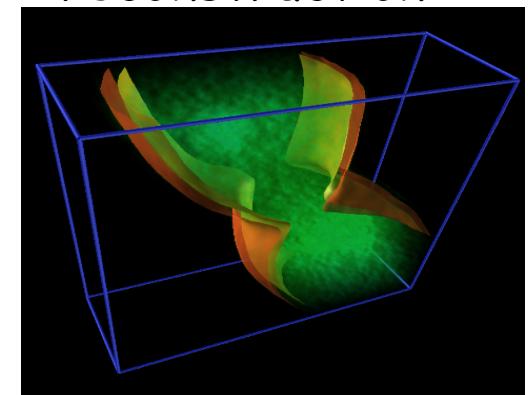
3D Cell Reconstruction



correspond
tilt images



compute 3D
reconstruction

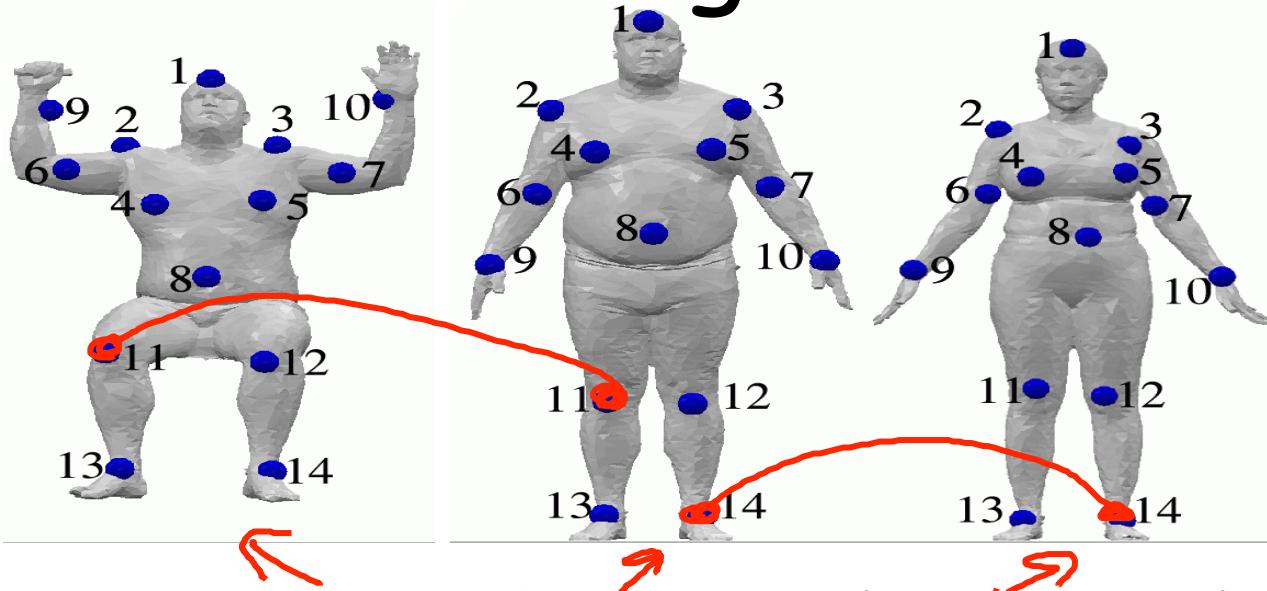


- Matching weights: similarity of location and local neighborhood appearance

Duchi, Tarlow, Elidan, and Koller, NIPS 2006. Amat, Moussavi, Comolli, Elidan, Downing, Horowitz, Journal of Structural Biology, 2006.

Daphne Koller

Mesh Registration



- Matching ~~pose~~ weights; similarity ~~of gap~~ of location and local neighborhood appearance

[Anguelov, Koller, Srinivasan, Thrun, Pang, Davis, NIPS 2004]

Daphne Koller

Associative potentials

- Arbitrary network over binary variables using only singleton θ_i and supermodular pairwise potentials θ_{ij}
 - Exact solution using algorithms for finding minimum cuts in graphs
- Many related variants admit efficient exact or approximate solutions
 - Metric MRFs ^{vision}

	0	1
0	a	b
1	c	d

$$a+d \geq b+c$$

Example: Depth Reconstruction



depth
reconstruction

denoising , infilling , FG/BG segmentation

Scharstein & Szeliski, "High-accuracy stereo depth maps using structured light"
Proc. IEEE CVPR 2003

Daphne Koller

Cardinality Factors

- A factor over arbitrarily many binary variables X_1, \dots, X_k
- Score(X_1, \dots, X_k) = $f(\sum_i X_i)$
- Example applications:
 - soft parity constraints
 - prior on # pixels in a given category
 - prior on # of instances assigned to a given cluster

A	B	C	D	score
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	1
0	1	0	1	2
0	1	1	0	3
0	1	1	1	4
1	0	0	0	0
1	0	0	1	1
1	0	1	0	2
1	0	1	1	3
1	1	0	0	1
1	1	0	1	2
1	1	1	0	3
1	1	1	1	4

Daphne Koller

Sparse Pattern Factors

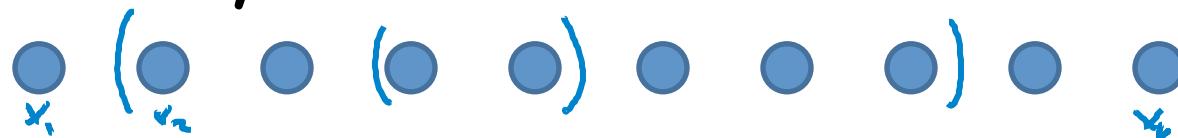
- A factor over variables X_1, \dots, X_k
 - $\text{Score}(X_1, \dots, X_k)$ specified for some small # of assignments x_1, \dots, x_k
 - Constant for all other assignments
- Examples: give higher score to combinations that occur in real data
 - In spelling, letter combinations that occur in dictionary
 - 5x5 image patches that appear in natural images

A	B	C	D	score
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Daphne Koller

Convexity Factors

- Ordered binary variables X_1, \dots, X_k
- Convexity constraints

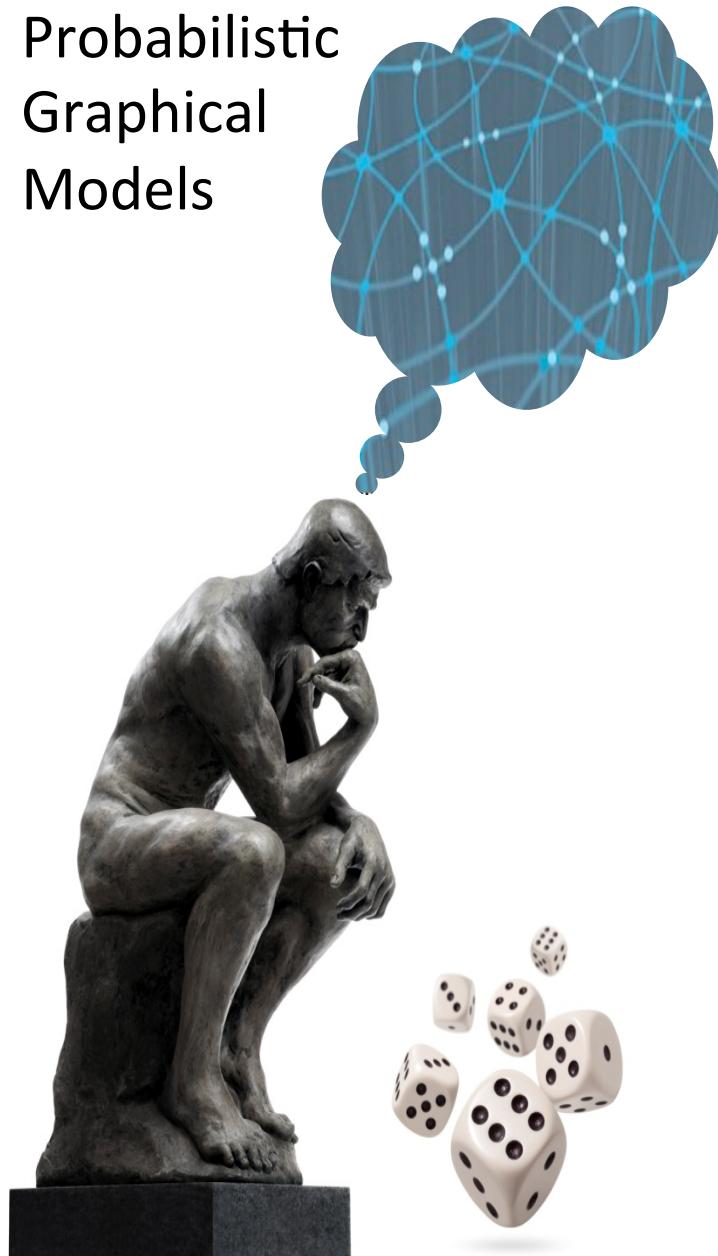


- Examples:
 - Convexity of “parts” in image segmentation
 - Contiguity of word labeling in text
 - Temporal contiguity of subactivities

Summary

- Many specialized models admit tractable MAP solution
 - Many do not have tractable algorithms for computing marginals
- These specialized models are useful
 - On their own
 - As a component in a larger model with other types of factors

Probabilistic
Graphical
Models



Inference

MAP

Dual
Decomposition

Problem Formulation

- Singleton factors $\theta_i(x_i)$
- Large factors $\theta_F(x_F)$

$$\text{MAP}(\boldsymbol{\theta}) = \max_{\boldsymbol{x}} \left(\sum_{i=1}^n \theta_i(x_i) + \sum_F \theta_F(x_F) \right)$$

Divide and Conquer

$$\text{MAP}(\theta) = \max_{\boldsymbol{x}} \left(\sum_{i=1}^n \theta_i(x_i) + \sum_F \theta_F(\boldsymbol{x}_F) \right)$$



$$\sum_{i=1}^n \max_{x_i} \theta_i(x_i) + \sum_F \overbrace{\max_{\boldsymbol{x}_F} \theta_F(\boldsymbol{x}_F)}$$

local decision making

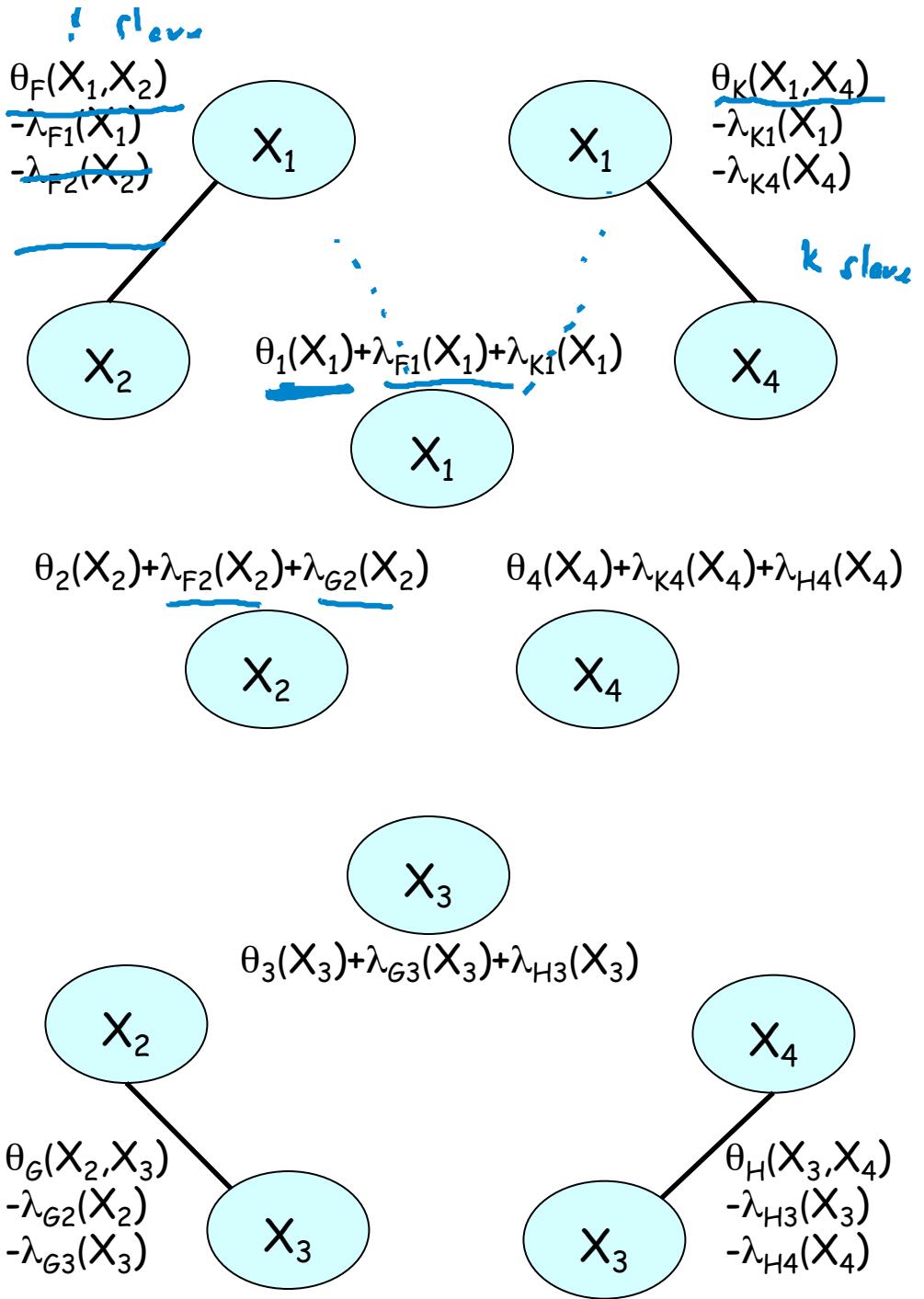
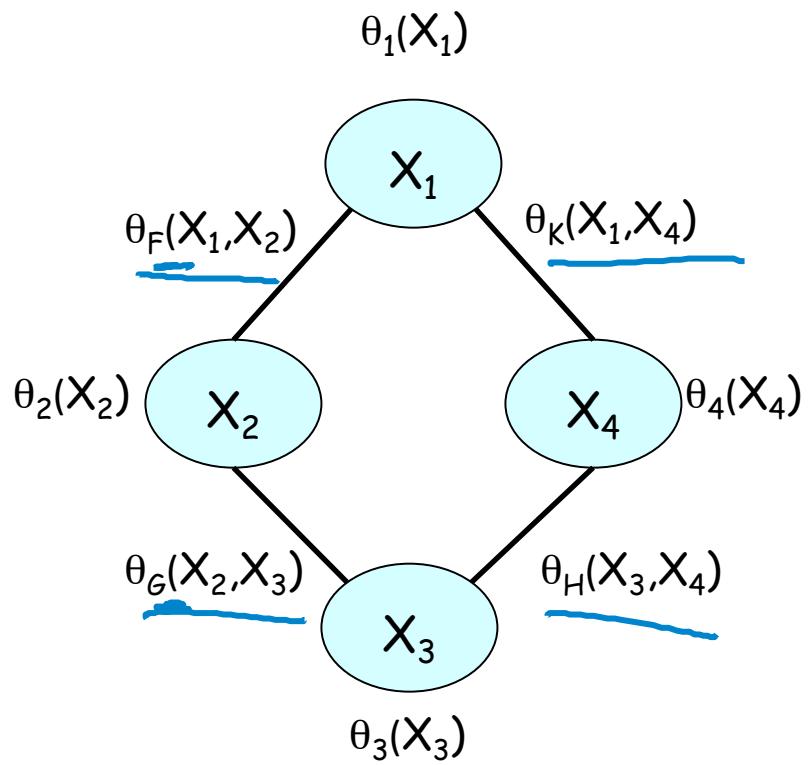
Divide and Conquer

$$\begin{aligned}
 \text{MAP}(\theta) &= \max_{\underline{x}} \left(\sum_{i=1}^n \theta_i(x_i) + \sum_F \theta_F(x_F) \right) \\
 &= \max_{\underline{x}} \left(\sum_{i=1}^n (\underbrace{\theta_i(x_i)}_{i: \text{ slave}} + \underbrace{\lambda_{F_i}(x_i)}_{F: i \in F}) + \sum_F \left(\theta_F(x_F) - \sum_{i \in F} \lambda_{F_i}(x_i) \right) \right) \\
 L(\lambda) &= \sum_{i=1}^n \max_{x_i} \left(\theta_i(x_i) + \sum_{F: i \in F} \lambda_{F_i}(x_i) \right) + \sum_F \max_{x_F} \left(\theta_F(x_F) - \sum_{i \in F} \lambda_{F_i}(x_i) \right)
 \end{aligned}$$

i: slave i ∈ F f slave
 messages between f and i
 agree with i slaves

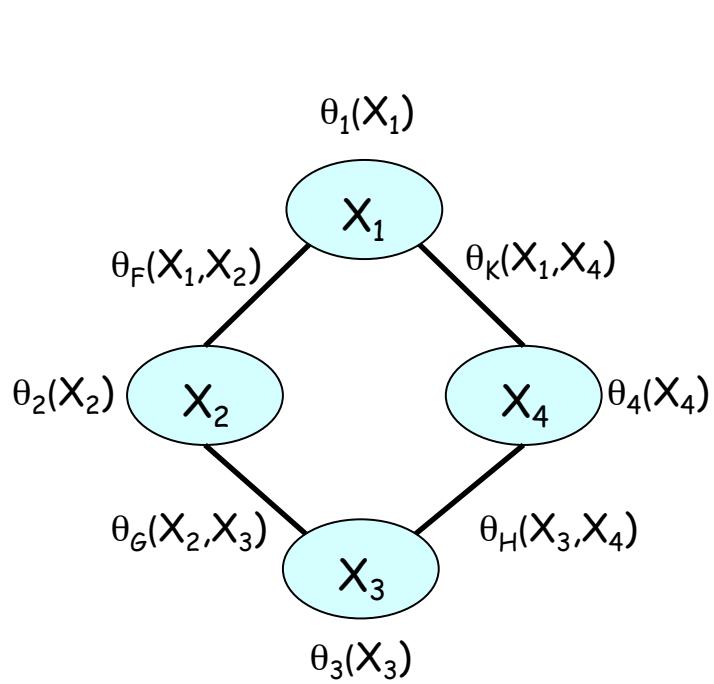
$\bar{\theta}_i^\lambda$ $\bar{\theta}_F^\lambda$

$L(\lambda)$ is upper bound on $\text{MAP}(\theta)$ for any setting of λ 's

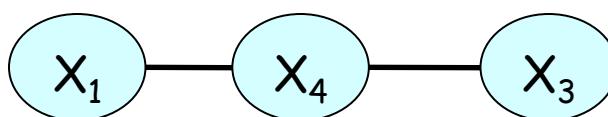


Divide and Conquer

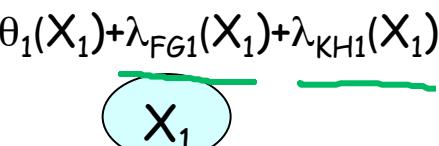
- Slaves don't have to be factors in original model
 - Subsets of factors that admit tractable solution to local maximization task



$$\begin{aligned} & \theta_F(X_1, X_2) + \theta_G(X_2, X_3) \\ & - \lambda_{FG1}(X_1) - \lambda_{FG2}(X_2) - \lambda_{FG3}(X_3) \end{aligned}$$



$$\begin{aligned} & \theta_K(X_1, X_4) + \theta_H(X_3, X_4) \\ & - \lambda_{KH1}(X_1) - \lambda_{KH3}(X_3) - \lambda_{KH4}(X_4) \end{aligned}$$



$$\theta_1(X_1) + \lambda_{FG1}(X_1) + \lambda_{KH1}(X_1)$$



$$\theta_3(X_3) + \lambda_{FG3}(X_3) + \lambda_{KH3}(X_3)$$



$$\theta_2(X_2) + \lambda_{FG2}(X_2)$$



$$\theta_4(X_4) + \lambda_{KH4}(X_4)$$



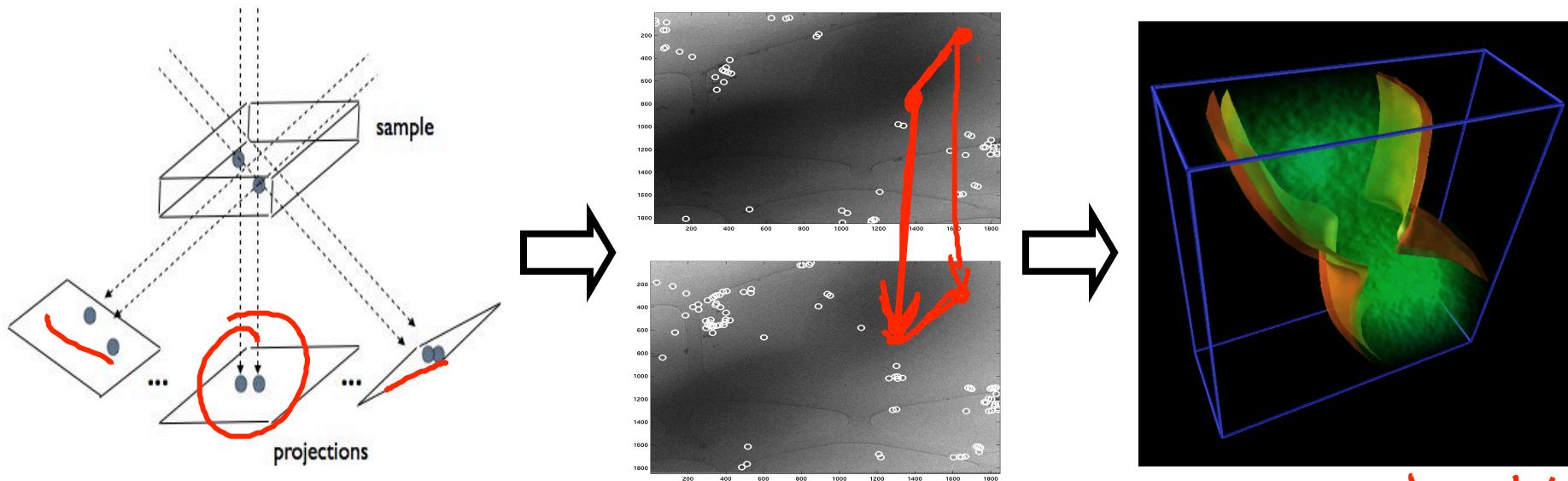
Divide and Conquer

- In pairwise networks, often divide factors into set of disjoint trees
 - Each edge factor assigned to exactly one tree
- Other tractable classes of factor sets
 - Matchings
 - Associative models
 - ...

Example: 3D Cell Reconstruction

correspond tilt
images

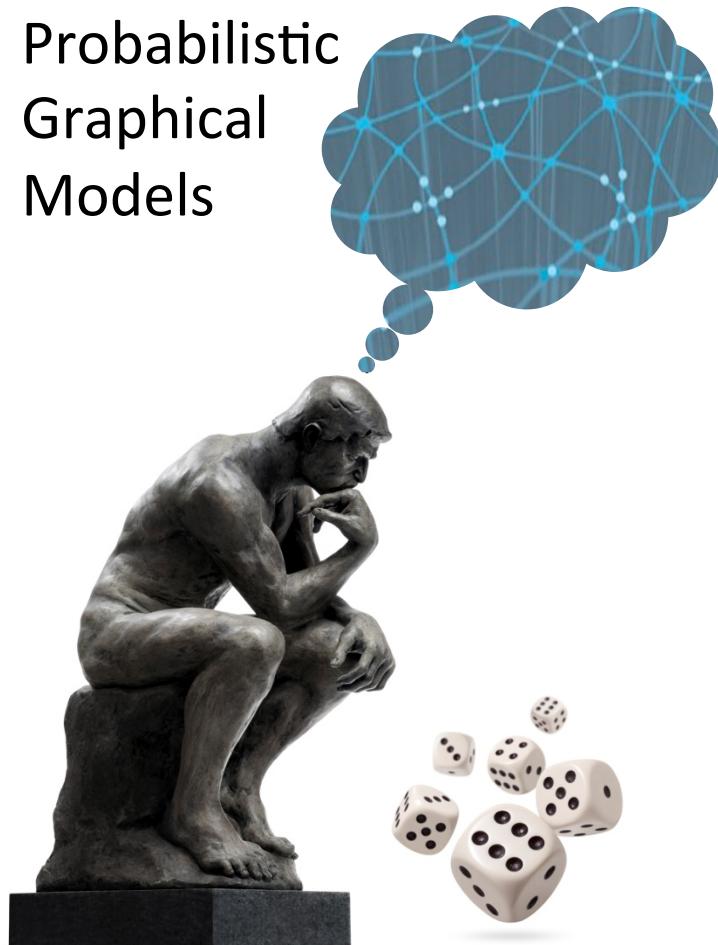
compute 3D
reconstruction



- Matching weights: similarity of location and local neighborhood appearance
- Pairwise potentials: approximate preservation of relative marker positions across images

Duchi, Tarlow, Elidan, and Koller, NIPS 2006. Amat, Moussavi, Comolli, Elidan, Downing, Horowitz, Journal of Structural Biology, 2006.

Probabilistic
Graphical
Models



Inference

MAP

Dual Decomposition Algorithm

Dual Decomposition Algorithm

$$\bar{\theta}_i^\lambda = \theta_i(x_i) + \sum_{F:i \in F} \lambda_{Fi}(x_i)$$

x_i $F:i \in F$

$$\bar{\theta}_F^\lambda = \theta_F(x_F) - \sum_{i \in F} \lambda_{Fi}(x_i)$$

F $i \in F$

- Initialize all λ 's to be 0

- Repeat for $t=1,2,\dots$

- Locally optimize all slaves:

- For all F and $i \in F$

$$x_F^* = \operatorname{argmax}_{x_F} \bar{\theta}_F^\lambda(x_F)$$

$$x_i^* = \operatorname{argmax}_{x_i} \bar{\theta}_i^\lambda(x_i)$$

- disagree* • If $x_{Fi}^* \neq x_i^*$ then

$$\alpha_t > 0$$

$$\lambda_{Fi}(x_i^*) := \lambda_{Fi}(x_i^*) - \alpha_t$$

$$\lambda_{Fi}(x_{Fi}^*) := \lambda_{Fi}(x_{Fi}^*) + \alpha_t$$

Dual Decomposition Convergence

- Under weak conditions on $\underline{\alpha}_+$, the λ 's are guaranteed to converge

$$-\sum_t \underline{\alpha}_+ = \underline{\infty}$$

$$-\sum_t \underline{\alpha}_+^2 < \infty$$

- Convergence is to a unique global optimum, regardless of initialization

At Convergence

- Each slave has a locally optimal solution over its own variables (in its scope)
- Solutions may not agree on shared variables
- If all slaves agree, the shared solution is a guaranteed MAP assignment
- Otherwise, we need to solve the decoding problem to construct a joint assignment

Options for Decoding x^*

- Several heuristics
 - If we use decomposition into spanning trees, can take MAP solution of any tree
 - Have each slave vote on X_i 's in its scope & for each X_i pick value with most votes
 - Weighted average of sequence of messages sent regarding each X_i
- Score θ is easy to evaluate for any x
- Best to generate many candidates and pick the one with highest score

Upper Bound

- $L(\lambda)$ is upper bound on $\text{MAP}(\theta)$

$$\underbrace{\text{score}(x)}_{\text{candidate}} \leq \text{MAP}(\theta) \leq \underline{L(\lambda)}$$

$$\underbrace{\text{MAP}(\theta) - \text{score}(x)}_{\text{small enough}} \leq \underbrace{L(\lambda) - \text{score}(x)}_{\text{small enough}}$$

Important Design Choices

- Division of problem into slaves
 - Larger slaves (with more factors) improve convergence and often quality of answers
- Selecting locally optimal solutions for slaves
 - Try to move toward faster agreement
- Adjusting the step size α_t
- Methods to construct candidate solutions

Summary: Algorithm

- Dual decomposition is a general-purpose algorithm for MAP inference
 - Divides model into tractable components
 - Solves each one locally
 - Passes “messages” to induce them to agree
- Any tractable MAP subclass can be used in this setting *as a slave*

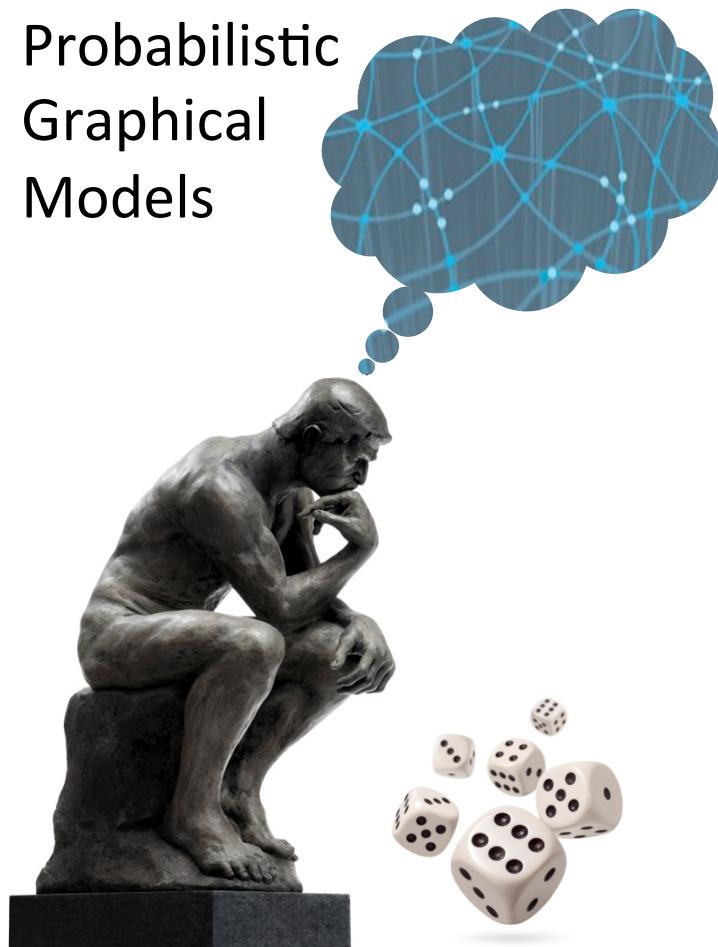
Summary: Theory

- Formally: a subgradient optimization algorithm on dual problem to MAP
- Provides important guarantees
 - Upper bound on distance to MAP
 - Conditions that guarantee exact MAP solution
- Even some analysis for which decomposition into slaves is better

Summary: Practice

- Pros:
 - Very general purpose
 - Best theoretical guarantees
 - Can use very fast, specialized MAP subroutines
for solving large model components
- Cons:
 - Not the fastest algorithm
 - Lots of tunable parameters / design choices

Probabilistic
Graphical
Models



Inference

Sampling Methods

Simple
Sampling

Sampling-Based Estimation

$\mathcal{D} = \{x[1], \dots, x[M]\}$ sampled IID from P
independent, identically distributed

If $P(X=1) = p = E_P[I_{X=1}]$

fraction of 1's



Estimator for p : $T_{\mathcal{D}} = \frac{1}{M} \sum_{m=1}^M I_{x[m]=1}$

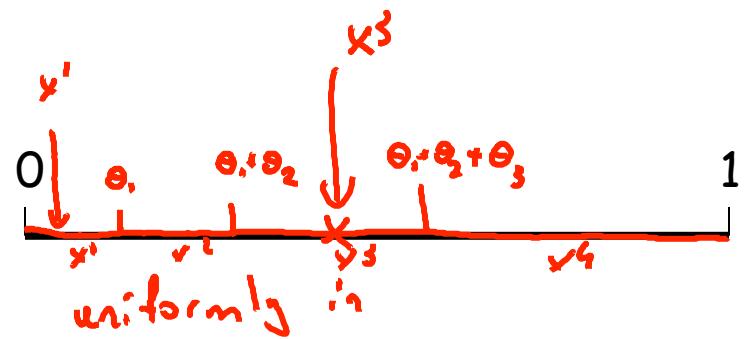
More generally, for any distribution P , function f :

$$E_P[f] \approx \frac{1}{M} \sum_{m=1}^M f(x[m])$$

indicator function f on samples
empirical expectation

Sampling from Discrete Distribution

$$\text{Val}(\underline{X}) = \{x^1, \dots, x^k\}$$
$$P(x^i) = \theta^i$$



Sampling-Based Estimation

Hoeffding Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p - \epsilon, p + \epsilon]) \leq 2e^{-2M\epsilon^2}$$

additive
samples
is ϵ -away from p
estimator
prob of sample set
a bad sample set

$$T_{\mathcal{D}} = \frac{1}{M} \sum_{m=1}^M X[m]$$

Chernoff Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p(1 - \epsilon), p(1 + \epsilon)]) \leq 2e^{-Mpe^2/3}$$

multiplication

Sampling-Based Estimation

Hoeffding Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p - \epsilon, p + \epsilon]) \leq 2e^{-2M\epsilon^2} < \delta$$

$$T_{\mathcal{D}} = \frac{1}{M} \sum_{m=1}^M X[m]$$

For additive bound ϵ on error with probability $> 1-\delta$:

$$M \geq \frac{\ln(2/\delta)}{2\epsilon^2}$$

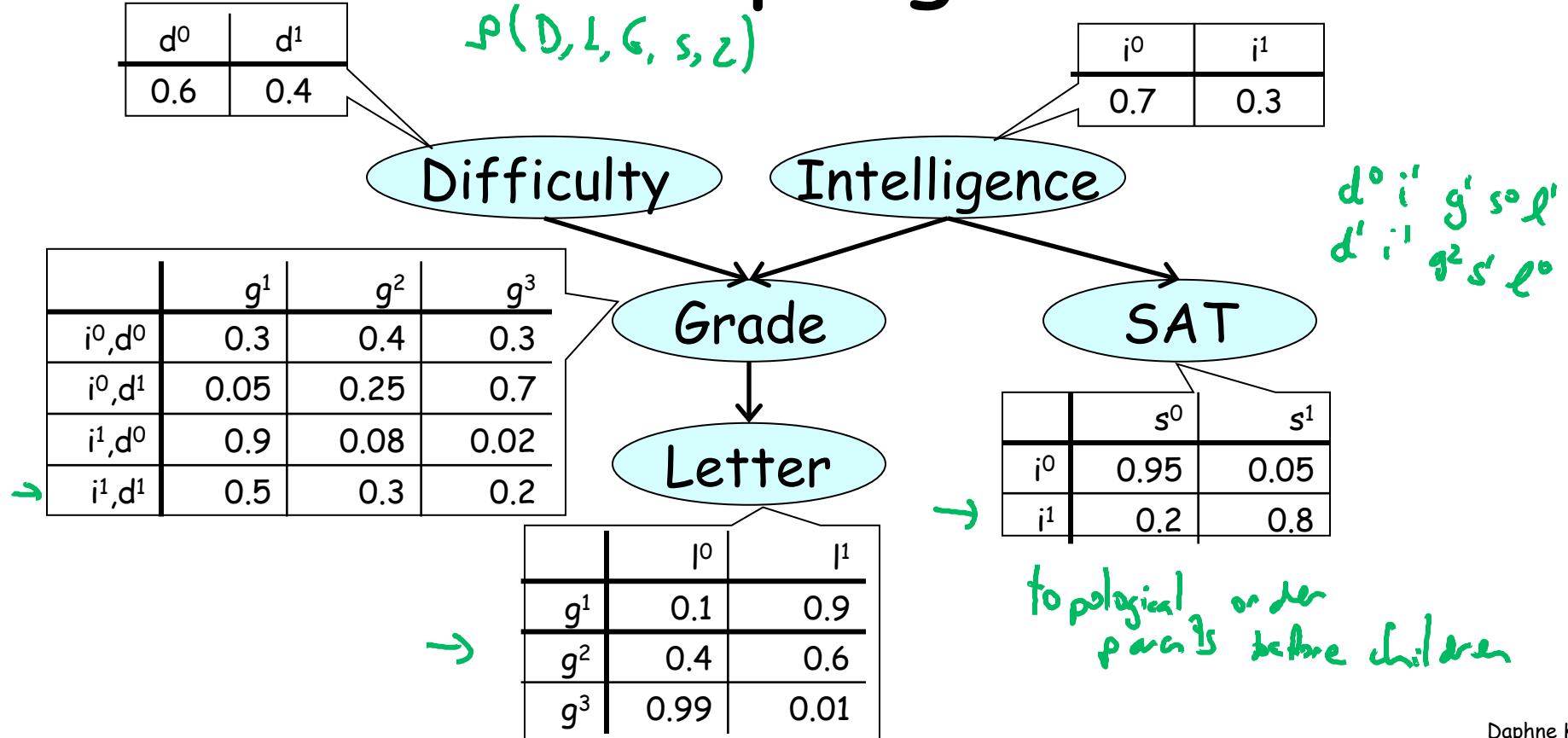
Chernoff Bound:

$$P_{\mathcal{D}}(T_{\mathcal{D}} \notin [p(1 - \epsilon), p(1 + \epsilon)]) \leq 2e^{-Mp\epsilon^2/3}$$

For multiplicative bound ϵ on error with probability $> 1-\delta$:

$$M \geq 3 \frac{\ln(2/\delta)}{p\epsilon^2}$$

Forward sampling from a BN



Forward Sampling for Querying

- Goal: Estimate $P(Y=y)$
 - Generate samples from BN
 - Compute fraction where $Y=y$

For additive bound ϵ on error with probability $> 1-\delta$: $M \geq \frac{\ln(2/\delta)}{2\epsilon^2}$

For multiplicative bound ϵ on error with probability $> 1-\delta$: $M \geq 3 \frac{\ln(2/\delta)}{P(y)\epsilon^2}$

Queries with Evidence

- Goal: Estimate $P(\underline{Y=y} \mid \underline{E=e})$
 - Rejection sampling algorithm
 - Generate samples from BN
 - Throw away all those where $\underline{E \neq e}$
 - Compute fraction where $\underline{Y=y}$
- remaining samples
are sampled
from P(Y=y | E=e)*

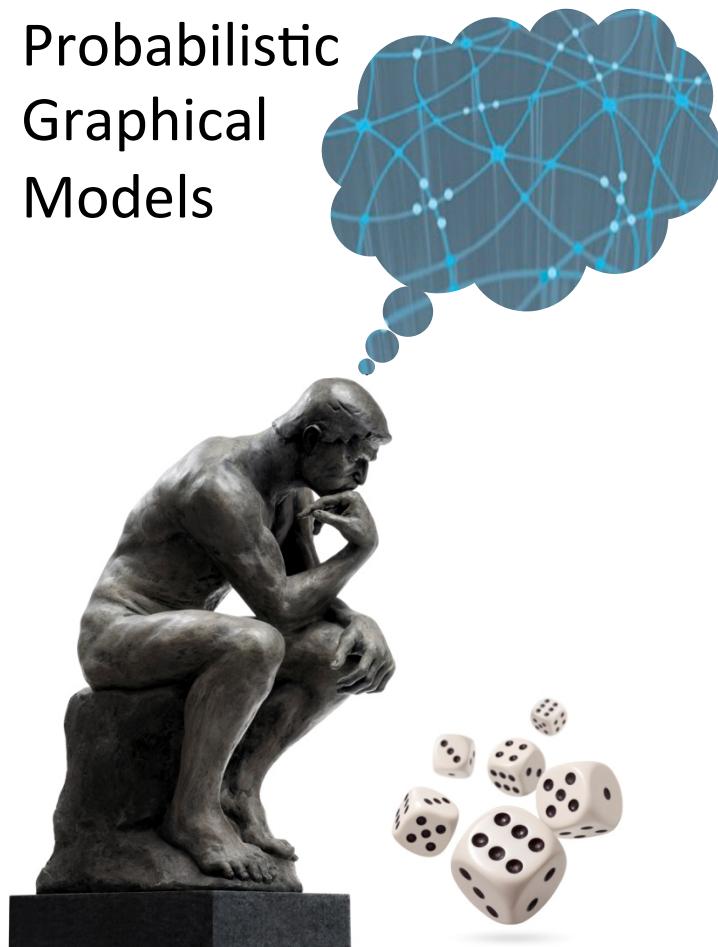
Expected fraction of samples kept $\sim P(e)$

samples needed grows exponentially
with # of observed variables

Summary

- Generating samples from a BN is easy
- (ε, δ) -bounds exist, but usefulness is limited:
 - Additive bounds: useless for low probability events
 - Multiplicative bounds: # samples grows as $1/P(y)$
- With evidence, # of required samples grows exponentially with # of observed variables
- Forward sampling generally infeasible for MNs

Probabilistic
Graphical
Models

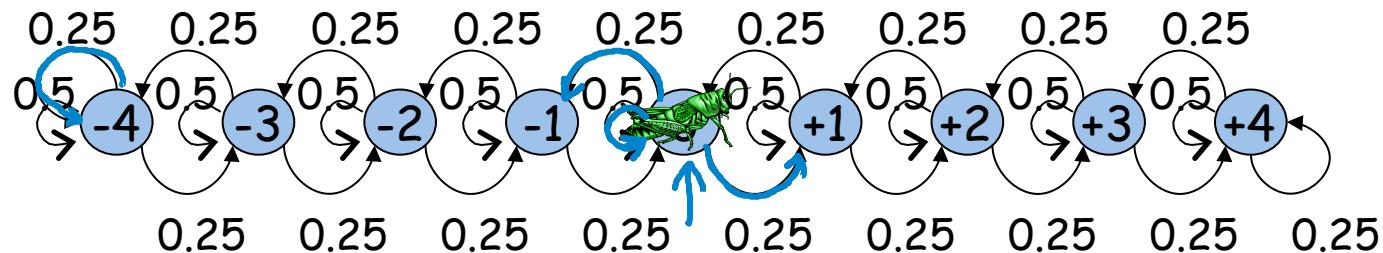


Inference

Sampling Methods

Markov Chain
Monte Carlo

Markov Chain

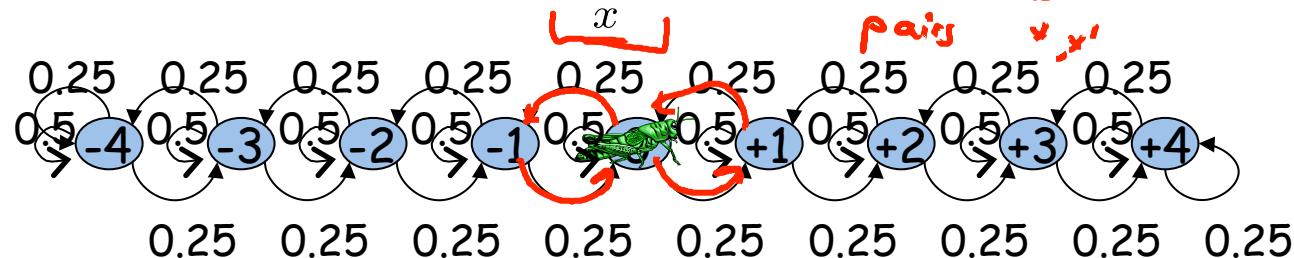


- A Markov chain defines a probabilistic transition model $T(x \rightarrow x')$ over states x :
 - for all x :

$$\sum_{x'} T(x \rightarrow x') = 1$$

Temporal Dynamics

$$P^{(t+1)}(X^{(t+1)} = x') = \sum P^{(t)}(X^{(t)} = x) T(x \rightarrow x')$$

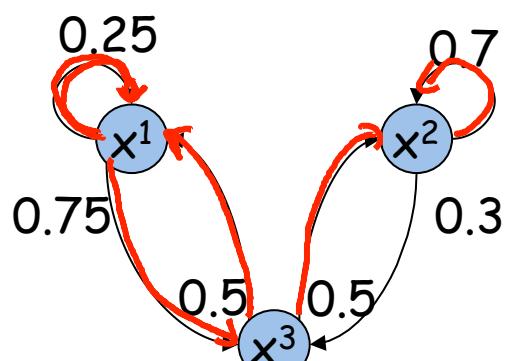


	-2	-1	0	+1	+2
$P^{(0)}$	0	0	1	0	0
$P^{(1)}$	0	.25	.5	.25	0
$P^{(2)}$	<u>$.25^2 =$</u> .0625	$2 \times (.5 \times .25) = .25$	<u>$.5^2 + 2 \times .25^2 = .375$</u>	$2 \times (.5 \times .25) = .25$	<u>$.25^2 =$</u> .0625

Stationary Distribution

$$P^{(t)}(x') \approx P^{(t+1)}(x') = \sum_x P^{(t)}(x) T(x \rightarrow x')$$

$$\pi(x') = \sum_x \pi(x) T(x \rightarrow x')$$



$$\underline{\pi(x^1)} = \underline{0.25\pi(x^1)} + \underline{0.5\pi(x^3)}$$

$$\underline{\pi(x^2)} = \underline{0.7\pi(x^2)} + \underline{0.5\pi(x^3)}$$

$$\underline{\pi(x^3)} = \underline{0.75\pi(x^1)} + \underline{0.3\pi(x^2)}$$

$$\pi(x^1) = 0.2$$

$$\pi(x^2) = 0.5$$

$$\pi(x^3) = 0.3$$

$$\pi(x^1) + \pi(x^2) + \pi(x^3) = 1$$

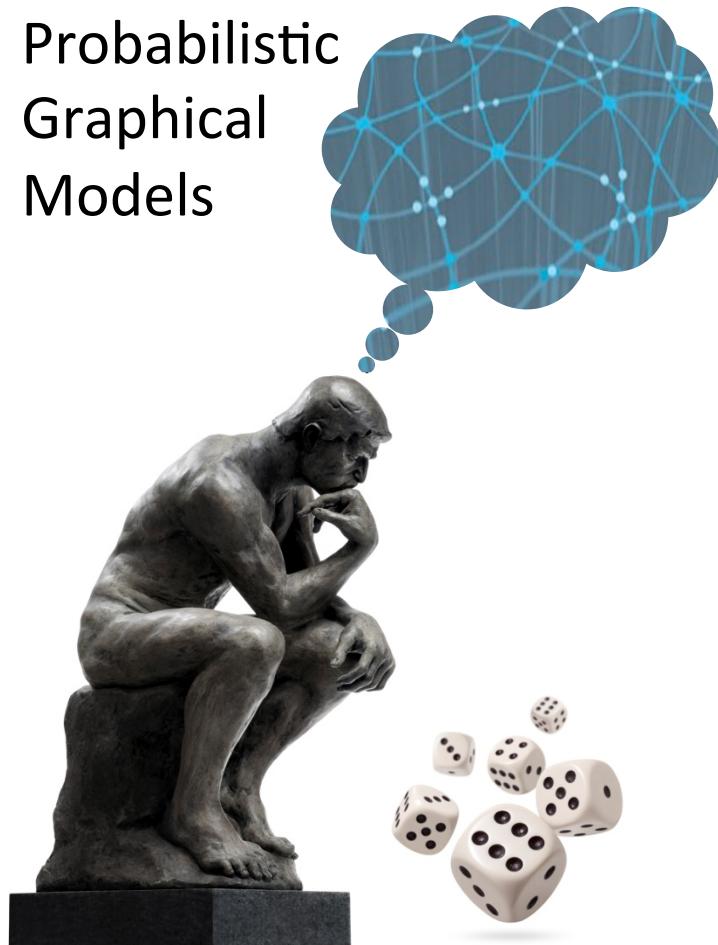
Regular Markov Chains

- A Markov chain is regular if there exists k such that, for every x, x' , the probability of getting from x to x' in exactly k steps is > 0
- Theorem: A regular Markov chain converges to a unique stationary distribution regardless of start state

Regular Markov Chains

- A Markov chain is regular if there exists k such that, for every x, x' , the probability of getting from x to x' in exactly k steps is > 0
 \leftarrow distance between furthest x, x'
- Sufficient conditions for regularity:
 - Every two states ^{x, y'} are connected with path of prob > 0
 - For every state, there is a self-transition

Probabilistic
Graphical
Models



Inference

Sampling Methods

Using a
Markov Chain

Using a Markov Chain

- Goal: compute $P(x \in S)$
 - but P is too hard to sample from directly
- Construct a Markov chain T whose unique stationary distribution is P
- Sample $\underline{x^{(0)}}$ from some $P^{(0)}$
- For $t = 0, 1, 2, \dots$
 - Generate $x^{(t+1)}$ from $T(x^{(t)} \rightarrow x')$

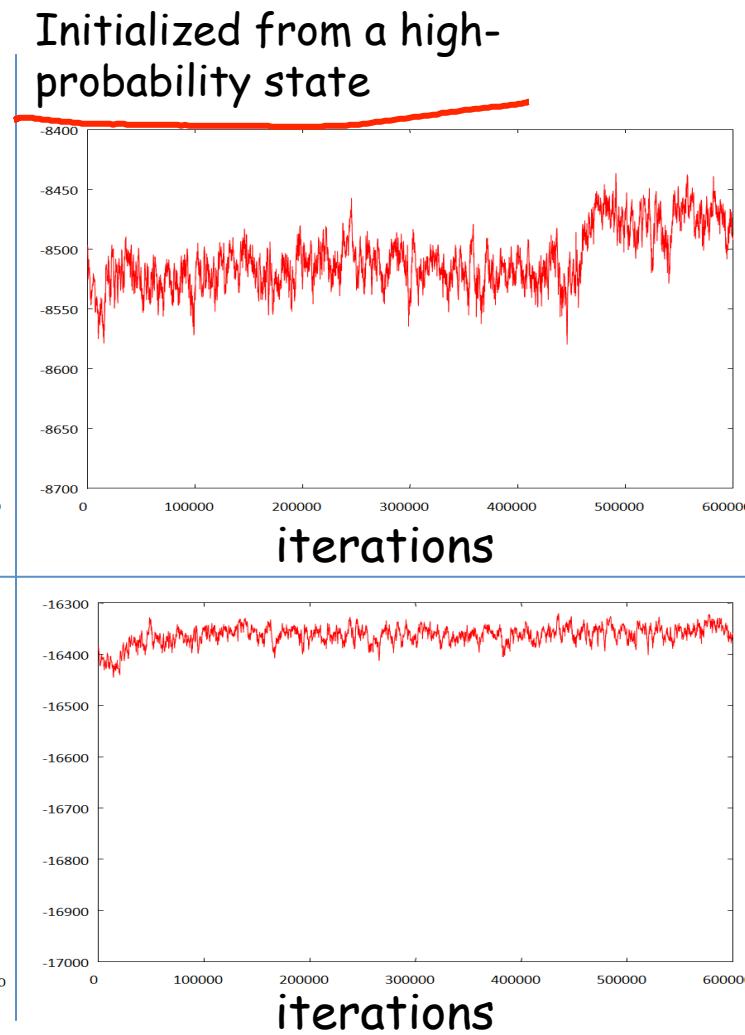
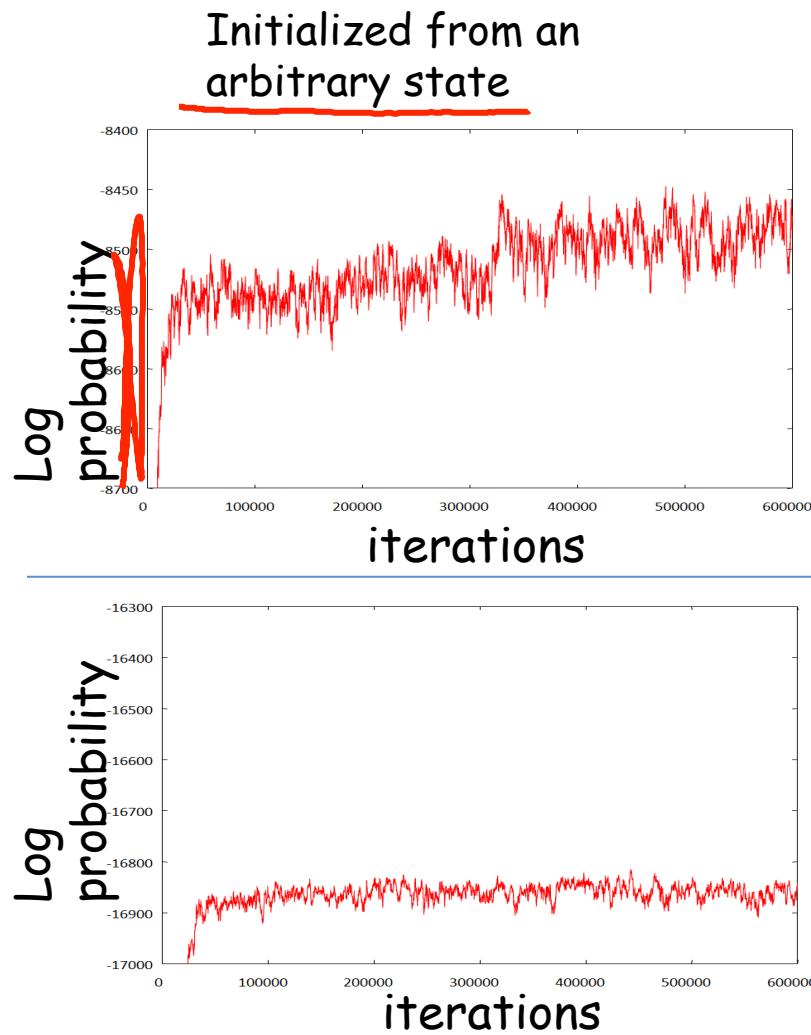
Using a Markov Chain

- We only want to use samples that are sampled from a distribution close to P
- At early iterations, $P^{(t)}$ is usually far from P
- Start collecting samples only after the chain has run long enough to "mix" $P^{(t)}$ close enough to P

Mixing

- How do you know if a chain has mixed or not?
 - In general, you can never “prove” a chain has mixed
 - But in many cases you can show that it has NOT
- How do you know a chain has not mixed?
 - Compare chain statistics in different windows within a single run of the chain
 - and across different runs initialized differently





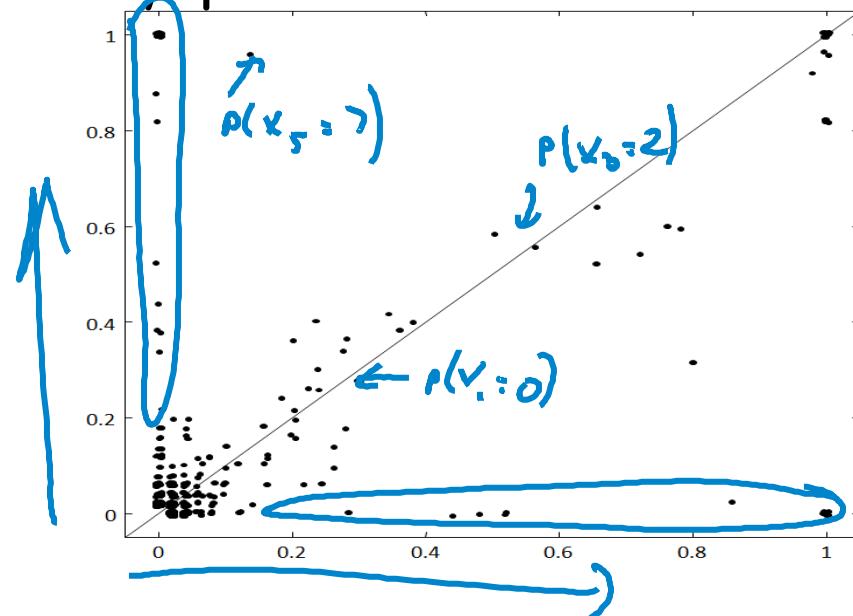
Mixing?

Maybe

NO

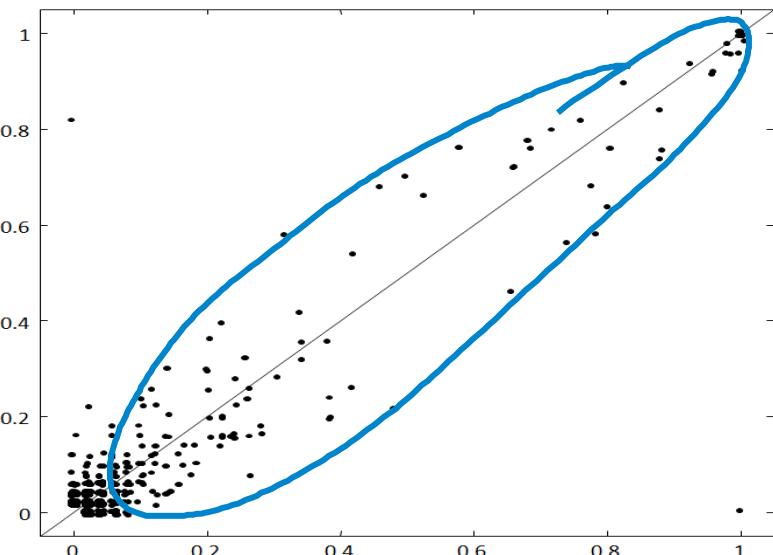


- Each dot is a statistic (e.g., $P(x \in S)$)
- x -position is its estimated value from chain 1
- y -position is its estimated value from chain 2



Mixing?

NO



Maybe

Using the Samples

- Once the chain mixes, all samples $x^{(t)}$ are from the stationary distribution π
 - So we can (and should) use all $x^{(t)}$ for $t > T_{\text{mix}}$
- However, nearby samples are correlated!
 - So we shouldn't overestimate the quality of our estimate by simply counting samples not IID
- The faster a chain mixes, the less correlated (more useful) the samples

MCMC Algorithm Summary I

- For $c=1, \dots, C$
 - Sample $x^{(c,0)}$ from $P^{(0)}$
- Repeat until mixing
 - For $c=1, \dots, C$
 - Generate $\underline{x^{(c,t+1)}}$ from $T(\underline{x^{(c,t)}} \rightarrow x')$
 - Compare window statistics in different chains to determine mixing
 - $t := t+1$

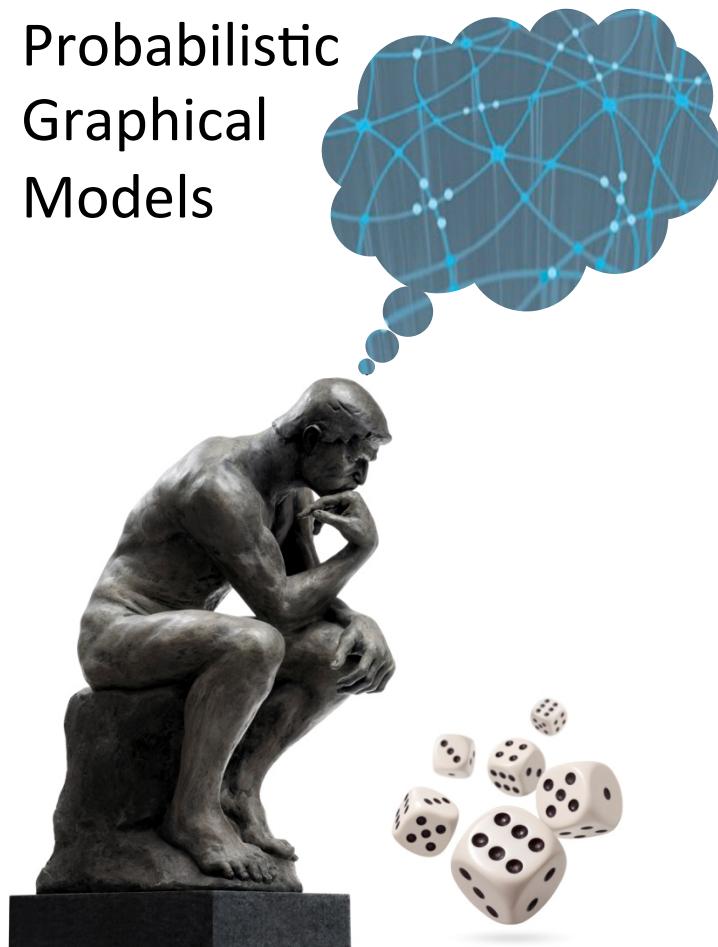
MCMC Algorithm Summary II

- Repeat until sufficient samples
 - $D := \emptyset$
 - For $c=1, \dots, C$
 - Generate $x^{(c, t+1)}$ from $T(x^{(c, t)} \rightarrow x')$
 - $D := D \cup \{x^{(c, t+1)}\}$
 - $t := t+1$
- Let $D = \{x[1], \dots, x[M]\}$
- Estimate $E_P[f] \approx \frac{1}{M} \sum_{m=1}^M f(x[m])$

Summary

- Pros:
 - Very general purpose
 - Often easy to implement
 - Good theoretical guarantees as $t \rightarrow \infty$
- Cons:
 - Lots of tunable parameters / design choices
 - Can be quite slow to converge
 - Difficult to tell whether it's working

Probabilistic
Graphical
Models



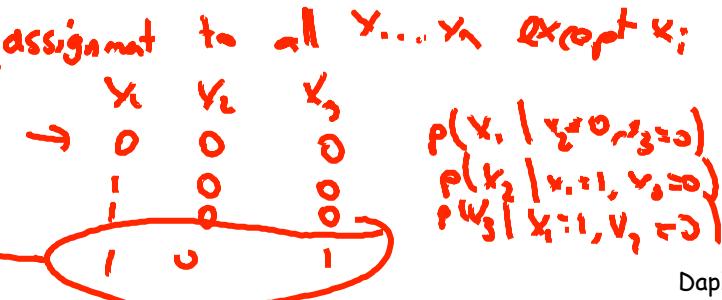
Inference

Sampling Methods

MCMC for
PGMs: The
Gibbs Chain

Gibbs Chain

- Target distribution $P_\Phi(X_1, \dots, X_n)$
- Markov chain state space: complete assignments x to X = $\{X_1, \dots, X_n\}$
- Transition model given starting state x :
 - For $i=1, \dots, n$
 - Sample $x_i \sim P_\Phi(X_i | x_{-i})$ assignment to all $x_1 \dots x_n$ except x_i
 - Set $x' = x$ ←



Daphne Koller

$$P(D | i^0, g^*, l^0, s^*)$$

	d^0	d^1
i^0	0.6	0.4
i^1		

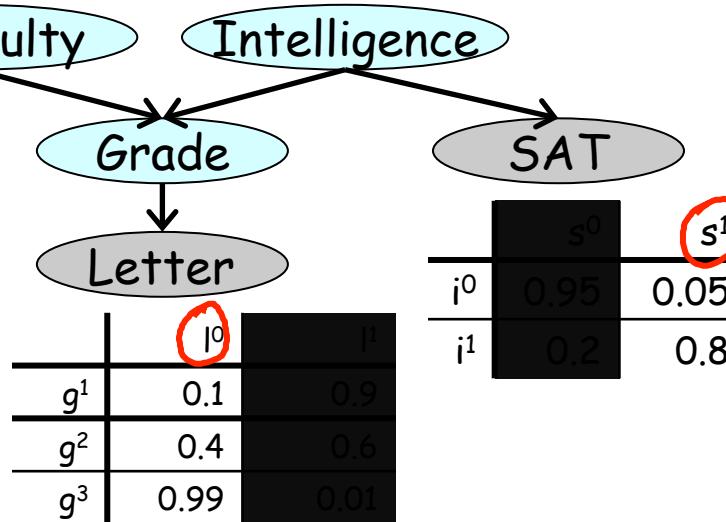
	g^1	g^2	g^3
i^0, d^0	0.3	0.4	0.3
i^0, d^1	0.05	0.25	0.7
i^1, d^0	0.9	0.08	0.02
i^1, d^1	0.5	0.3	0.2

$$P(G | d^1, i^1, l^0, s^*)$$

Example

$$P(i^1 | d^1, g^*, l^0, s^*)$$

	i^0	i^1
i^0	0.7	0.3
i^1		



$$P_i(s, l, G | s, x^0)$$

$$\begin{array}{l} d^0 \ i^0 \ g^* \\ d^1 \ i^0 \ g^* \\ d^1 \ i^1 \ g^* \\ \curvearrowleft d^1 \ i^1 \ g^* \end{array}$$

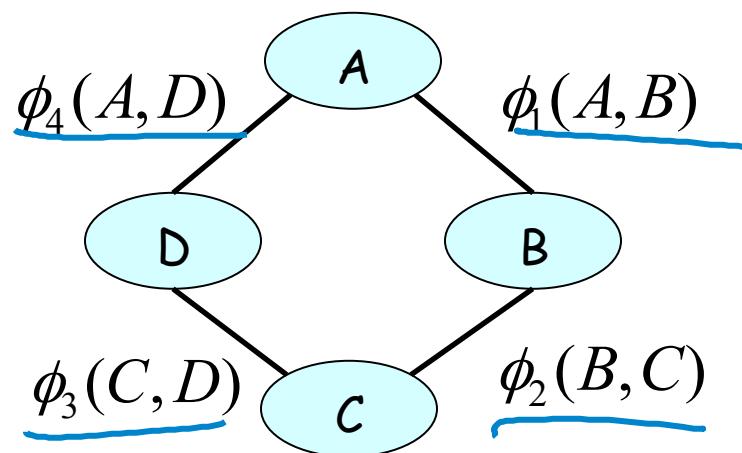
Computational Cost

- For $i=1, \dots, n$
 - Sample $x_i \sim P_\Phi(X_i | x_{-i})$

$$\underline{P_\Phi(X_i | x_{-i})} = \frac{P_\Phi(\underline{X_i}, \underline{x_{-i}})}{P_\Phi(\underline{x_{-i}})} = \frac{\cancel{\tilde{P}_\Phi(X_i, x_{-i})}}{\cancel{\tilde{P}_\Phi(x_{-i})}}$$

complete assignment
product at factors

Another Example



$$P_{\Phi}(A | b, c, d) = \frac{\tilde{P}_{\Phi}(a, b, c, d)}{\sum_{A'} \tilde{P}_{\Phi}(A', b, c, d)}$$

$$\frac{\phi_1(A, b)\phi_2(b, c)\phi_3(c, d)\phi_4(A, d)}{\sum_{A'} \phi_1(A', b)\phi_2(b, c)\phi_3(c, d)\phi_4(A', d)}$$

normalizing constant
 $\propto \phi_1(A, b)\phi_4(A, d)$

factors that involve A

Computational Cost Revisited

- For $i=1, \dots, n$
 - Sample $x_i \sim P_\Phi(X_i | \mathbf{x}_{-i})$

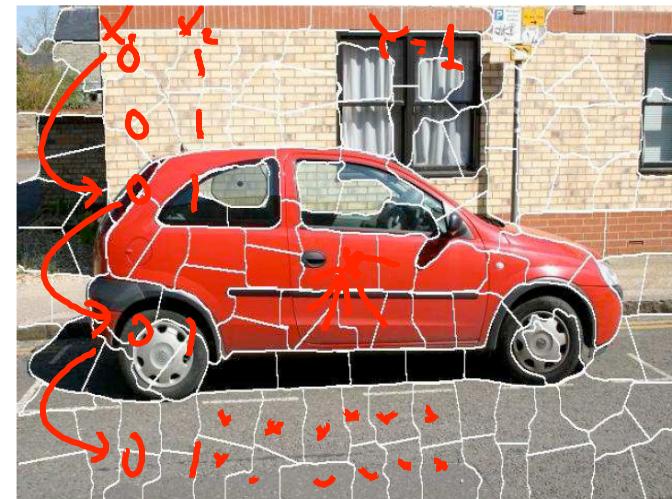
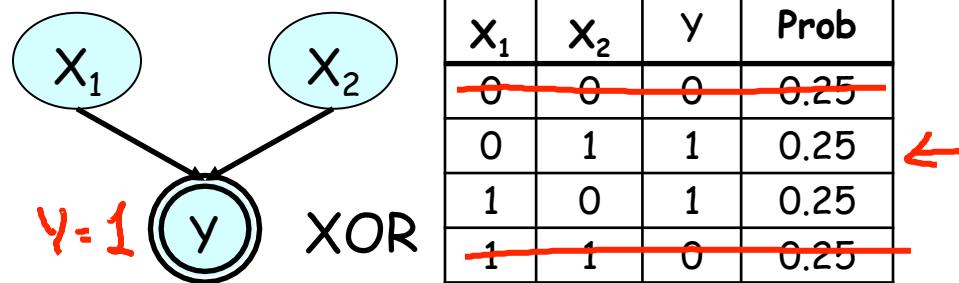
$$P_\Phi(\underline{X_i} | \underline{\mathbf{x}_{-i}}) = \frac{P_\Phi(X_i, \mathbf{x}_{-i})}{P_\Phi(\mathbf{x}_{-i})} = \frac{\tilde{P}_\Phi(X_i, \mathbf{x}_{-i})}{\tilde{P}_\Phi(\mathbf{x}_{-i})}$$

only x_i and
its neighbors

$$\propto \prod_{j: X_i \in \text{Scope}[C_j]} \phi_j(X_i, \mathbf{x}_{j,-i})$$

factored + hat
involve x_i

Gibbs Chain and Regularity

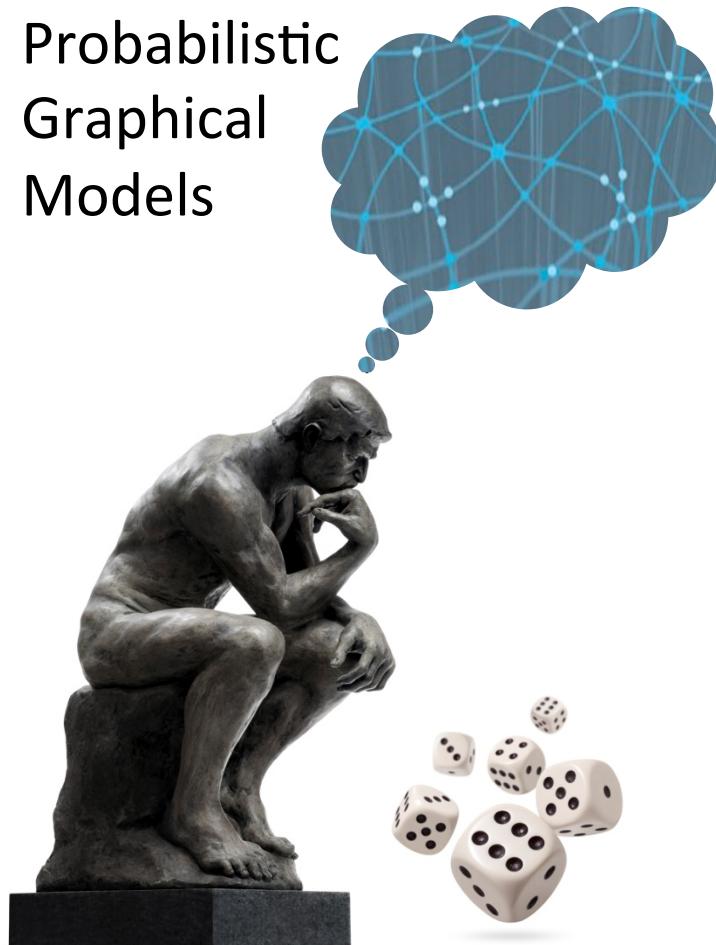


- If all factors are positive, Gibbs chain is regular
- However, mixing can still be very slow

Summary

- Converts the hard problem of inference to a sequence of "easy" sampling steps
- Pros:
 - Probably the simplest Markov chain for PGMs
 - Computationally efficient to sample
- Cons:
 - Often slow to mix, esp. when probabilities are peaked
 - Only applies if we can sample from product of factors

Probabilistic
Graphical
Models

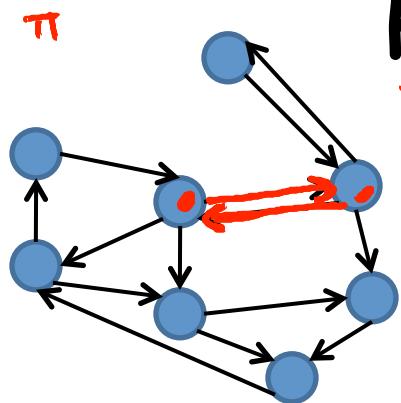


Inference

Sampling Methods

Metropolis-
Hastings
Algorithm

Reversible Chains



$$\boxed{\pi(x)T(x \rightarrow x')} = \boxed{\pi(x')T(x' \rightarrow x)}$$

detailed balance

Theorem: If detailed balance holds, and T is regular, then T has a unique stationary distribution $\underline{\pi}$

Proof:

$$\sum_x \pi(x)T(x \rightarrow x') = \sum_x \pi(x')T(x' \rightarrow x) = \underline{\pi(x)} \cdot \underbrace{\sum_x T(x \rightarrow x')}_1$$

definition of π

Metropolis Hastings Chain

Proposal distribution $Q(x \rightarrow x')$



Acceptance probability: $A(x \rightarrow x')$

- At each state x , sample x' from $Q(x \rightarrow x')$
- Accept proposal with probability $A(x \rightarrow x')$
 - If proposal accepted, move to x'
 - Otherwise stay at x

$$T(x \rightarrow x') = Q(x \rightarrow x') A(x \rightarrow x') \quad \text{if } x' \neq x$$

$$T(x \rightarrow x) = Q(x \rightarrow x) + \sum_{x' \neq x} Q(x \rightarrow x') (1 - A(x \rightarrow x'))$$

Acceptance Probability

$$\underline{\pi(x)T(x \rightarrow x')} = \underline{\pi(x')T(x' \rightarrow x)}$$

construct A s.t. \leftarrow holds for Q, π

$$x \neq x' \quad \pi(x)Q(x \rightarrow x')\underline{A(x \rightarrow x')} = \pi(x')Q(x' \rightarrow x)\underline{A(x' \rightarrow x)}$$

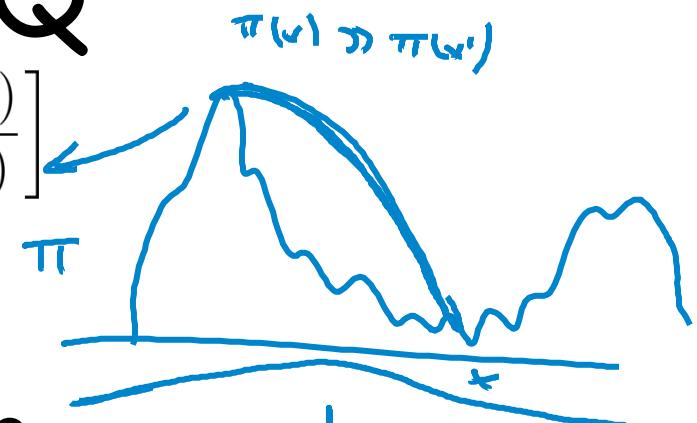
$$\begin{aligned} A(x \rightarrow x') &= p \\ A(x' \rightarrow x) &= 1 \end{aligned}$$

$$\frac{A(x \rightarrow x')}{A(x' \rightarrow x)} = \boxed{\frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} = p^{-1}}$$

$$\boxed{A(x \rightarrow x') = \min \left[1, \frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} \right]}$$

Choice of Q

$$A(x \rightarrow x') = \min \left[1, \frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} \right]$$



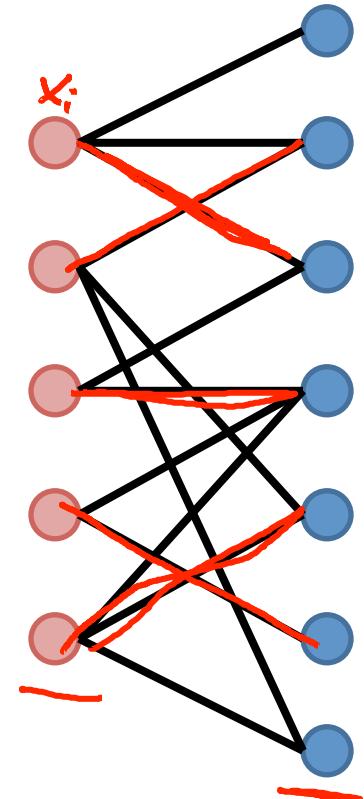
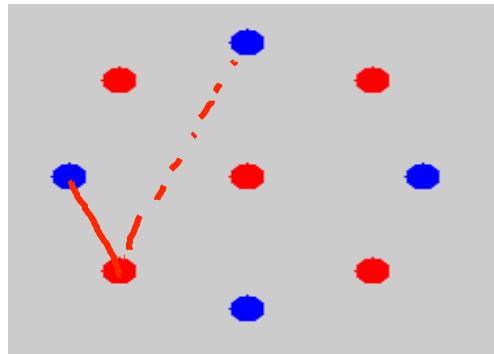
- Q must be reversible:
 - $Q(x \rightarrow x') > 0 \Rightarrow Q(x' \rightarrow x) > 0$
- Opposing forces
 - Q should try to spread out, to improve mixing
 - But then acceptance probability often low

MCMC for Matching

$X_i = j$ if i matched to j

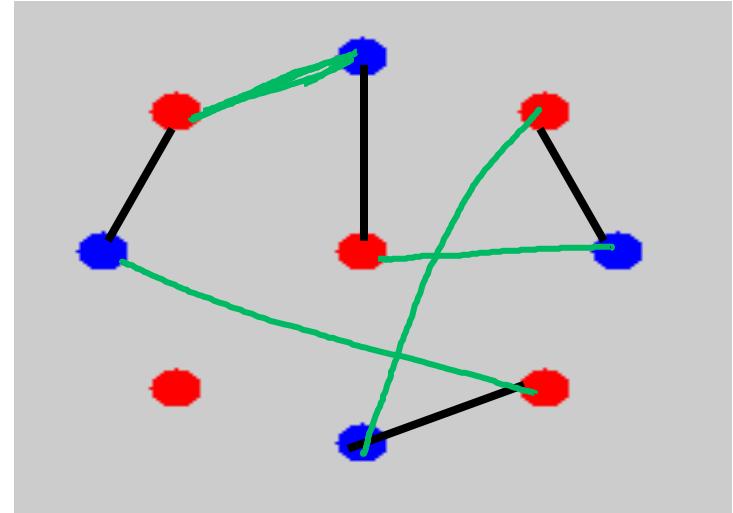
$$P(X_1 = v_1, \dots, X_4 = v_4) \propto$$

$$\begin{cases} \exp\left(-\sum_i \text{dist}(i, v_i)\right) & \text{if every } X_i \text{ has different value} \\ 0 & \text{otherwise} \end{cases}$$



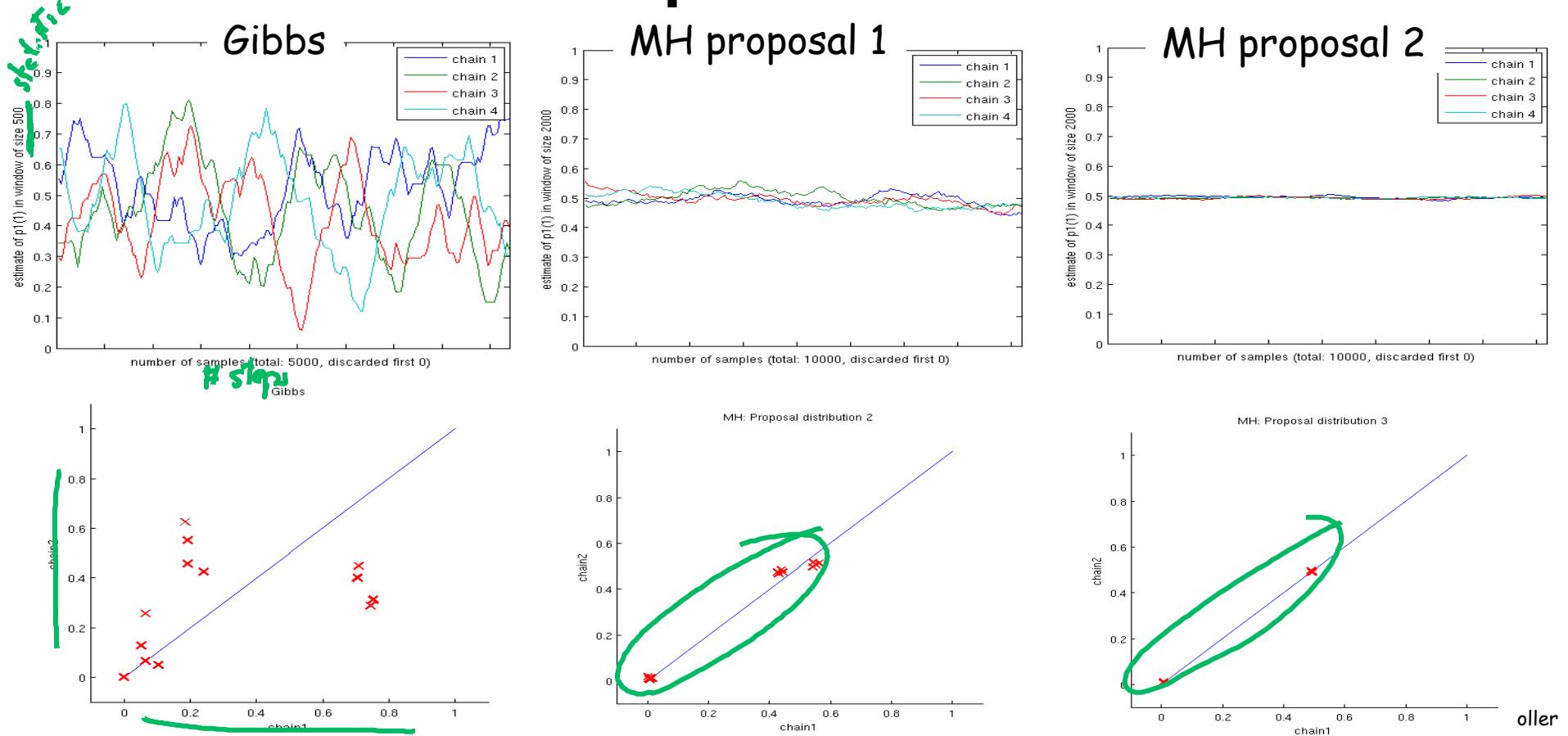
Daphne Koller

MH for Matching: Augmenting Path



- 1) randomly pick one variable X_i
 - 2) sample X_i , pretending that all values are available
 - 3) pick the variable whose assignment was taken (conflict), and return to step 2
- When step 2 creates no conflict, modify assignment to flip augmenting path

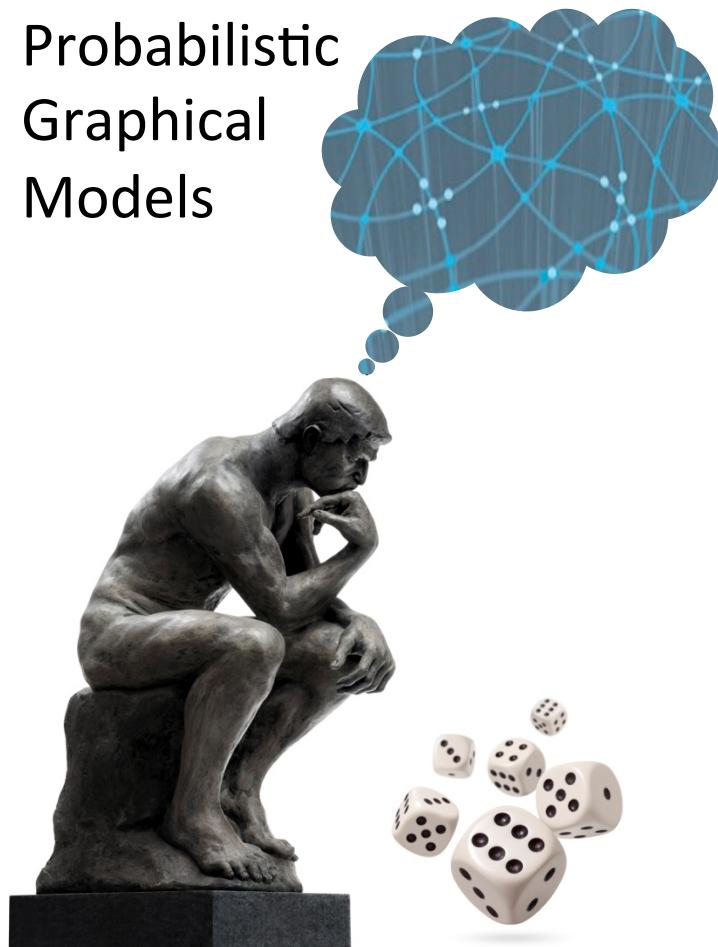
Example Results



Summary

- MH is a general framework for building Markov chains with a particular stationary distribution
 - Requires a proposal distribution
 - Acceptance computed via detailed balance
- Tremendous flexibility in designing proposal distributions that explore the space quickly
 - But proposal distribution makes a big difference
 - and finding a good one is not always easy

Probabilistic
Graphical
Models

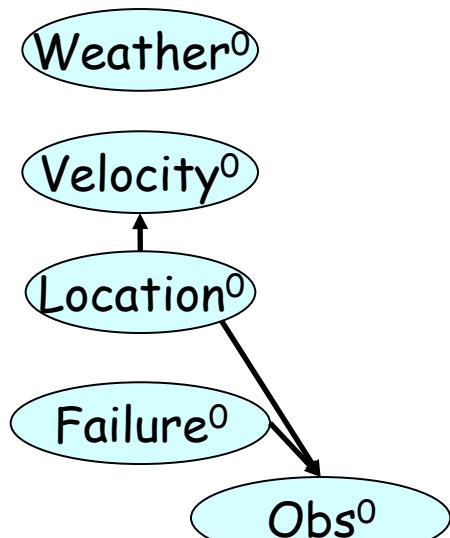


Inference

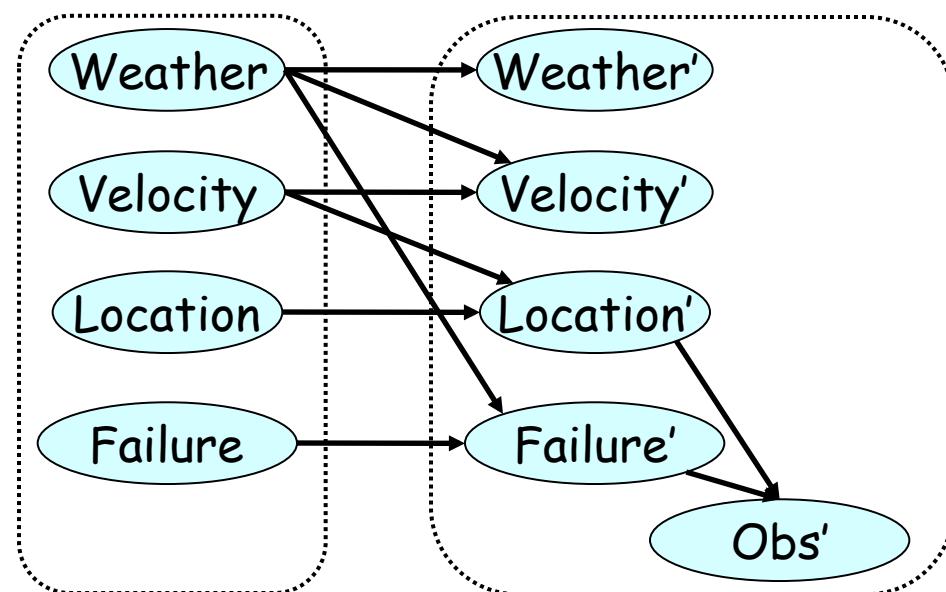
Sampling Methods

Inference In
Template
Models

DBN Template Specification



Time slice 0

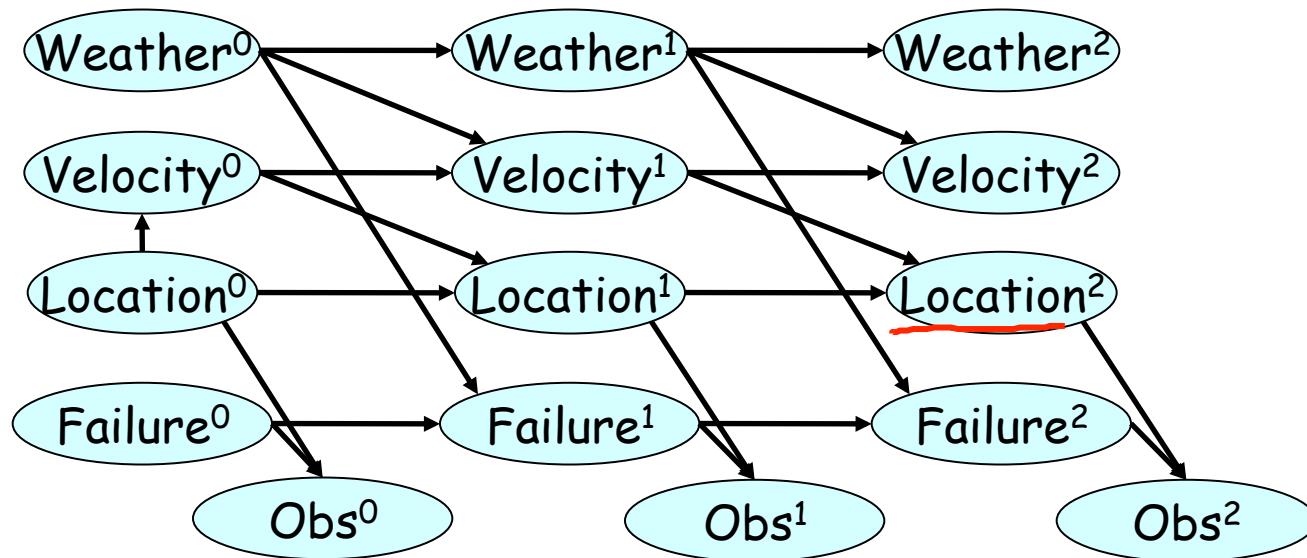


Time slice t

Time slice t+1

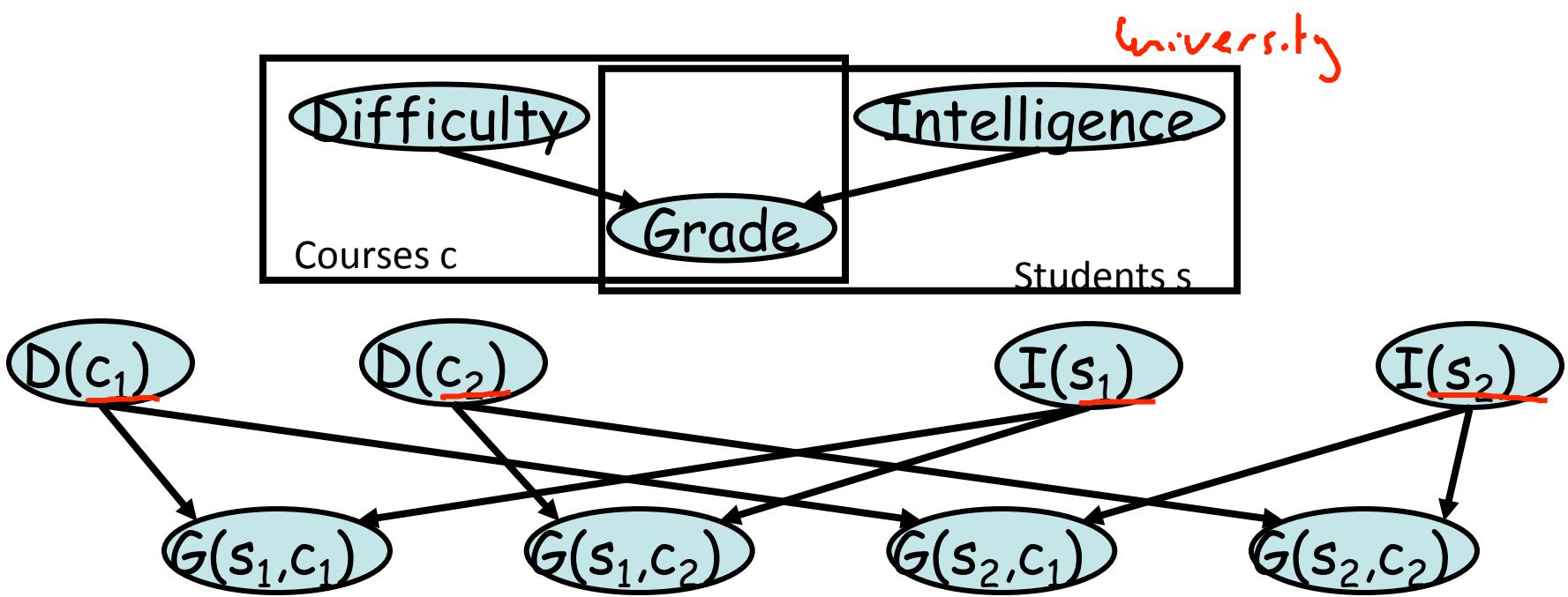
2 TBAI

Ground Bayesian Network



Can unroll DBN for given trajectory
and run inference over ground network

Plate Model



Can unroll plate model for given set of objects
and run inference over ground network

Belief State Tracking

o^1, o^2, \dots, o^t

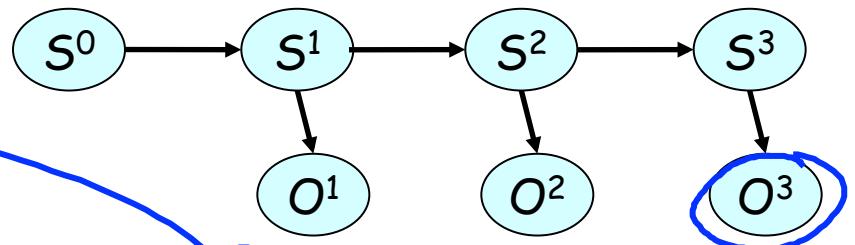
$$\begin{aligned} \underline{\sigma^{(t)}(S^{(t)})} &= P(S^{(t)} | \underline{o^{(1:t)}}) \\ \underline{\sigma^{(t+1)}(S^{(t+1)})} &\triangleq P(S^{(t+1)} | \underline{o^{(1:t)}}) \\ &= \sum_{\underline{S^{(t)}}} P(S^{(t+1)} | \underline{S^{(t)}}, \underline{o^{(1:t)}}) P(S^{(t)} | \underline{o^{(1:t)}}) \\ &= \sum_{S^{(t)}} P(S^{(t+1)} | S^{(t)}) \underline{\sigma^{(t)}(S^{(t)})} \end{aligned}$$

transition model

Belief State Tracking

$$\sigma^{(t)}(S^{(t)}) = P(S^{(t)} | o^{(1:t)})$$

$$\underline{\sigma^{(\cdot t+1)}(S^{(t+1)})} \stackrel{\Delta}{=} P(S^{(t+1)} | o^{(1:t)})$$



$$\underline{\sigma^{(t+1)}(S^{(t+1)})} = P(S^{(t+1)} | o^{(1:t)}, o^{(t+1)})$$

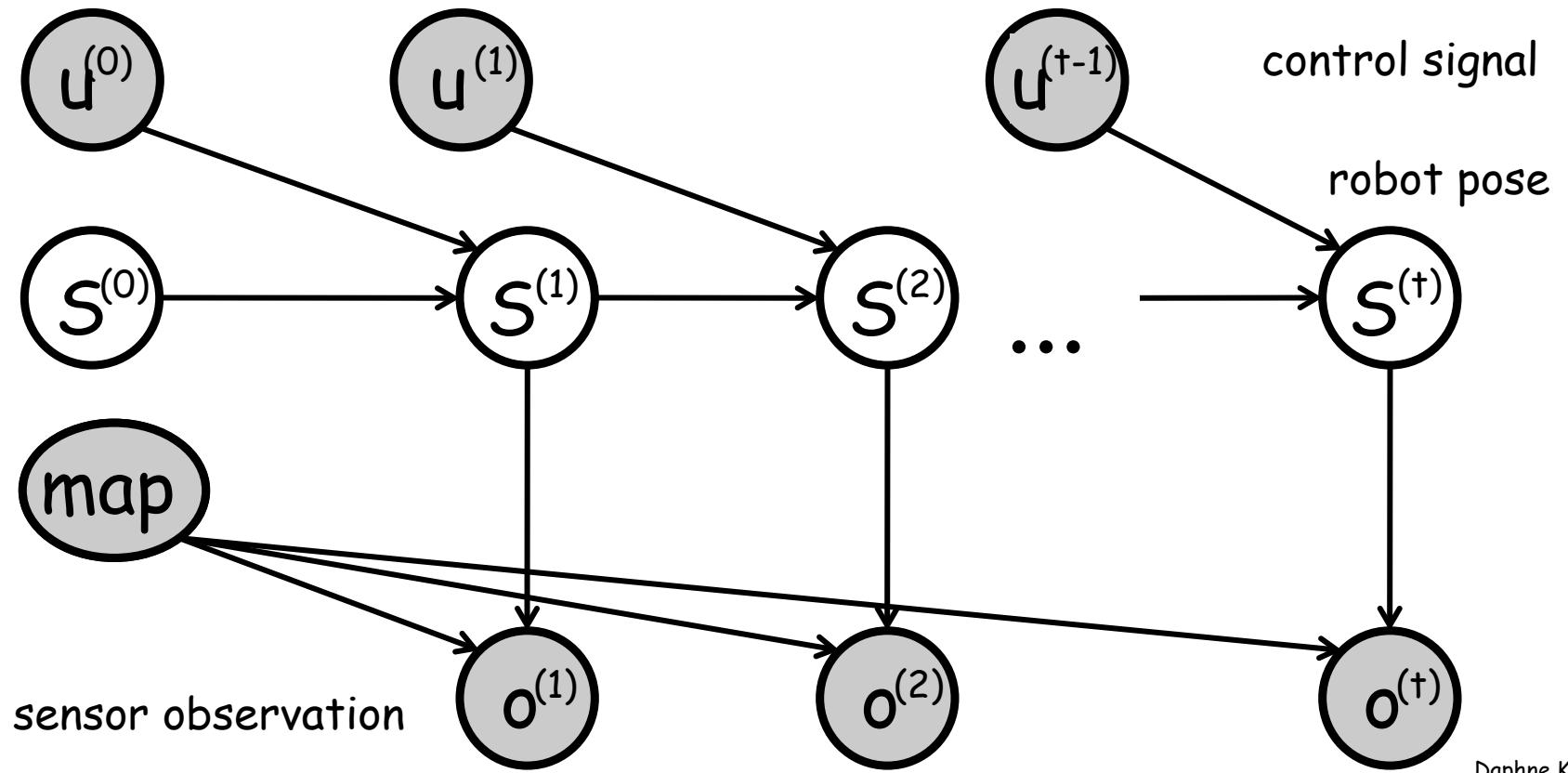
$$\rightarrow \frac{P(o^{(t+1)} | S^{(t+1)}, o^{(1:t)})}{\cancel{P(S^{(t+1)} | o^{(1:t)})}} P(S^{(t+1)} | o^{(1:t)})$$

observation $\underline{P(o^{(t+1)} | o^{(1:t)})}$

$$= \frac{P(o^{(t+1)} | S^{(t+1)}) \underline{\sigma^{(\cdot t+1)}(S^{(t+1)})}}{\underline{P(o^{(t+1)} | o^{(1:t)})}}$$

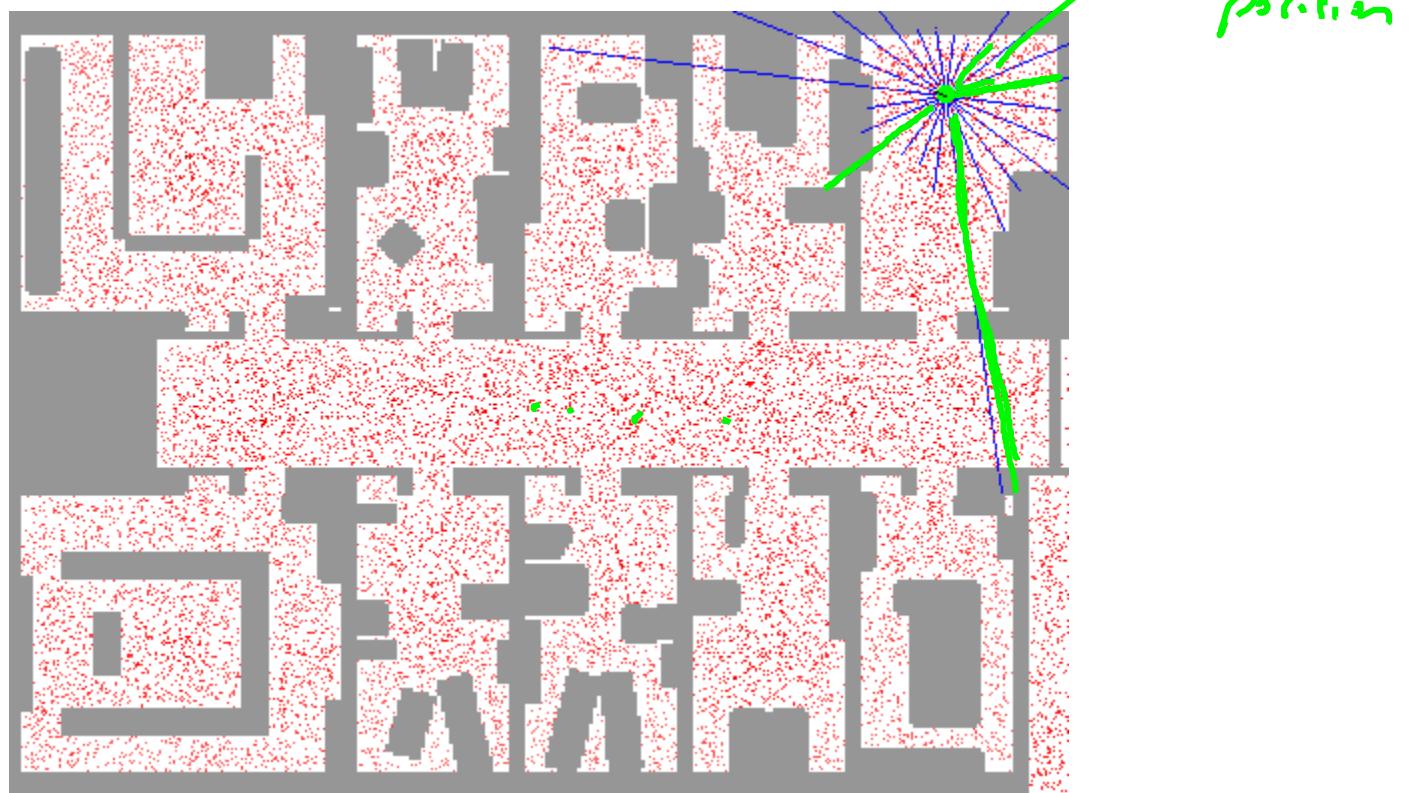
numerатор
+ normalizing
normalizing constant

Robot Localization



Daphne Koller

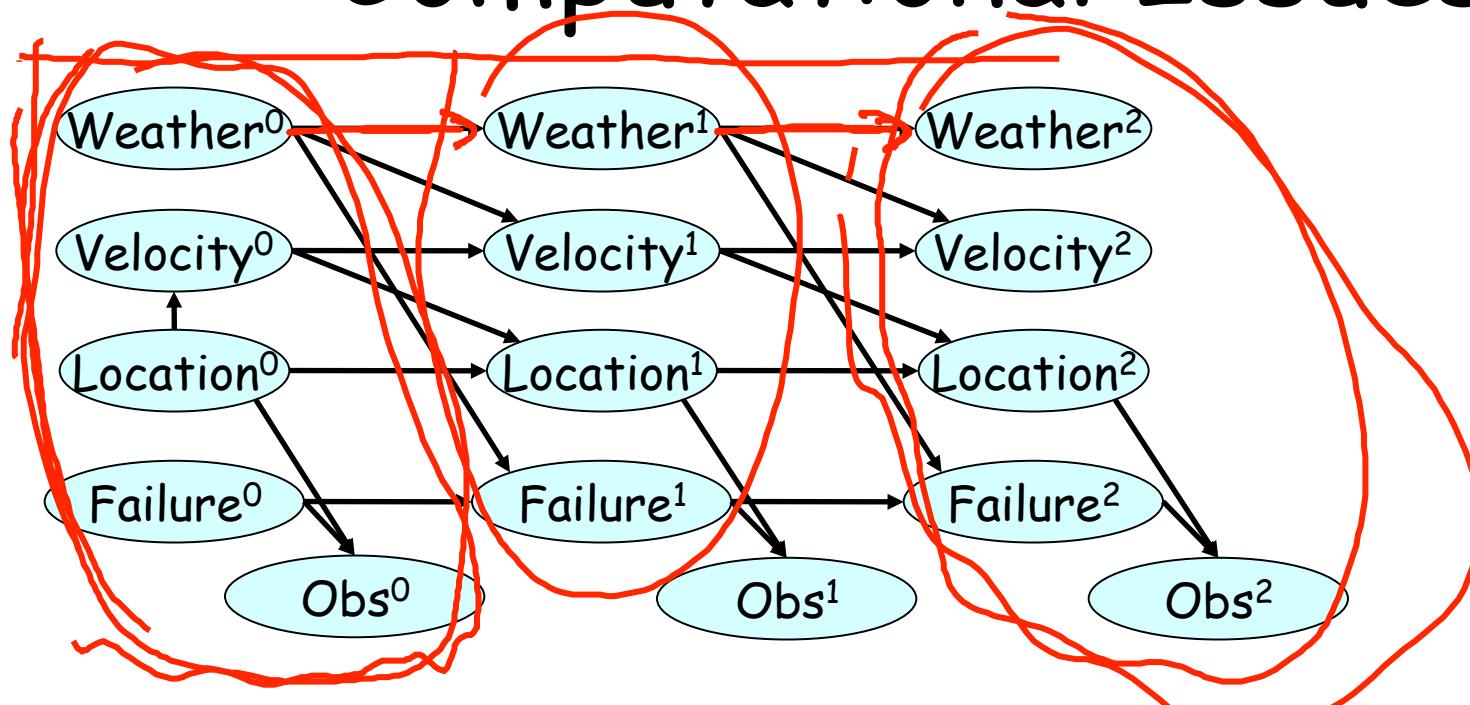
Robot Localization



Fox, Burgard, Thrun

Daphne Koller

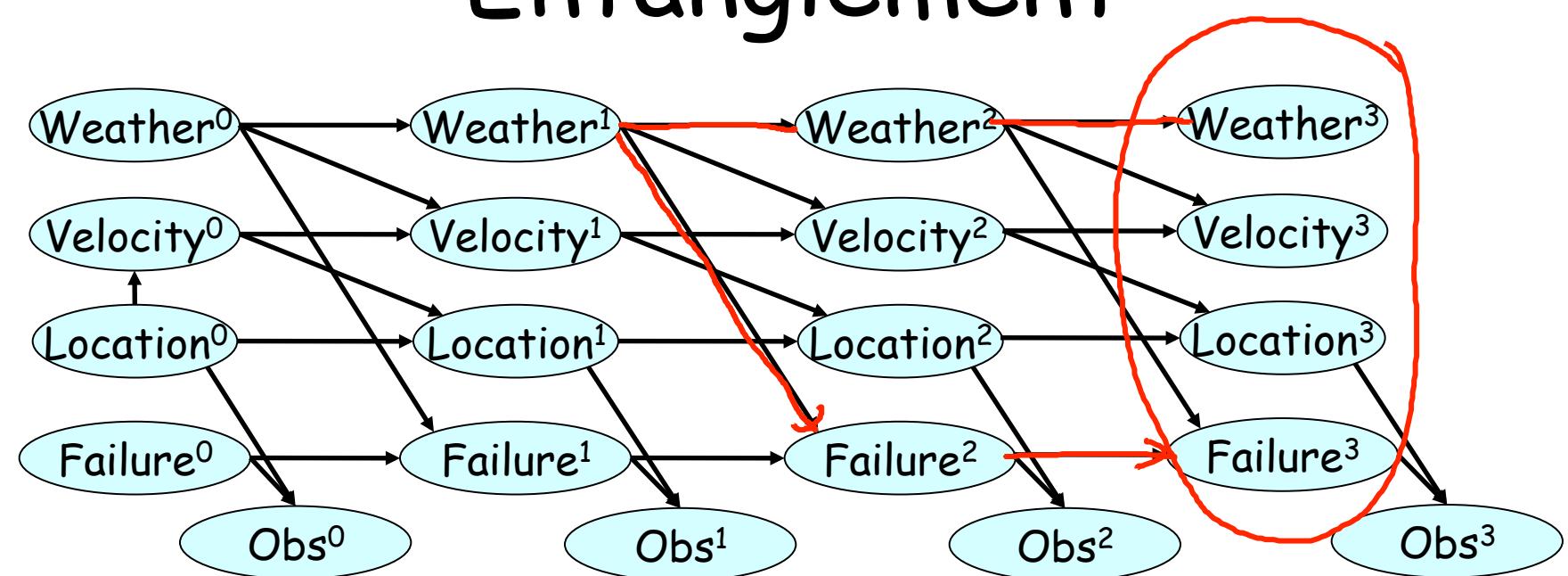
Computational Issues



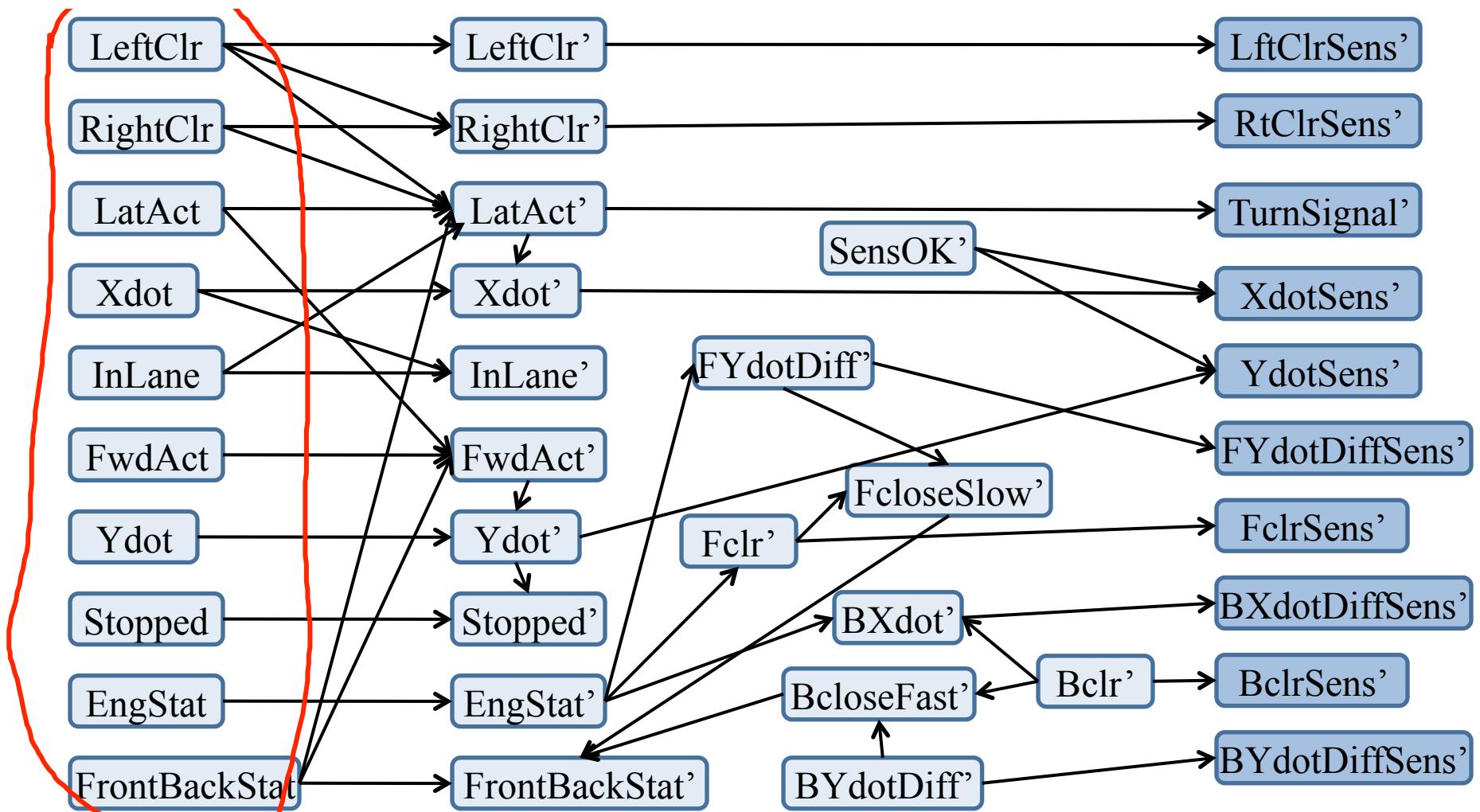
Minimal sepset must separate future from past

⇒ must involve at least all of the persistent variables

Entanglement



exact belief state is fully correlated in most cases, no conditional independence,





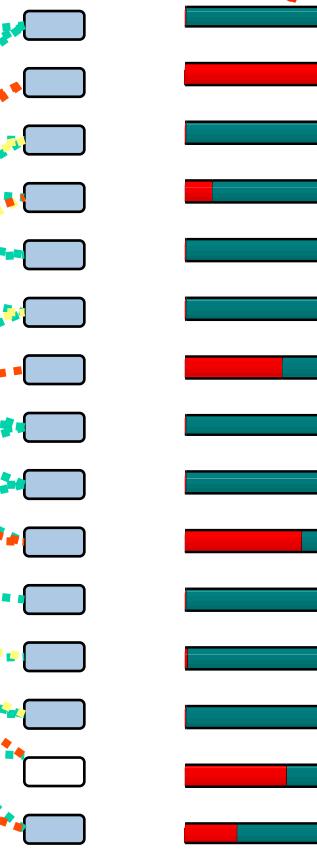
difficulty



easy / hard

bipartite
 $\min(m, n)$

intelligence

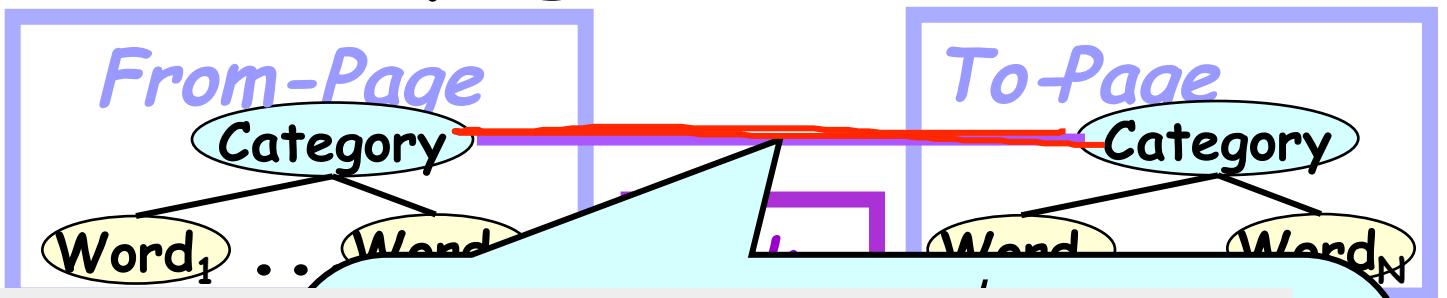
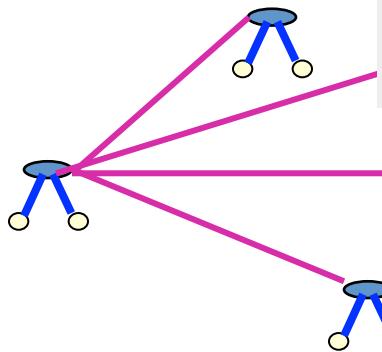
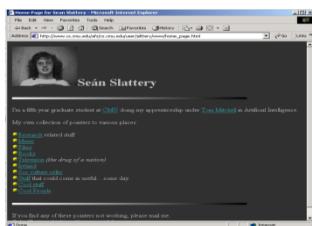


low / high

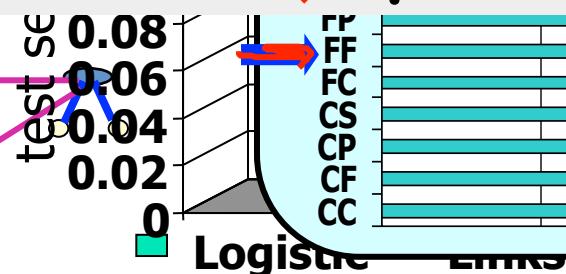
Daphne Koller

Webkinz (mitchell) (Craven et al, Proc AAAI98; Tasker et al, UAI2002)

Collective Webpage Classification



Classify all pages *collectively*,
maximizing the *joint label*
probability

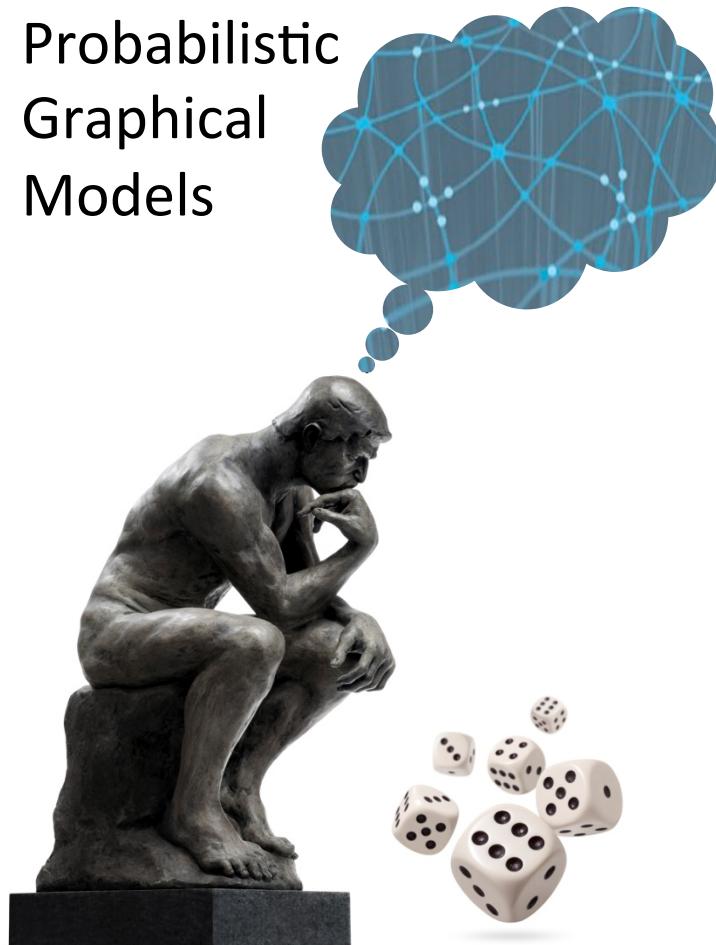


Daphne Koller

Summary

- Inference in template and temporal models can be done by unrolling the ground network and using standard methods
- Temporal models also raise new inference tasks, such as real-time tracking, which require that we adapt our methods
- Moreover, ground network is often large and densely connected, requiring careful algorithm design and use of approximate methods

Probabilistic
Graphical
Models



Inference

Summary

Inference Methods and Evaluation

MAP vs Marginals

Marginals

- Less fragile
- Confidence in answers
- Supports decision making

MAP

- Coherent joint assignment
- More tractable model classes
- Some theoretical guarantees

Approximate inference

- Errors are often attenuated
- Ability to gauge whether algorithm is working

Algorithms for Marginals

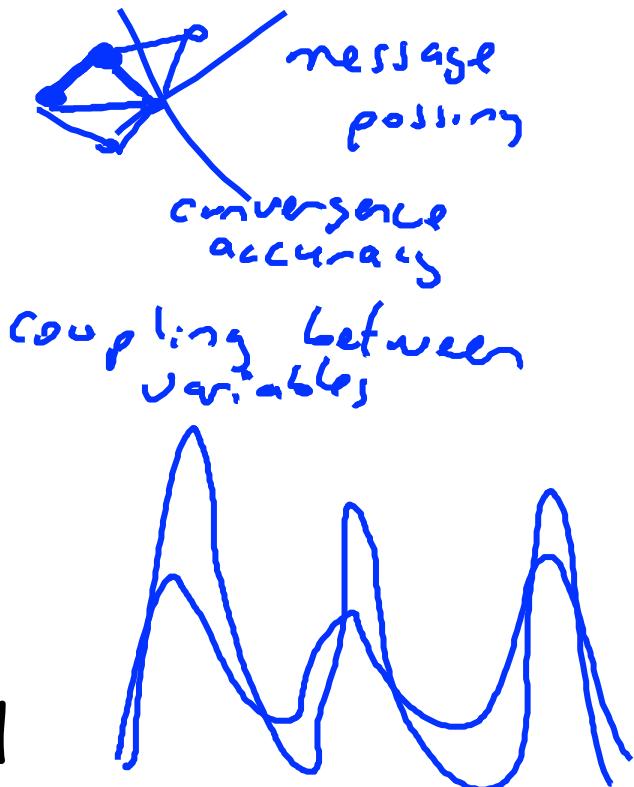
- Exact inference
fits in memory \Rightarrow exact inference
- Loopy message passing
- Sampling methods

Algorithms for MAP

- Exact inference
*low trees: dFL
associative models
...*
- Optimization methods:
– exact or approximate (dual decomposition)
- Search-based methods (including sampling)
hill-climbing nmc

Factors in Approximate Inference

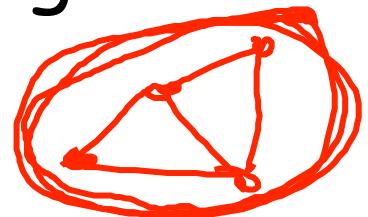
- Connectivity structure
- Strength of influence
- Opposing influences
- Multiple peaks in likelihood



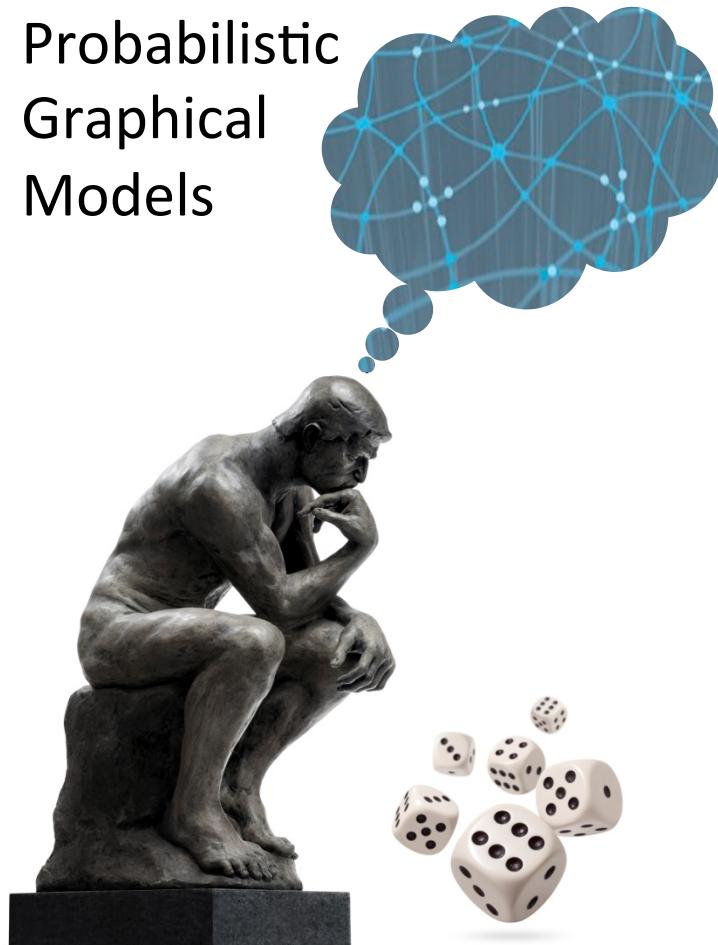
Daphne Koller

So, now what?

- Identify "problem regions" in network
- Try to make inference in these regions more exact
 - Larger clusters in cluster graph
 - Proposal moves over multiple variables
 - Larger "slave" in dual decomposition



Probabilistic
Graphical
Models

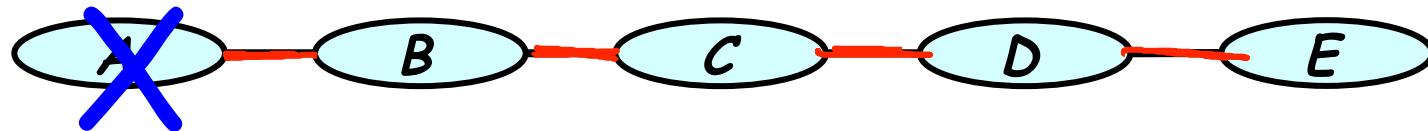


Inference

Variable Elimination

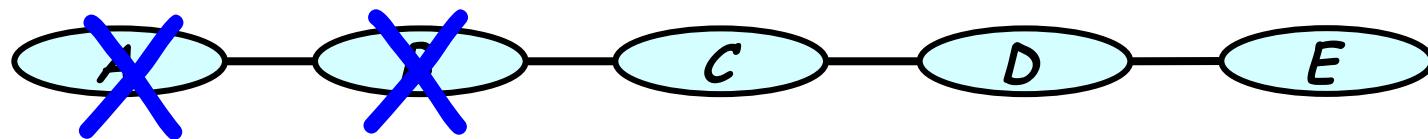
Variable Elimination Algorithm

Elimination in Chains



$$\begin{aligned}
 \underline{P(E)} &\propto \sum_D \sum_C \sum_B \sum_A \widetilde{P}(A, B, C, D, E) \\
 &= \sum_D \sum_C \sum_B \sum_A \phi_1(A, B) \phi_2(B, C) \phi_3(C, D) \phi_4(D, E) \\
 &= \sum_D \sum_C \sum_B \phi_2(B, C) \phi_3(C, D) \phi_4(D, E) \sum_A \phi_1(A, B) \tau_1(B) \\
 &= \sum_D \sum_C \sum_B \phi_2(B, C) \phi_3(C, D) \phi_4(D, E) \tau_1(B)
 \end{aligned}$$

Elimination in Chains



$$\begin{aligned} P(E) &\propto \sum_D \sum_C \sum_B \phi_2(B, C) \phi_3(C, D) \phi_4(D, E) \tau_1(B) \\ &= \sum_D \sum_C \phi_3(C, D) \phi_4(D, E) \left(\sum_B \phi_2(B, C) \tau_1(B) \right) \\ &= \sum_D \sum_C \phi_3(C, D) \phi_4(D, E) \tau_2(C) \end{aligned}$$

Annotations: Red arrows point from the terms $\phi_3(C, D)$ and $\phi_4(D, E)$ in the second line to the term $\phi_3(C, D) \phi_4(D, E)$ in the third line. A red bracket labeled $\tau_2(c)$ groups the term $\sum_B \phi_2(B, C) \tau_1(B)$.

Variable Elimination

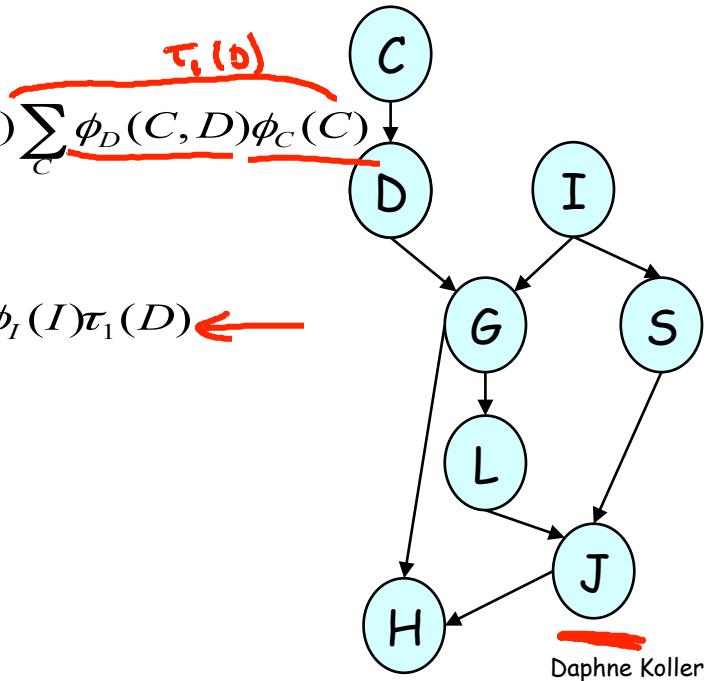
- Goal: $P(J)$
- Eliminate: C,D,I,H,G,S,L

$$\sum_{L,S,G,H,I,D,C} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_G(G, I, D) \phi_H(H, G, J) \phi_I(I) \phi_D(C, D) \phi_C(C)$$

$$\sum_{L,S,G,H,I,D} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_G(G, I, D) \phi_H(H, G, J) \phi_I(I)$$

Compute $\tau_1(D) = \sum_C \phi_C(C) \phi_D(C, D)$

$$= \sum_{L,S,G,H,I,D} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_G(G, I, D) \phi_H(H, G, J) \phi_I(I) \tau_1(D) \leftarrow$$



Variable Elimination

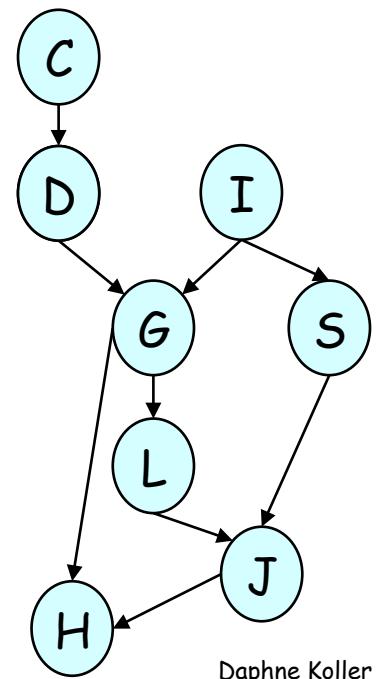
- Goal: $P(J)$
- Eliminate: D,I,H,G,S,L

$$\sum_{L,S,G,H,I,D} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_G(G, I, D) \phi_H(H, G, J) \phi_I(I) \tau_1(D)$$

$$= \sum_{L,S,G,H,I} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_H(H, G, J) \phi_I(I) \sum_D \phi_G(G, I, D) \tau_1(D)$$

Compute $\tau_2(G, I) = \sum_D \phi_G(G, I, D) \tau_1(D)$

$$= \sum_{L,S,G,H,I} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_H(H, G, J) \phi_I(I) \tau_2(G, I)$$



Variable Elimination

- Goal: $P(J)$
- Eliminate: I, H, G, S, L

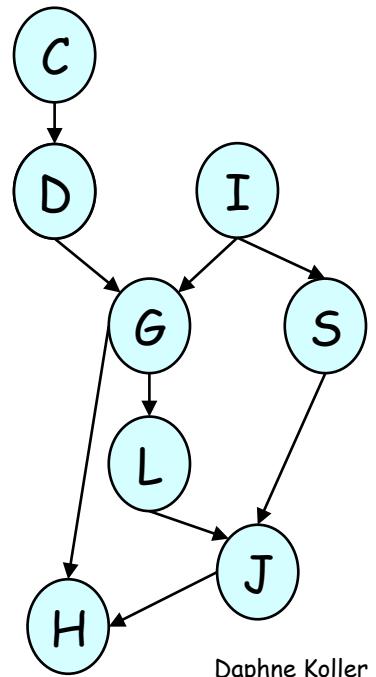
$$\sum_{L,S,G,H,I} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_H(H, G, J) \phi_I(I) \tau_2(G, I)$$

G, I, S

$$= \sum_{L,S,G,H} \phi_J(J, L, S) \phi_L(L, G) \phi_H(H, G, J) \sum_I \phi_S(S, I) \phi_I(I) \tau_2(G, I)$$

Compute $\tau_3(S, G)$ = $\sum_I \phi_S(S, I) \phi_I(I) \tau_2(G, I)$

$$= \sum_{L,S,G,H} \phi_J(J, L, S) \phi_L(L, G) \phi_H(H, G, J) \tau_3(S, G)$$



Variable Elimination

- Goal: $P(J)$
- Eliminate: H, G, S, L

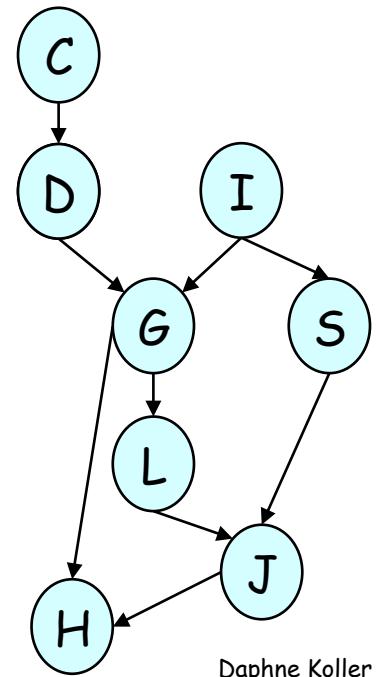
$$\sum_{L,S,G,H} \phi_J(J, L, S) \phi_L(L, G) \phi_H(H, G, J) \tau_3(S, G)$$

$$\sum_{L,S,G} \phi_J(J, L, S) \phi_L(L, G) \tau_3(S, G) \underbrace{\sum_H \phi_H(H, G, J)}_{\tau_4(G, J)}$$

$\cancel{\sum_H \phi_H(H, G, J) = 1}$

Compute $\tau_4(G, J) = \sum_H \phi_H(H, G, J)$

$$\sum_{L,S,G} \phi_J(J, L, S) \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$$



Variable Elimination

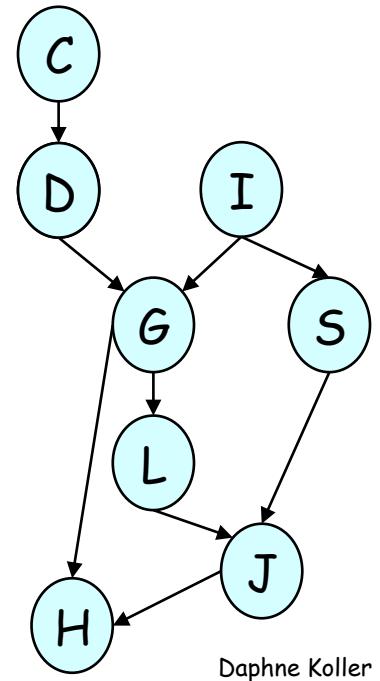
- Goal: $P(J)$
- Eliminate: G, S, L

$$\sum_{L,S,G} \phi_J(J, L, S) \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$$

$$\sum_{L,S} \phi_J(J, L, S) \sum_G \phi_L(L, G) \tau_4(G, J) \tau_3(S, G)$$

Compute $\tau_5(L, J) = \sum_G \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$

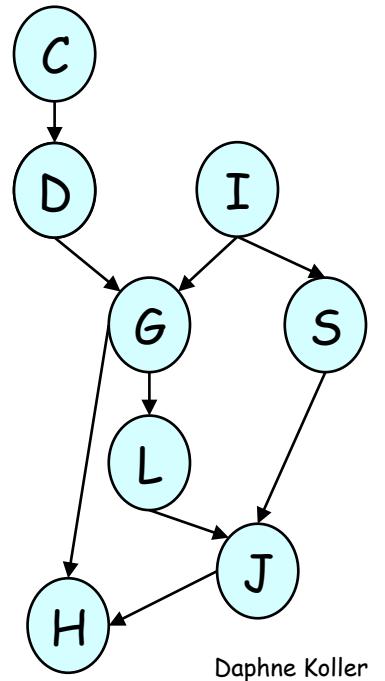
$$\sum_{L,S} \phi_J(J, L, S) \tau_5(L, J)$$



Variable Elimination

- Goal: $P(J)$
- Eliminate: S, L

$$\sum_{L,S} \phi_J(J, L, S) \tau_S(L, J)$$



Variable Elimination with evidence

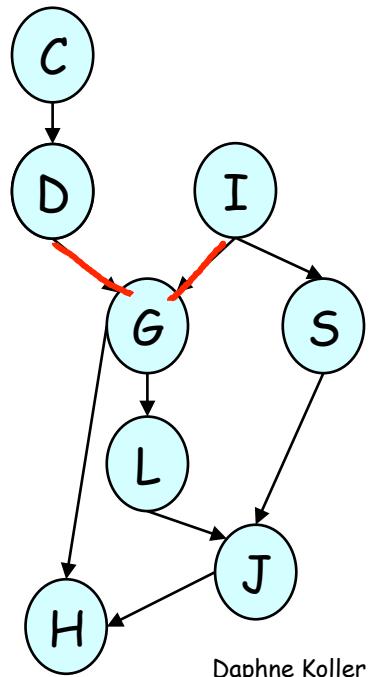
- Goal: $P(J, I=i, H=h)$
- Eliminate: $C, D, G, S, L \rightsquigarrow H, I$
 $P(I=i, D) \Phi_I(\cdot)$

$$\sum_{L, S, G, D, C} \phi_J(J, L, S) \phi_L(L, G) \phi_S(S) \underbrace{\phi_G(G, D)}_{\Phi_G(\cdot)} \phi_H(H, J) \underbrace{\phi_I(I)}_{\Phi_I(\cdot)} \phi_D(D, C) \phi_C(C)$$

elimination as before

How do we get $P(J | I=i, H=h)$?

normalize
 $P(I=i, H=h)$ is the normalizing constant



Variable Elimination in MNs

- Goal: $P(D)$
- Eliminate: A, B, C

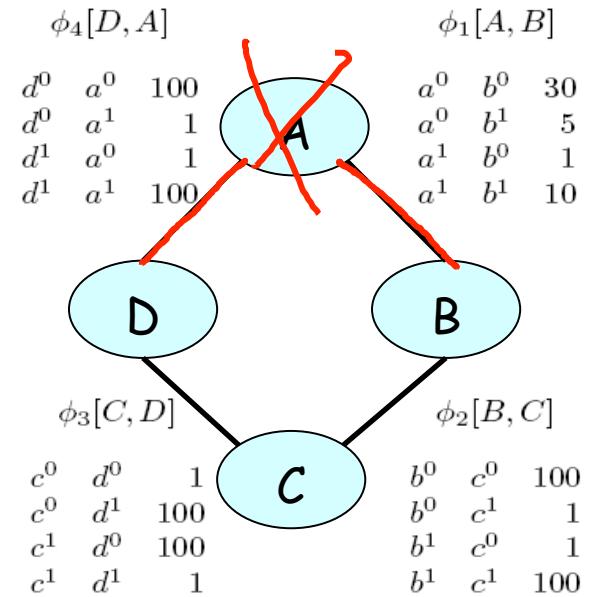
$$\sum_{A,B,C} \phi_1(A, B) \phi_2(B, C) \phi_3(C, D) \phi_4(A, D)$$

$$\sum_{B,C} \phi_2(B, C) \phi_3(C, D) \sum_A \phi_1(A, B) \phi_4(A, D)$$

$$\sum_{B,C} \phi_2(B, C) \phi_3(C, D) \tau_1(B, D)$$

$$= \tilde{P}(D)$$

At the end of elimination get $\underline{\tau_3(D)} \propto P(D)$
renormalize



Eliminate-Var Z from Φ

$$\underline{\Phi'} = \{\phi_i \in \Phi : \underline{Z \in \text{Scope}[\phi_i]}\}$$

all factors that involve z

$$\psi = \prod_{\phi_i \in \Phi'} \phi_i$$

multiply them

$$\tau = \sum_Z \psi$$

fixed sum out z

$$\Phi := \Phi - \underline{\Phi'} \cup \underline{\{\tau\}}$$

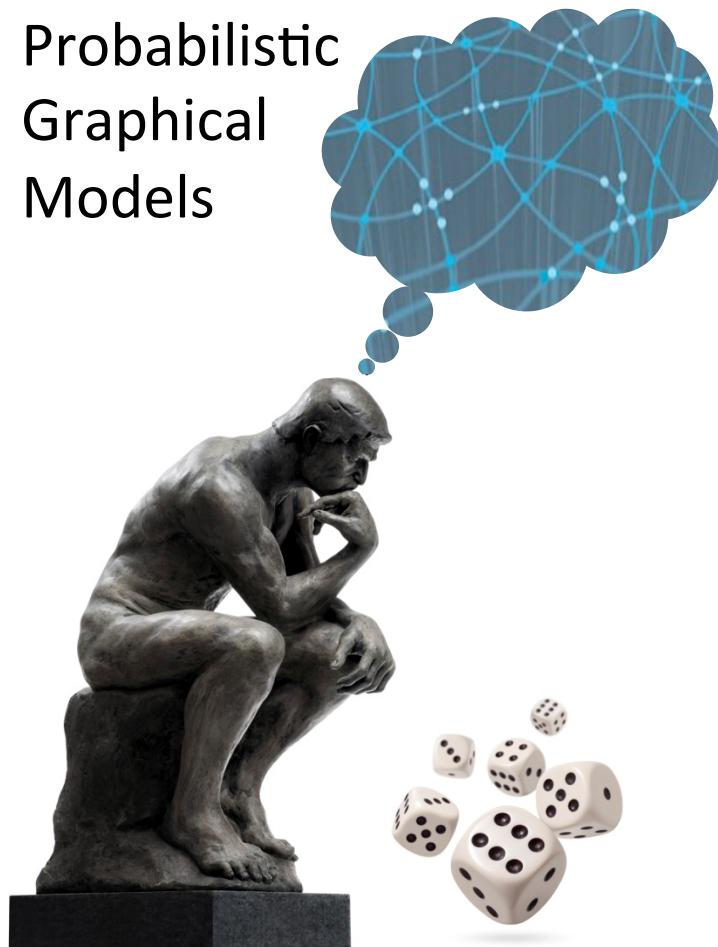
VE Algorithm Summary

- Reduce all factors by evidence
 - Get a set of factors $\underline{\Phi}$
- For each non-query variable Z
 - Eliminate-Var Z from $\underline{\Phi}$ adds removes $\underline{\Phi}'$
adds \underline{z}
- Multiply all remaining factors
- Renormalize to get distribution

Summary

- Simple algorithm
- Works for both BNs and MNs
- Factor product and summation steps can be done in any order, subject to:
 - when Z is eliminated, all factors involving Z have been multiplied in

Probabilistic
Graphical
Models



Inference

Variable Elimination

Complexity
Analysis

Eliminating Z

$$\psi_k(\mathbf{X}_k) = \prod_{i=1}^{m_k} \phi_i \quad \text{factor product}$$

$$\tau_k(\mathbf{X}_k - \{Z\}) = \sum_Z \psi_k(\mathbf{X}_k) \quad \text{marginalization}$$

Reminder: Factor Product

$$\psi_k(\mathbf{X}_k) = \prod_{i=1}^{m_k} \phi_i$$

*each row
 m_{k-1} products*

$$N_k = |\text{Val}(\mathbf{X}_k)|$$

a ¹	b ¹	0.5
a ¹	b ²	0.8
a ²	b ¹	0.1
a ²	b ²	0
a ³	b ¹	0.3
a ³	b ²	0.9

b ¹	c ¹	0.5
b ¹	c ²	0.7
b ²	c ¹	0.1
b ²	c ²	0.2

B C



a ¹	b ¹	c ¹	0.5 · 0.5 = 0.25
a ¹	b ¹	c ²	0.5 · 0.7 = 0.35
a ¹	b ²	c ¹	0.8 · 0.1 = 0.08
a ¹	b ²	c ²	0.8 · 0.2 = 0.16
a ²	b ¹	c ¹	0.1 · 0.5 = 0.05
a ²	b ¹	c ²	0.1 · 0.7 = 0.07
a ²	b ²	c ¹	0 · 0.1 = 0
a ²	b ²	c ²	0 · 0.2 = 0
a ³	b ¹	c ¹	0.3 · 0.5 = 0.15
a ³	b ¹	c ²	0.3 · 0.7 = 0.21
a ³	b ²	c ¹	0.9 · 0.1 = 0.09
a ³	b ²	c ²	0.9 · 0.2 = 0.18

Cost: $(m_k - 1)N_k$ multiplications

Reminder: Factor Marginalization

$$\tau_k(\underline{X}_k - \{Z\}) = \sum_Z \psi_k(\underline{X}_k)$$

$$N_k = |\text{Val}(\underline{X}_k)|$$

Cost: $\sim N_k$ additions

each number used exactly once

marg B

a ¹	b ¹	c ¹	0.25
a ¹	b ¹	c ²	0.35
a ¹	b ²	c ¹	0.08
a ¹	b ²	c ²	0.16
a ²	b ¹	c ¹	0.05
a ²	b ¹	c ²	0.07
a ²	b ²	c ¹	0
a ²	b ²	c ²	0
a ³	b ¹	c ¹	0.15
a ³	b ¹	c ²	0.21
a ³	b ²	c ¹	0.09
a ³	b ²	c ²	0.18

a ¹	c ¹	0.33
a ¹	c ²	0.51
a ²	c ¹	0.05
a ²	c ²	0.07
a ³	c ¹	0.24
a ³	c ²	0.39

Complexity of Variable Elimination

- Start with m factors
 - $m \leq n$ for Bayesian networks *(one for every variable)*
 - can be larger for Markov networks
- At each elimination step generate 1 factor,
- At most n elimination steps
- Total number of factors: $m^* \leq m + n$

Complexity of Variable Elimination

- $\underline{N} = \max(N_k) = \text{size of the largest factor}$
- Product operations: $\sum_k (m_k - 1)N_k \leq N \sum_k (m_k - 1)$
each factor multiply in at most one
 $N_{\text{mt}} \leq m^*$
- Sum operations: $\leq \sum_k N_k \leq N \cdot \# \text{elimination steps} \leq \underline{N \cdot n}$
- Total work is linear in \underline{N} and m^*

Complexity of Variable Elimination

- Total work is linear in N and m exponential blowup
- $N_k = |\text{Val}(X_k)| = O(d^{r_k})$ where # variables in kth factor
 - $d = \max(|\text{Val}(X_i)|)$ d values in their scope
 - $r_k = |X_k|$ = cardinality of the scope of the kth factor

Complexity Example

$$\tau_1(D) = \sum_C \phi_C(C) \phi_D(C, D)$$
2

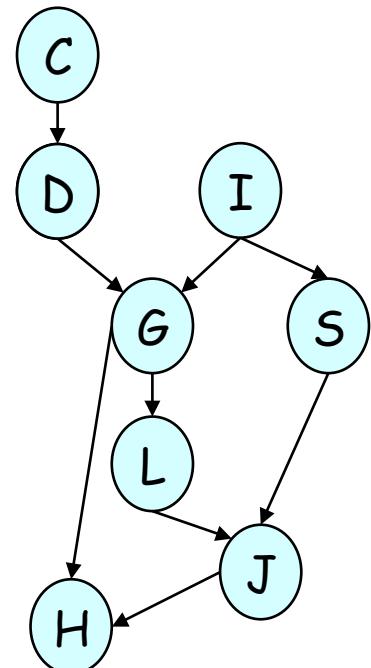
$$\tau_2(G, I) = \sum_D \phi_G(G, I, D) \tau_1(D)$$
3

$$\tau_3(S, G) = \sum_I \phi_S(S, I) \phi_I(I) \tau_2(G, I)$$
3

$$\tau_4(G, J) = \sum_H \phi_H(H, G, J)$$
3

$$\tau_5(J, L, S) = \sum_G \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$$
4


$$\tau_6(J) = \sum_{L, S} \phi_J(J, L, S) \tau_5(J, L, S)$$
3



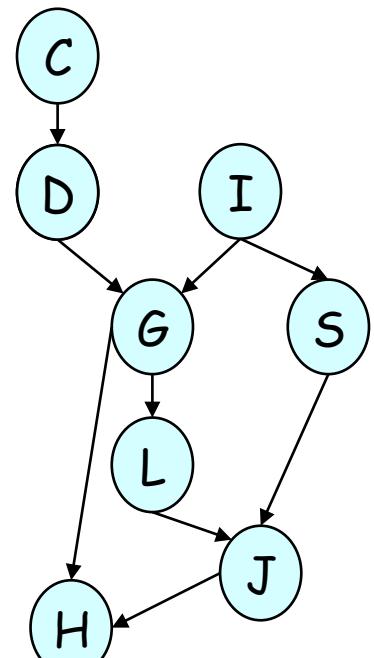
Complexity and Elimination Order

$$\sum_{L,S,G,H,I,D,C} \phi_J(J, L, S) \underbrace{\phi_L(L, G)}_{6} \phi_S(S, I) \underbrace{\phi_G(G, I, D)}_{6} \underbrace{\phi_H(H, G, J)}_{6} \phi_I(I) \phi_D(C, D) \phi_C(C)$$

- Eliminate: G

$\overbrace{L, G, I, D, H, J}^6$

$$\sum_G \phi_L(L, G) \phi_G(G, I, D) \phi_H(H, G, J)$$



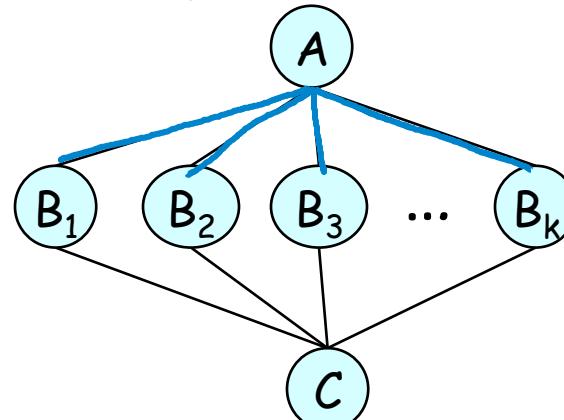
Daphne Koller

Complexity and Elimination Order

Eliminate A first:

$$\{A, B_1, \dots, B_k\}$$

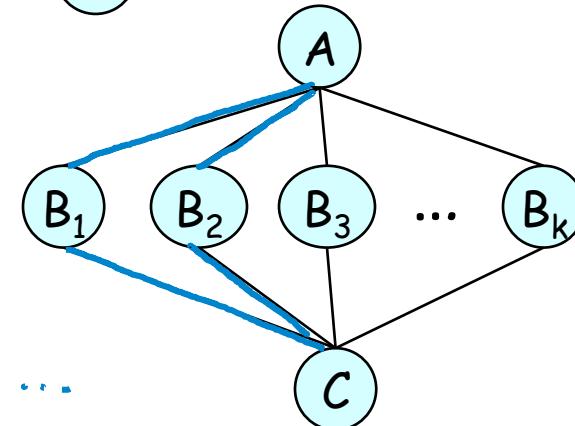
size of factor is exp in k



$$\prod_i \tau_i(A, c)$$

Eliminate Bi's first:

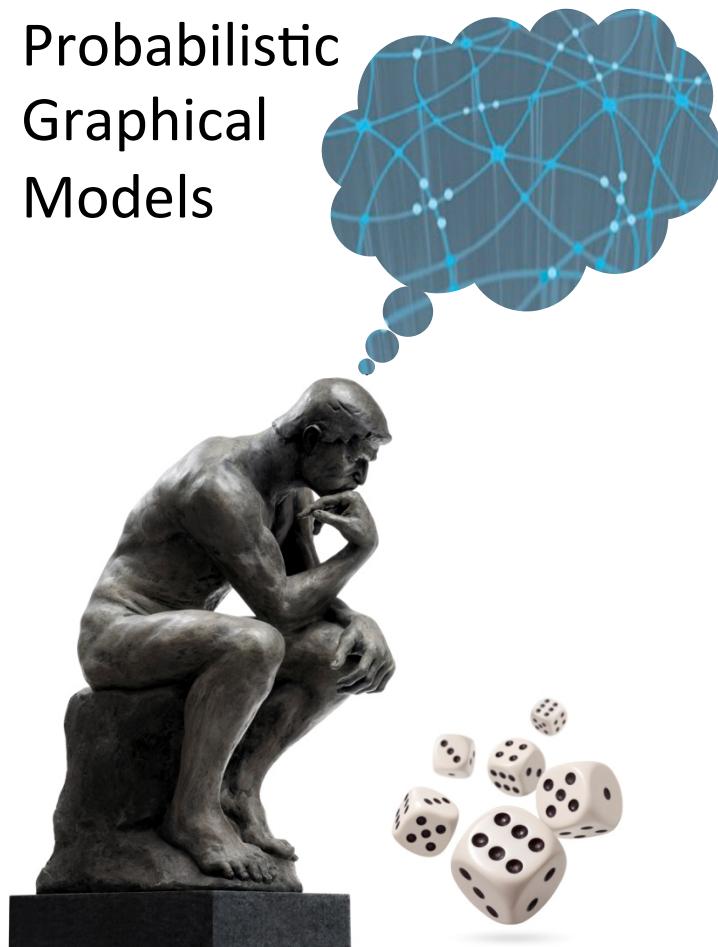
$$\underbrace{\phi_{i+1}(A, B_i) \cdot \phi_{i+1}(C, B_i)}_{\text{Scope } A, B_i, C} \Rightarrow \tau_i(A, c) \quad \tau_2(A, c) \dots$$



Summary

- Complexity of variable elimination linear in
 - size of the model (# factors, # variables)
 - size of the largest factor generated
- Size of factor is exponential in its scope
- Complexity of algorithm depends heavily on elimination ordering

Probabilistic
Graphical
Models



Inference

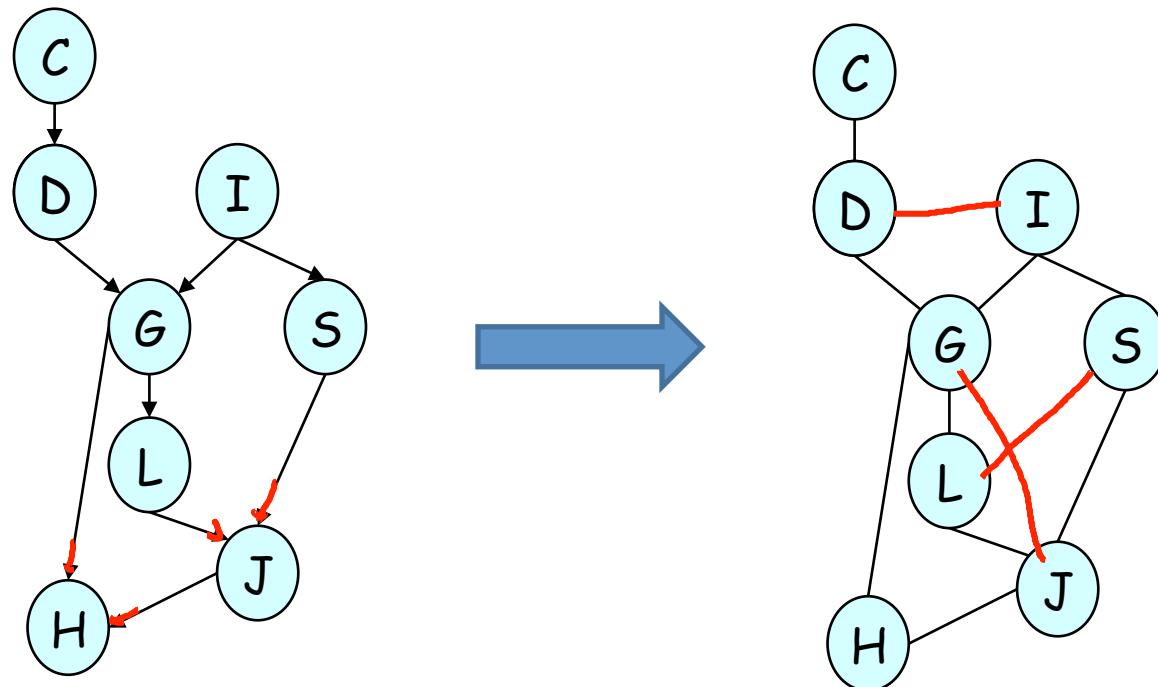
Variable Elimination

Graph-Based
Perspective

Initial Graph

$\phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_G(G, I, D) \phi_H(H, G, J) \phi_I(I) \phi_D(C, D) \phi_C(C)$

moralization



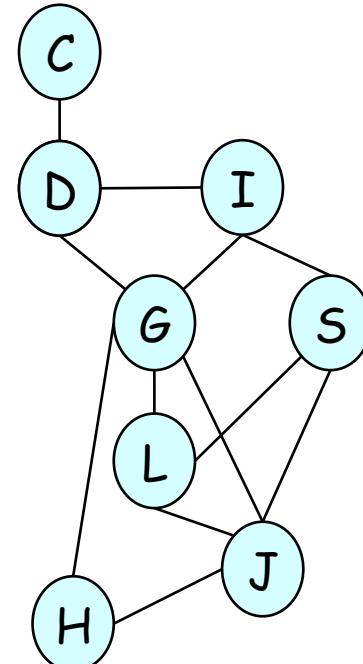
Daphne Koller

Elimination as Graph Operation

$$\phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \phi_G(G, I, D) \phi_H(H, G, J) \phi_I(I) \cancel{\phi_D(C, D)} \cancel{\phi_C(C)}$$

- Eliminate: C

$$\underline{\tau_1(D)} = \sum_C \phi_C(C) \phi_D(C, D)$$



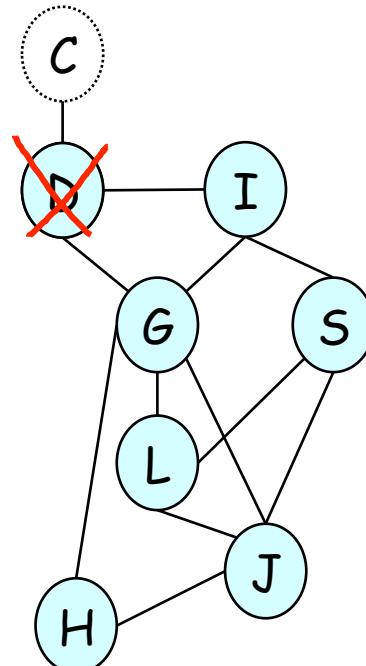
Induced Markov network for the current set of factors

Elimination as Graph Operation

$$\phi_J(J, L, S) \phi_L(L, G) \phi_S(S, I) \underline{\phi_G(G, I, D)} \phi_H(H, G, J) \phi_I(I) \underline{\tau_1(D)}$$

- Eliminate: D

$$\tau_2(G, I) = \sum_D \phi_G(G, I, D) \tau_1(D)$$



Induced Markov network for the current set of factors

Daphne Koller

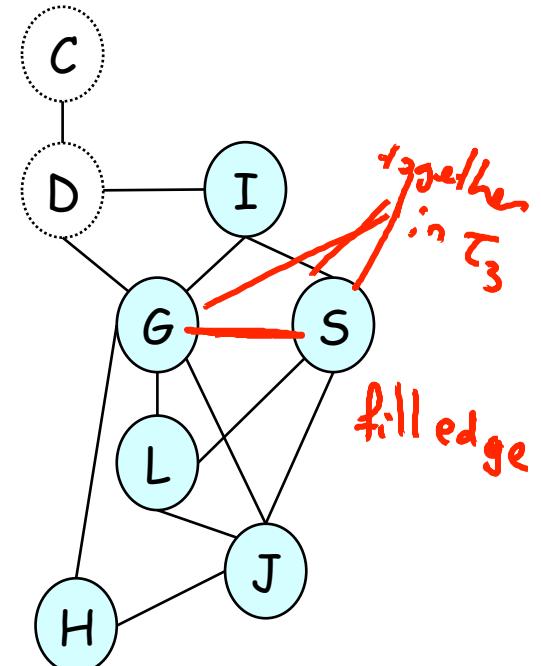
Elimination as Graph Operation

$$\phi_J(J, L, S) \phi_L(L, G) \underline{\phi_S(S, I)} \underline{\phi_I(I)} \phi_H(H, G, J) \underline{\tau_2(G, I)}$$

- Eliminate: I

$$\tau_3(S, G) = \sum_I \phi_S(S, I) \phi_I(I) \tau_2(G, I)$$

all variables connected to I become
connected directly



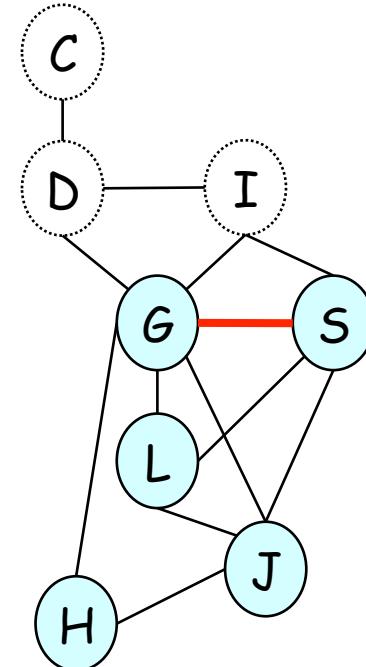
Induced Markov network for the current set of factors

Elimination as Graph Operation

$$\phi_J(J, L, S) \phi_L(L, G) \phi_H(H, G, J) \tau_3(S, G)$$

- Eliminate: H

$$\tau_4(G, J) = \sum_H \phi_H(H, G, J)$$



Induced Markov network for the current set of factors

Daphne Koller

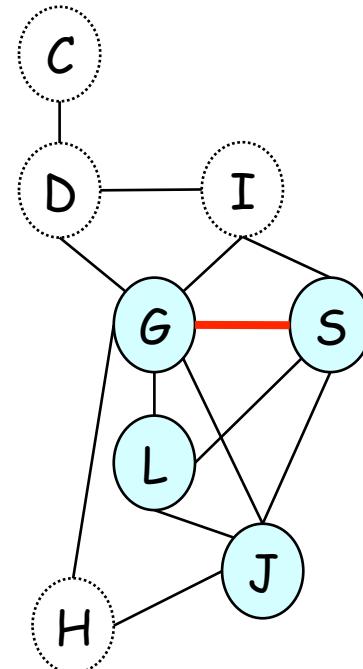
Elimination as Graph Operation

$$\phi_J(J, L, S) \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$$

- Eliminate: G

$$\tau_5(L, J) = \sum_G \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$$

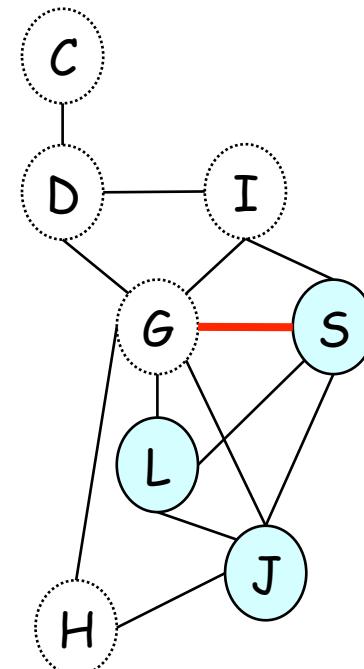
Induced Markov network for the current set of factors



Elimination as Graph Operation

$$\phi_J(J, L, S) \tau_5(L, J)$$

- Eliminate: L, S



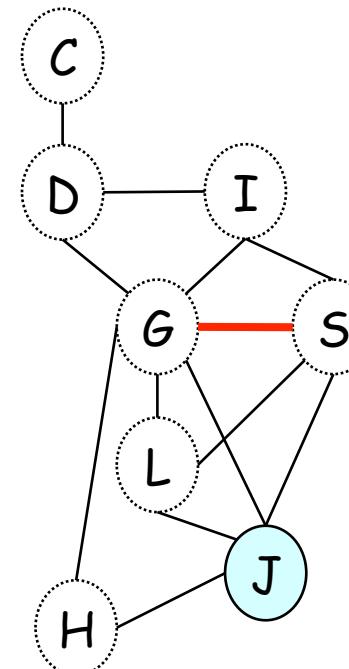
Induced Markov network for the current set of factors

Daphne Koller

Elimination as Graph Operation

$$\phi_J(J, L, S) \tau_5(L, J)$$

- Eliminate: L, S

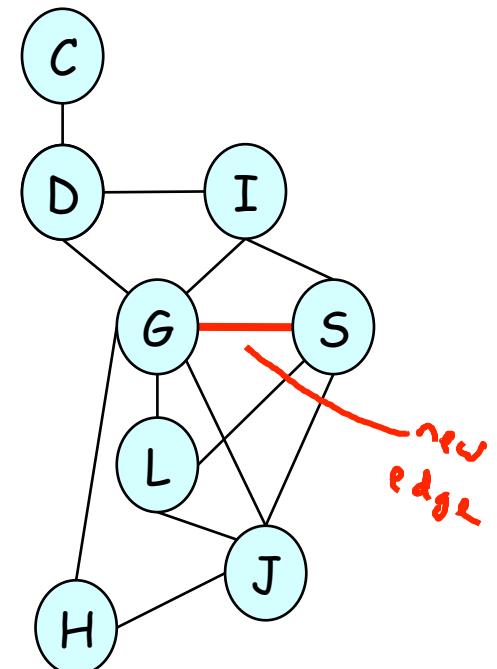


Induced Markov network for the current set of factors

Daphne Koller

Induced Graph

- The induced graph $I_{\Phi, \alpha}$ over factors Φ and ordering α :
 - Undirected graph
 - X_i and X_j are connected if they appeared in the same factor in a run of the VE algorithm using α as the ordering



Daphne Koller

Cliques in the Induced Graph

maximal fully connected subgraph

- ~~Theorem~~: Every factor produced during VE is a clique in the induced graph

$$\tau_1(D) = \sum_C \phi_C(C) \phi_D(C, D)$$

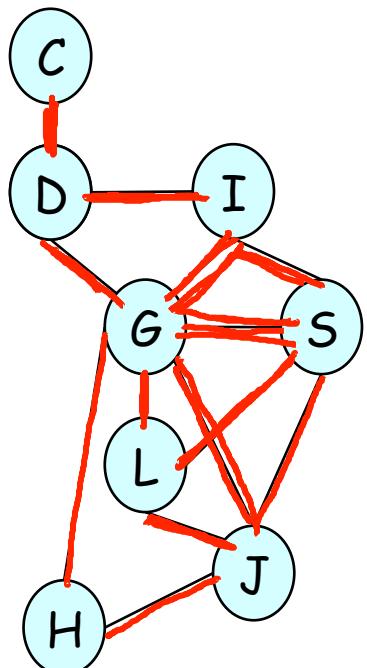
$$\tau_2(G, I) = \sum_D \phi_G(G, I, D) \tau_1(D)$$

$$\tau_3(S, G) = \sum_I \phi_S(S, I) \phi_I(I) \tau_2(G, I)$$

$$\tau_4(G, J) = \sum_H \phi_H(H, G, J)$$

$$\tau_5(L, J) = \sum_G \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$$

$$\tau_6 = \sum_{L, S} \phi_J(J, L, S) \tau_5(L, J)$$



Daphne Koller

Cliques in the Induced Graph

- **Theorem:** Every (maximal) clique in the induced graph is a factor produced during VE

$$\tau_1(D) = \sum_C \phi_C(C) \phi_D(C, D)$$

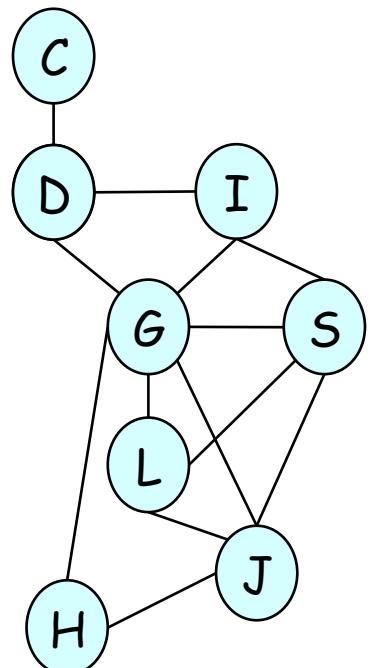
$$\tau_2(G, I) = \sum_D \phi_G(G, I, D) \tau_1(D)$$

$$\tau_3(S, G) = \sum_I \phi_S(S, I) \phi_I(I) \tau_2(G, I)$$

$$\tau_4(G, J) = \sum_H \phi_H(H, G, J)$$

$$\tau_5(L, J) = \sum_G \phi_L(L, G) \tau_3(S, G) \tau_4(G, J)$$

$$\tau_6 = \sum_{L, S} \phi_J(J, L, S) \tau_5(L, J)$$



Daphne Koller

Cliques in the Induced Graph

- Theorem: Every (maximal) clique in the induced graph is a factor produced during VE

Consider a max clique -

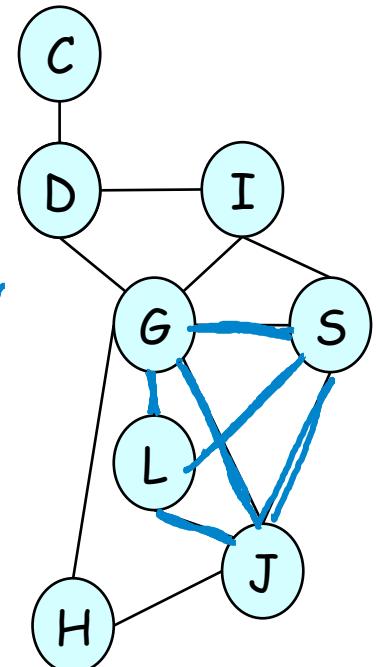
some variable is first to be eliminated

once a variable is eliminated - no new neighbor

⇒ when eliminated it already had all the
clique members as neighbors

⇒ participated in factors with all these other variables

⇒ when multiplied together, we have a factor
over all of them



Daphne Koller

Induced Width

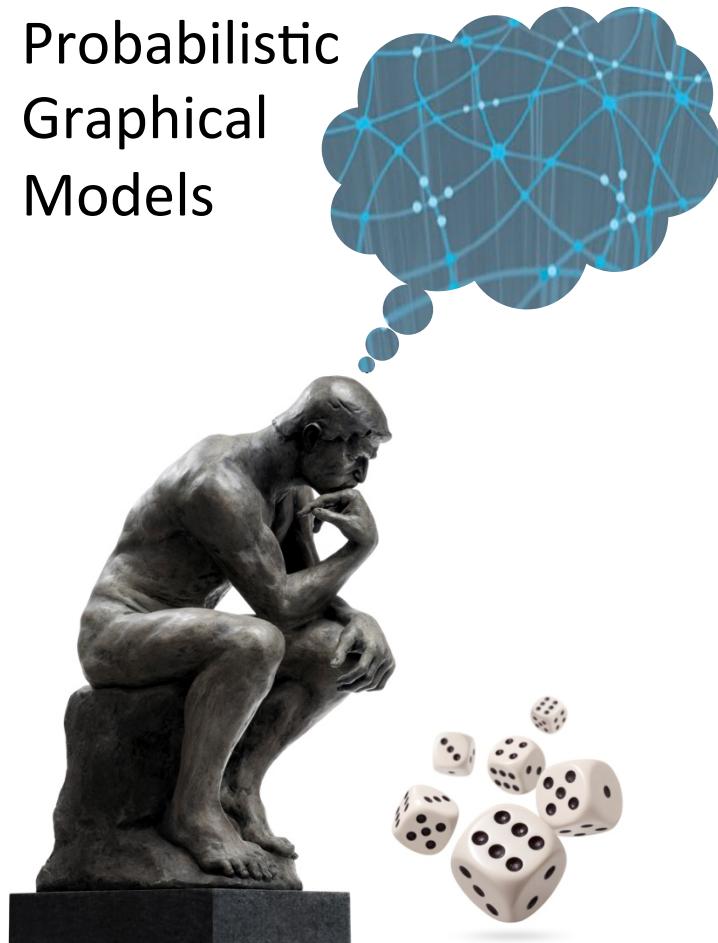
- The width of an induced graph is the number of nodes in the largest clique in the graph minus 1
- Minimal induced width of a graph K is $\min_{\alpha}(\text{width}(I_{K,\alpha}))$
- Provides a lower bound on best performance of VE to a model factorizing over K

Summary

- Variable elimination can be viewed as transformations on undirected graph
 - Elimination connects all node's current neighbors
- Cliques in resulting induced graph directly correspond to algorithm's complexity



Probabilistic
Graphical
Models



Inference

Variable Elimination

Finding
Elimination
Orderings

Finding Elimination Orderings

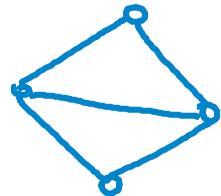
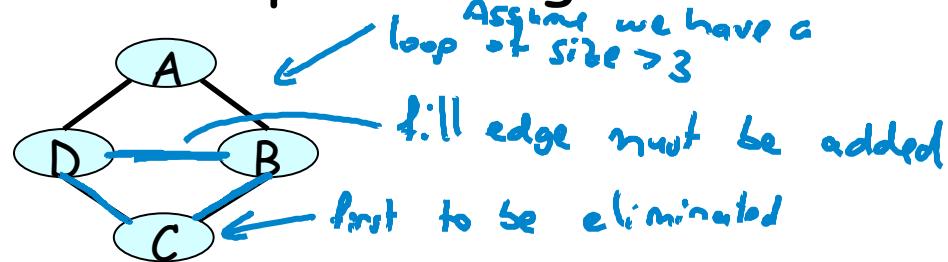
- **Theorem:** For a graph H , determining whether there exists an elimination ordering for H with induced width $\leq K$ is NP-complete
- **Note:** This NP-hardness result is distinct from the NP-hardness result of inference
 - Even given the optimal ordering, inference may still be exponential

Finding Elimination Orderings

- Greedy search using heuristic cost function
 - At each point, eliminate node with smallest cost
- Possible cost functions:
 - min-neighbors: # neighbors in current graph
 - min-weight: weight (# values) of factor formed
 - min-fill: number of new fill edges
 - weighted min-fill: total weight of new fill edges
(edge weight = product of weights of the 2 nodes)
smallest factor

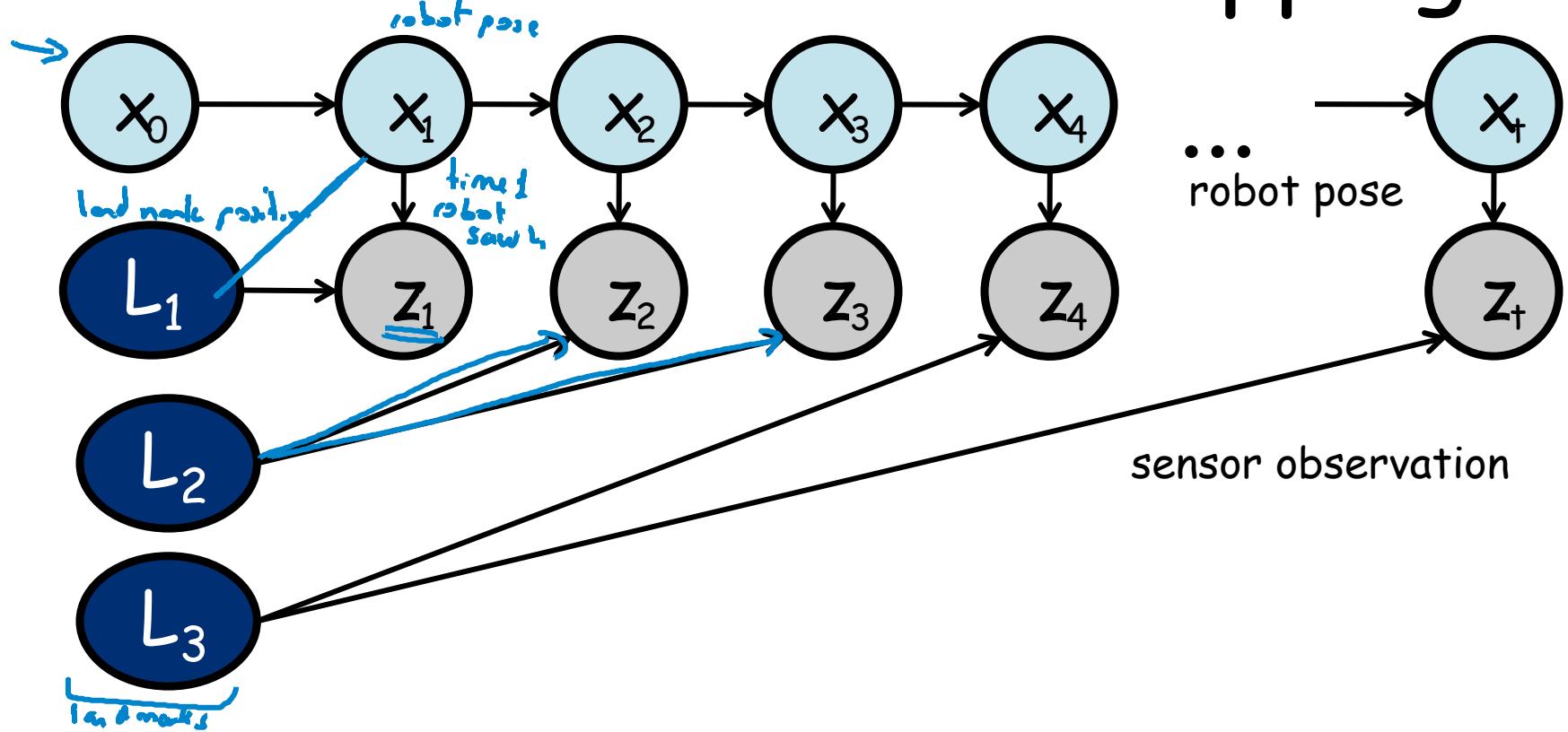
Finding Elimination Orderings

- **Theorem:** The induced graph is triangulated
 - No loops of length > 3 without a "bridge"



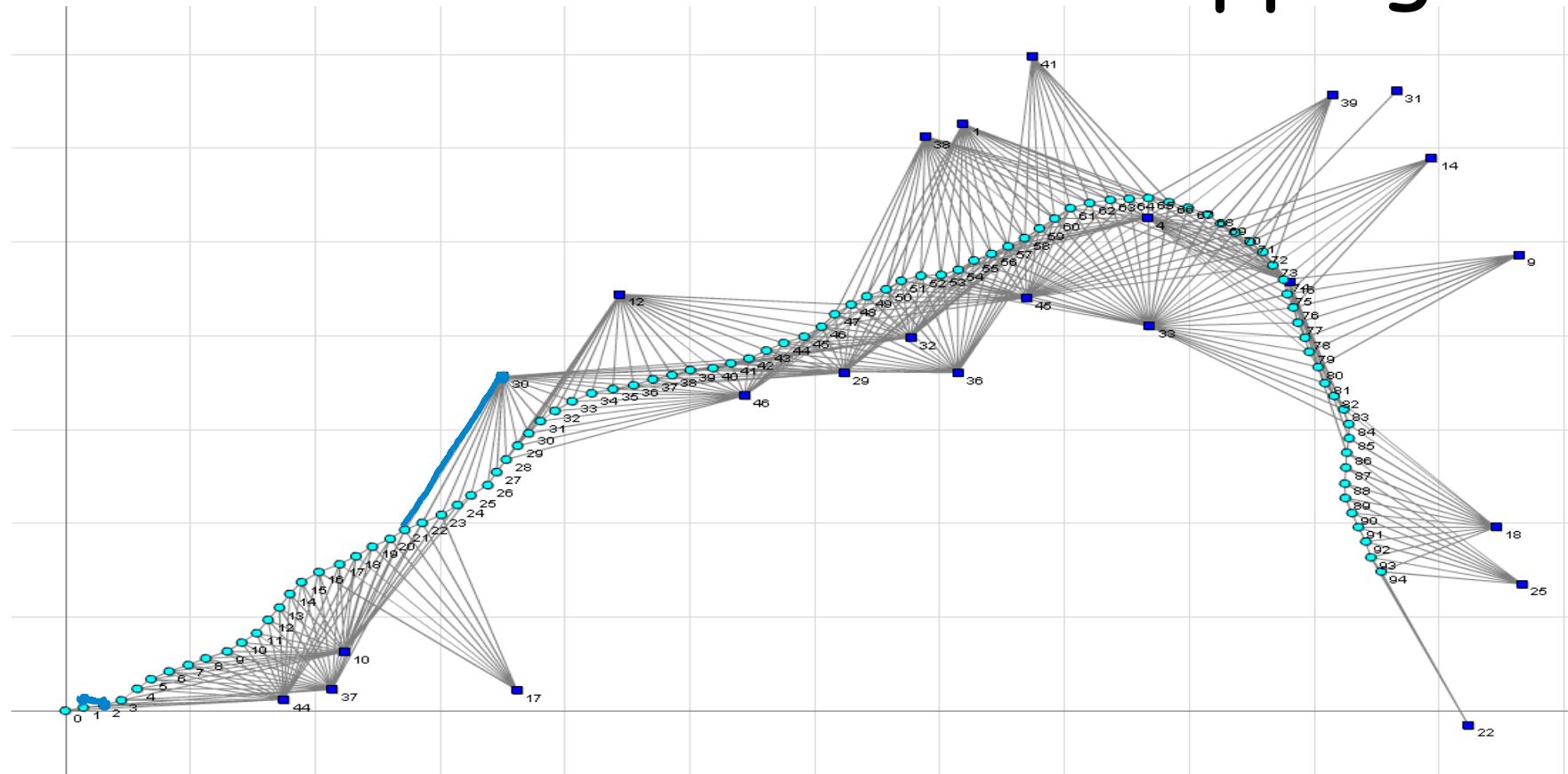
- Can find elimination ordering by finding a low-width triangulation of original graph H_Φ

Robot Localization & Mapping



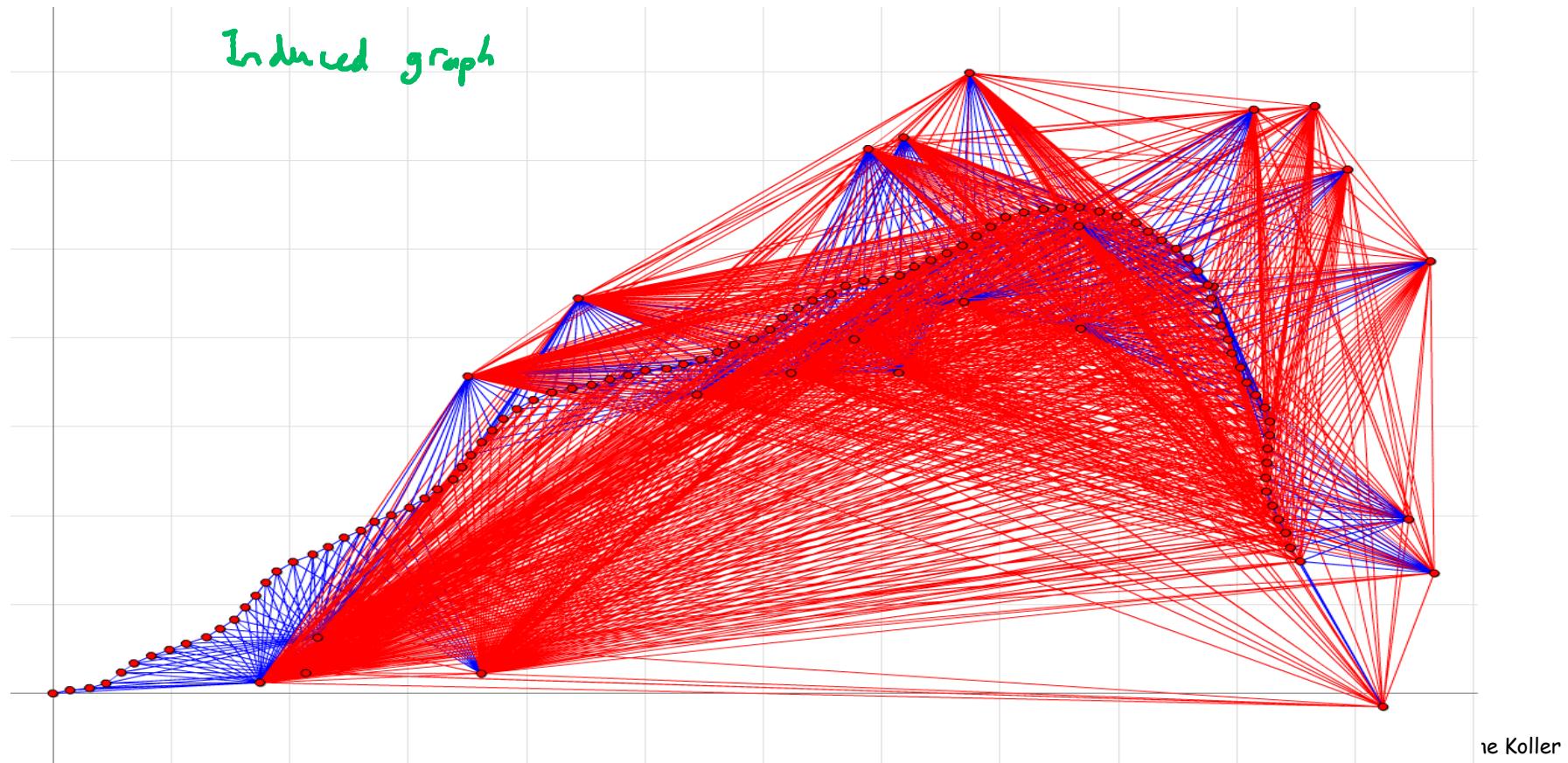
Square Root SAM, F. Dellaert and M. Kaess, IJRR, 2006

Robot Localization & Mapping



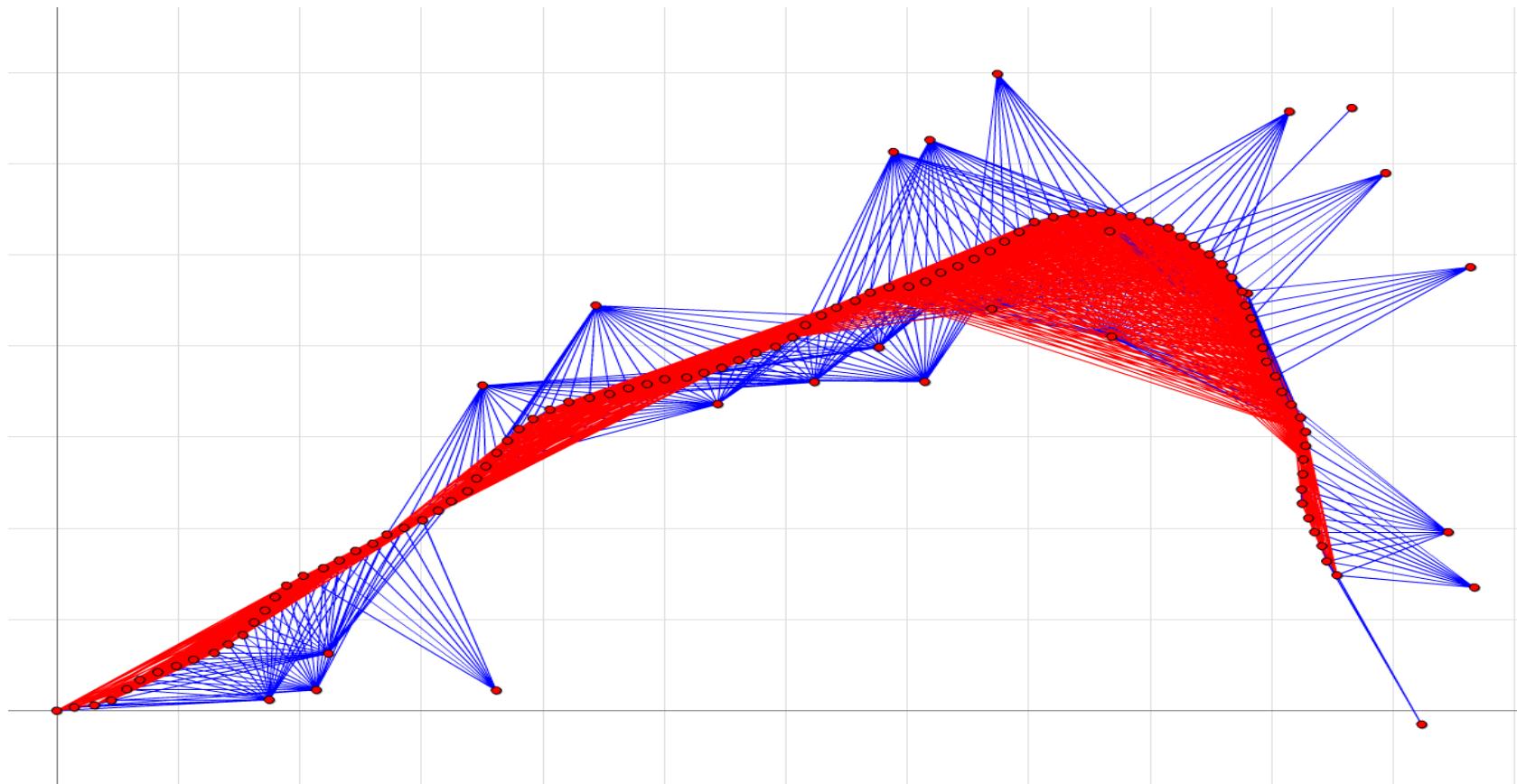
Square Root SAM, F. Dellaert and M. Kaess, IJRR, 2006

Eliminate Poses then Landmarks



Square Root SAM, F. Dellaert and M. Kaess, IJRR, 2006

Eliminate Landmarks then Poses

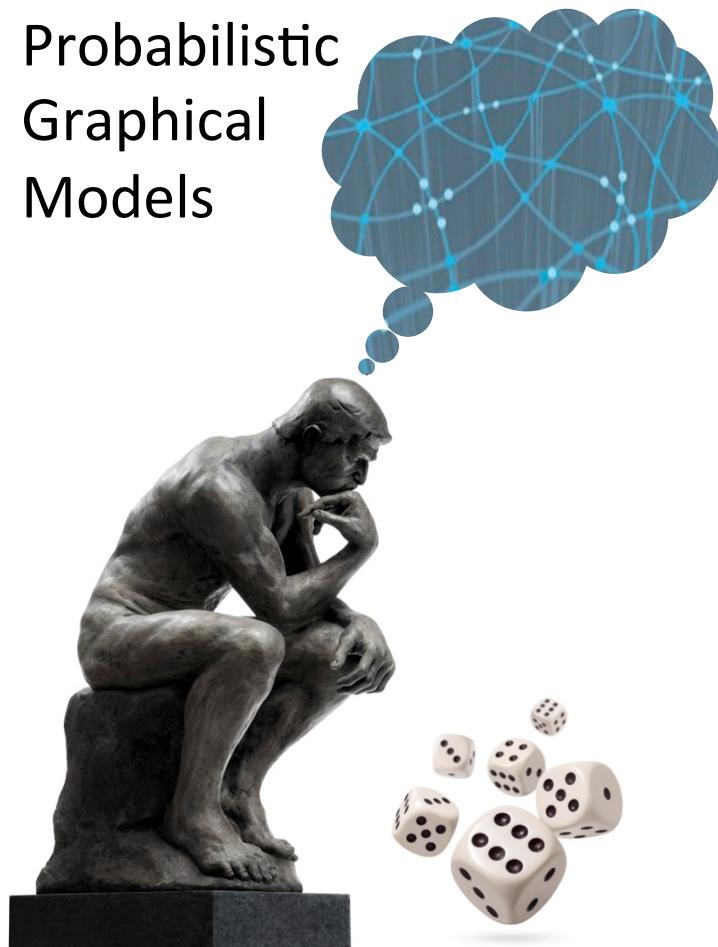


hne Koller

Summary

- Finding the optimal elimination ordering is NP-hard
- Simple heuristics that try to keep induced graph small often provide reasonable performance

Probabilistic
Graphical
Models



Acting

Decision Making

Maximum
Expected
Utility

Simple Decision Making

A simple decision making situation \mathcal{D} :

- A set of possible actions $\text{Val}(\underline{A}) = \{\underline{a^1}, \dots, \underline{a^K}\}$
- A set of states $\text{Val}(\underline{X}) = \{\underline{x^1}, \dots, \underline{x^N}\}$
- A distribution $\underline{P(X | A)}$
- A utility function $\underline{U(X, A)}$

Expected Utility

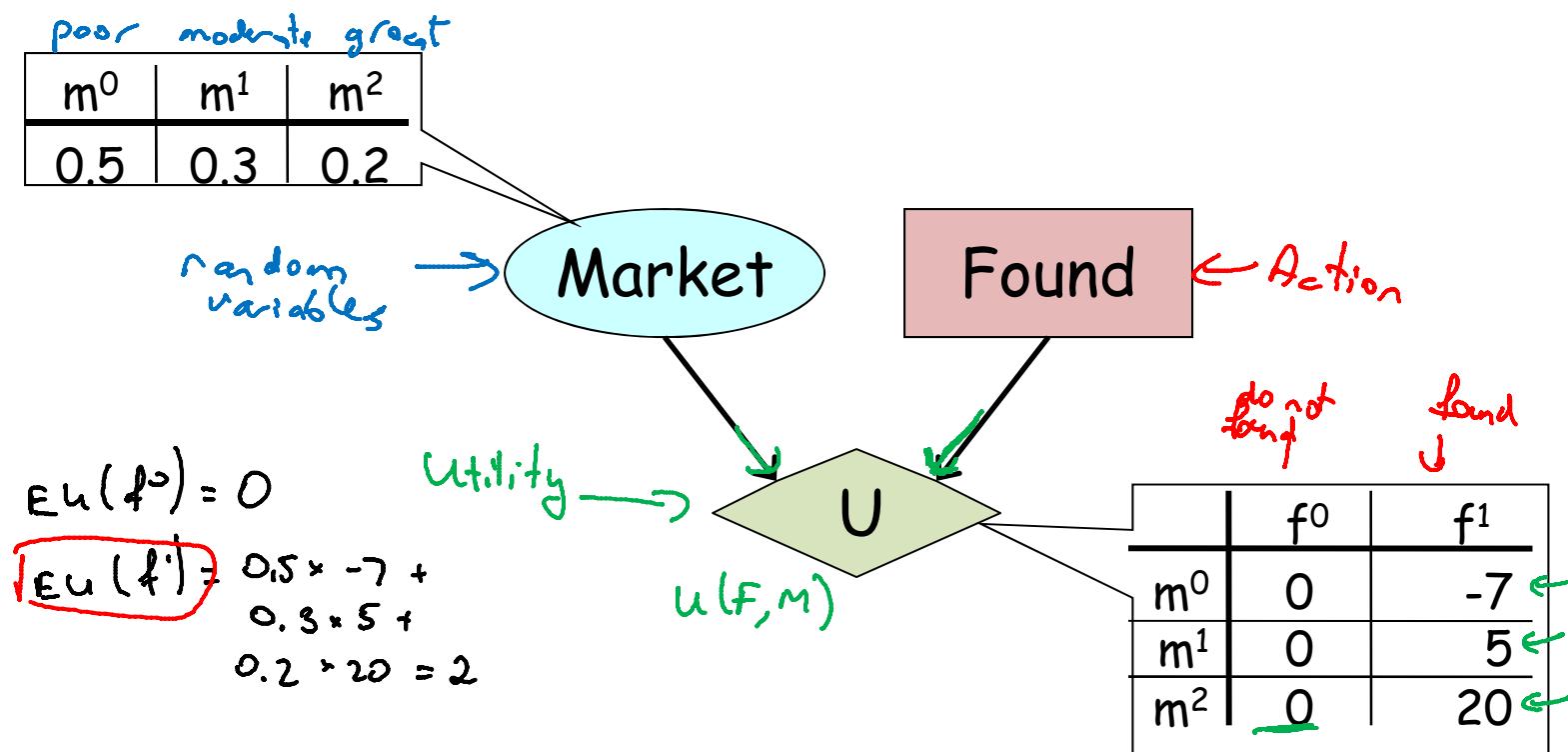
$$\text{EU}[\underline{\mathcal{D}}[a]] = \sum_x P(x | a) \underline{U(x, a)}$$

- Want to choose action \hat{a} that maximizes the expected utility

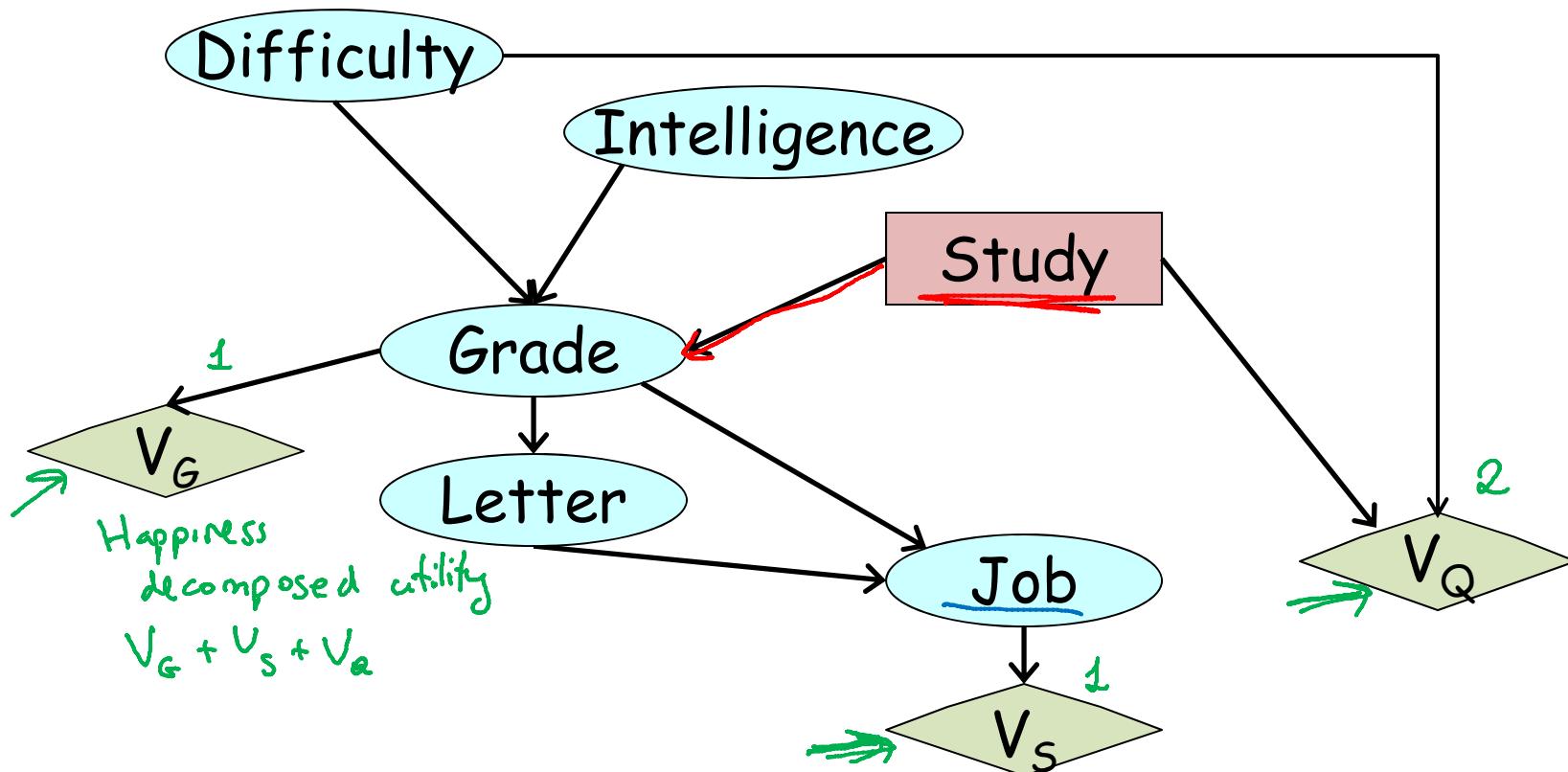
Max. expected utility

$$a^* = \operatorname{argmax}_a \text{EU}[\mathcal{D}[a]]$$

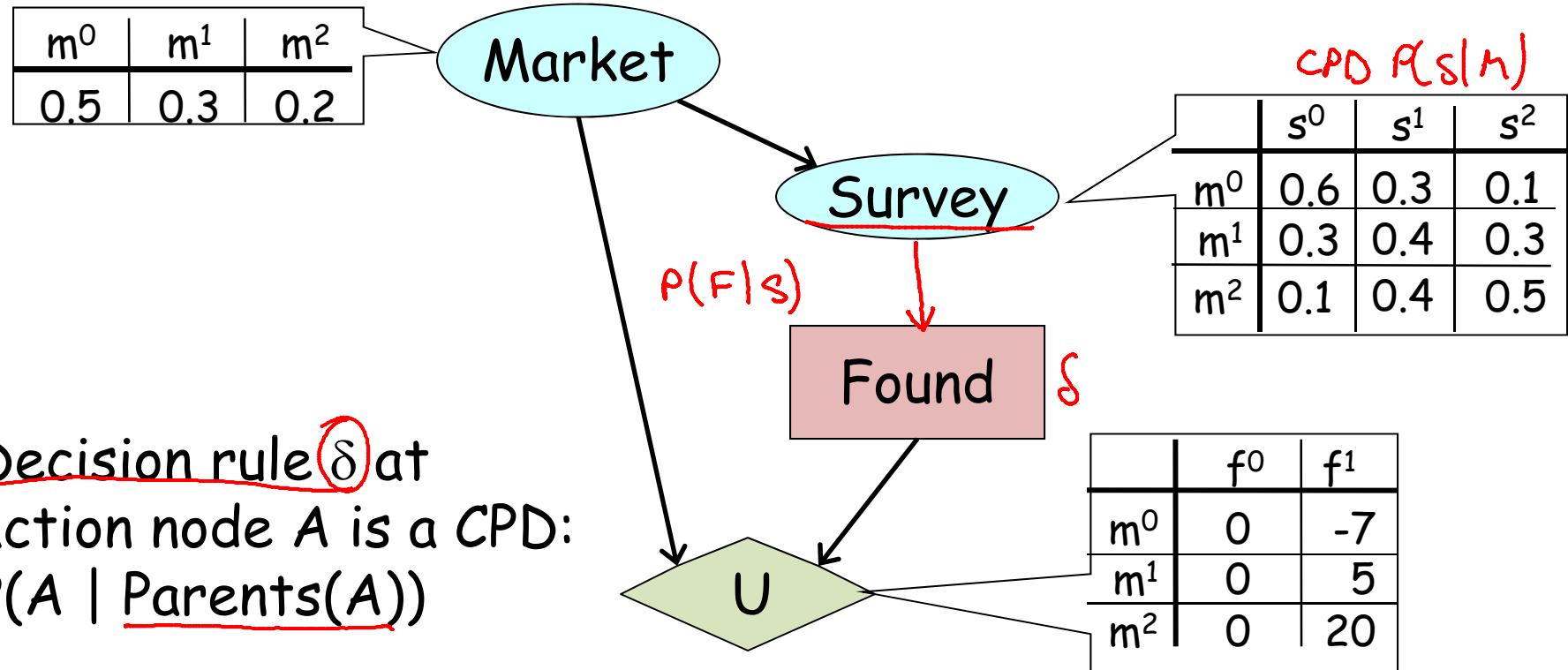
Simple Influence Diagram



More Complex Influence Diagram



Information Edges



Expected Utility with Information

$$\text{EU}[\mathcal{D}[\delta_A]] = \sum_{x,a} \underbrace{P_{\delta_A}(x,a)}_{\text{joint prob. dist over } X \cup \{A\}} \underbrace{U(x,a)}$$

- Want to choose the decision rule δ_A that maximizes the expected utility

$$\operatorname{argmax}_{\delta_A} \text{EU}[\mathcal{D}[\delta_A]]$$

$$\text{MEU}(\mathcal{D}) = \max_{\delta_A} \text{EU}[\mathcal{D}[\delta_A]]$$

Finding MEU Decision Rules

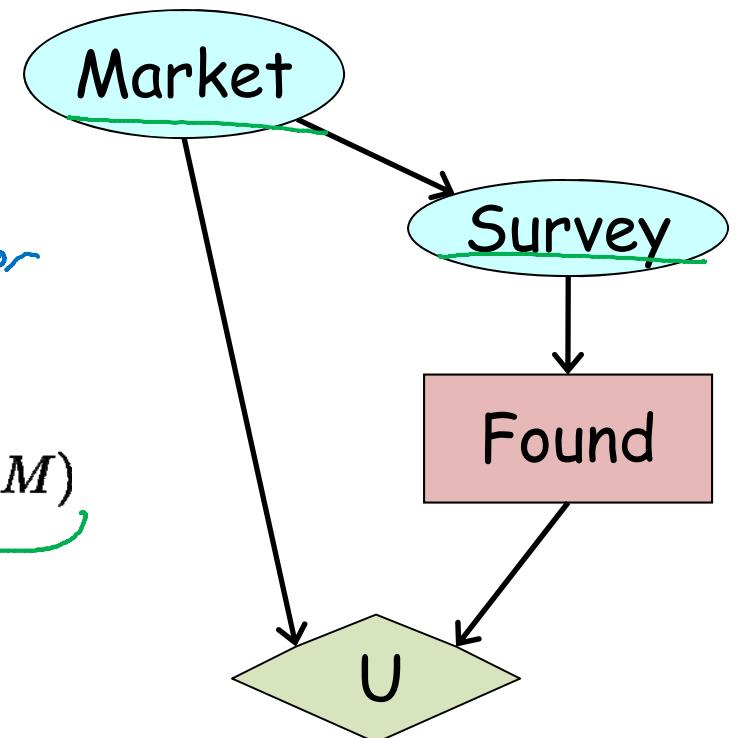
$$\text{EU}[\mathcal{D}[\delta_A]] = \sum_{x,a} P_{\delta_A}(x, a) \underline{U(x, a)}$$

optimize

$$\sum_{M,S,F} \underbrace{P(M)P(S | M)}_{\text{factor}} \underbrace{\delta_F(F | S)U(F, M)}_{\text{factor}} =$$

$$= \sum_{S,F} \underbrace{\delta_F(F | S)}_{\text{factor}} \underbrace{\sum_M P(M)P(S | M)U(F, M)}_{\text{factor}}$$

$$= \sum_{S,F} \underbrace{\delta_F(F | S)}_{\text{factor}} \underbrace{\mu(F, S)}_{\text{factor}}$$



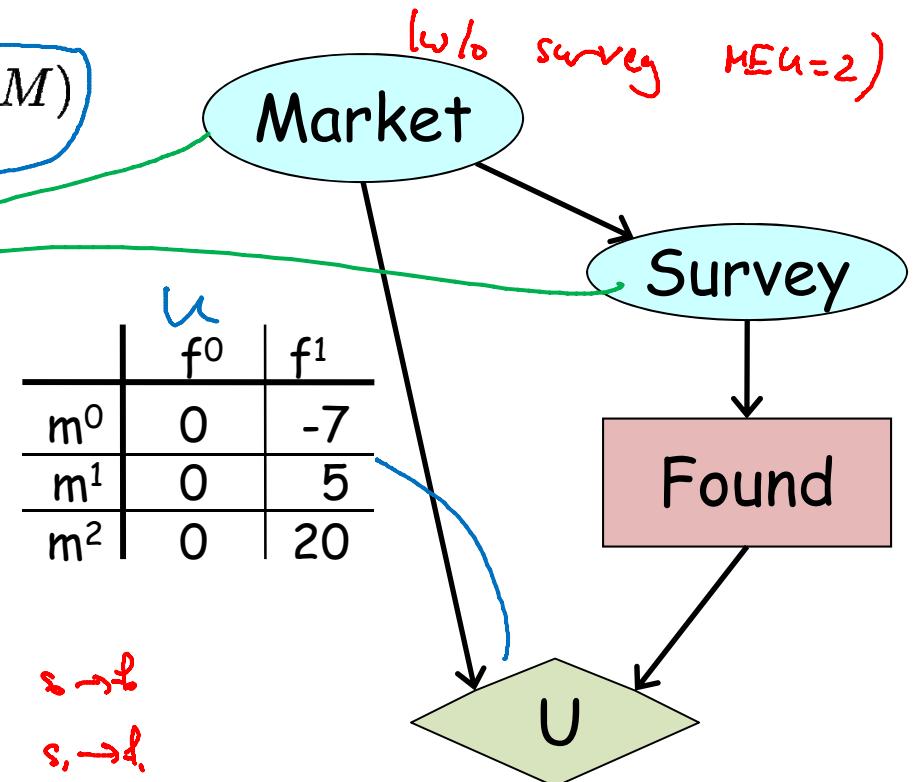
Finding MEU Decision Rules

$$\begin{aligned}
 & \sum_{S,F} \delta_F(F | S) \sum_M P(M) P(S | M) U(F, M) \\
 &= \sum_{S,F} \delta_F(F | S) \mu(F, S)
 \end{aligned}$$

$m^0 \quad m^1 \quad m^2$
 \hline
 $m^0 \quad 0.6 \quad 0.3 \quad 0.1$
 $m^1 \quad 0.3 \quad 0.4 \quad 0.3$
 $m^2 \quad 0.1 \quad 0.4 \quad 0.5$

$f^0 \quad f^1$
 \hline
 $s^0 \quad 0 \quad -1.25$
 $s^1 \quad 0 \quad 1.15$
 $s^2 \quad 0 \quad 2.1$

$+ 0$
 $+ 1.15$
 $+ 2.1$
 $\hline 2.15$



More Generally

$$\text{EU}[\mathcal{D}[\delta_A]] = \sum_{\mathbf{x}, a} P_{\delta_A}(\mathbf{x}, a) \overbrace{U(\mathbf{x}, a)}^{\text{joint dist.}}$$

$$\frac{\mathbf{Z} = \text{Pa}_A}{\mathbf{W} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\} - \mathbf{Z}} \begin{matrix} \text{observations} \\ \text{prior to } A \end{matrix}$$

$$= \sum_{X_1, \dots, X_n, A} \left(\left(\prod_i P(X_i | \text{Pa}_{X_i}) \right) U(\text{Pa}_U) \underbrace{\delta_A(A | \mathbf{Z})}_{\text{prob}} \right)$$

$$= \sum_{\mathbf{Z}, A} \underbrace{\delta_A(A | \mathbf{Z})}_{\text{prob}} \sum_{\mathbf{W}} \left(\left(\prod_i P(X_i | \text{Pa}_{X_i}) \right) U(\text{Pa}_U) \right)$$

$$= \sum_{\mathbf{Z}, A} \delta_A(A | \mathbf{Z}) \underbrace{\mu(A, \mathbf{Z})}_{\text{prob}}$$

$$\delta_A^*(a | \mathbf{z}) = \begin{cases} 1 & a = \text{argmax}_A \mu(A, \mathbf{z}) \\ 0 & \text{otherwise} \end{cases}$$

MEU Algorithm Summary

- To compute MEU & optimize decision at A :
 - Treat A as random variable with arbitrary CPD
 - Introduce utility factor with scope P_{AU}
 - Eliminate all variables except A, Z (A 's parents) to produce factor $\mu(A, Z)$
 - For each z , set:

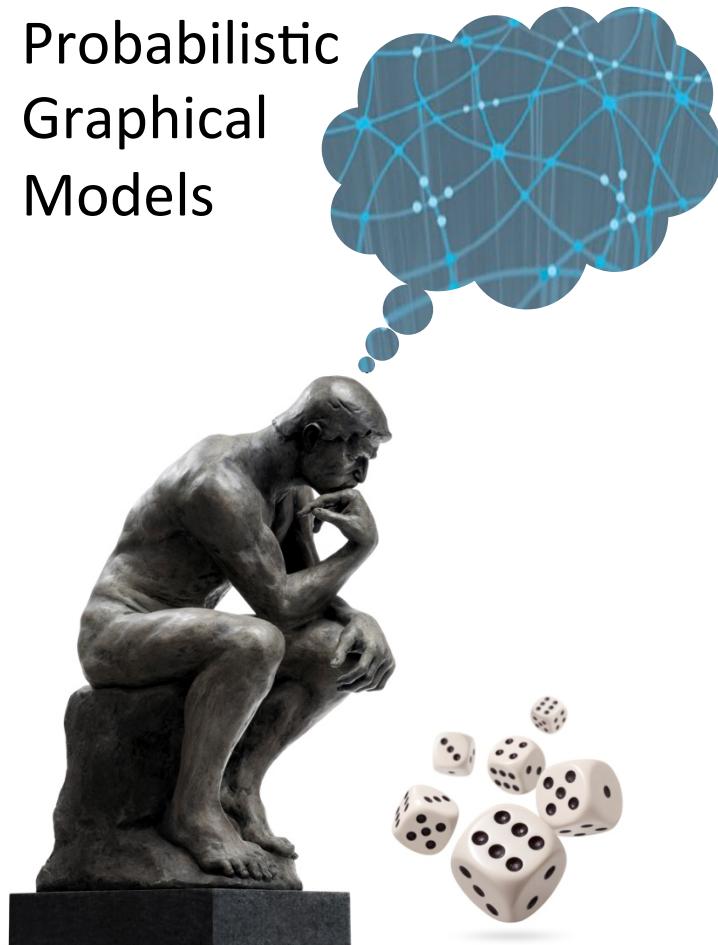
$$\delta_A^*(a | z) = \begin{cases} 1 & a = \operatorname{argmax}_A \mu(A, z) \\ 0 & \text{otherwise} \end{cases}$$

VE

Decision Making under Uncertainty

- MEU principle provides rigorous foundation
- PGMs provide structured representation for probabilities, actions, and utilities
- PGM inference methods (VE) can be used for
 - Finding the optimal strategy
 - Determining overall value of the decision situation
- Efficient methods also exist for:
 - Multiple utility components
 - Multiple decisions

Probabilistic
Graphical
Models



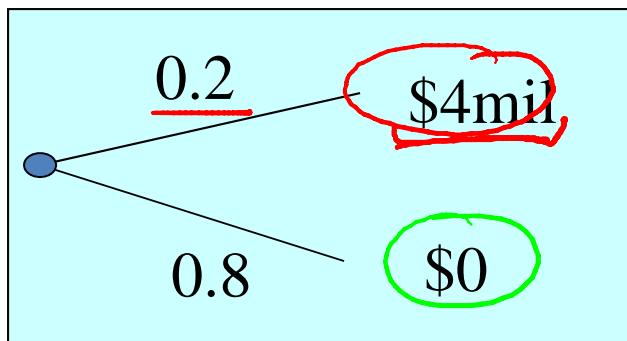
Acting

Decision Making

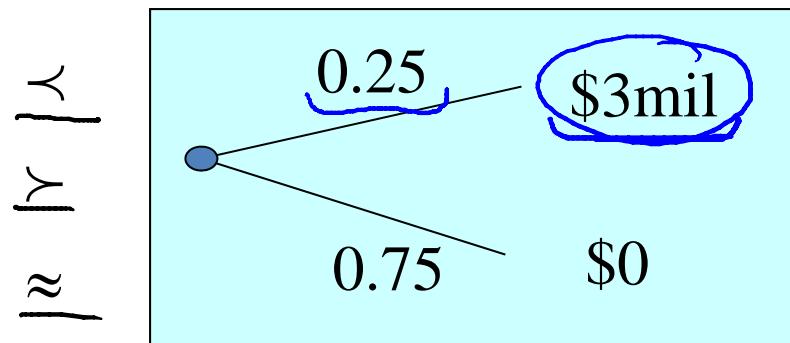
Utility
Functions

Utilities and Preferences

lotteries

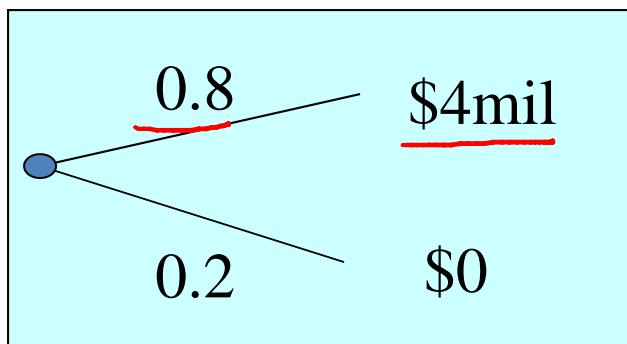


$$0.2 \cdot u(4) + 0.8 \cdot u(0)$$



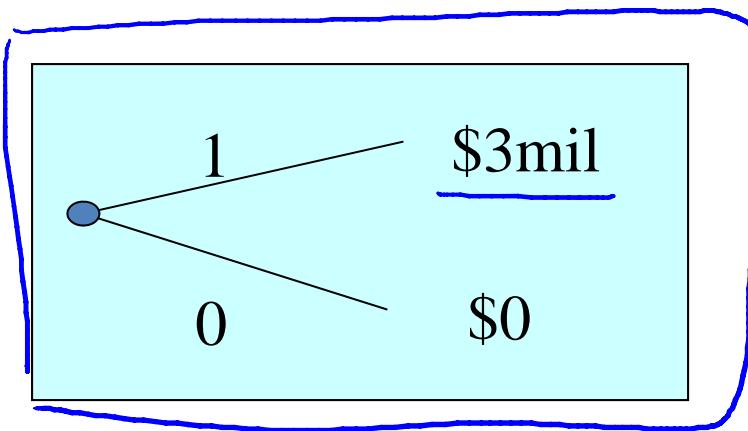
$$0.25 \cdot u(3) + 0.75 \cdot u(0)$$

Utility = Payoff?



$$\begin{aligned} \$4\text{mil} \times 0.8 &= \\ \$3.2\text{mil} & \end{aligned}$$

≈ γ λ



\$3 mil

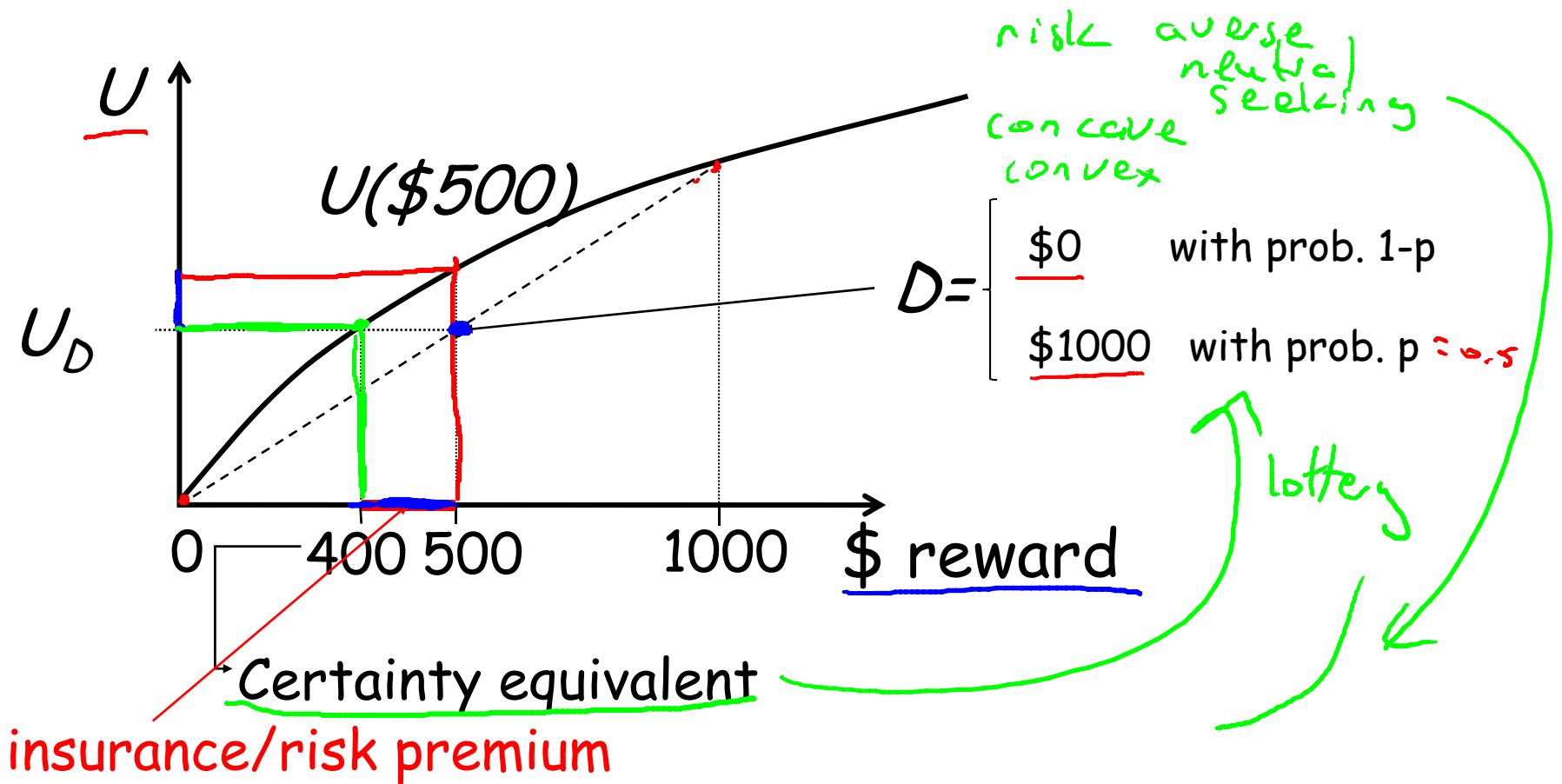
St. Petersburg Paradox



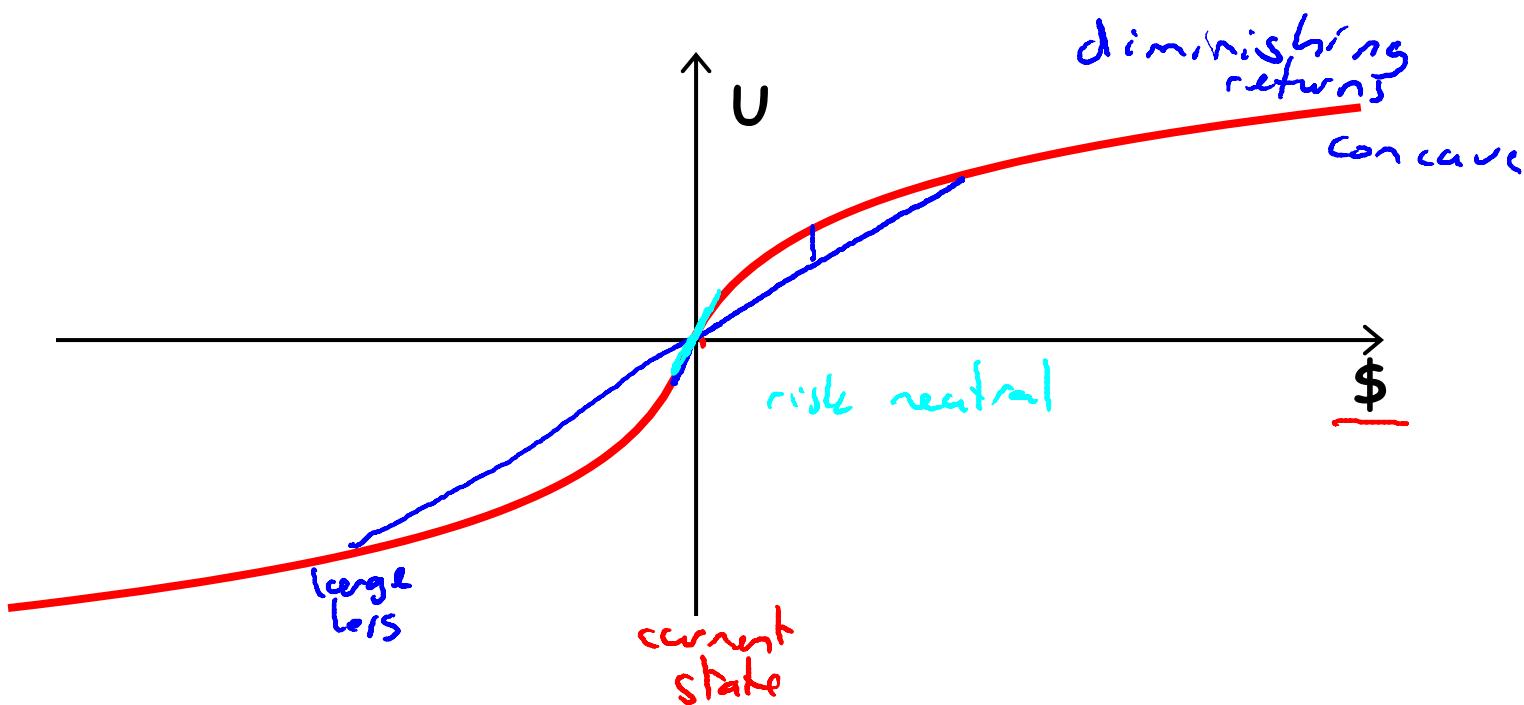
- Fair coin is tossed repeatedly until it comes up heads, say on the n^{th} toss
- Payoff = $\$2^n$

$$\frac{1}{2} \times 2 + \frac{1}{4} \times 4 + \frac{1}{8} \times 8 + \dots = \infty$$

most people value $\approx \$2$



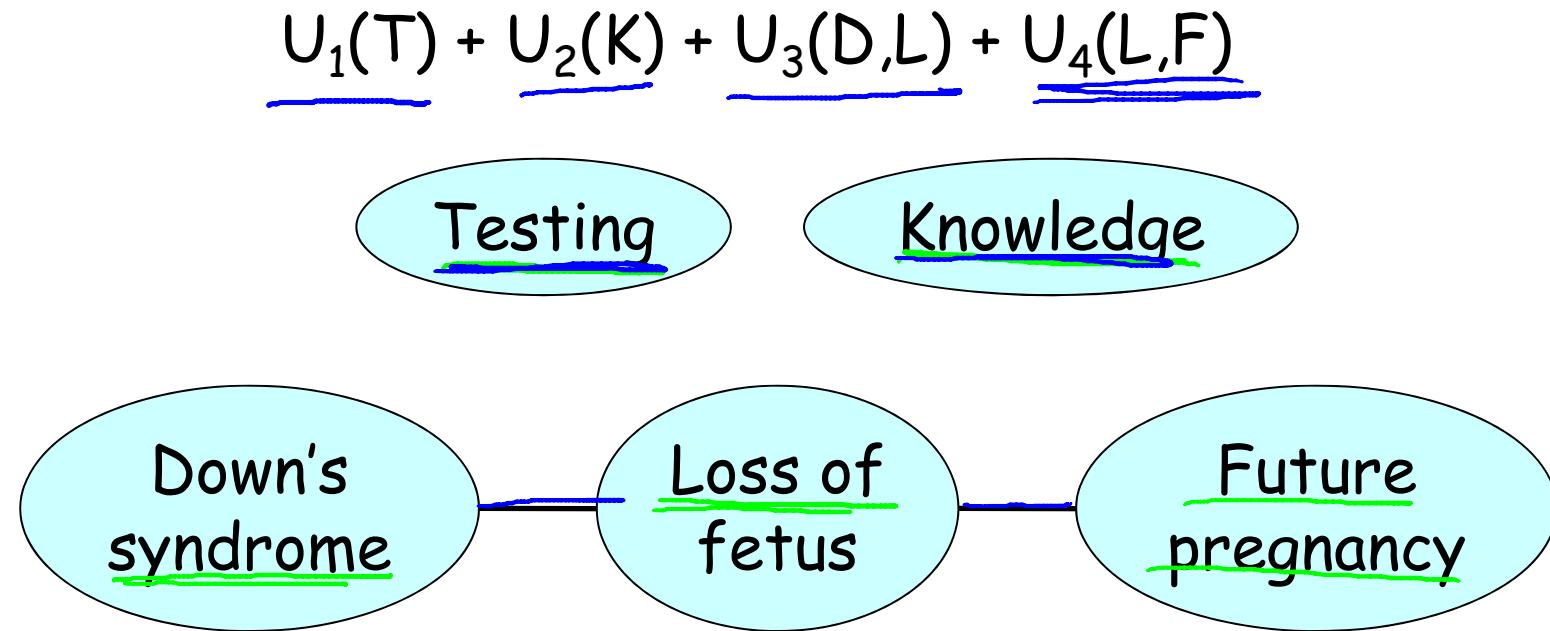
Typical Utility Curve



Multi-Attribute Utilities

- All attributes affecting preferences must be integrated into one utility function
money, time, pleasure, ...
- Human life
 - Micromorts $\frac{1}{100000}$ chance of death $\approx \$20,000$
 - QALY (quality-adjusted life year)

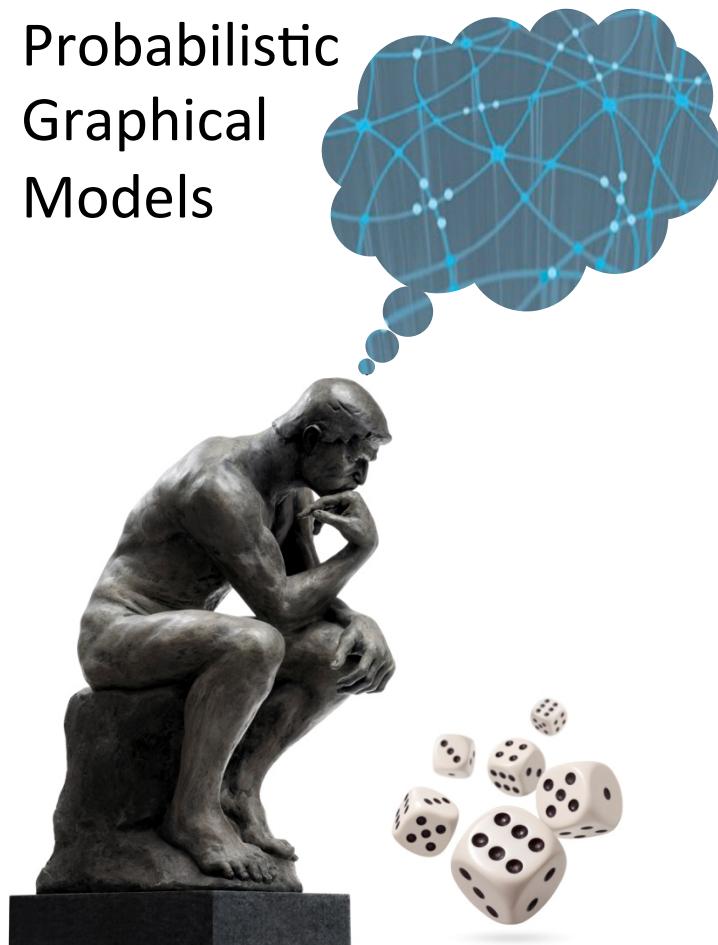
Example: Prenatal diagnosis



Summary

- Our utility function determines our preferences about decisions that involve uncertainty
- Utility generally depends on multiple factors
 - Money, time, chances of death, ...
- Relationship is usually nonlinear
 - Shape of utility curve determines attitude to risk
- Multi-attribute utilities can help decompose high-dimensional function into tractable pieces

Probabilistic
Graphical
Models



Acting

Decision Making

Value of
Perfect
Information

Value of Information

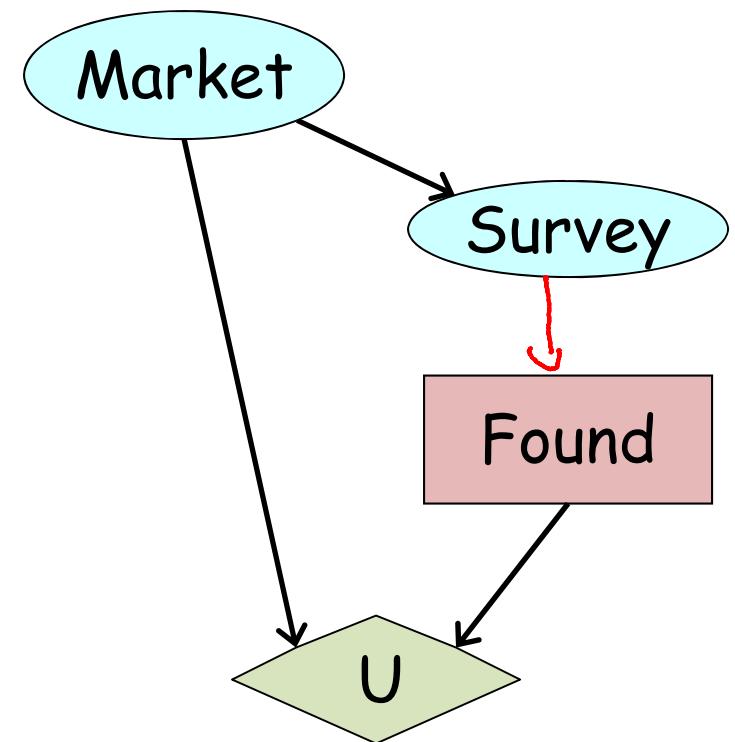
- VPI(A | X) is the value of observing X before choosing an action at A
value of perfect information
- \mathcal{D} = original influence diagram
- $\mathcal{D}_{X \rightarrow A}$ = influence diagram with edge $X \rightarrow A$

$$\text{VPI}(A | X) := \underline{\text{MEU}(\mathcal{D}_{X \rightarrow A}) - \text{MEU}(\mathcal{D})}$$

Finding MEU Decision Rules

$$mEU(D_{S \rightarrow F}) - mEU(D)$$

3.25 2 = 1.25



Value of Information

$$VPI(A | X) := \underbrace{\text{MEU}(\mathcal{D}_{X \rightarrow A}) - \text{MEU}(\mathcal{D})}_{\substack{\text{optimizing } \delta(A|\bar{z}, x) \\ \text{optimizing } \delta(A|z)}}$$

- Theorem:

- $VPI(A | X) \geq 0$
- $VPI(A | X) = 0$ if and only if the optimal decision rule for \mathcal{D} is still optimal for $\mathcal{D}_{X \rightarrow A}$

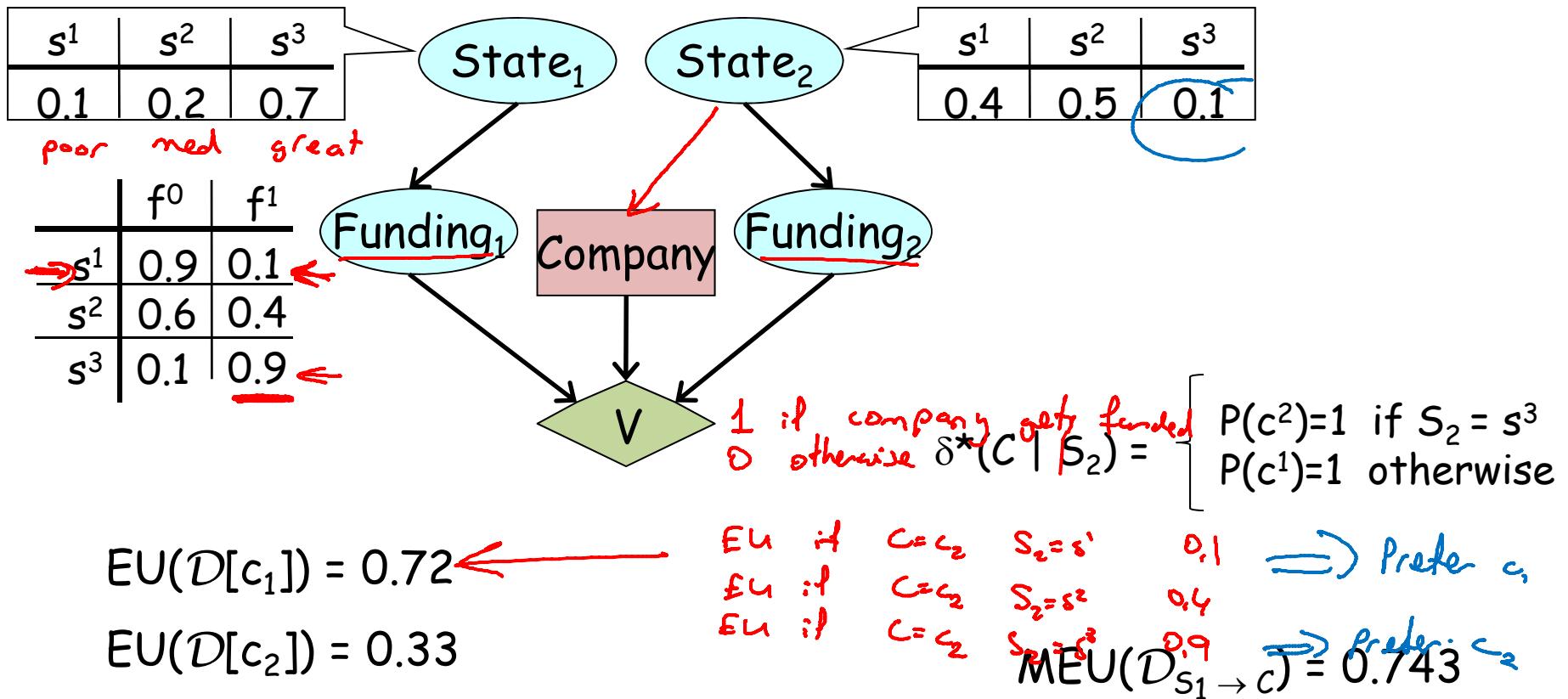
Any CPO $\delta(A|\bar{z})$ is also a CPO $\delta(A|\bar{z}, x)$

Clear notion of when information worth



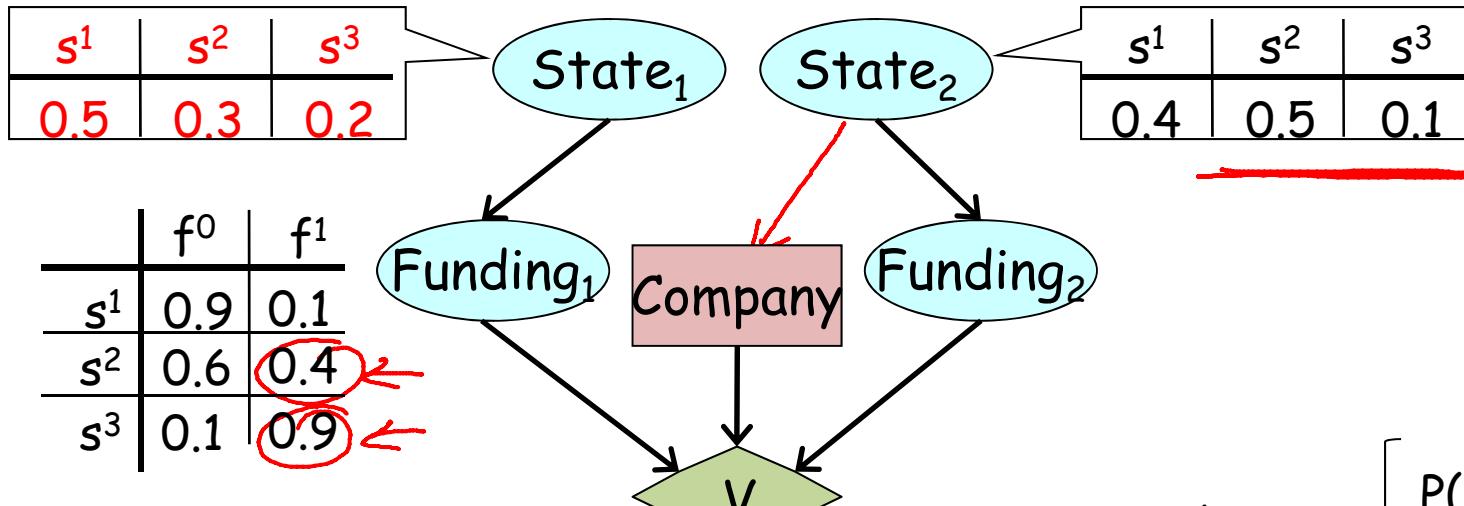
it changes my decision

Value of Information Example



Daphne Koller

Value of Information Example



$$\delta^*(C | S_2) = \begin{cases} P(c^2)=1 & \text{if } S_2 = s^2, s^3 \\ P(c^1)=1 & \text{otherwise} \end{cases}$$

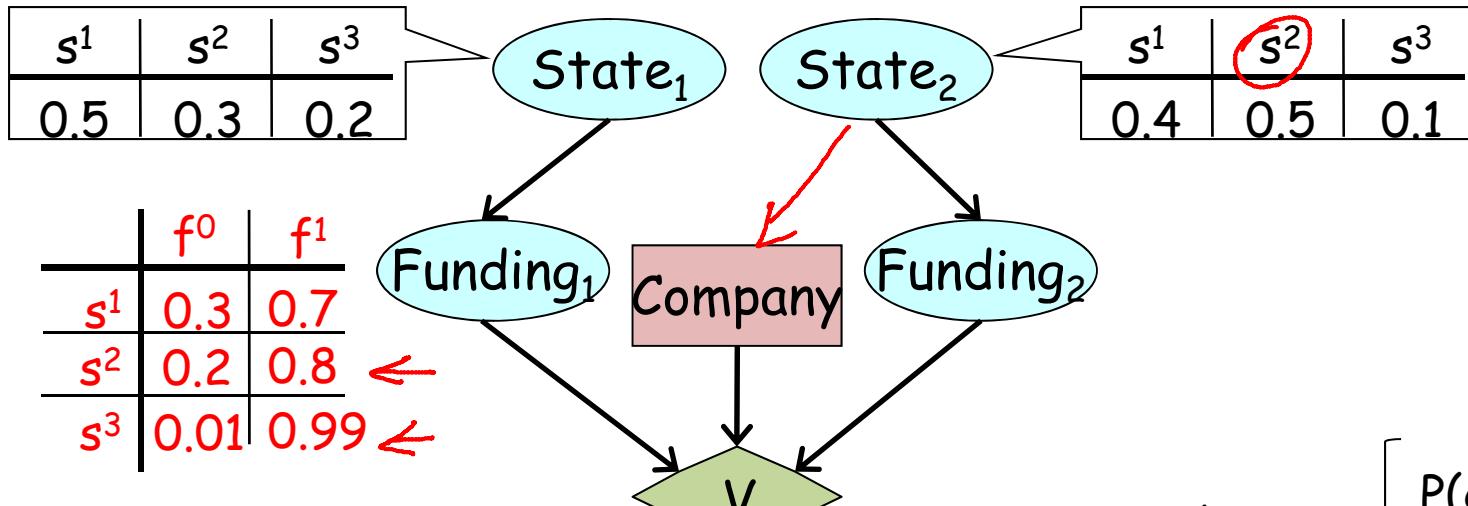
$$EU(D[c_1]) = 0.35$$

$$EU(D[c_2]) = 0.33$$

$$MEU(D_{S_2 \rightarrow C}) = \underline{\underline{0.43}}$$

Daphne Koller

Value of Information Example



$$\delta^*(C | S_2) = \begin{cases} P(c^2)=1 & \text{if } S_2 = s^2, s^3 \\ P(c^1)=1 & \text{otherwise} \end{cases}$$

$$EU(D[c_1]) = 0.788$$

$$EU(D[c_2]) = 0.779$$

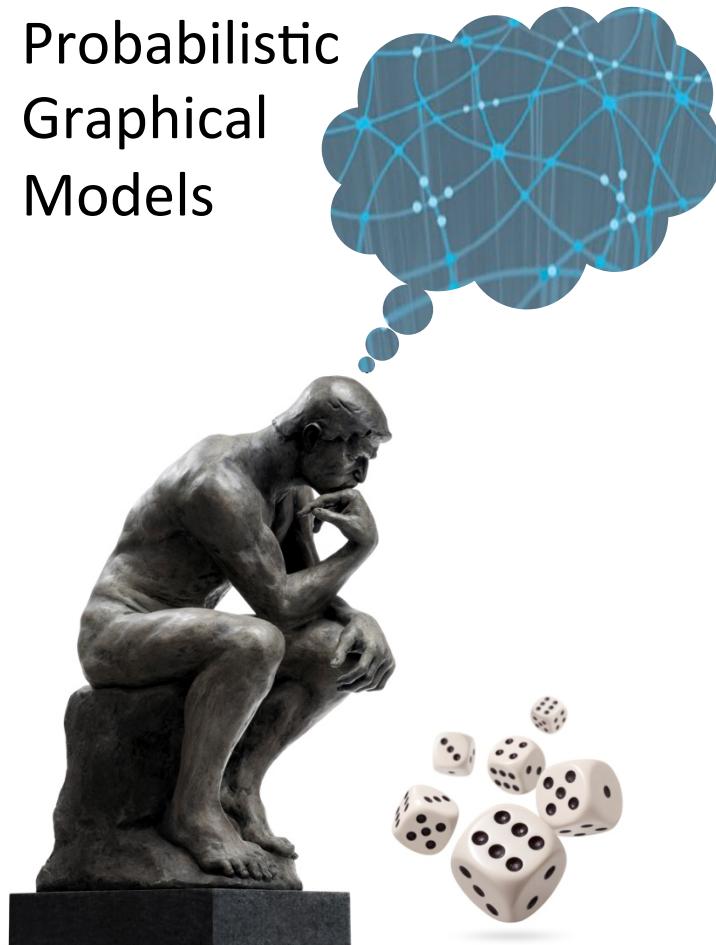
$$MEU(D_{S_1 \rightarrow C}) = \underline{0.8142}$$

Daphne Koller

Summary

- Influence diagrams provide clear and coherent semantics for the value of making an observation
 - Difference between values of two IDs
- Information is valuable if and only if it induces a change in action in at least one context

Probabilistic
Graphical
Models



Learning

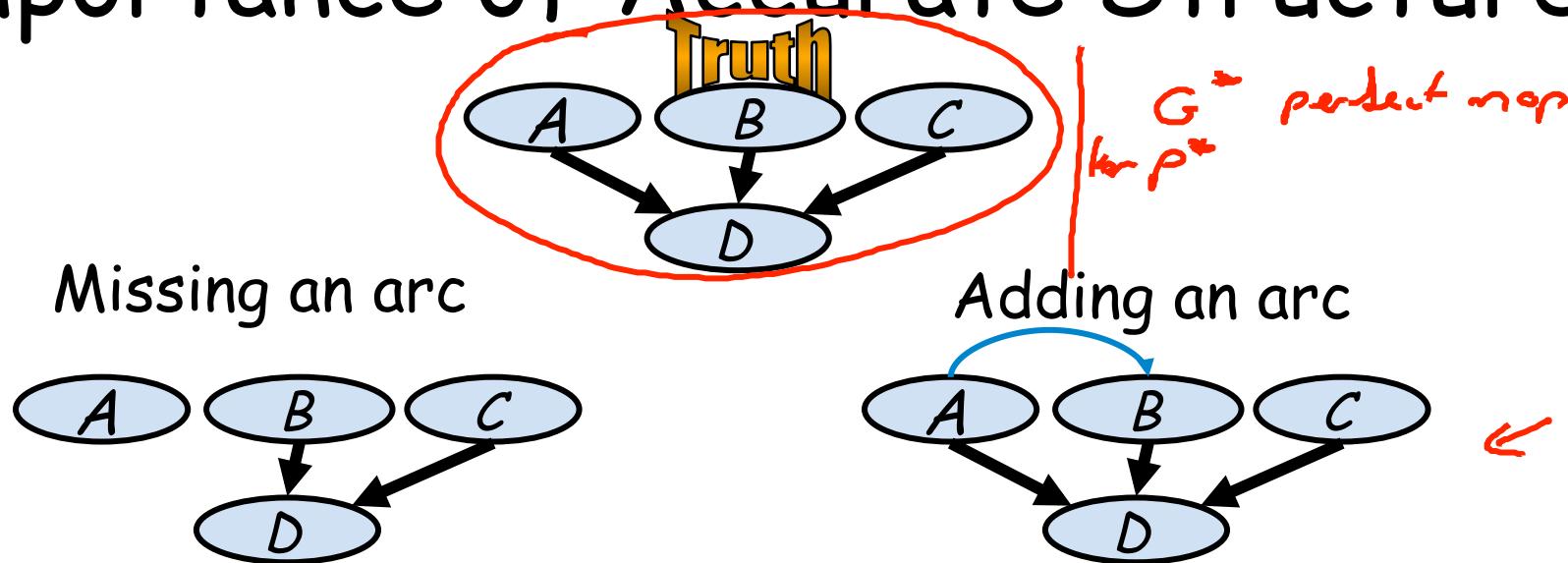
BN Structure

Structure
Learning

Why Structure Learning

- To learn model for new queries, when domain expertise is not perfect
- For structure discovery, when inferring network structure is goal in itself

Importance of Accurate Structure

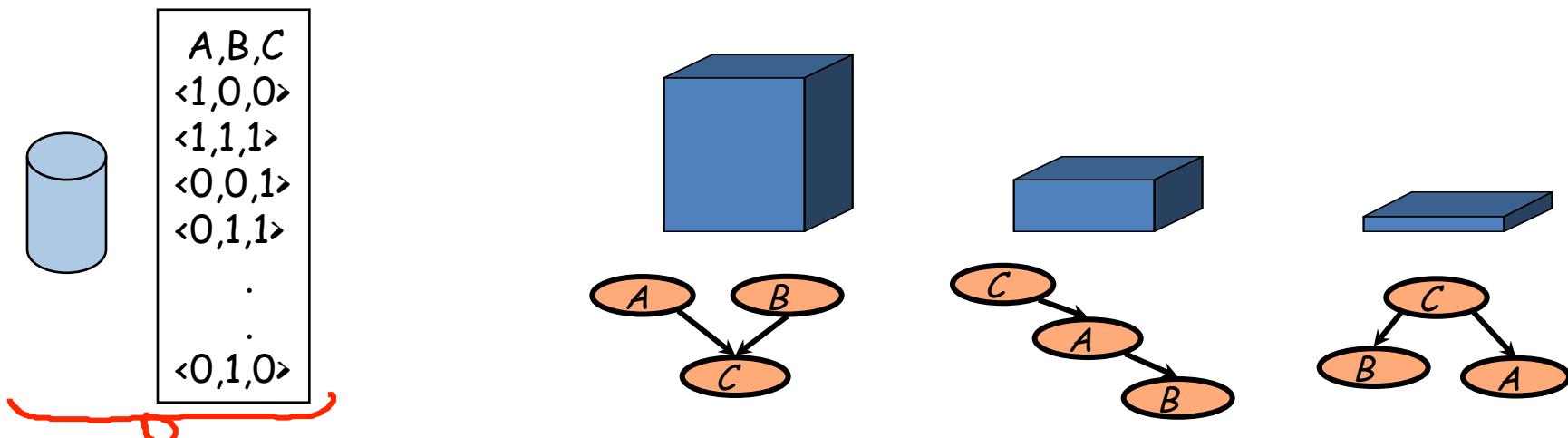


- Incorrect independencies
- Correct distribution P^* cannot be learned
- But could generalize better
- Spurious dependencies
- Can correctly learn P^*
- Increases # of parameters
- Worse generalization

Score-Based Learning

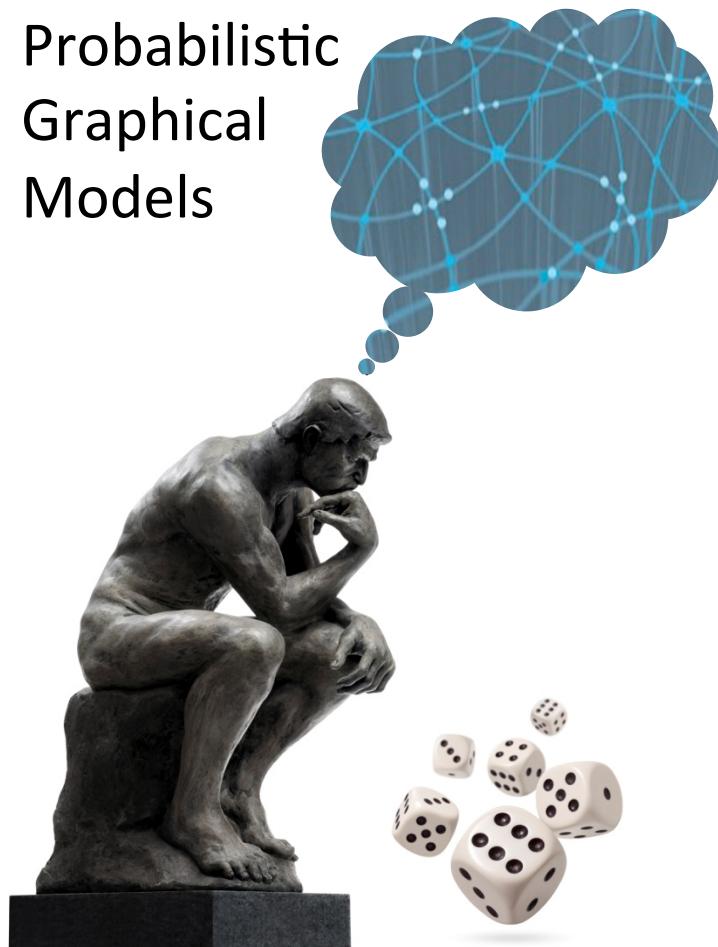
optimization

Define scoring function that evaluates how well a structure matches the data



Search for a structure that maximizes the score

Probabilistic
Graphical
Models



Learning

BN Structurds

Likelihood
Structure
Score

Likelihood Score

- Find (G, θ) that maximize the likelihood

$$\underline{\text{score}_L(G : \mathcal{D})} = \underline{\ell((\hat{\theta}, G) : \mathcal{D})}$$

$\hat{\theta}$ = MLE of params. given G and \mathcal{D}

Example

$$\mathcal{G}_0 \quad \begin{array}{c} X \\ Y \end{array}$$

$$\text{score}_L(\mathcal{G}_0 : \mathcal{D}) = \sum_m (\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]})$$

$$\underline{\text{score}_L(\mathcal{G}_1 : \mathcal{D}) - \text{score}_L(\mathcal{G}_0 : \mathcal{D})} = \sum (\log \hat{\theta}_{y[m]|x[m]} - \log \hat{\theta}_{y[m]})$$

$$= \sum_{x,y} M[x,y] \log \hat{\theta}_{y|x} - \sum_y M[y] \log \hat{\theta}_y$$

$$= M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y|x) - M \sum_y \hat{P}(y) \log \hat{P}(y)$$

$$= M \left(\sum_{x,y} \hat{P}(x,y) \log \hat{P}(y|x) - \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y) \right)$$

$$= M \left(\sum_{x,y} \hat{P}(x,y) \log \frac{\hat{P}(x,y)}{\hat{P}(x)\hat{P}(y)} \right) = M \cdot \underline{\mathbf{I}_{\hat{P}}(X;Y)}$$

$$\mathcal{G}_1 \quad \begin{array}{c} X \rightarrow Y \end{array}$$

$$\text{score}_L(\mathcal{G}_1 : \mathcal{D}) = \sum_m (\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]})$$

\hat{P} = empirical distribution

$$M[x,y] = M \hat{P}[x,y]$$

$$\sum_x \hat{P}(x,y) = \hat{P}[y]$$

mutual information

General Decomposition

- The Likelihood score decomposes as:

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^n \mathbf{I}_{\hat{P}}(X_i; \text{Pa}_{X_i}^{\mathcal{G}}) - M \sum_i H_{\hat{P}}(X_i)$$

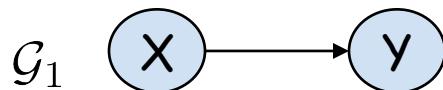
mutual information *independent
and G*

$$\mathbf{I}_P(X; Y) = \sum_{x,y} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}$$

*Score is higher
if x_i is correlated
with parents*

$$H_P(X) = - \sum_x P(x) \log P(x)$$

Limitations of Likelihood Score



$$\underline{\text{score}_L(\mathcal{G}_1 : \mathcal{D})} - \underline{\text{score}_L(\mathcal{G}_0 : \mathcal{D})} = \underline{\text{MI}_{\hat{P}}(X; Y)}$$

- Mutual information is always ≥ 0
- Equals 0 iff X, Y are independent
 - In empirical distribution \hat{P} $I_{\hat{P}}(x; y) > 0$ almost always
- Adding edges can't hurt, and almost always helps
- Score maximized for fully connected network

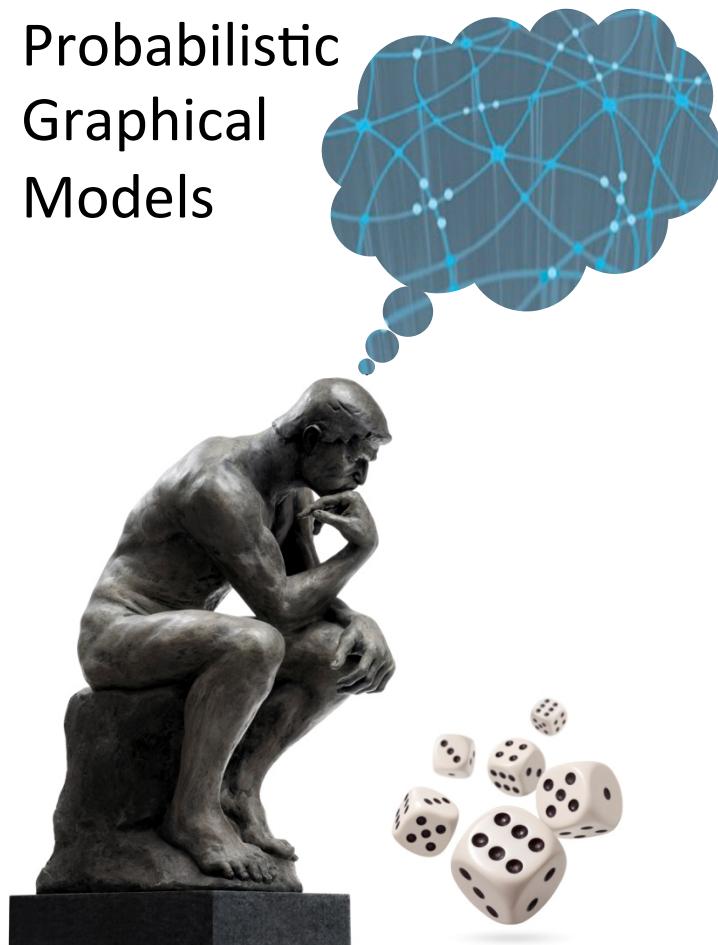
Avoiding Overfitting

- Restricting the hypothesis space
 - restrict # of parents, or # of parameters
- Scores that penalize complexity:
 - Explicitly ←
 - Bayesian score averages over all possible parameter values

Summary

- Likelihood score computes log-likelihood of D relative to G , using MLE parameters $\hat{\theta}$ for G
 - Parameters optimized for D
- Nice information-theoretic interpretation in terms of (in)dependencies in G
- Guaranteed to overfit the training data (if we don't impose constraints)

Probabilistic
Graphical
Models



Learning

BN Structure

BIC Score and
Asymptotic
Consistency

Penalizing Complexity

Bayesian information criterion

$$\text{score}_{BIC}(\mathcal{G} : \mathcal{D}) = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2} \frac{\text{# training instances}}{\text{Dim}[\mathcal{G}]}$$

Score_L(G : D) # independent params,

- Tradeoff between fit to data and model complexity

*(MDL criterion)
minimum description length*

Asymptotic Behavior

$$\ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2} \text{Dim}[\mathcal{G}]$$

$$M \sum_{i=1}^n \mathbf{I}_{\hat{P}}(X_i; \mathbf{Pa}_{X_i}^G) - M \sum_i \mathbf{H}_{\hat{P}}(X_i) - \frac{\log M}{2} \text{Dim}[\mathcal{G}]$$

independent of \mathcal{G}

- Mutual information grows linearly with M while complexity grows logarithmically with M
 - As M grows, more emphasis is given to fit to data

$\hat{P} \rightarrow P^*$

Consistency

G^*

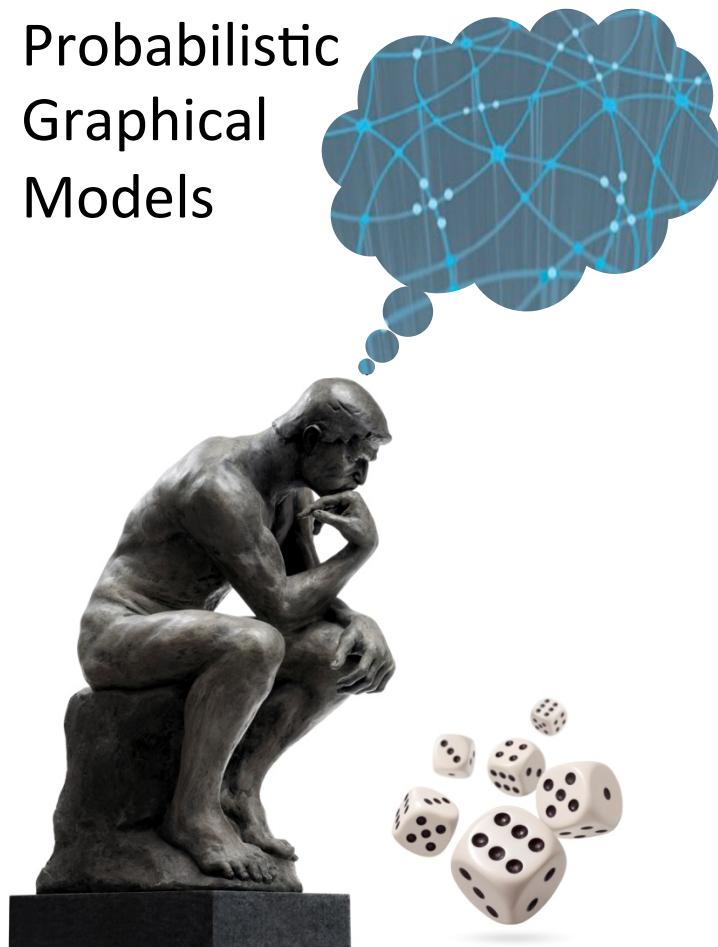
$$\begin{aligned} & \rightarrow M \sum_{i=1}^n I_{\hat{P}}(X_i; \text{Pa}_{X_i}^{G^*}) - M \sum_i H_{\hat{P}}(X_i) - \frac{\log M}{2} \text{Dim}[G] \\ & \rightarrow \sim L_{\hat{P}}(x_i; P_{\text{Pa}_{X_i}^{G^*}}) \end{aligned}$$

- As $M \rightarrow \infty$, the true structure G^* (or any I-equivalent structure) maximizes the score
 - Asymptotically, spurious edges will not contribute to likelihood and will be penalized
 - Required edges will be added due to linear growth of likelihood term compared to logarithmic growth of model complexity

Summary

- BIC score explicitly penalizes model complexity (# of independent parameters)
 - Its negation often called MDL
- BIC is asymptotically consistent:
 - If data generated by G^* , networks I-equivalent to G^* will have highest score as M grows to ∞

Probabilistic
Graphical
Models



Learning

BN Structure

Bayesian
Score

Bayesian Score

Marginal likelihood

Prior over structures

$$\rightarrow P(\mathcal{G} | \mathcal{D}) = \frac{P(\mathcal{D} | \mathcal{G})P(\mathcal{G})}{P(\mathcal{D})}$$

*independent
of \mathcal{G}*

Marginal probability of Data

$$\underline{\text{score}_B(\mathcal{G} : \mathcal{D})} = \underline{\log P(\mathcal{D} | \mathcal{G})} + \underline{\log P(\mathcal{G})}$$

Marginal Likelihood of Data Given \mathcal{G}

$$\text{score}_B(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D} | \mathcal{G}) + \log P(\mathcal{G})$$

$$P(\mathcal{D} | \mathcal{G}) = \int P(\mathcal{D} | \mathcal{G}, \theta_{\mathcal{G}}) P(\theta_{\mathcal{G}} | \mathcal{G}) d\theta_{\mathcal{G}}$$

Diagram illustrating the components of the Marginal Likelihood:

- Likelihood: $P(\mathcal{D} | \mathcal{G}, \theta_{\mathcal{G}})$
- Priors over parameters: $P(\theta_{\mathcal{G}} | \mathcal{G})$

The diagram shows the Marginal Likelihood formula with annotations. A red underline is placed under the entire integral. Two blue callout boxes point to the terms: one labeled "Likelihood" pointing to the first term $P(\mathcal{D} | \mathcal{G}, \theta_{\mathcal{G}})$, and another labeled "Prior over parameters" pointing to the second term $P(\theta_{\mathcal{G}} | \mathcal{G})$.

Marginal Likelihood Intuition

$$\underline{P(\mathcal{D} \mid \mathcal{G})} = \underline{P(x[1], \dots, x[M] \mid \mathcal{G})}$$

$$P(x[1] \mid \mathcal{G}) \leftarrow$$

$$P(x[2] \mid x[1], \mathcal{G}) \leftarrow$$

...

$$P(\underline{x[M]} \mid \underline{x[1], \dots, x[M-1]}, \mathcal{G}) \leftarrow$$

$\hat{\theta}_g$ depends
on all of \mathcal{D}

Marginal Likelihood: BayesNets

$$P(\underline{\mathcal{D}} \mid \mathcal{G}) = \prod_i \left(\prod_{\mathbf{u}_i \in Val(\text{Pa}_{X_i}^{\mathcal{G}})} \underbrace{\frac{\Gamma(\alpha_{X_i} | \mathbf{u}_i)}{\Gamma(\alpha_{X_i} | \mathbf{u}_i + M[\mathbf{u}_i])}}_{\text{sub stats}} \right) \underbrace{\prod_{x_i^j \in Val(X_i)} \left[\frac{\Gamma(\alpha_{x_i^j} | \mathbf{u}_i + M[x_i^j, \mathbf{u}_i])}{\Gamma(\alpha_{x_i^j} | \mathbf{u}_i)} \right]}_{\text{prior}} \Bigg)$$

variables

prior

sub stats

sub stats

prior

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

$$\Gamma(x) = x \cdot \Gamma(x - 1)$$

Marginal Likelihood Decomposition

$$P(\mathcal{D} \mid \mathcal{G}) = \prod_i \left(\prod_{\mathbf{u}_i \in Val(\mathbf{Pa}_{X_i}^{\mathcal{G}})} \frac{\Gamma(\alpha_{X_i|\mathbf{u}_i})}{\Gamma(\alpha_{X_i|\mathbf{u}_i} + M[\mathbf{u}_i])} \prod_{x_i^j \in Val(X_i)} \left[\frac{\Gamma(\alpha_{x_i^j|\mathbf{u}_i} + M[x_i^j, \mathbf{u}_i])}{\Gamma(\alpha_{x_i^j|\mathbf{u}_i})} \right] \right)$$

$$\log P(\mathcal{D} \mid \mathcal{G}) = \sum_i \text{FamScore}_B(X_i \mid \mathbf{Pa}_{X_i}^{\mathcal{G}} : \mathcal{D})$$

Structure Priors

$$\text{score}_B(\mathcal{G} : \mathcal{D}) = \underbrace{\log P(\mathcal{D} | \mathcal{G})}_{\text{Evidence Term}} + \underbrace{\log P(\mathcal{G})}_{\text{Structure Prior}}$$

- **Structure prior $P(\mathcal{G})$**
 - Uniform prior: $P(\mathcal{G}) \propto \text{constant}$
 - Prior penalizing # of edges: $P(\mathcal{G}) \propto c^{|\mathcal{G}|}$ ($0 < c < 1$)
 - Prior penalizing # of parameters
- Normalizing constant across networks is $\rho(s)$
similar and can thus be ignored

Parameter Priors

- Parameter prior $P(\theta|G)$ is usually the BDe prior
 - α : equivalent sample size
 - B_0 : network representing prior probability of events
 - Set $\alpha(x_i, pa_i^G) = \alpha P(x_i, pa_i^G | B_0)$
 - Note: pa_i^G are not the same as parents of X_i in B_0)
- A single network provides priors for all candidate networks
- Unique prior with the property that I-equivalent networks have the same Bayesian score

BDe and BIC

- As $M \rightarrow \infty$, a network G with Dirichlet priors satisfies

$$\log P(\mathcal{D} | \mathcal{G}) = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2} \text{Dim}[\mathcal{G}] + O(1)$$

likelihood score of \mathcal{D} given MLE $\hat{\theta}_{\mathcal{G}}$

#instances

#independent parameters

constant relative to m grows w.r.t m

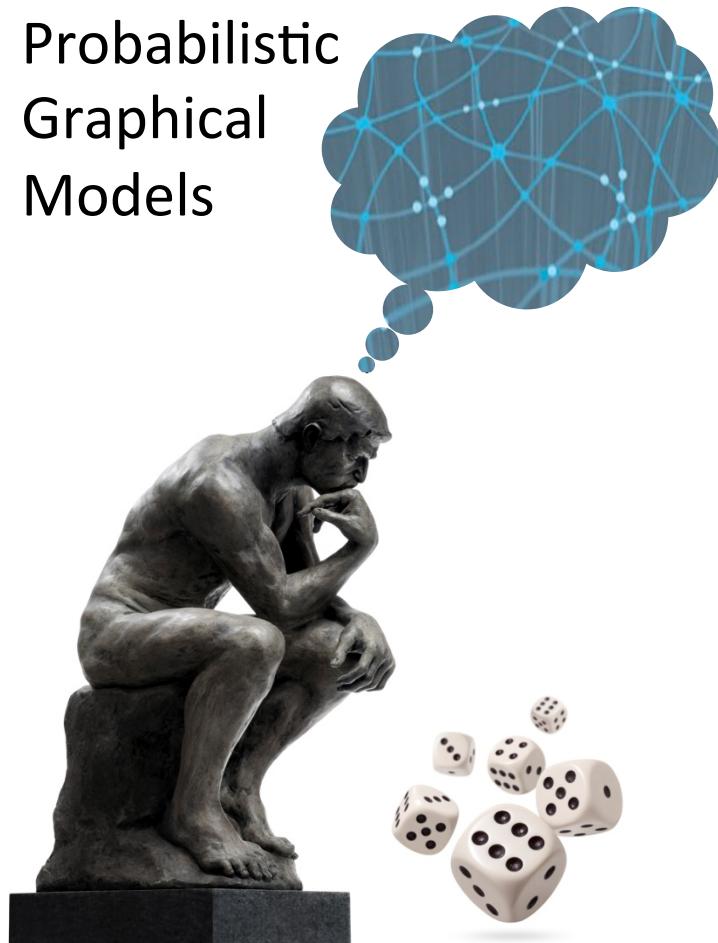
$$\text{score}_{BIC}(\mathcal{G} : \mathcal{D}) = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2} \text{Dim}[\mathcal{G}]$$

as $m \rightarrow \infty$ score is consistent

Summary

- Bayesian score averages over parameters to avoid overfitting
- Most often instantiated as BDe
 - BDe requires assessing prior network
 - Can naturally incorporate prior knowledge
 - I-equivalent networks have same score
- Bayesian score
 - Asymptotically equivalent to BIC (as $m \rightarrow \infty$)
 - Asymptotically consistent learn, current network $\xrightarrow{m \rightarrow \infty}$
 - But for small M, BIC tends to underfit

Probabilistic
Graphical
Models



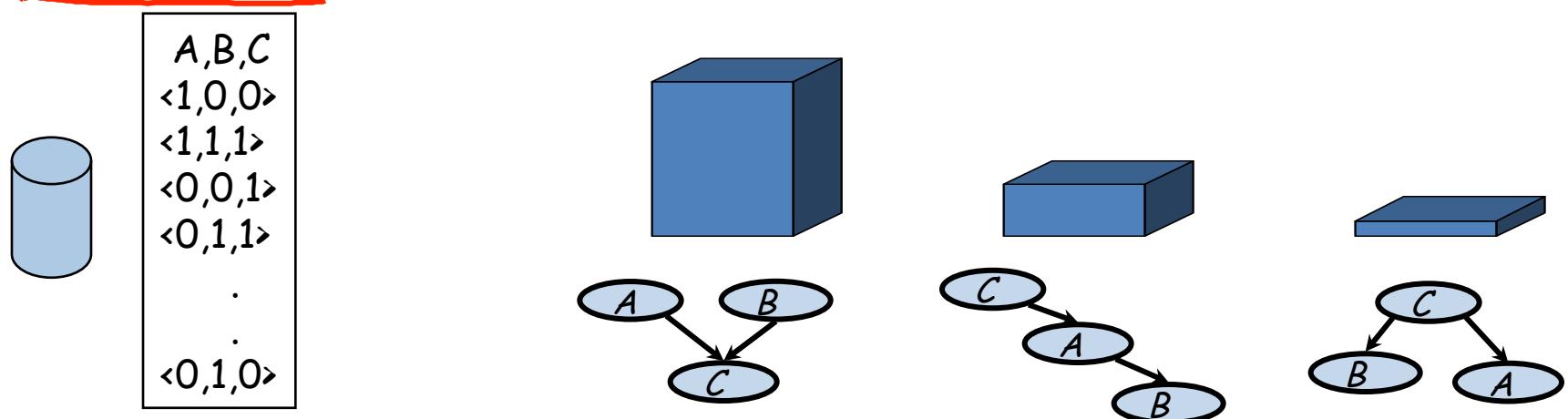
Learning

BN Structure

Structure
Learning In
Trees

Score-Based Learning

Define scoring function that evaluates how well a structure matches the data



Search for a structure that maximizes the score

Optimization Problem

Input:

- Training data
- Scoring function (including priors, if needed)
- Set of possible structures

Output: A network that maximizes the score

Key Property: Decomposability

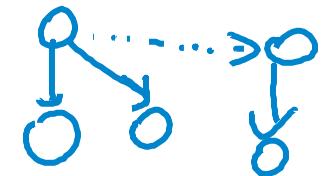
$$\text{score}(\mathcal{G} : \mathcal{D}) = \sum_i \text{score}(X_i | \text{Pa}_{X_i}^{\mathcal{G}} : \mathcal{D})$$

Daphne Koller

Learning Trees/Forests

- **Forests**

- At most one parent per variable



- Why trees?

- Elegant math

- Efficient optimization

- Sparse parameterization \Rightarrow overfit less
more small relative to n

Learning Forests

- $p(i)$ = parent of X_i , or 0 if X_i has no parent

$$\text{score}(\mathcal{G} : \mathcal{D}) = \sum_i \text{score}(X_i | \text{Pa}_{X_i}^{\mathcal{G}} : \mathcal{D})$$

$$\begin{aligned}
 &= \sum_{i:p(i)>0} \text{score}(X_i | X_{p(i)} : \mathcal{D}) + \sum_{i:p(i)=0} \text{score}(X_i : \mathcal{D}) \\
 &= \sum_{i:p(i)>0} (\text{score}(X_i | X_{p(i)} : \mathcal{D}) - \text{score}(X_i : \mathcal{D})) + \sum_{i=1}^n \text{score}(X_i : \mathcal{D})
 \end{aligned}$$

Improvement over "empty" network
Score of "empty" network

have parent → parent
 same for all trees

- Score = sum of edge scores + constant

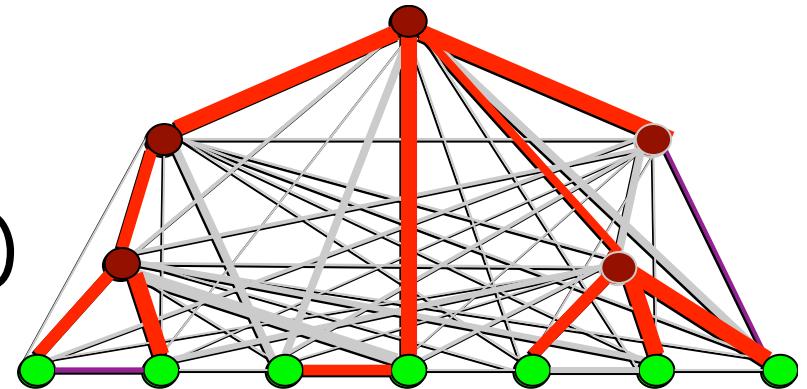
Learning Forests I

- Set $w(i \rightarrow j) = \boxed{\text{Score}(\underline{X}_j | \underline{X}_i) - \text{Score}(\underline{X}_j)}$
- For likelihood score, $w(i \rightarrow j) = M \hat{I}_P(X_i; X_j)$,
and all edge weights are nonnegative
 - ⇒ Optimal structure is always a tree
- For BIC or BDe, weights can be negative
 - ⇒ Optimal structure might be a forest

Learning Forests II

- A score satisfies score equivalence if I-equivalent structures have the same score
 - Such scores include likelihood, BIC, and BDe
- For such a score, we can show $w(i \rightarrow j) = w(j \rightarrow i)$, and use an undirected graph

Learning Forests III (for score-equivalent scores)

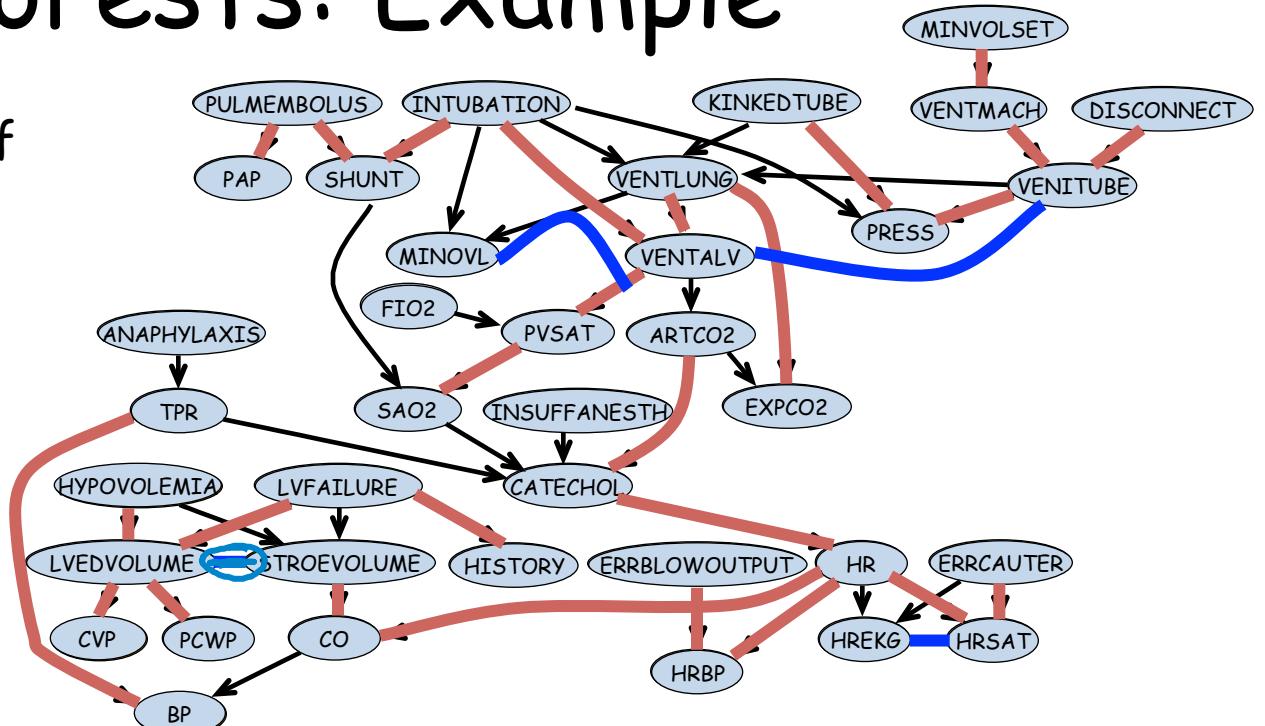


- Define undirected graph with nodes $\{1, \dots, n\}$
- Set $w(i,j) = \max[Score(X_j | X_i) - Score(X_j), 0]$
- Find forest with maximal weight
 - Standard algorithms for max-weight spanning trees (e.g., Prim's or Kruskal's) in $\underline{O(n^2)}$ time
 - Remove all edges of weight 0 to produce a forest

Learning Forests: Example

Tree learned from data of
Alarm network

- Correct edges
- Spurious edges

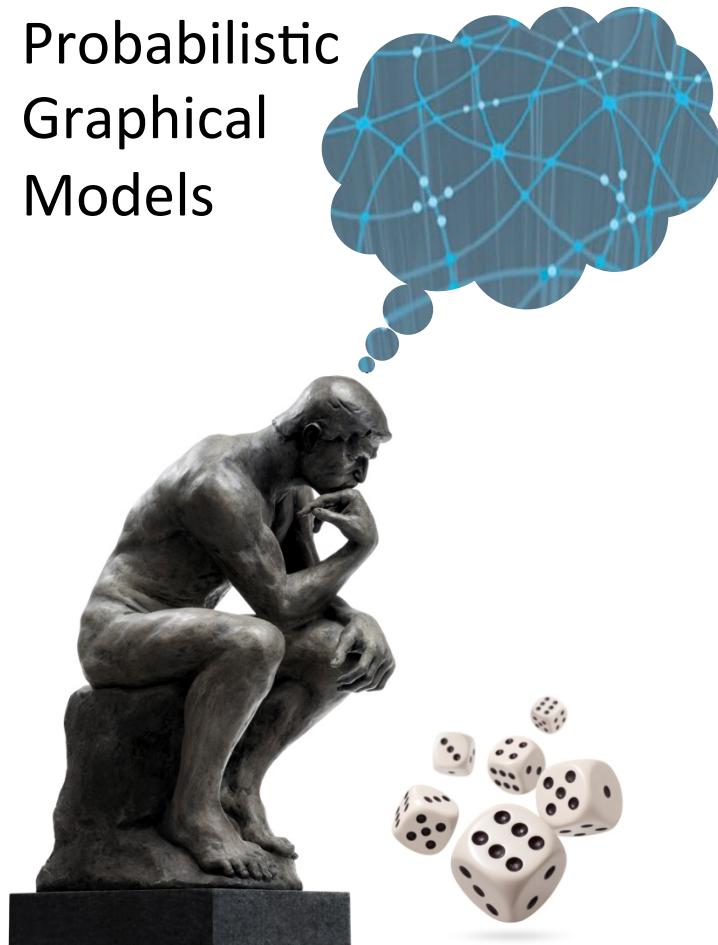


- Not every edge in tree is in the original network
- Inferred edges are undirected - can't determine direction

Summary

- Structure learning is an optimization over the combinatorial space of graph structures
- Decomposability \Rightarrow network score is a sum of terms for different families
- Optimal tree-structured network can be found using standard MST algorithms
- Computation takes quadratic time

Probabilistic
Graphical
Models



Learning

BN Structure

General
Graphs: Search

Optimization Problem

Input:

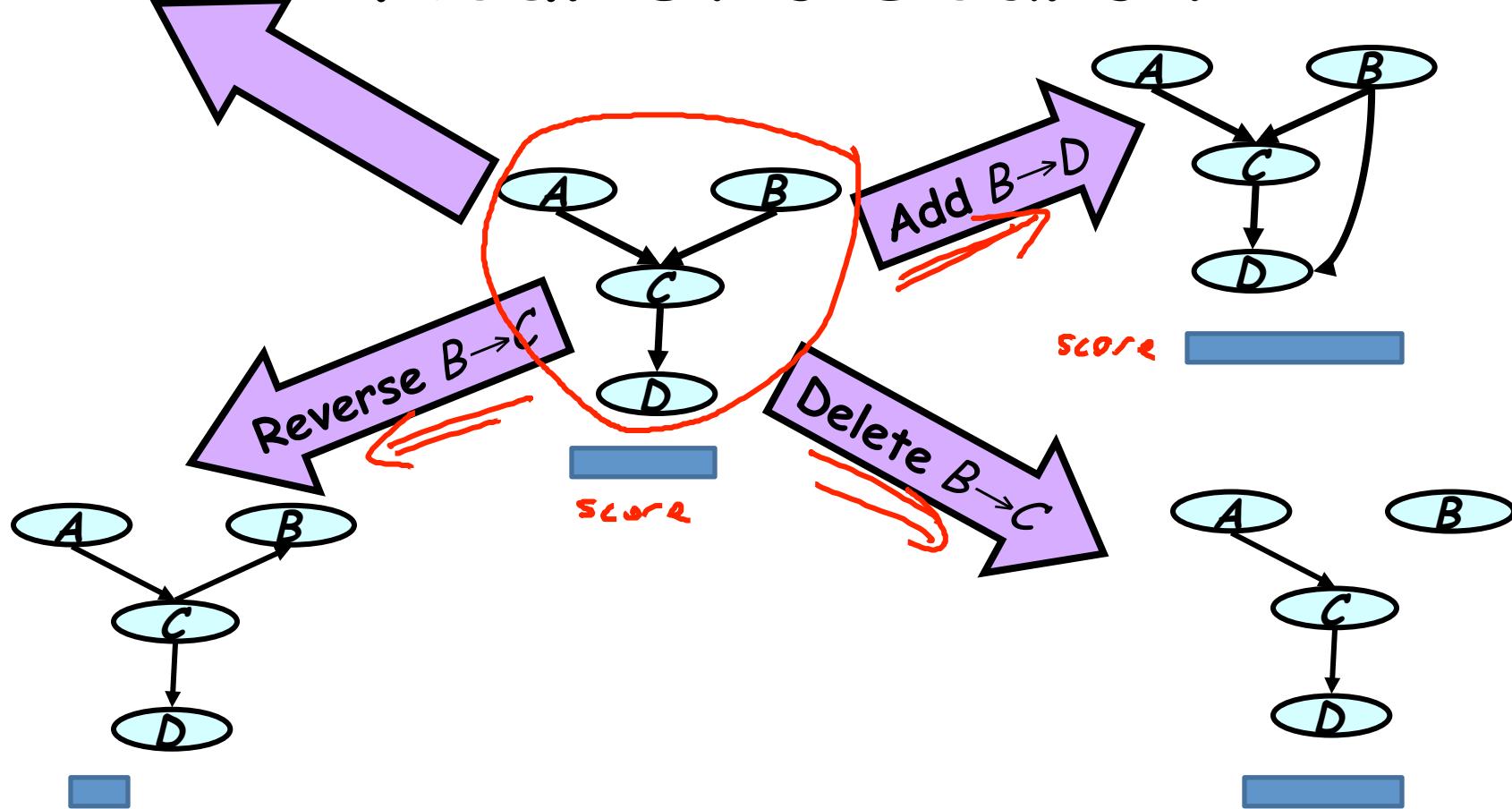
- Training data
- Scoring function
- Set of possible structures ↙

Output: A network that maximizes the score

Beyond Trees

- Problem is not obvious for general networks
 - Example: Allowing two parents, greedy algorithm is no longer guaranteed to find the optimal network
- Theorem:
 - Finding maximal scoring network structure with at most k parents for each variable is NP-hard for $k > 1$

Heuristic Search



Daphne Koller

Heuristic Search

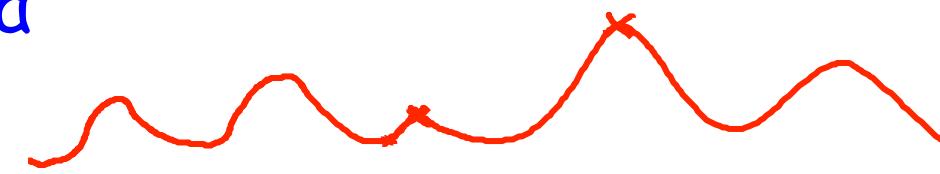
- Search operators:
 - local steps: edge addition, deletion, reversal
 - global steps
- Search techniques:
 - Greedy hill-climbing
 - Best first search
 - Simulated Annealing
 - ...

Search: Greedy Hill Climbing

- Start with a given network
 - empty network
 - best tree
 - a random network
 - prior knowledge
- At each iteration
 - Consider score for all possible changes
 - Apply change that most improves the score greedy
- Stop when no modification improves score local max.

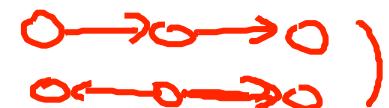
Greedy Hill Climbing Pitfalls

- Greedy hill-climbing can get stuck in:
 - Local maxima



- Plateaux

- Typically because equivalent networks are often neighbors in the search space

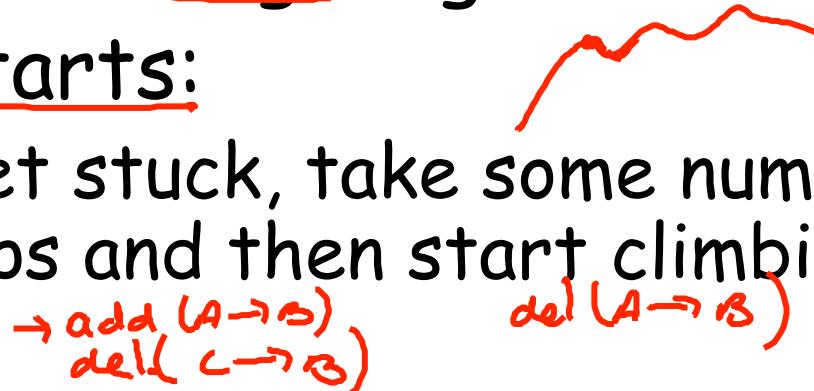


Why Edge Reversal,

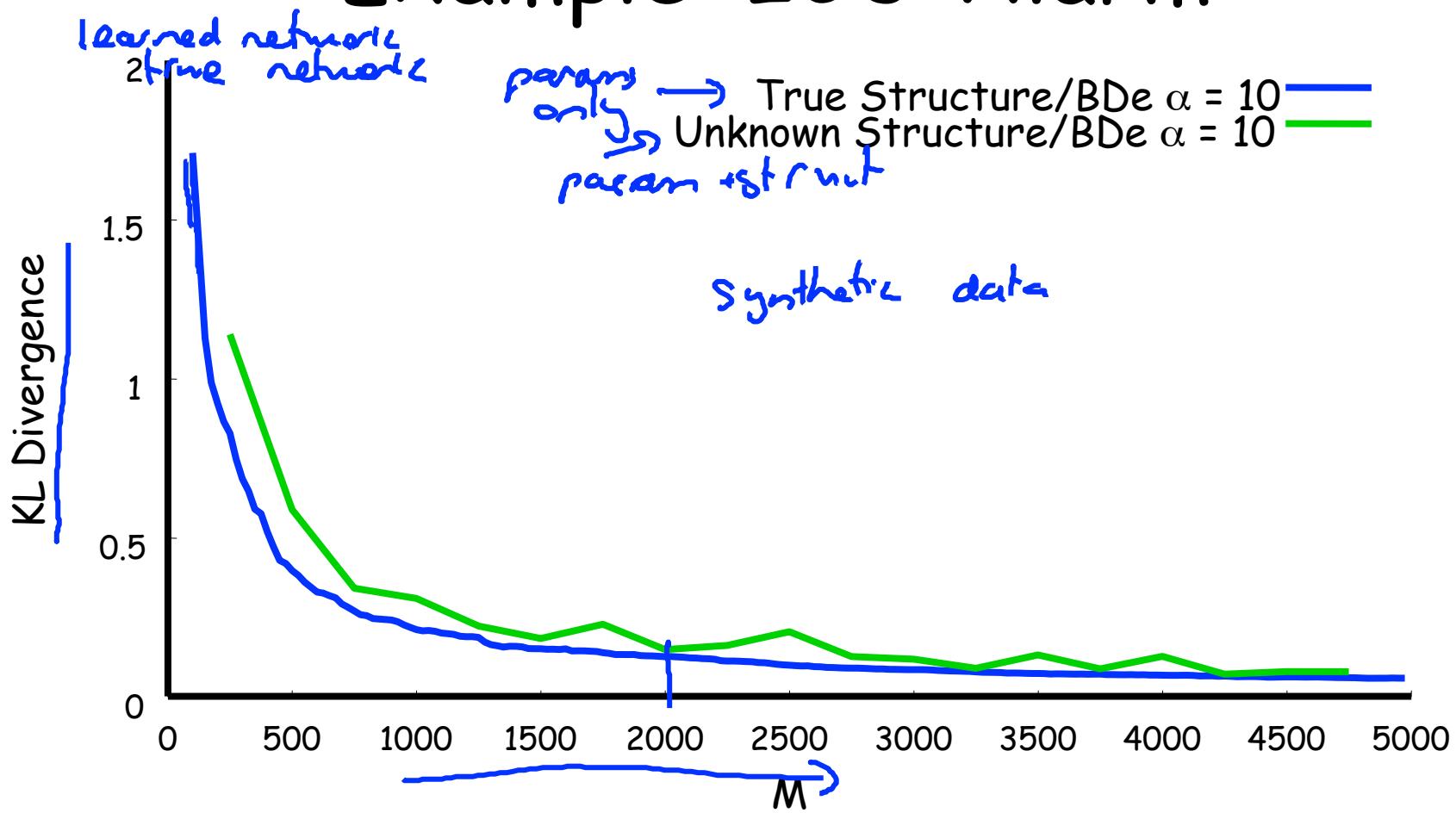


A Pretty Good, Simple Algorithm

- Greedy hill-climbing, augmented with:
 - Random restarts:
 - When we get stuck, take some number of random steps and then start climbing again
- Tabu list:
 - Keep a list of K steps most recently taken
 - Search cannot reverse any of these steps

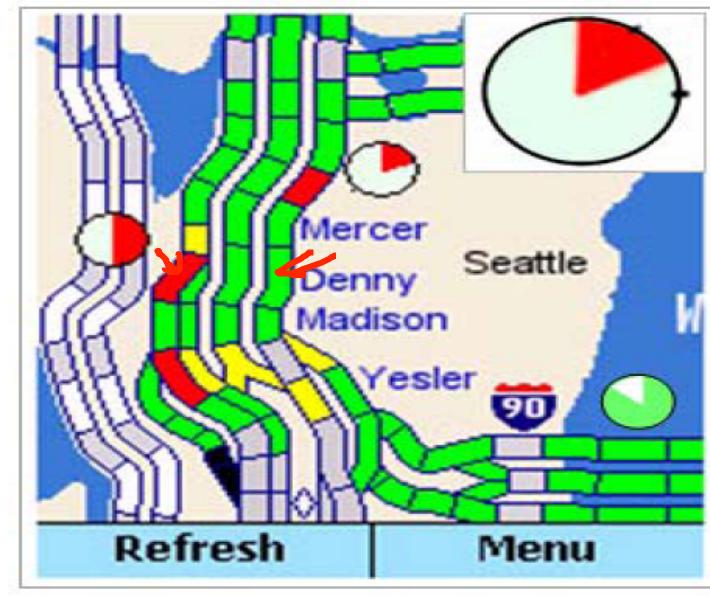


Example: ICU-Alarm



Daphne Koller

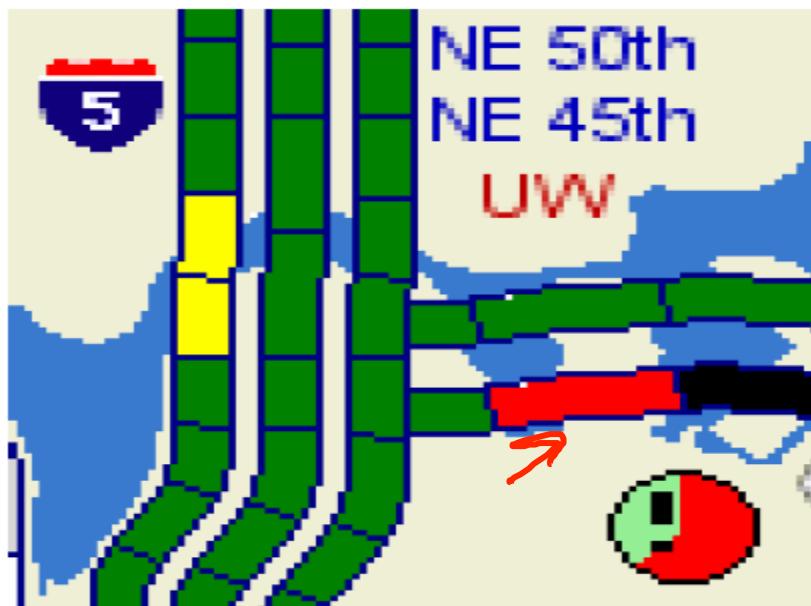
JamBayes



Horvitz, Apacible, Sarin, & Liao, UAI 2005

Daphne Koller

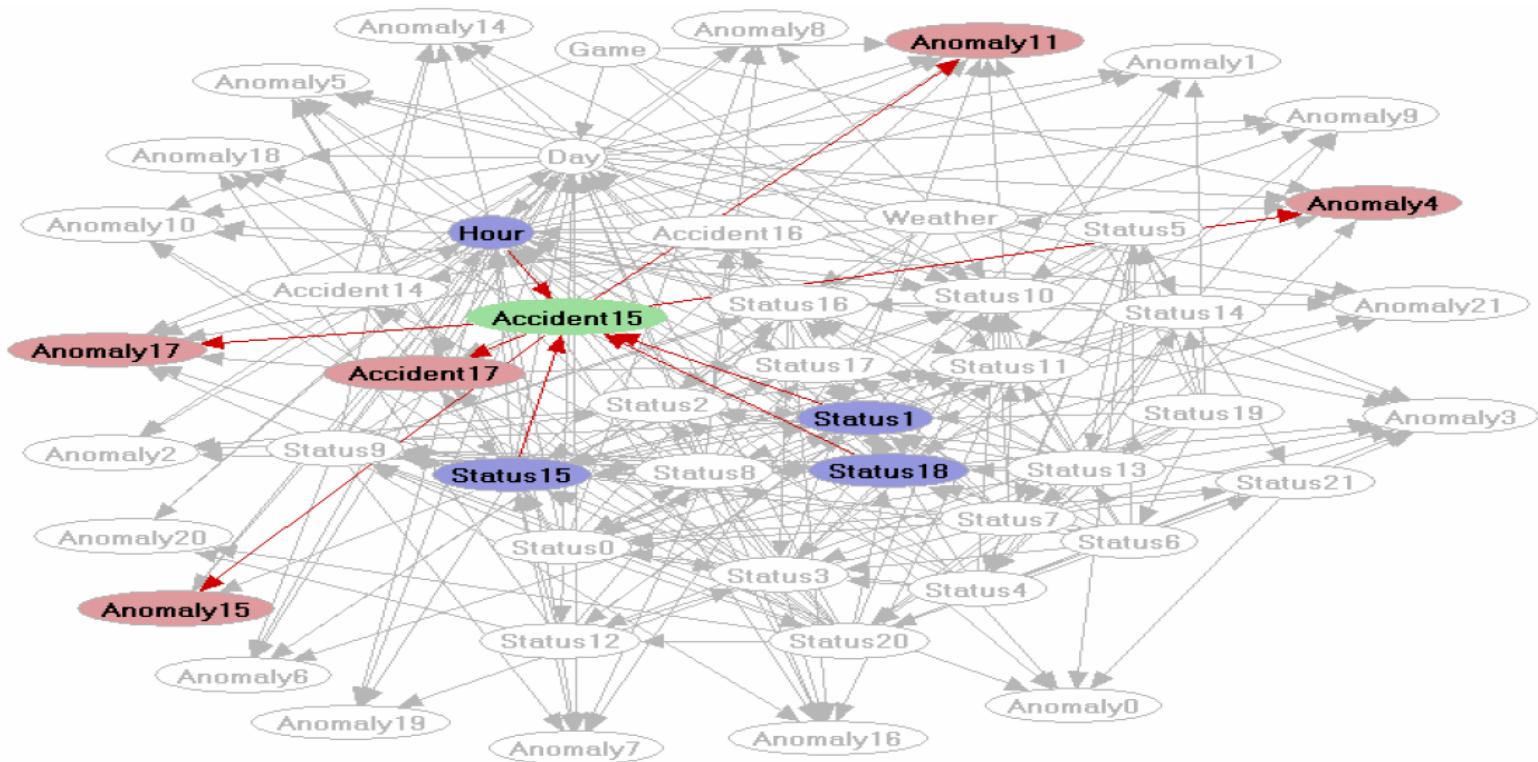
Predicting Surprises



Horvitz, Apacible, Sarin, & Liao, UAI 2005

Daphne Koller

Learned Model



Horvitz, Apacible, Sarin, & Liao, UAI 2005

Daphne Koller

Influences in Learned Model

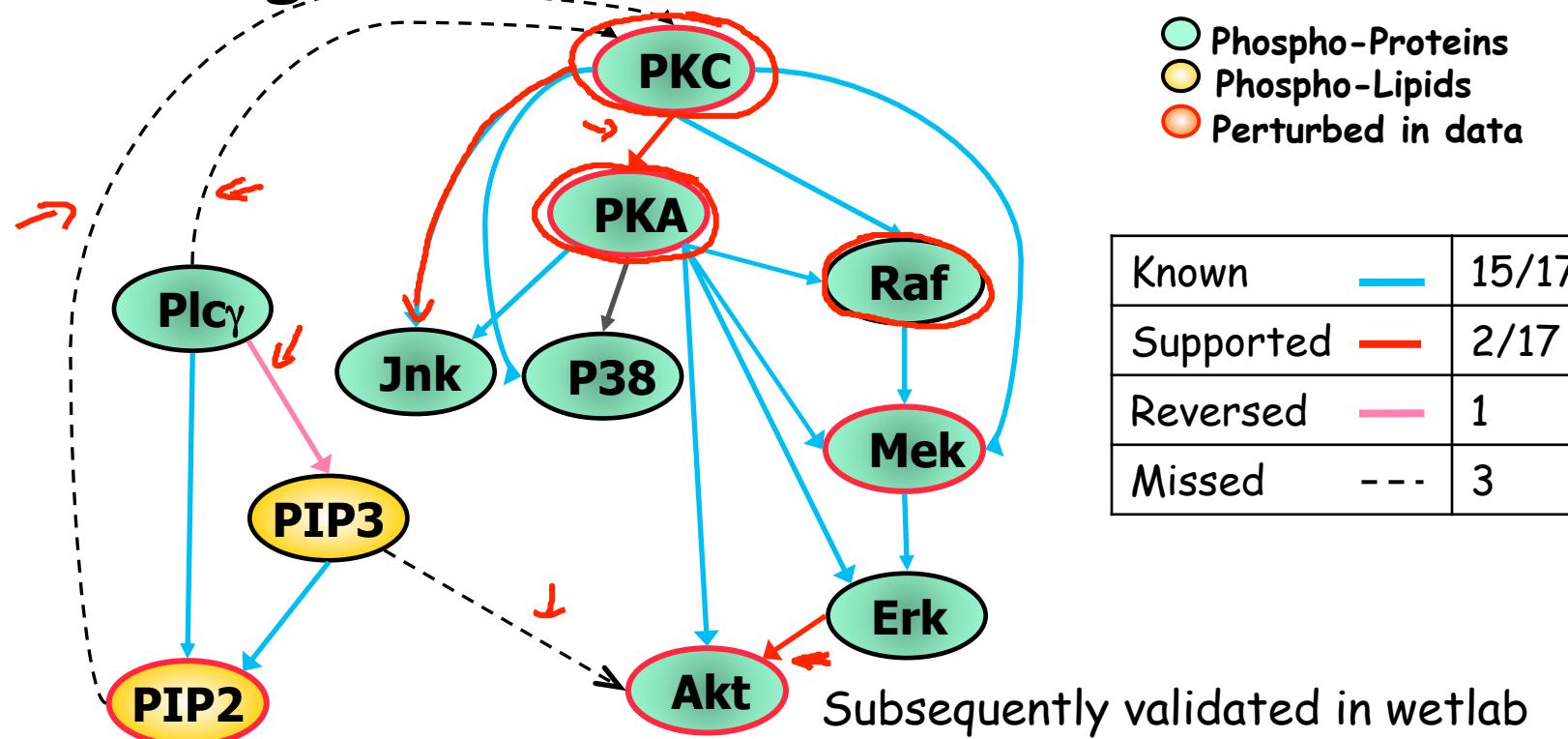


Horvitz, Apacible, Sarin, & Liao, UAI 2005

Daphne Koller

This figure may be used for non-commercial and classroom purposes only.
Any other uses require the prior written permission from AAAS

Biological Network Reconstruction



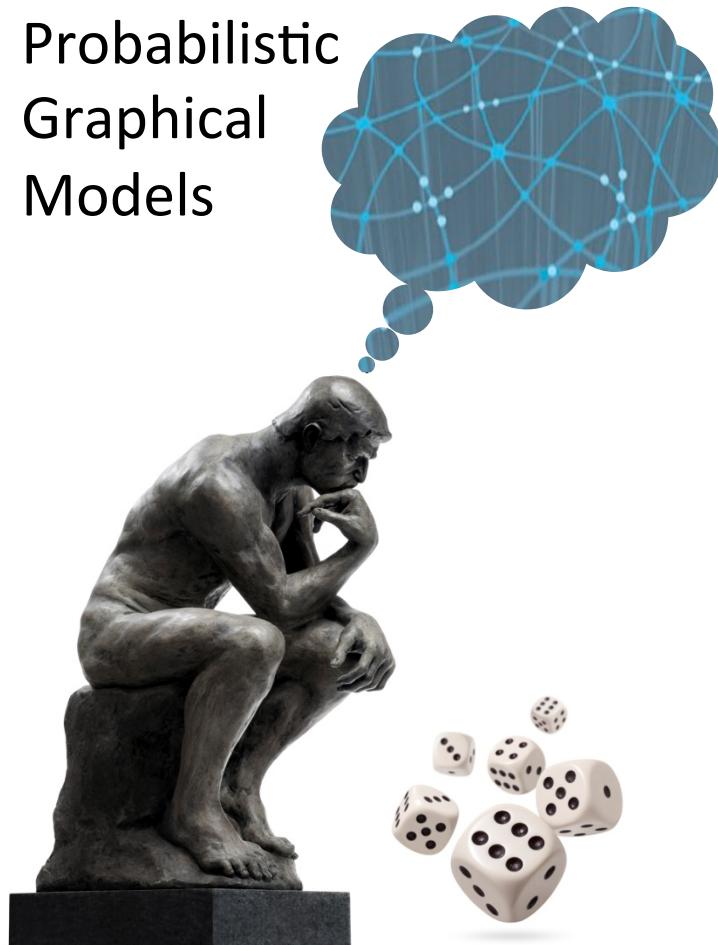
From “Causal protein-signaling networks derived from multiparameter single-cell data”
Sachs et al., Science 308:523, 2005. Reprinted with permission from AAAS.

Daphne Koller

Summary

- Useful for building better predictive models:
 - when domain experts don't know the structure
 - for knowledge discovery
- Finding highest-scoring structure is NP-hard
- Typically solved using simple heuristic search
 - local steps: edge addition, deletion, reversal
 - hill-climbing with tabu lists and random restarts
- But there are better algorithms

Probabilistic
Graphical
Models

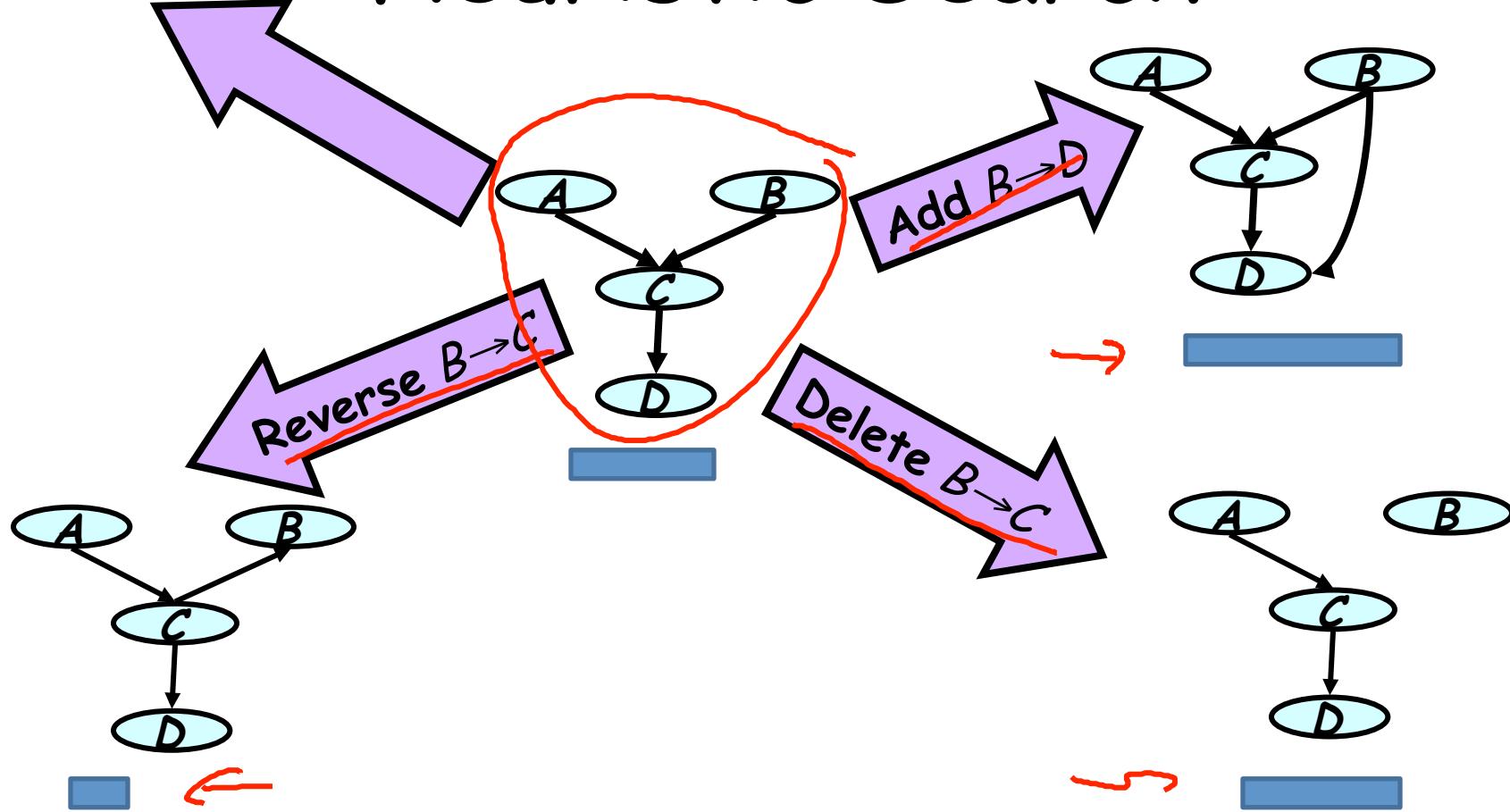


Learning

BN Structure

General Graphs:
Decomposability

Heuristic Search

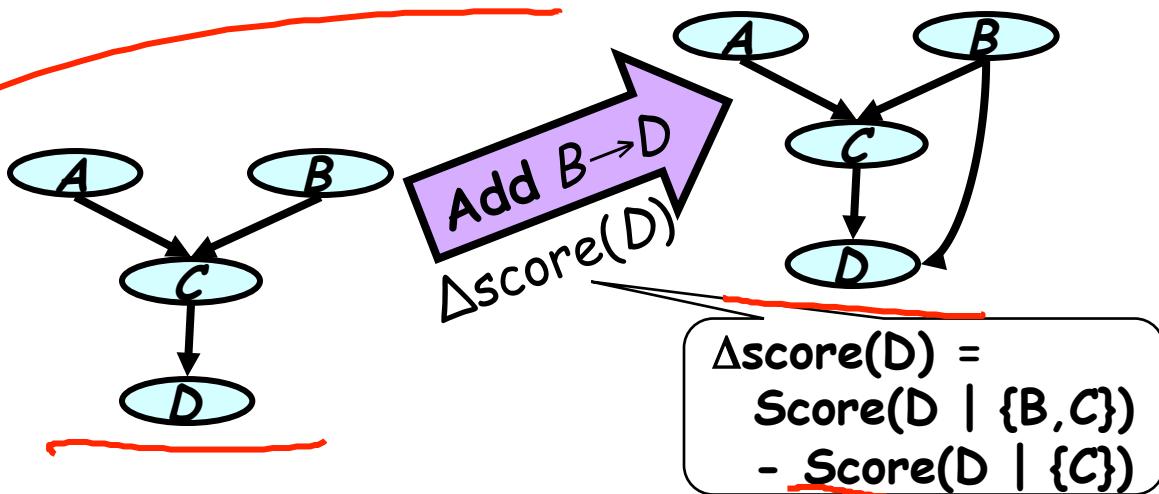


Daphne Koller

Naïve Computational Analysis

- Operators per search step:
 $n(n-1)$ possible edges $O(n^2)$
present ← delete 2
reverse ← assert-add 1
- Cost per network evaluation:
 - Components in score n components $O(n \cdot m)$
 - Compute sufficient statistics $O(m)$
 - Acyclicity check $O(m) \leftarrow \# \text{edges}$
- Total: $O(\underline{n^2} (\underline{Mn} + \underline{m}))$ per search step

Exploiting Decomposability

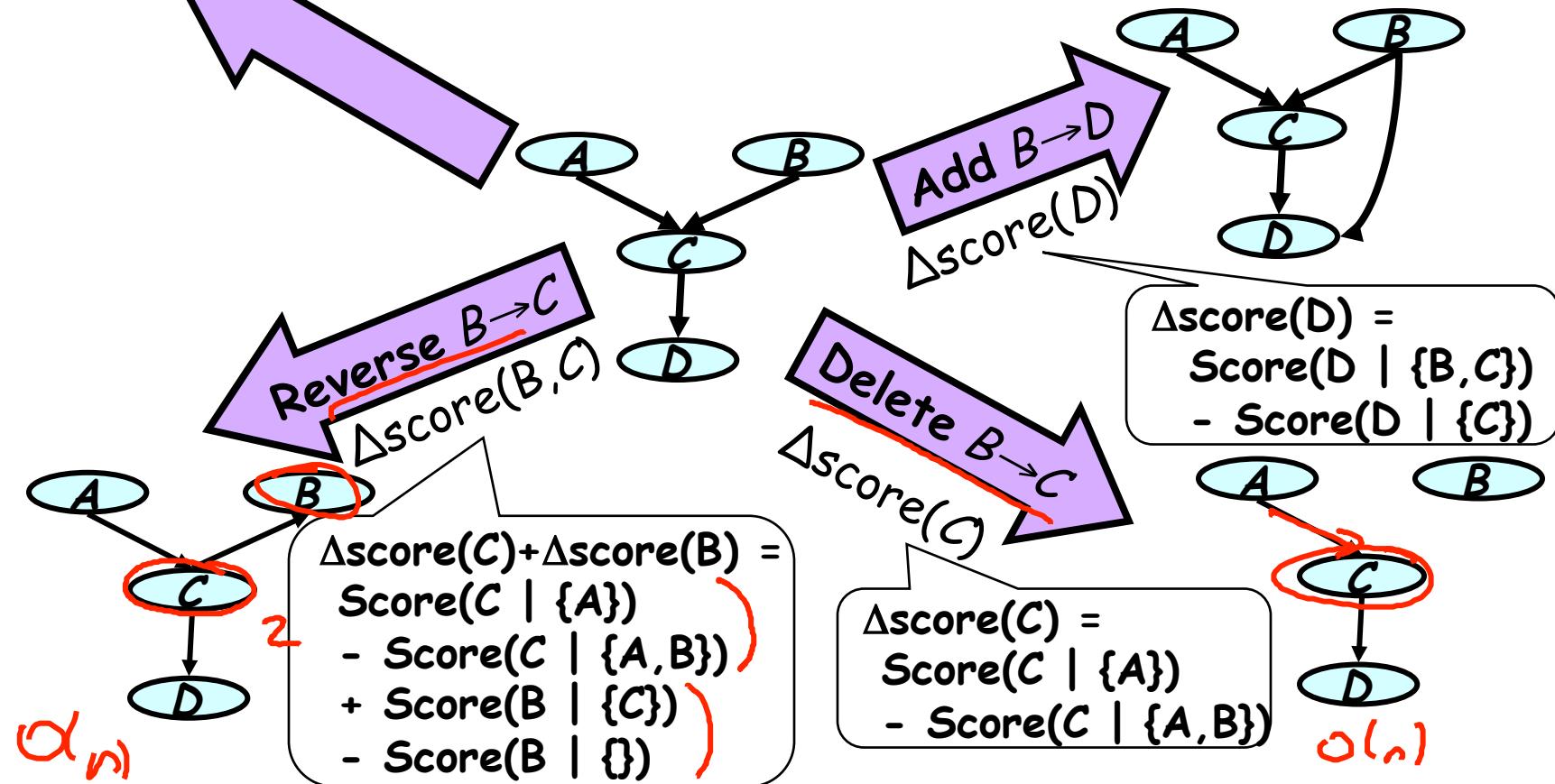


$$\text{score} = \underline{\text{Score}(A \mid \emptyset)} + \underline{\text{Score}(B \mid \emptyset)} + \underline{\text{Score}(C \mid \{A, B\})} + \underline{\text{Score}(D \mid \{C\})}$$

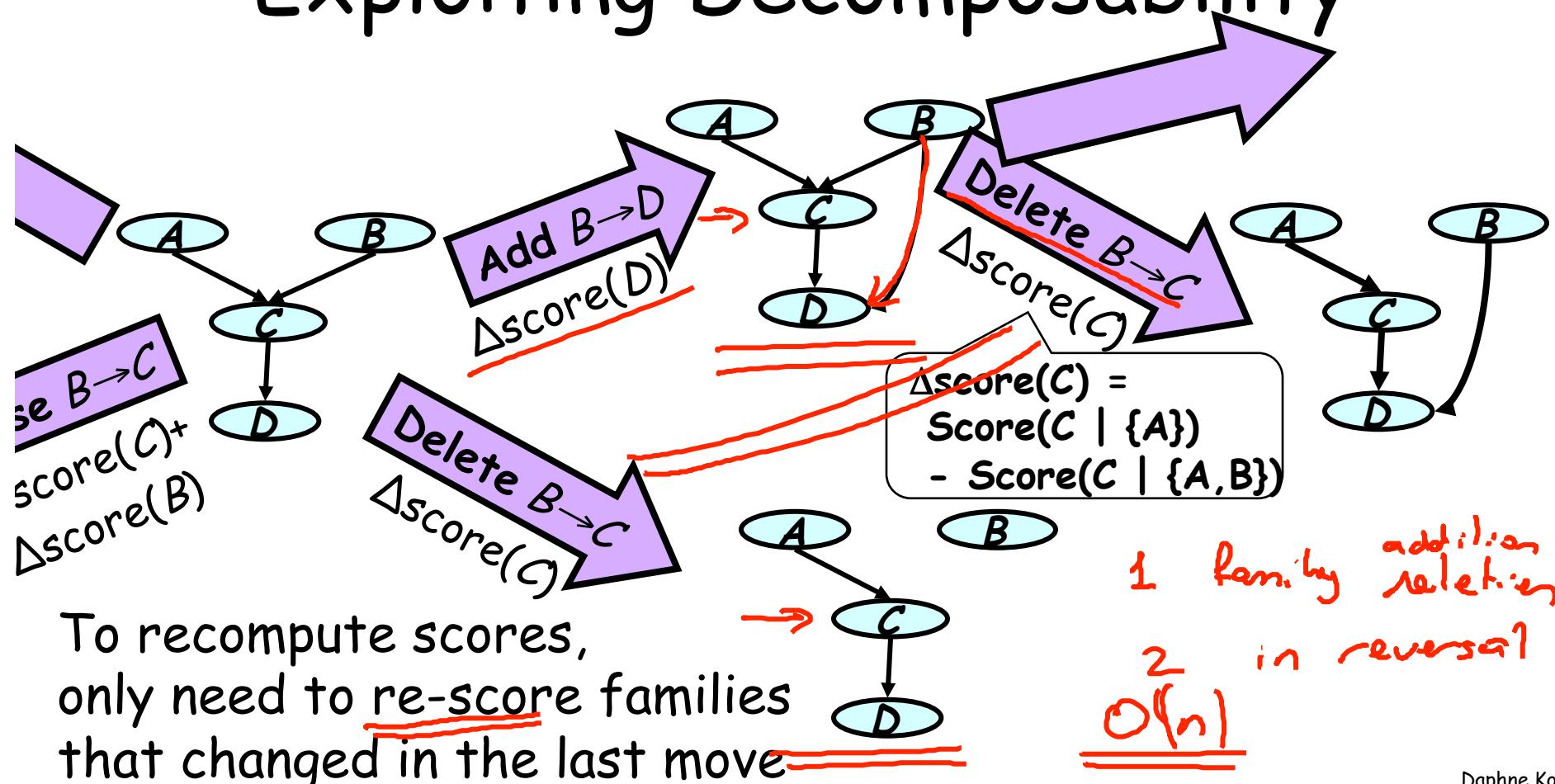
$$\text{score} = \underline{\text{Score}(A \mid \emptyset)} + \underline{\text{Score}(B \mid \emptyset)} + \underline{\text{Score}(C \mid \{A, B\})} + \boxed{\underline{\text{Score}(D \mid \{B, C\})}}$$

$\mathcal{O}(n)$ Savings

Exploiting Decomposability



Exploiting Decomposability



Computational Cost

- Cost per move
 - Compute $O(n)$ delta-scores damaged by move
 - Each one takes $O(M)$ time
- Keep priority queue of operators sorted by delta-score - $O(n \log n)$ $O(nM + n \log n)$

More Computational Efficiency

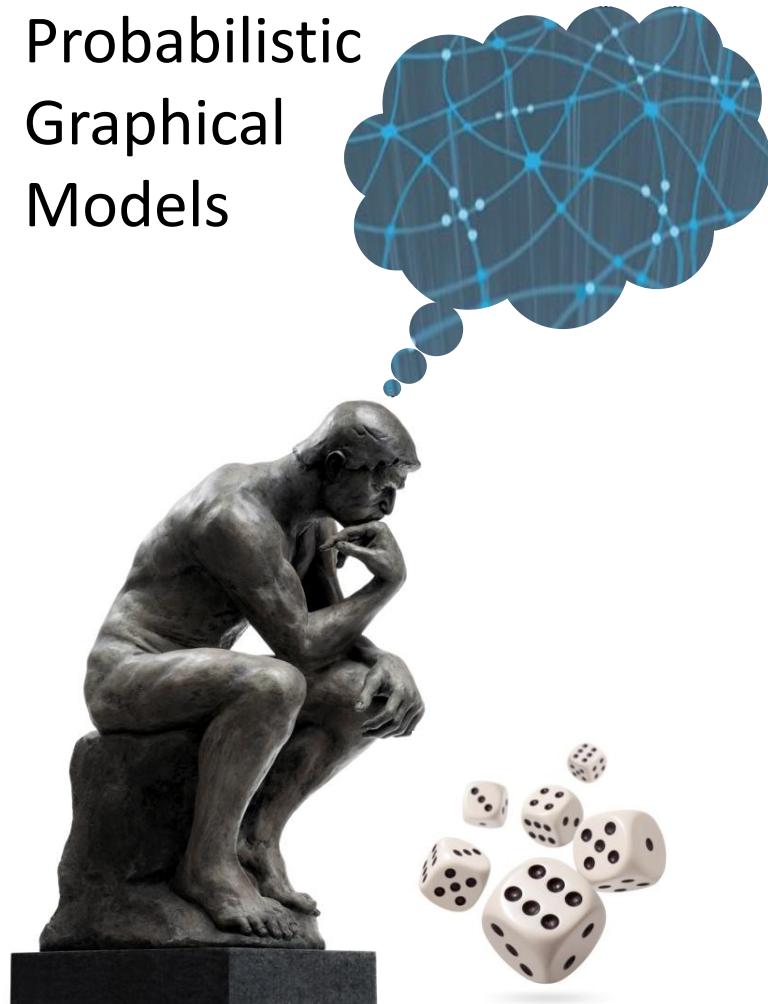
- Reuse and adapt previously computed sufficient statistics
- Restrict in advance the set of operators considered in the search

$$\text{A, B, C} \quad m[A, B, C]$$
$$m[A, B] = \sum C m[A, B, C]$$

Summary

- Even heuristic structure search can get expensive for large n
- Can exploit decomposability to get orders of magnitude reduction in cost
- Other tricks are also used for scaling

Probabilistic
Graphical
Models



Learning

Parameter Estimation

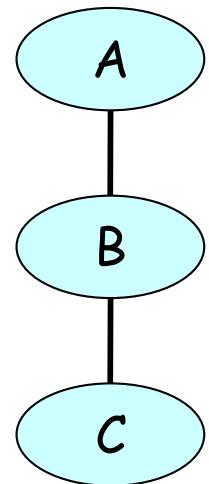
Max Likelihood
for Log-Linear
Models

Log-Likelihood for Markov Nets

$$P_{\phi}(\alpha, \beta, \gamma) = \frac{1}{Z} \phi_1(\alpha, \beta) \cdot \phi_2(\beta, \gamma)$$

$$\begin{aligned}\ell(\theta : \mathcal{D}) &= \sum_m (\ln \phi_1(a[m], b[m]) + \ln \phi_2(b[m], c[m]) - \ln Z(\theta)) \\ &= \sum_{a,b} [M[a, b] \ln \phi_1(a, b)] + \sum_{b,c} [M[b, c] \ln \phi_2(b, c)] - M \ln Z(\theta)\end{aligned}$$

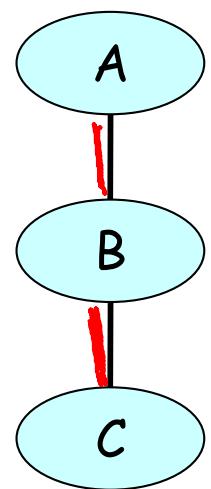
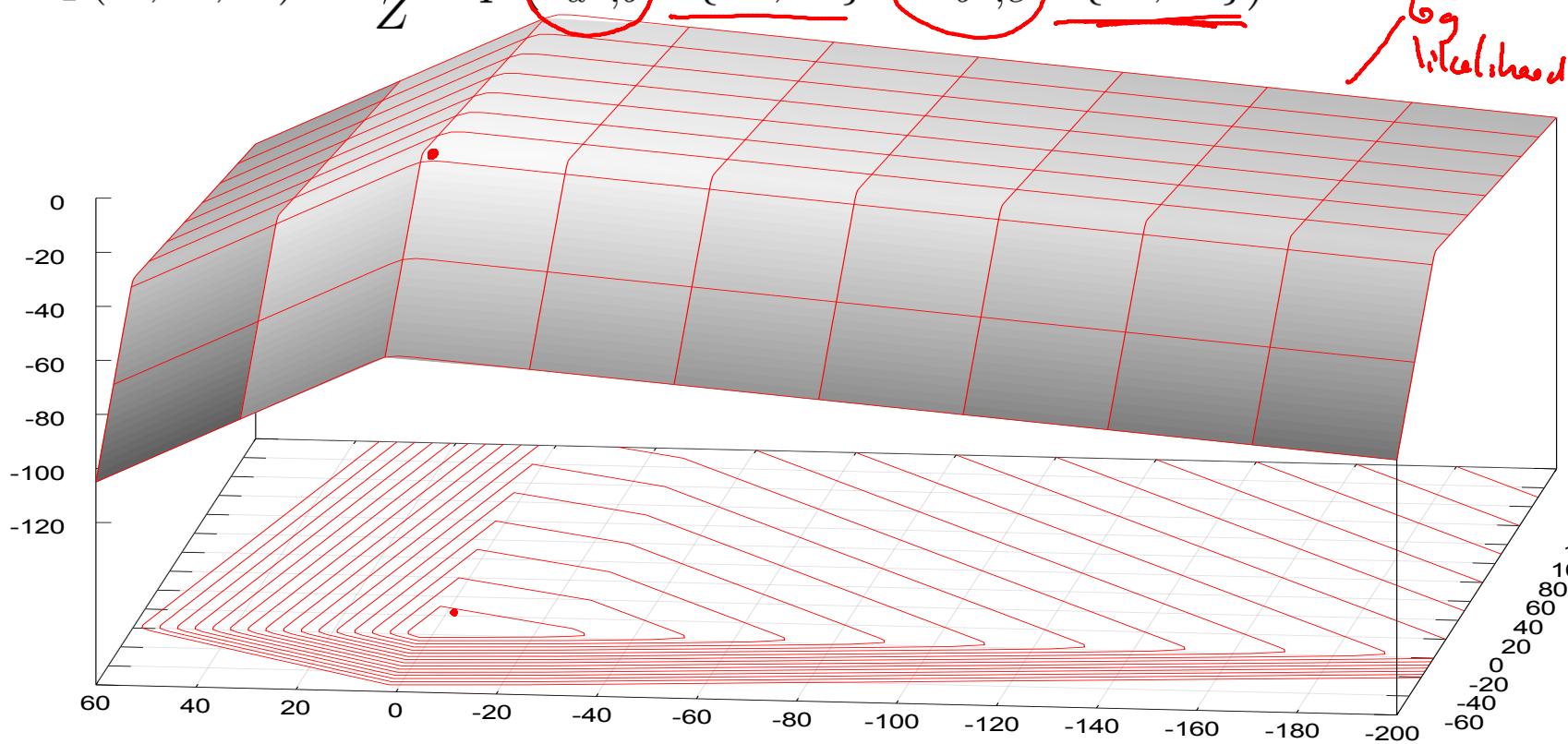
$Z(\theta) = \sum_{a,b,c} \phi_1(a, b) \phi_2(b, c)$



- Partition function couples the parameters
 - No decomposition of likelihood
 - No closed form solution

Example: Log-Likelihood Function

$$P_{\Phi}(A, B, C) = \frac{1}{Z} \exp(\theta_{a^1, b^1} \mathbf{1}\{a^1, b^1\} + \theta_{b^0, c^1} \mathbf{1}\{b^0, c^1\})$$



200
180
160
140
120
100
80
60
40
20

Daphne Koller

Log-Likelihood for Log-Linear Model

$$P(X_1, \dots, X_n : \theta) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{i=1}^k \theta_i f_i(D_i) \right\}$$

parameters ↗ *features*

$$\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(x[m]) \right) - M \ln Z(\theta)$$

↓
 Feature f_i applied to the m th instance
 ↓
 partition function

$$\ln Z(\theta) = \ln \sum_x \exp \left\{ \sum_i \theta_i f_i(x) \right\}$$

x
exponentially
large space

log-sum-exp

The Log-Partition Function

Theorem: $\frac{\partial}{\partial \theta_i} \ln Z(\theta) = \underbrace{E_{\theta}[f_i]}_{\substack{\text{vector} \\ \text{at} \\ \text{derivative}}} \quad \text{expectation at } P_{\theta} = \sum_x P_{\theta}(x) f_i(x)$

(Hessian) $\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln Z(\theta) = \underbrace{\text{Cov}_{\theta}[f_i; f_j]}_{\substack{\text{matrix} \\ \text{(Hessian)}}}$

Proof:
$$\begin{aligned} \frac{\partial}{\partial \theta_i} \ln Z(\theta) &= \frac{1}{Z(\theta)} \sum_x \frac{\partial}{\partial \theta_i} \exp \left\{ \sum_j \theta_j f_j(x) \right\} \\ &= \frac{1}{Z(\theta)} \sum_x f_i(x) \exp \left\{ \sum_j \theta_j f_j(x) \right\} \\ &= \sum_x \underbrace{\frac{1}{Z(\theta)} \exp \left\{ \sum_j \theta_j f_j(x) \right\}}_{P_{\theta}(x)} f_i(x) = \sum_x \underbrace{P_{\theta}(x)}_{\text{blue}} \underbrace{f_i(x)}_{\text{blue}} \end{aligned}$$

$\frac{\partial}{\partial \theta_i} f_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$

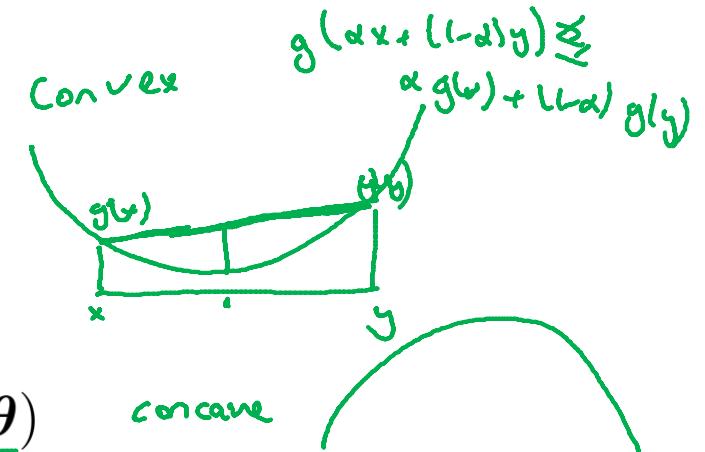
The Log-Partition Function

Theorem: $\frac{\partial}{\partial \theta_i} \ln Z(\theta) = E_{\theta}[f_i]$

Hessian $\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln Z(\theta) = \text{Cov}_{\theta}[f_i; f_j]$

$$\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(x[m]) \right) - \underline{M \ln Z(\theta)}$$

linear in θ



- Log likelihood function

- No local optima
 - Easy to optimize

Maximum Likelihood Estimation

$$\frac{1}{M} \ell(\boldsymbol{\theta} : \mathcal{D}) = \sum_i \theta_i \left(\underbrace{\frac{1}{M} \sum_m f_i(x[m])}_{\text{empirical expectation of } f_i \text{ in } \mathcal{D}} \right) - \ln Z(\boldsymbol{\theta})$$
$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\boldsymbol{\theta} : \mathcal{D}) = E_{\mathcal{D}}[f_i(\mathbf{X})] - E_{\boldsymbol{\theta}}[f_i] \quad \begin{matrix} \text{expectation of } f_i \\ \text{in } \mathcal{D} \\ \text{expectation relative to } \boldsymbol{\theta} \end{matrix}$$

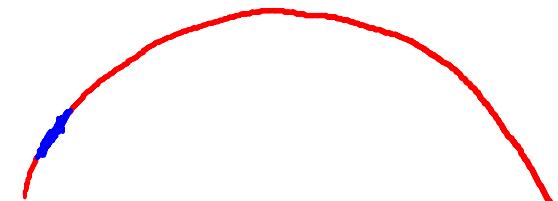
Theorem: $\hat{\boldsymbol{\theta}}$ is the MLE if and only if

$$E_{\mathcal{D}}[f_i(\mathbf{X})] = E_{\hat{\boldsymbol{\theta}}}[f_i]$$

expectation in \mathcal{D} = expectation relative to $\boldsymbol{\theta}$

Computation: Gradient Ascent

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\boldsymbol{\theta} : \mathcal{D}) = \boxed{\mathbf{E}_{\mathcal{D}}[f_i(\mathbf{X})]} - \boxed{\mathbf{E}_{\boldsymbol{\theta}}[f_i]}$$

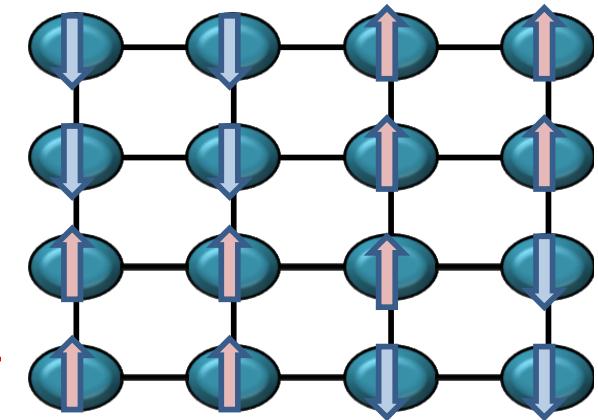


- Use gradient ascent:
 - typically L-BFGS – a quasi-Newton method
- For gradient, need expected feature counts:
 - in data
 - relative to current model
- Requires inference at each gradient step

Example: Ising Model

$$E(x_1, \dots, x_n) = - \sum_{i < j} w_{i,j} x_i x_j - \sum_i u_i x_i$$

$$\boxed{\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\boldsymbol{\theta} : \mathcal{D}) = \mathbf{E}_{\mathcal{D}}[f_i(\mathbf{X})] - \mathbf{E}_{\boldsymbol{\theta}}[f_i]}$$



$$x_i \in \{-1, +1\}$$

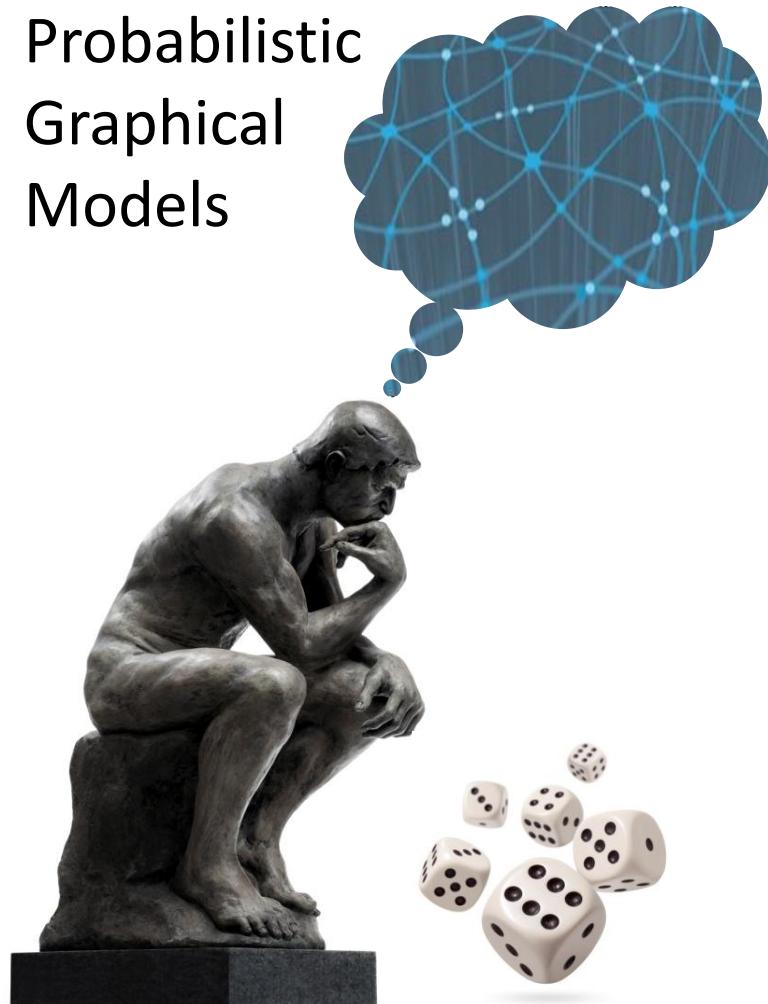
$$\frac{\partial}{\partial u_i} = \frac{1}{M} \sum_m \underbrace{x_i[m]}_{+1} - (\underbrace{P_{\boldsymbol{\theta}}(X_i = 1)}_{+1} - \underbrace{P_{\boldsymbol{\theta}}(X_i = -1)}_{-1})$$

$$\frac{\partial}{\partial w_{ij}} = \frac{1}{M} \sum_m x_i[m] x_j[m] - \left(\underbrace{P_{\boldsymbol{\theta}}(X_i = 1, X_j = 1)}_{-P_{\boldsymbol{\theta}}(X_i = 1, X_j = -1)} + \underbrace{P_{\boldsymbol{\theta}}(X_i = -1, X_j = -1)}_{-P_{\boldsymbol{\theta}}(X_i = -1, X_j = 1)} \right)$$

Summary

- Partition function couples parameters in likelihood
- No closed form solution, but convex optimization
 - Solved using gradient ascent (usually L-BFGS) global opt.
- Gradient computation requires inference at each gradient step to compute expected feature counts
- Features are always within clusters in cluster-graph or clique tree due to family preservation
 - One calibration suffices for all feature expectations

Probabilistic
Graphical
Models



Learning

Parameter Estimation

Max Likelihood
for CRFs

Estimation for CRFs

$$P_{\theta}(\mathbf{Y} | \mathbf{x}) = \frac{1}{Z_{\mathbf{x}}(\theta)} \tilde{P}_{\theta}(\mathbf{x}, \mathbf{Y})$$

Z_{\mathbf{x}}(\theta)

\tilde{P}_{\theta}(\mathbf{x}, \mathbf{Y})

log conditional likelihood

$$\mathcal{D} = \{(\mathbf{x}[m], \mathbf{y}[m])\}_{m=1}^M \quad \ell_{\mathbf{Y}|\mathbf{X}}(\theta : \mathcal{D}) = \sum_{m=1}^M \ln P_{\theta}(\mathbf{y}[m] | \mathbf{x}[m], \theta)$$

$$\ell_{\mathbf{Y}|\mathbf{X}}(\theta : (\mathbf{x}[m], \mathbf{y}[m])) = \left(\sum_i \theta_i f_i(\mathbf{x}[m], \mathbf{y}[m]) \right) - \ln Z_{\mathbf{x}[m]}(\theta)$$

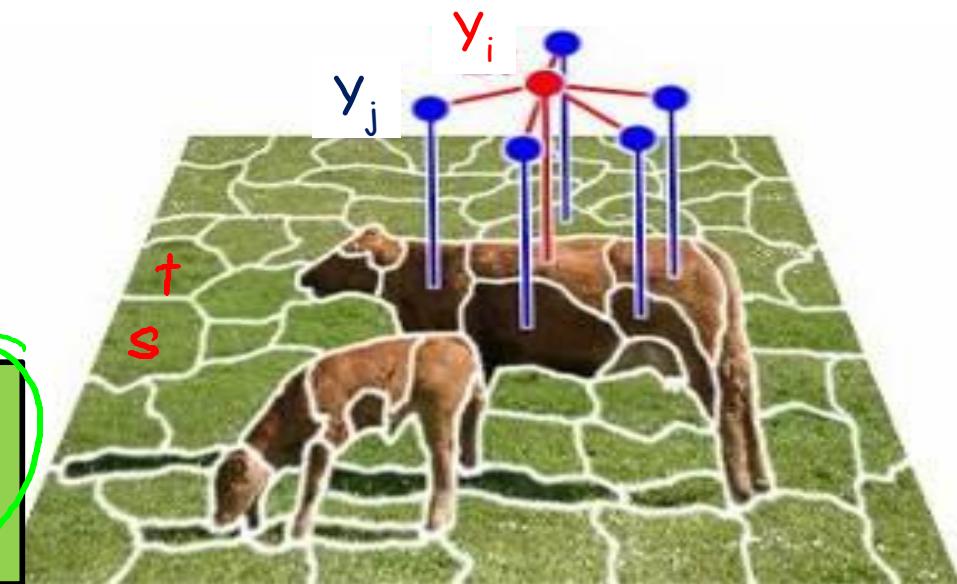
$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell_{\mathbf{Y}|\mathbf{X}}(\theta : \mathcal{D}) = \frac{1}{M} \sum_{m=1}^M (f_i(\mathbf{x}[m], \mathbf{y}[m]) - E_{\theta}[f_i(\mathbf{x}[m], \mathbf{Y})])$$

Example

$$\underline{f_1(y_s, X_s)} = \underline{\mathbf{1}(y_s = g)} \times G_s$$

$$\underline{f_2(y_s, y_t)} = \underline{\mathbf{1}(y_s = y_t)}$$

average intensity of
green channel for
pixels in superpixel s



$$\frac{\partial}{\partial \theta_i} \ell_{\mathbf{Y}|\mathbf{X}}(\boldsymbol{\theta} : (\mathbf{x}[m], \mathbf{y}[m])) = (f_i(\mathbf{x}[m], \mathbf{y}[m]) - \mathbf{E}_{\boldsymbol{\theta}}[f_i(\mathbf{x}[m], \mathbf{Y})])$$

$$\frac{\partial}{\partial \theta_1} = \sum_s \underline{\mathbf{1}\{y_s[m] = g\}} \underline{G_s[m]} - \sum_s \underline{P_{\boldsymbol{\theta}}(Y_s = g \mid \mathbf{x}[m])} \underline{G_s[m]}$$

$$\frac{\partial}{\partial \theta_2} = \sum_{(s,t) \in \mathcal{N}} \underline{\mathbf{1}\{y_s[m] = y_t[m]\}} - \sum_{(s,t) \in \mathcal{N}} \underline{P_{\boldsymbol{\theta}}(Y_s = Y_t \mid \mathbf{x}[m])}$$

Computation

MRF

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\boldsymbol{\theta} : \mathcal{D}) = \mathbf{E}_{\mathcal{D}}[f_i(\mathbf{X})] - \underline{\mathbf{E}_{\boldsymbol{\theta}}[f_i]}$$

- Requires inference at each gradient step

CRF

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell_{\mathbf{Y}|\mathbf{X}}(\boldsymbol{\theta} : \mathcal{D}) = \frac{1}{M} \sum_{m=1}^M \underline{(f_i(\mathbf{x}[m], \mathbf{y}[m]) - \boxed{\mathbf{E}_{\boldsymbol{\theta}}[f_i(\mathbf{x}[m], \mathbf{Y})]})}$$

- Requires inference for each $\mathbf{x}[m]$ at each gradient step
 $m = \# \text{ training instances}$

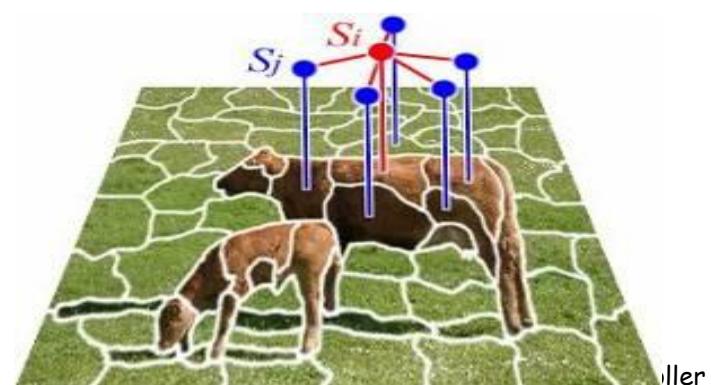
However...

- For inference of $P(Y | x)$, we need to compute distribution only over \underline{Y}
- If we learn an MRF, need to compute $\underline{P(Y, X)}$, which may be much more complex

$$f_1(y_s, \underline{x}_s) = \underline{1}(y_s = g) \times G_s$$

$f_2(y_s, y_t) = 1(y_s = y_t)$

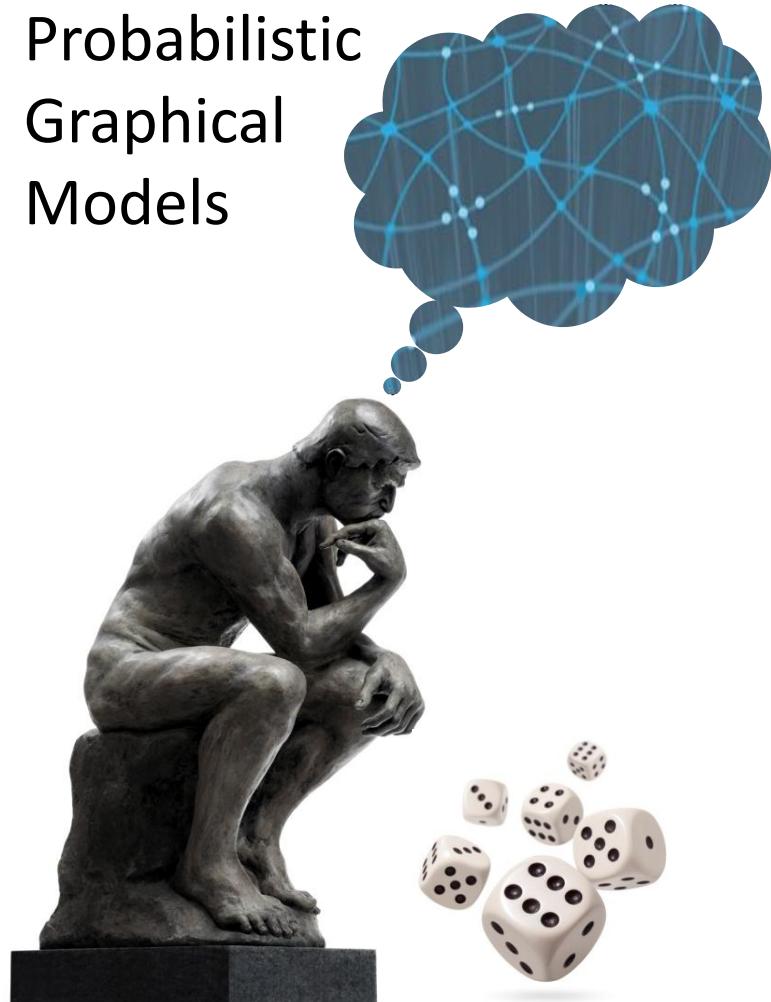
average intensity of
green channel for
pixels in superpixel i



Summary

- CRF learning very similar to MRF learning
 - Likelihood function is concave
 - Optimized using gradient ascent (usually L-BFGS)
- Gradient computation requires inference: one per gradient step, data instance
 - c.f., once per gradient step for MRFs
- But conditional model is often much simpler, so inference cost for CRF, MRF is not the same

Probabilistic
Graphical
Models



Learning

Parameter Estimation

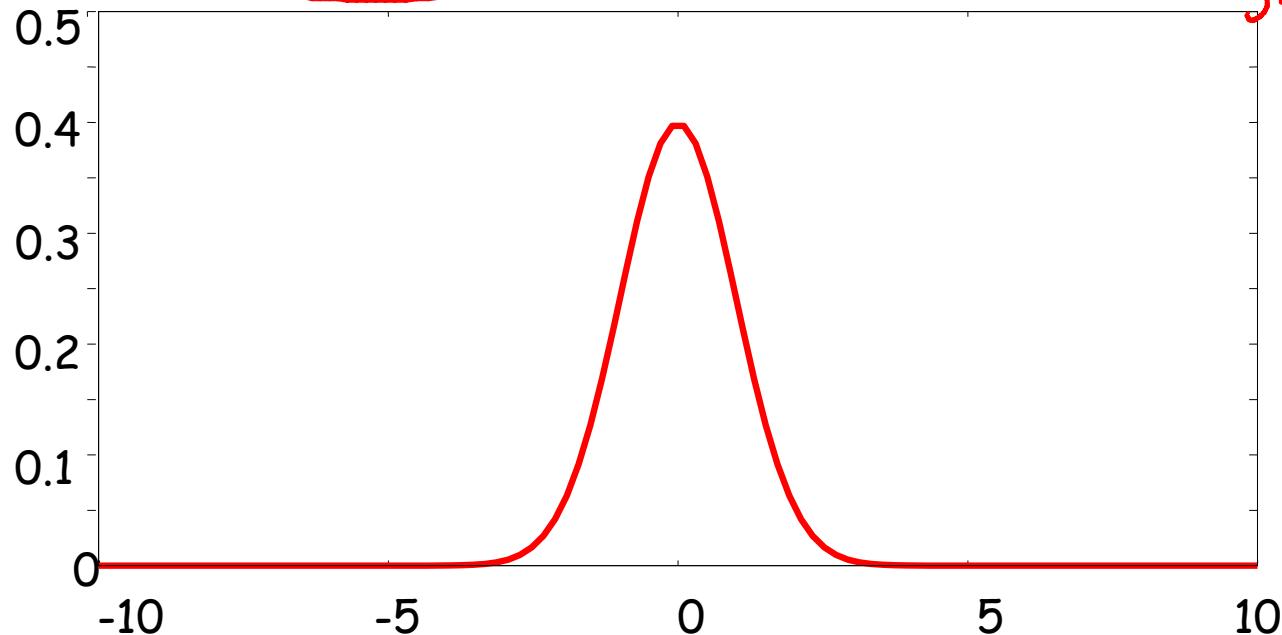
MAP

Estimation for
MRFs, CRFs

Gaussian Parameter Prior

$$P(\theta : \sigma^2) = \prod_{i=1}^k \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{\theta_i^2}{2\sigma^2} \right\}$$

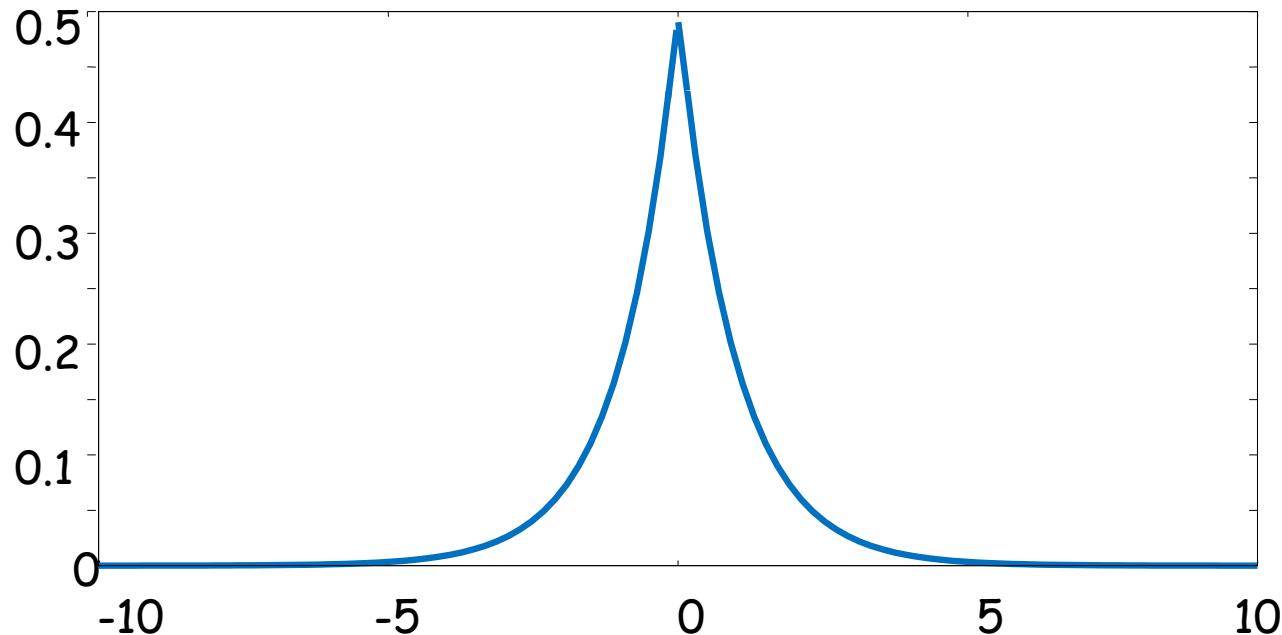
one mean univariate Gaussian
hyperparameter



Laplacian Parameter Prior

$$P(\theta : \beta) = \prod_{i=1}^k \frac{1}{2\beta} \exp \left\{ -\frac{|\theta_i|}{\beta} \right\}$$

hypoparameter



MAP Estimation & Regularization

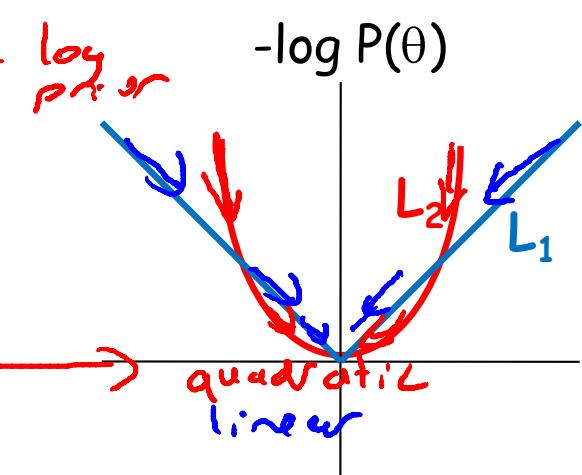
$$P(\boldsymbol{\theta} : \sigma^2) = \prod_{i=1}^k \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{\theta_i^2}{2\sigma^2} \right\}$$

$$P(\boldsymbol{\theta} : \beta) = \prod_{i=1}^k \frac{1}{2\beta} \exp \left\{ -\frac{|\theta_i|}{\beta} \right\}$$

$$\begin{aligned} \text{argmax}_{\boldsymbol{\theta}} P(\mathcal{D}, \boldsymbol{\theta}) &= \text{argmax}_{\boldsymbol{\theta}} P(\mathcal{D} | \boldsymbol{\theta}) P(\boldsymbol{\theta}) \\ &= \text{argmax}_{\boldsymbol{\theta}} (\ell(\boldsymbol{\theta} : \mathcal{D}) + \log P(\boldsymbol{\theta})) \end{aligned}$$

↑ likelihood ↓ prior
 ← log likelihood ← log prior

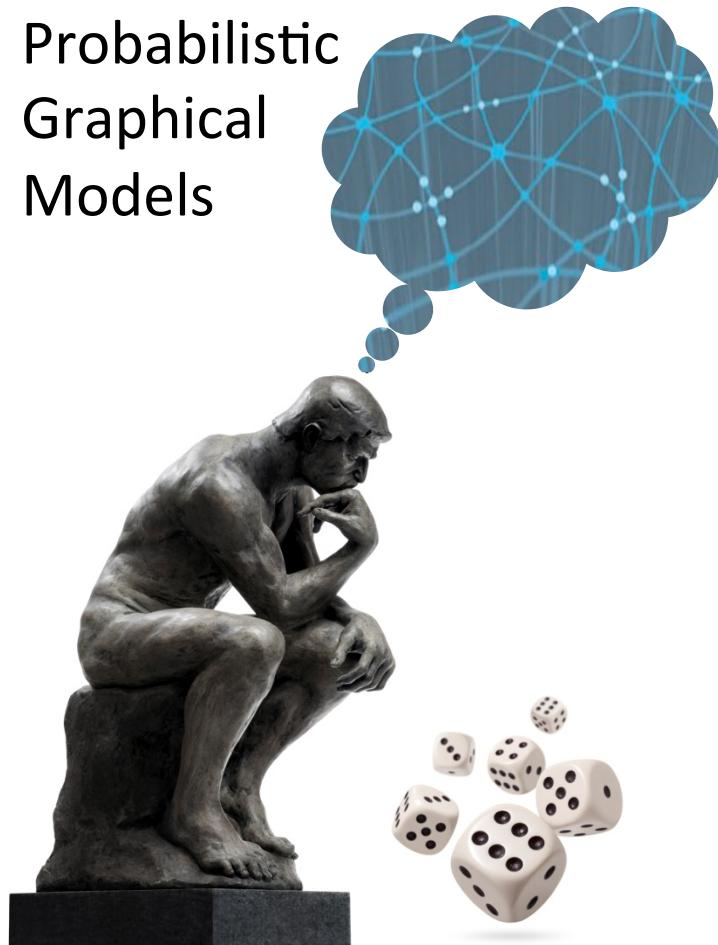
many $\theta_i \neq 0$
 dense → L_2 -regularization
 sparse L_1 -regularization



Summary

- In undirected models, parameter coupling prevents efficient Bayesian estimation
- However, can still use parameter priors to avoid overfitting of MLE MAP
- Typical priors are L_1 , L_2
 - Drive parameters toward zero
- L_1 provably induces sparse solutions
 - Performs feature selection / structure learning

Probabilistic
Graphical
Models



Learning
Incomplete Data

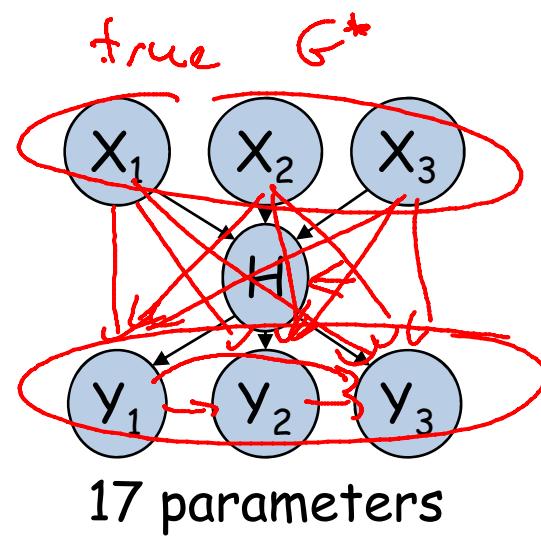
Overview

Incomplete Data

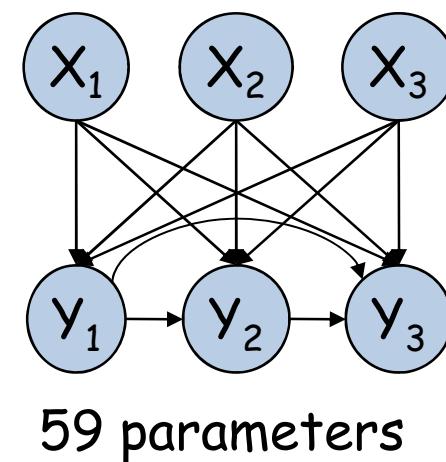
- Multiple settings:
 - Hidden variables
 - Missing values
- Challenges
 - Foundational – is the learning task well defined?
 - Computational – how can we learn with incomplete data?

Why latent variables?

- Model sparsity

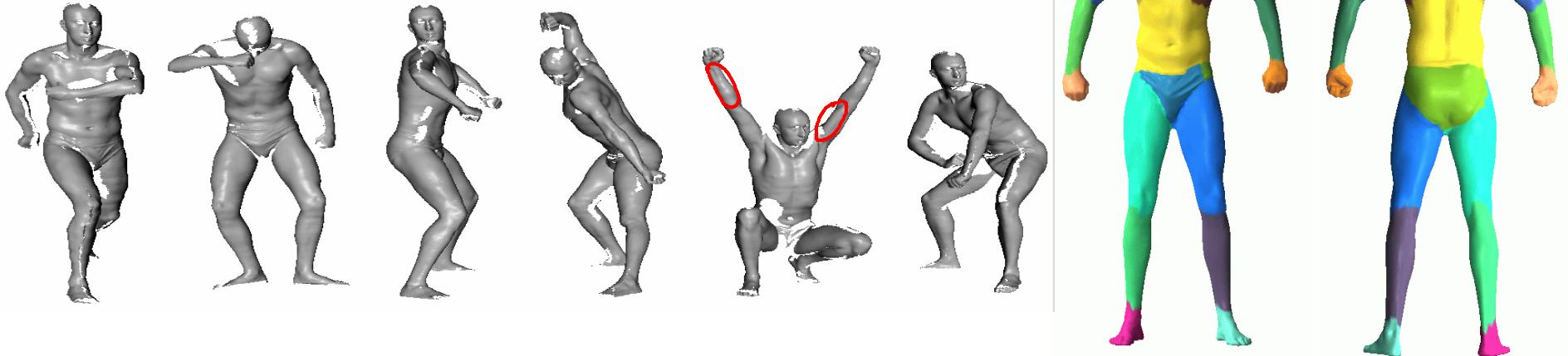


$$p(x_1, x_2, x_3, y_1, y_2, y_3 | z_1, z_2, z_3)$$



Why latent variables?

- Discovering clusters in data



Daphne Koller

Treating Missing Data

Sample sequence: H,T,?,?,H,?,H

- **Case I:** A coin is tossed on a table, occasionally it drops and measurements are not taken

H T H H

- **Case II:** A coin is tossed, but sometimes tails are not reported

H T T + H T H



We need to consider the missing data mechanism

Modeling Missing Data Mechanism

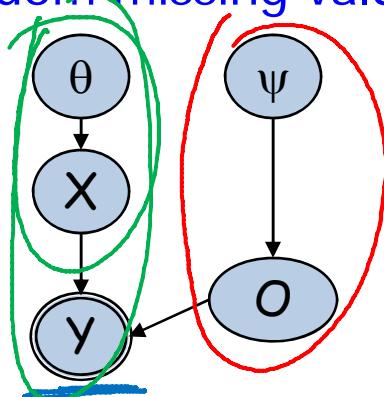
- $X = \{X_1, \dots, X_n\}$ are random variables
- $O = \{O_1, \dots, O_n\}$ are *observability variables*
 - Always observed $O_i = \begin{cases} 1 & X_i \text{ observed} \\ 0 & \text{otherwise} \end{cases}$
- $Y = \{Y_1, \dots, Y_n\}$ new random variables
 - $\text{Val}(Y_i) = \text{Val}(X_i) \cup \{?\}$
 - Always observed
 - Y_i is a deterministic function of X_i and O_i :

$$Y_i = \begin{cases} X_i & O_i = o^1 \\ ? & O_i = o^0 \end{cases}$$

Modeling Missing Data Mechanism

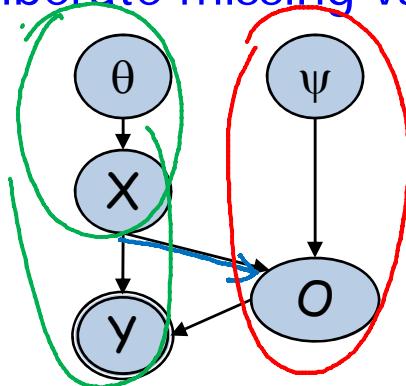
Case I

(random missing values)



Case II

(deliberate missing values)

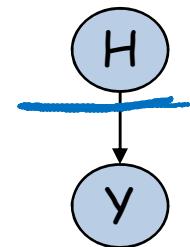


- When can we ignore the missing data mechanism and focus only on the likelihood?
- Missing at Random (MAR)

$$P_{missing} \models (O \perp H \mid d)$$

Identifiability

- Likelihood can have multiple global maxima
- Example:
 - We can rename the values of the hidden variable H
 - If H has two values, likelihood has two global maxima
- With many hidden variables, there can be an exponential number of global maxima
- Multiple local and global maxima can also occur with missing data (not only hidden variables)



Likelihood for Complete Data

Input Data:

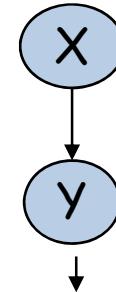
x	y
x^0	y^0
x^0	y^1
x^1	y^0

x^0	x^1
θ_{x^0}	θ_{x^1}

- Likelihood decomposes by variables
- Likelihood decomposes within CPDs

Likelihood:

$$\begin{aligned}
 L(D : \theta) &= P(x[1], y[1]) \cdot P(x[2], y[2]) \cdot P(x[3], y[3]) \\
 &= P(x^0, y^0) \cdot P(x^0, y^1) \cdot P(x^1, y^0) \\
 &= \underbrace{\theta_{x^0} \cdot \theta_{y^0|x^0}}_{\theta_{x^0} \cdot \theta_{y^0|x^0}} \cdot \underbrace{\theta_{x^0} \cdot \theta_{y^1|x^0}}_{\theta_{x^0} \cdot \theta_{y^1|x^0}} \cdot \underbrace{\theta_{x^1} \cdot \theta_{y^0|x^1}}_{\theta_{x^1} \cdot \theta_{y^0|x^1}} \\
 &= (\theta_{x^0} \cdot \theta_{x^0} \cdot \theta_{x^1}) \cdot (\theta_{y^0|x^0} \cdot \theta_{y^1|x^0}) \cdot (\theta_{y^0|x^1})
 \end{aligned}$$



x	$P(Y/x)$	
	y^0	y^1
x^0	$\theta_{y^0 x^0}$	$\theta_{y^1 x^0}$
x^1	$\theta_{y^0 x^1}$	$\theta_{y^1 x^1}$

Likelihood for Incomplete Data

Input Data:

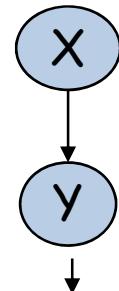
x	y
?	y^0
x^0	y^1
?	y^0

x^0	x^1
θ_{x^0}	θ_{x^1}

Likelihood:

- Likelihood does not decompose by variables
- Likelihood does not decompose within CPDs
- Computing likelihood requires inference!

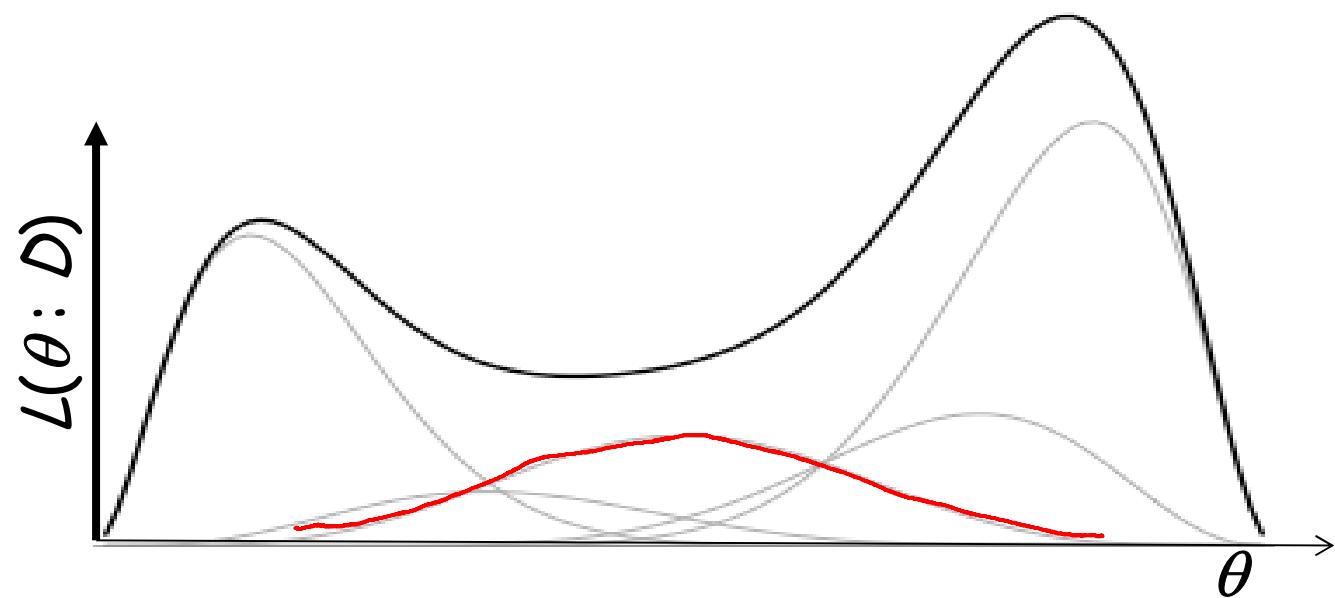
$$\begin{aligned}
 L(D : \theta) &= P(y^0) \cdot P(x^0, y^1) \cdot P(y^0) \\
 &= \left(\sum_{x \in Val(X)} P(x, y^0) \right)^2 \cdot P(x^0, y^1) \cdot \\
 &= \left(\theta_{x^0} \cdot \theta_{y^0|x^0} + \theta_{x^1} \cdot \theta_{y^0|x^1} \right)^2 \cdot \theta_{x^0} \cdot \theta_{y^1|x^0} \\
 &= \left(\theta_{x^0} \cdot \theta_{y^0|x^0} + \theta_{x^1} \cdot \theta_{y^0|x^1} \right)^2 \cdot \theta_{x^0} \cdot \theta_{y^1|x^0}
 \end{aligned}$$



x	$P(Y/X)$	
	y^0	y^1
x^0	$\theta_{y^0 x^0}$	$\theta_{y^1 x^0}$
x^1	$\theta_{y^0 x^1}$	$\theta_{y^1 x^1}$

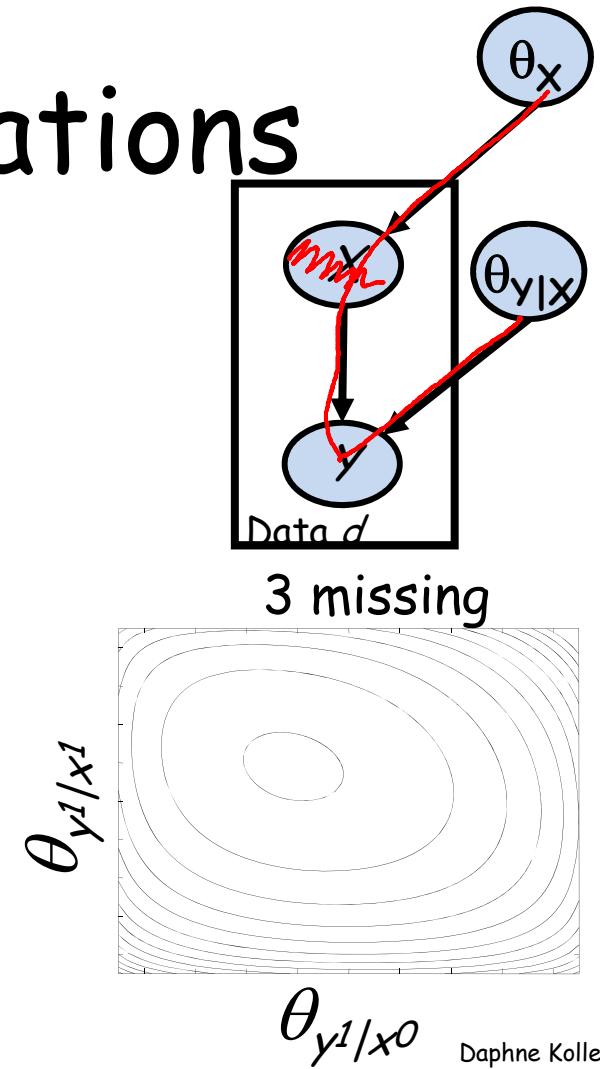
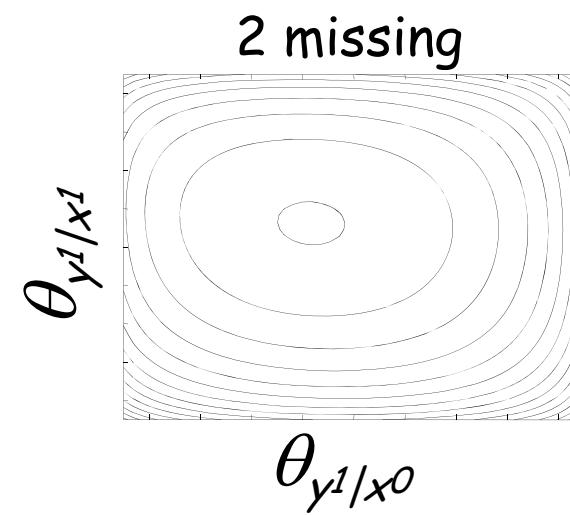
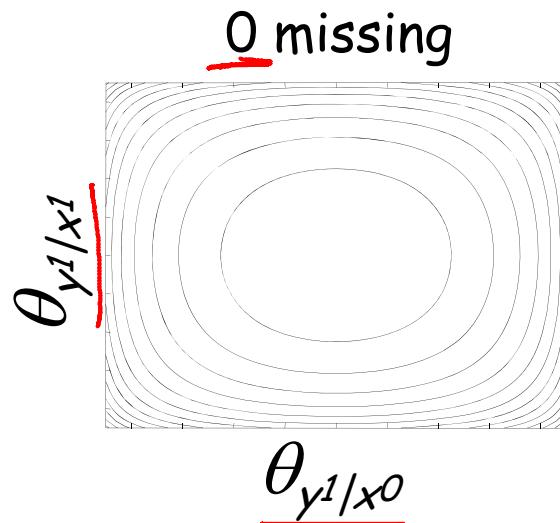
Daphne Koller

Multimodal Likelihood



Parameter Correlations

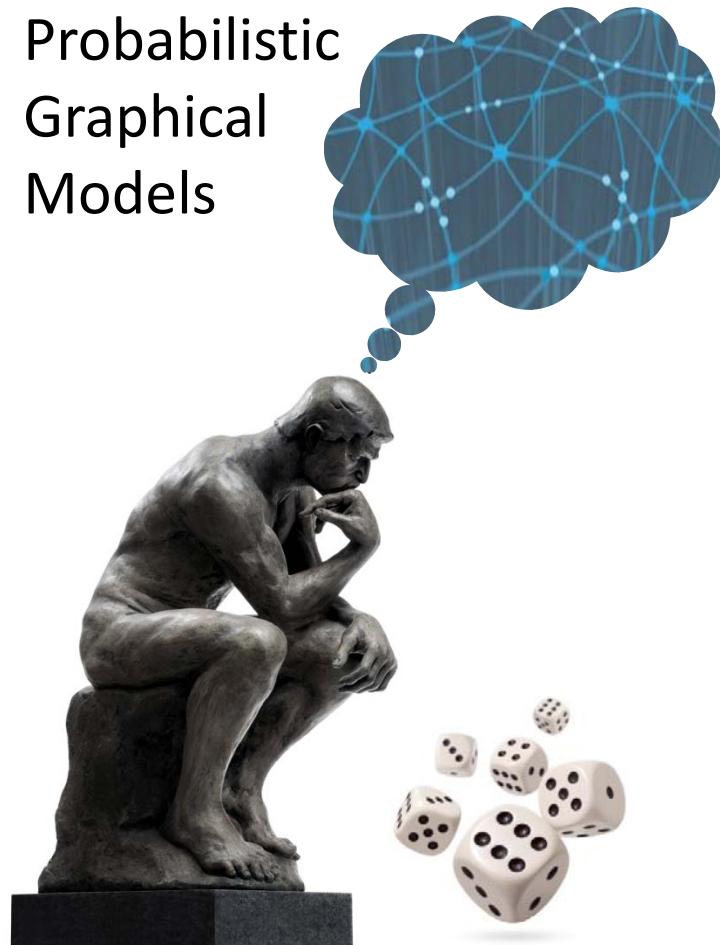
- Total of 8 data points
- Some X's unobserved



Summary

- Incomplete data arises often in practice
- Raises multiple challenges & issues:
 - The mechanism for missingness
 - Identifiability
 - Complexity of likelihood function

Probabilistic
Graphical
Models

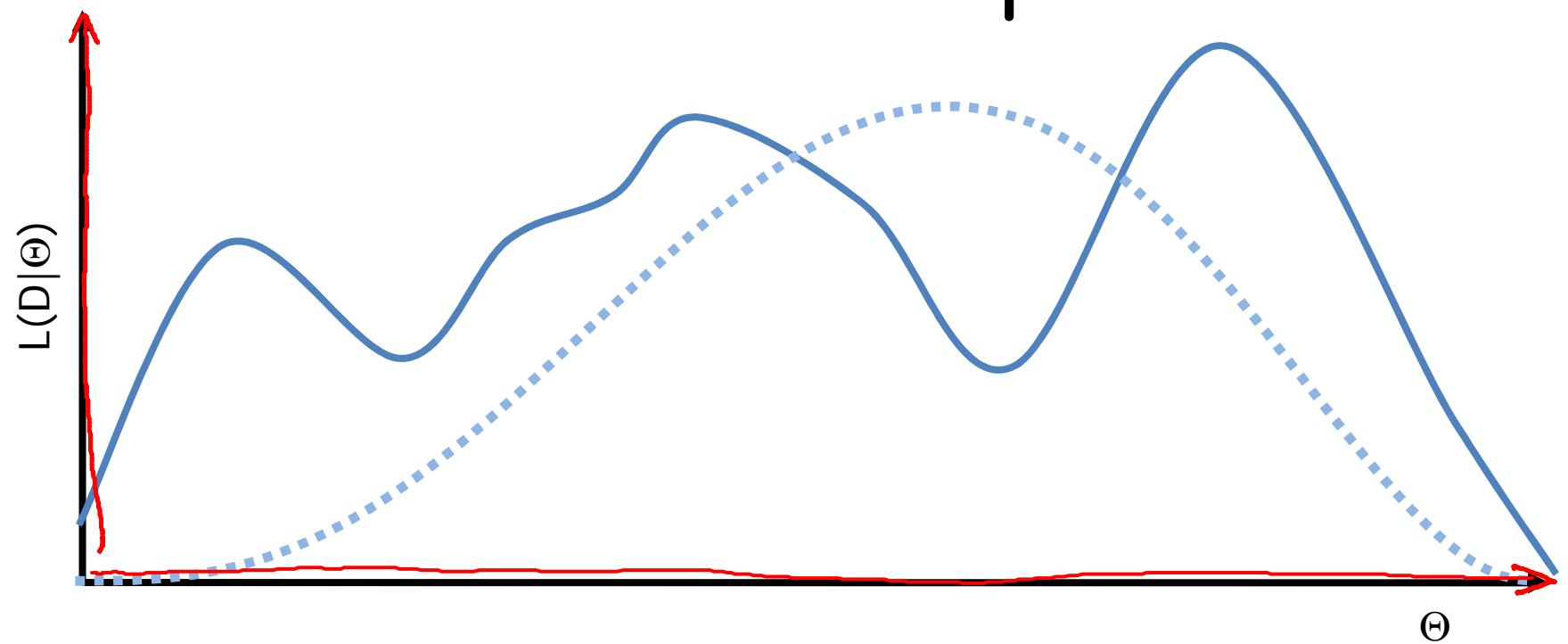


Learning

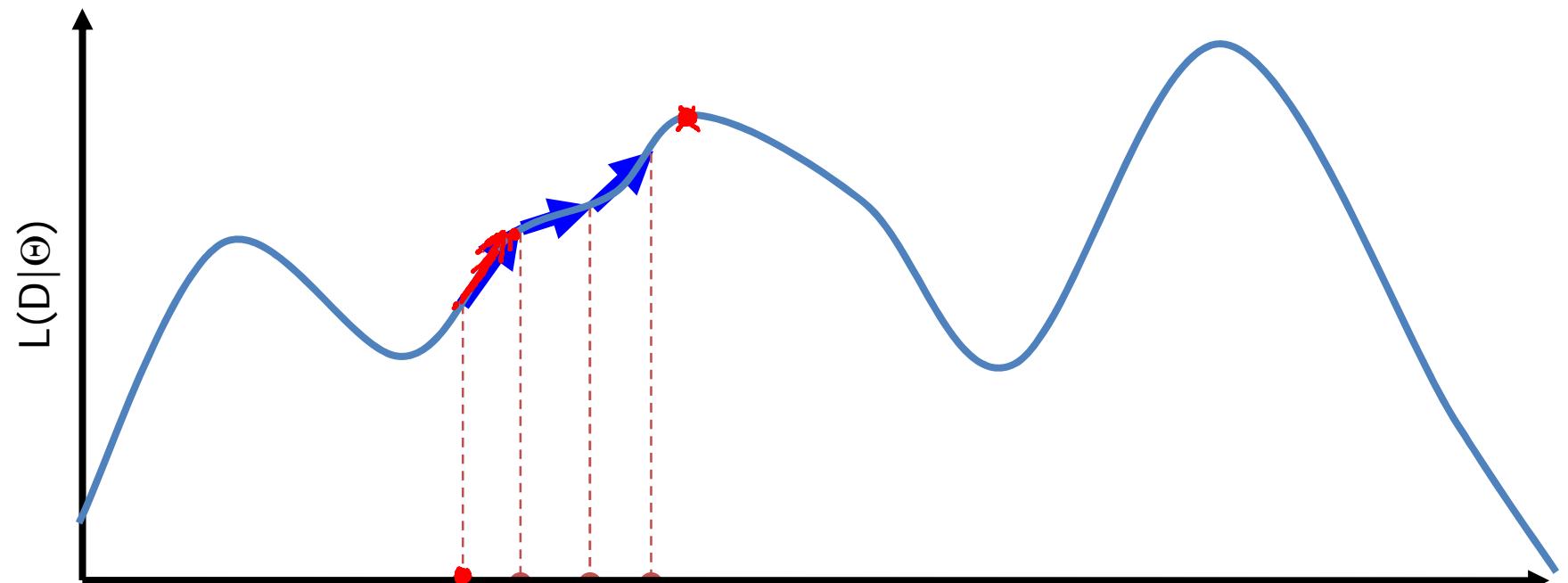
Incomplete Data

Likelihood
Optimization
Methods

Likelihood with Incomplete Data



Gradient Ascent



- Follow gradient of likelihood w.r.t. parameters
- Line search & conjugate gradient methods for fast convergence

θ

Daphne Koller

Gradient Ascent

- Theorem:

$$\frac{\partial \log P(D | \Theta)}{\partial \theta_{x_i | u_i}} = \frac{1}{\theta_{x_i | u_i}} \sum_m P(x_i, u_i | d[m], \Theta)$$

data instances m

evidence in mth instance
current param value

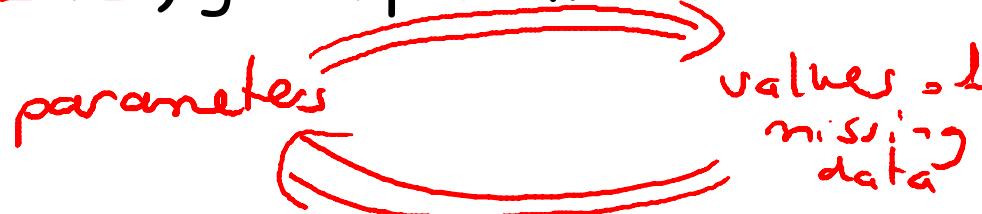
- Requires computing $P(X_i, U_i | d[m], \Theta)$ for all i, m
- Can be done with clique-tree algorithm, since X_i, U_i are in the same clique

Gradient Ascent Summary

- Need to run inference over each data instance at every iteration
 - Pros
 - Flexible, can be extended to non table CPDs
 - Cons
 - Constrained optimization: need to ensure that parameters define legal CPDs
 - For reasonable convergence, need to combine with advanced methods (conjugate gradient, line search)
- chain rule for deriving

Expectation Maximization (EM)

- Special-purpose algorithm designed for optimizing likelihood functions
- Intuition
 - Parameter estimation is easy given complete data
 - Computing probability of missing data is “easy” (=inference) given parameters



EM Overview

- Pick a starting point for parameters
- Iterate:
 - E-step (Expectation): “Complete” the data using current parameters
 - M-step (Maximization): Estimate parameters relative to data completion
- Guaranteed to improve $L(\theta : D)$ at each iteration

Expectation Maximization (EM)

- **Expectation (E-step):**

- For each data case $\underline{d[m]}$ and each family $\underline{X, U}$ compute
- Compute the expected sufficient statistics for each x, u

$$\overline{M}_{\theta^t}[x, u] = \sum_{m=1}^M P(x, u | d[m], \theta^t)$$

soft completion

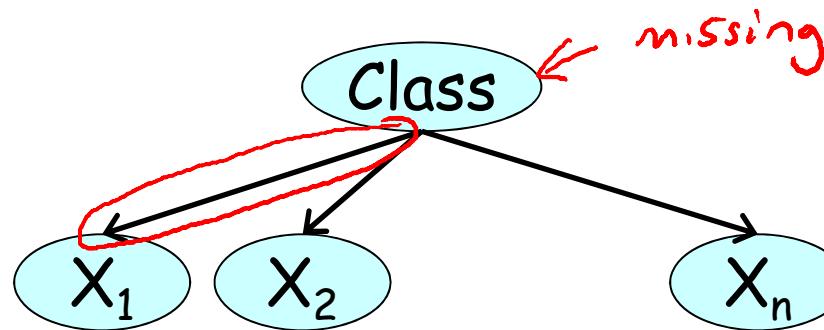
$$\underbrace{P(X, U | d[m], \theta^t)}_{m \in \{x, u\}}$$

- **Maximization (M-step):**

- Treat the expected sufficient statistics (ESS) as if real
- Use MLE with respect to the ESS

$$\underline{\theta}_{x|u}^{t+1} = \frac{\overline{M}_{\theta^t}[x, u]}{\overline{M}_{\theta^t}[u]}$$

Example: Bayesian Clustering

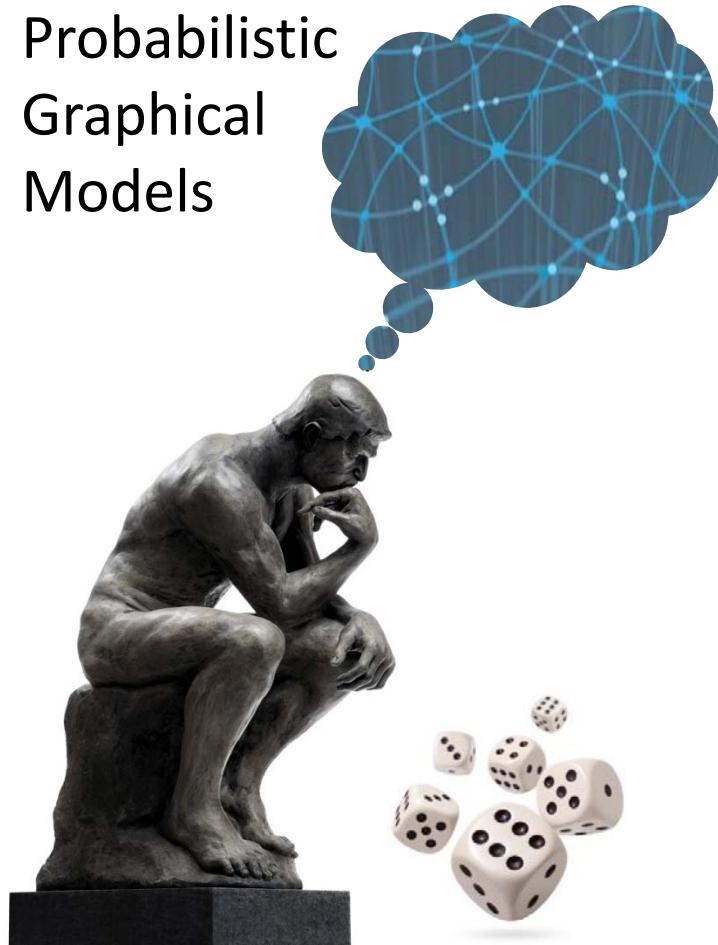


$$\begin{aligned} \bar{M}_{\theta}[c] &:= \sum_m P(c | \underline{x_1[m]}, \dots, \underline{x_n[m]}, \theta^t) & \theta_c^{t+1} &= \frac{\bar{M}_{\theta}[c]}{M} \\ \bar{M}_{\theta}[x_i, c] &:= \sum_m P(c, x_i | \underline{x_1[m]}, \dots, \underline{x_n[m]}, \theta^t) & \theta_{x_i|c}^{t+1} &:= \frac{\bar{M}_{\theta}[x_i, c]}{\bar{M}_{\theta}[c]} \end{aligned}$$

EM Summary

- Need to run inference over each data instance at every iteration
- Pros
 - Easy to implement on top of MLE for complete data
 - Makes rapid progress, especially in early iterations
- Cons
 - Convergence slows down at later iterations

Probabilistic
Graphical
Models

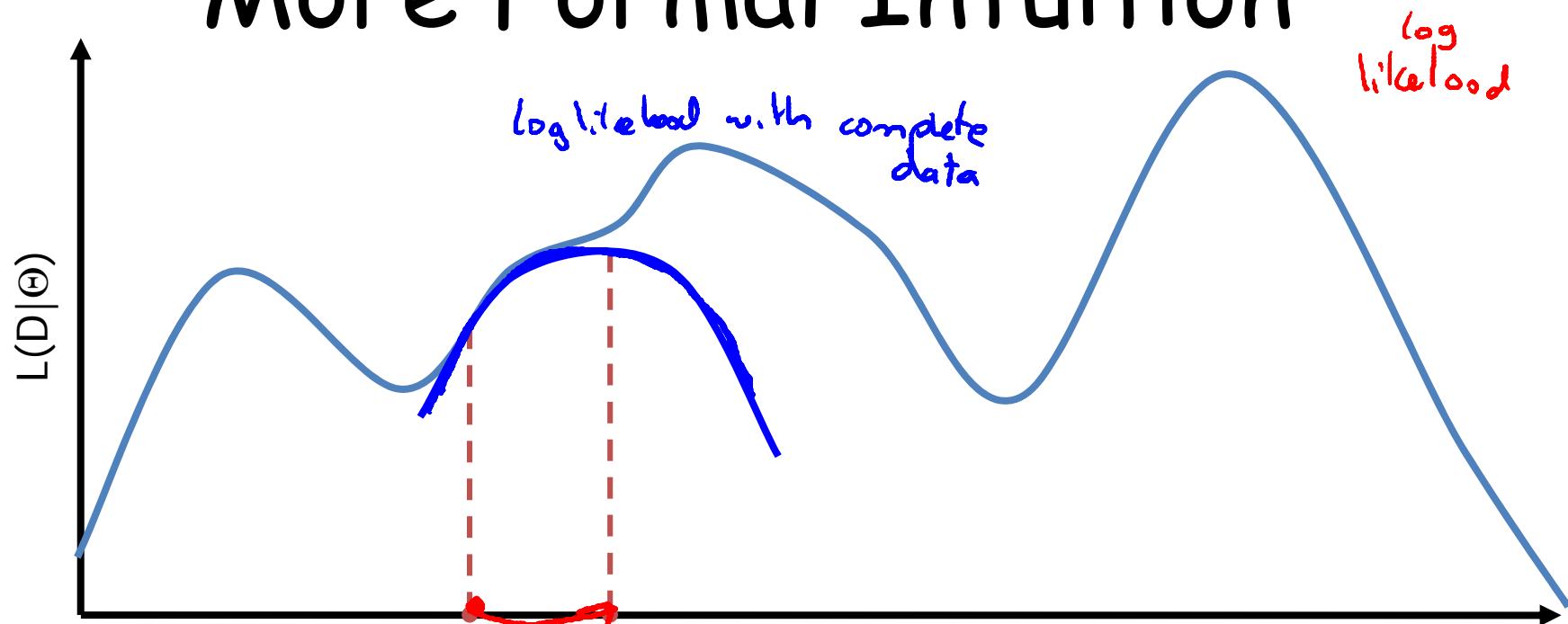


Learning

Incomplete Data

EM Analysis

More Formal Intuition



- Use current point to construct local approximation ^{$\hat{\theta}$}
- Maximize new function in closed form

More Formal Intuition

- \underline{d} : observed data in instance
- \underline{H} : hidden variables in instance
- $Q(H)$: distribution over hidden variables

$$\ell(\theta : \langle d, h \rangle) = \sum_{i=1}^n \sum_{(x_i, u_i) \in Val(X_i, \text{Pa}_{X_i})} 1_{\langle d, h \rangle}[x_i, u_i] \log \theta_{x_i | u_i}$$

assignment to H *log parameter*

$$E_{Q(H)}[\ell(\theta : \langle d, H \rangle)] = \sum_{i=1}^n \sum_{(x_i, u_i) \in Val(X_i, \text{Pa}_{X_i})} E_{Q(H)}[1_{\langle d, H \rangle}[x_i, u_i]] \log \theta_{x_i | u_i}$$

$$= \sum_{i=1}^n \sum_{(x_i, u_i) \in Val(X_i, \text{Pa}_{X_i})} Q(x_i, u_i) \log \theta_{x_i | u_i}$$

More Formal Intuition

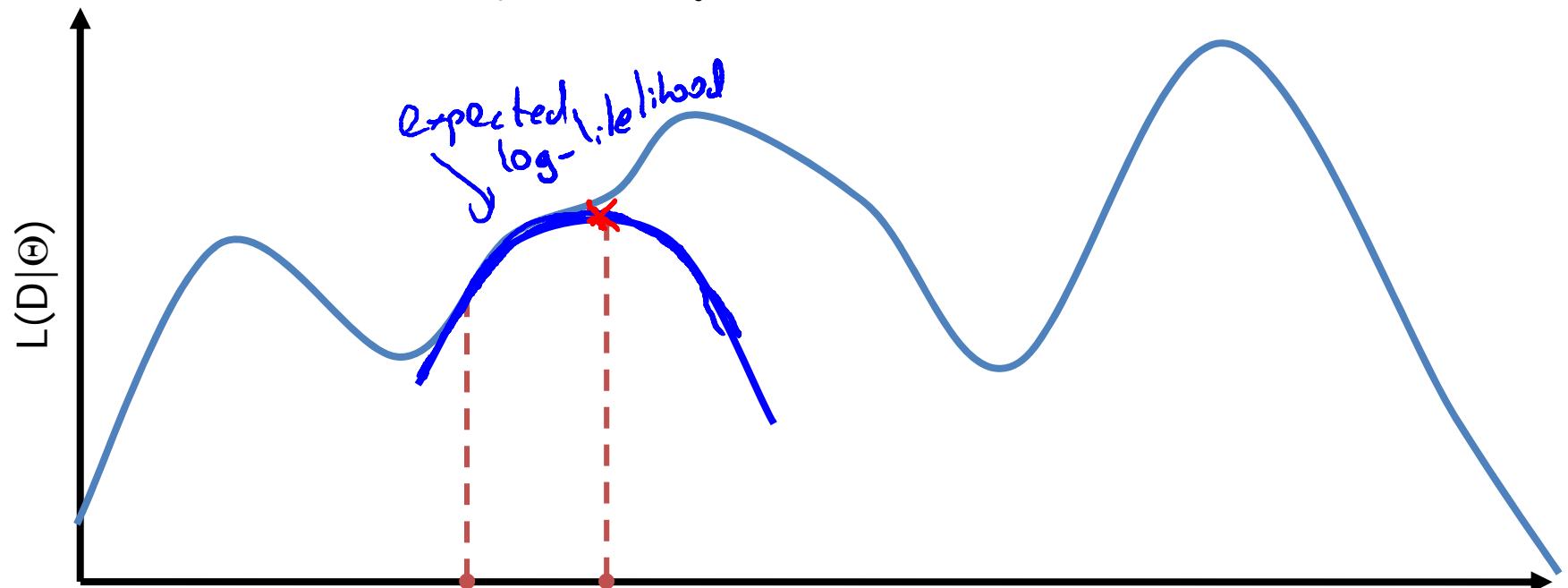
$$E_{Q(\mathbf{H})}[\ell(\boldsymbol{\theta} : \langle \mathbf{d}, \mathbf{H} \rangle)] = \sum_{i=1}^n \sum_{(x_i, u_i)} Q(x_i, u_i) \log \theta_{x_i | u_i}$$

$$\underline{Q_m^t(\mathbf{H}[m])} = \underline{P(\mathbf{H}[m] \mid \mathbf{d}[m], \boldsymbol{\theta}^t)}$$

$$\begin{aligned} & \sum_{m=1}^M \underline{E_{Q_m^t(\mathbf{H}[m])}[\ell(\boldsymbol{\theta} : \langle \mathbf{d}[m], \mathbf{H}[m] \rangle)]} \\ &= \sum_{i=1}^n \sum_{(x_i, u_i)} \left(\sum_{m=1}^M \underline{P(x_i, u_i \mid \mathbf{d}[m], \boldsymbol{\theta}^t)} \right) \underline{\log \theta_{x_i | u_i}} \\ &= \sum_{i=1}^n \sum_{(x_i, u_i)} \underline{\bar{M}_{\boldsymbol{\theta}^t}[x_i, u_i]} \underline{\text{ESS}} \end{aligned}$$

expected sum
stats
log likelihood for
complete data
using ESS

More Formal Intuition



- Use current point to construct local approximation
- Maximize new function in closed form

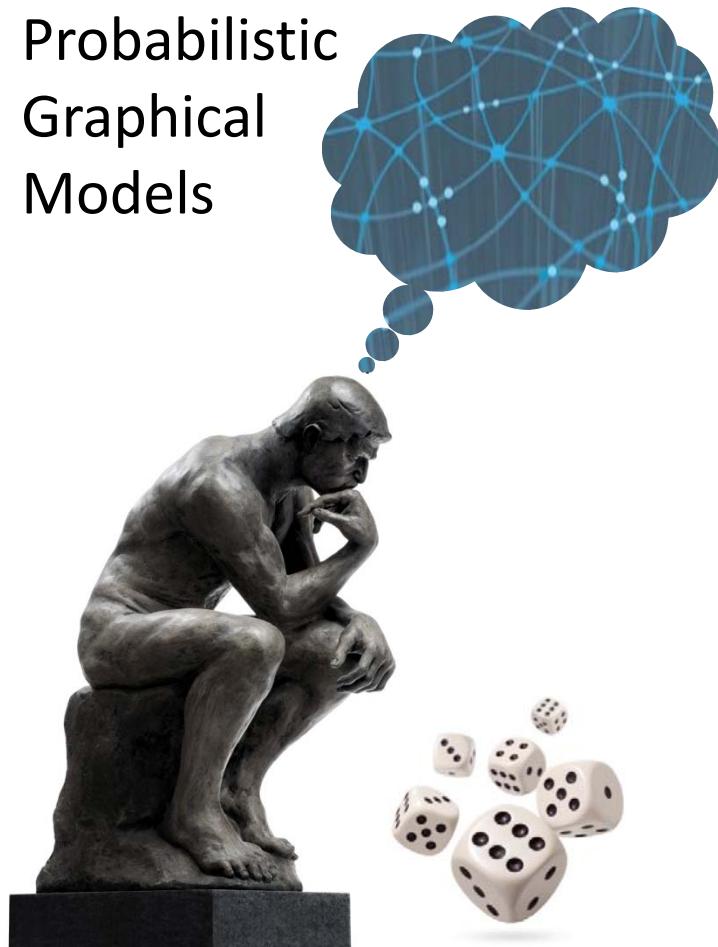
EM Guarantees

- $\underline{L(D : \theta^{t+1})} \geq \underline{L(D : \theta^t)}$
 - Each iteration improves the likelihood
- If $\underline{\theta^{t+1}} = \underline{\theta^t}$, then $\underline{\theta^t}$ is a stationary point of $\underline{L(D : \theta)}$
 - Usually, this means a local maximum

gradient is zero



Probabilistic
Graphical
Models

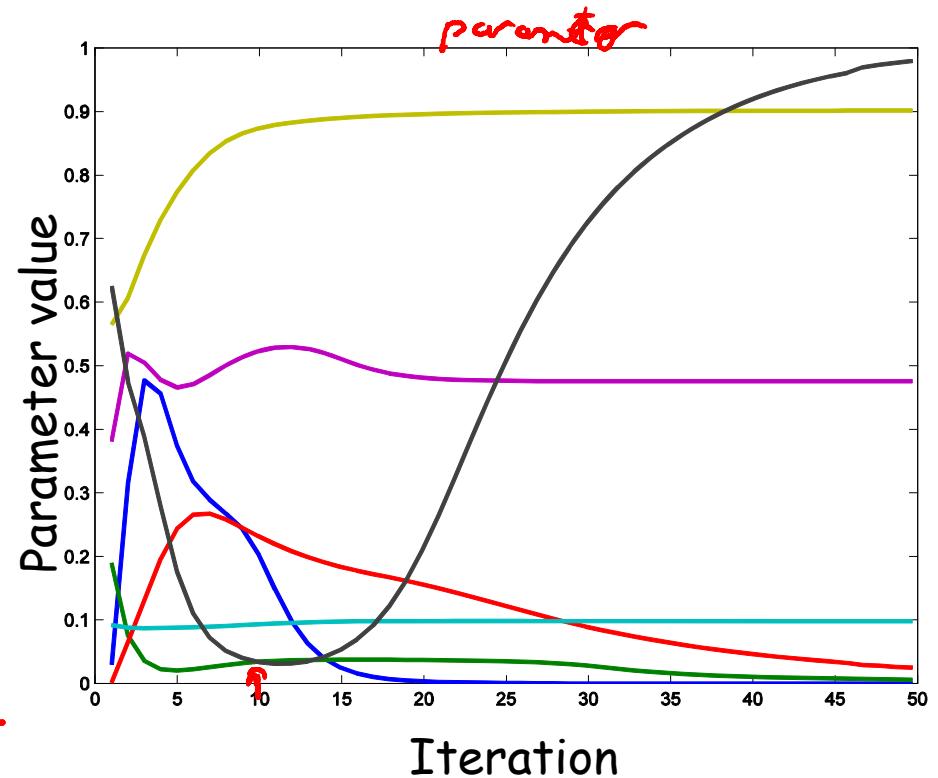
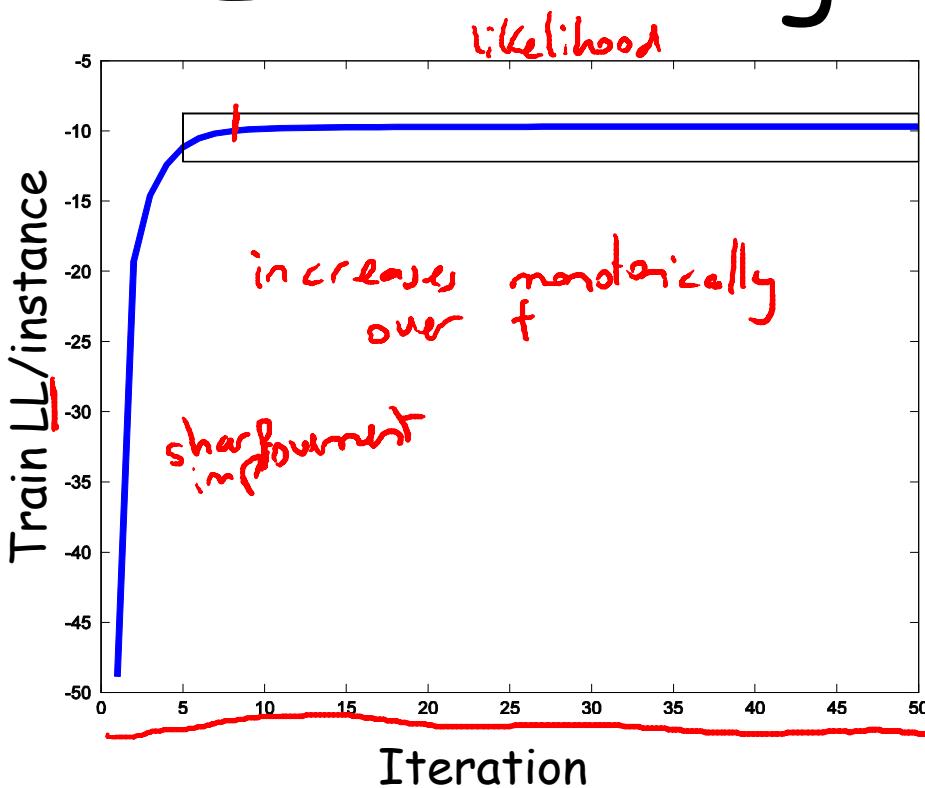


Learning

Incomplete Data

EM in Practice

EM Convergence in Practice

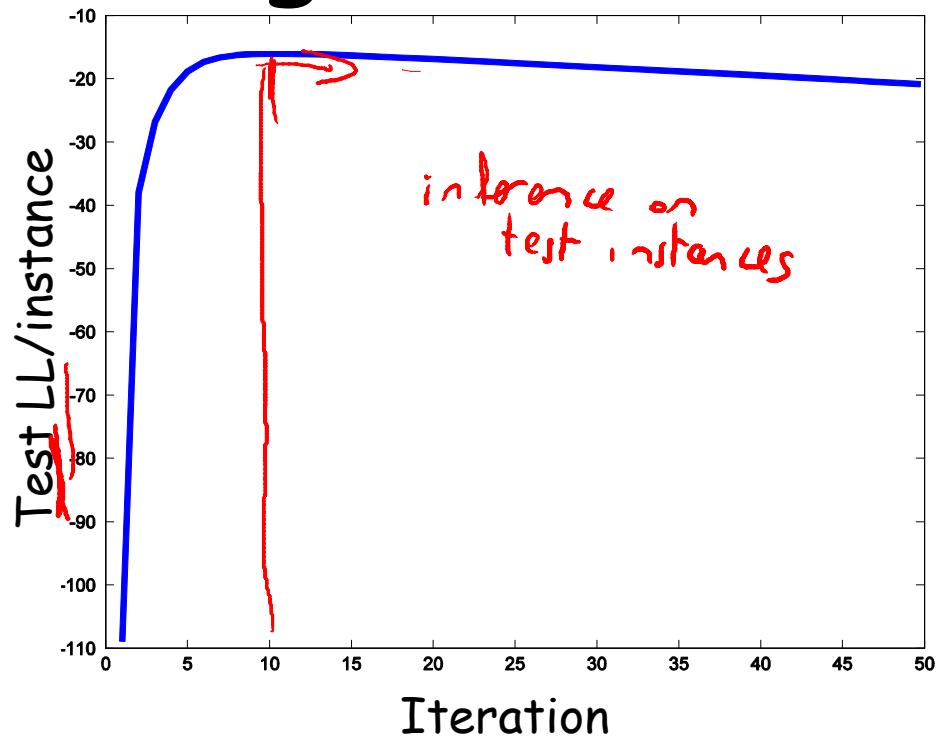
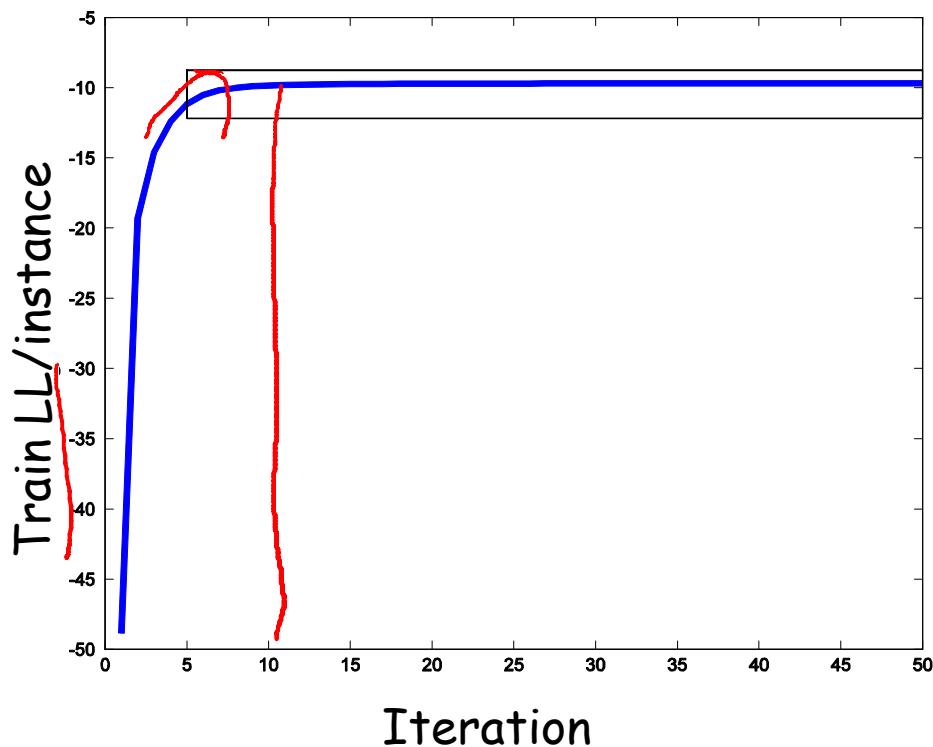


G. Elidan

Daphne Koller

Overfitting

(numerical)

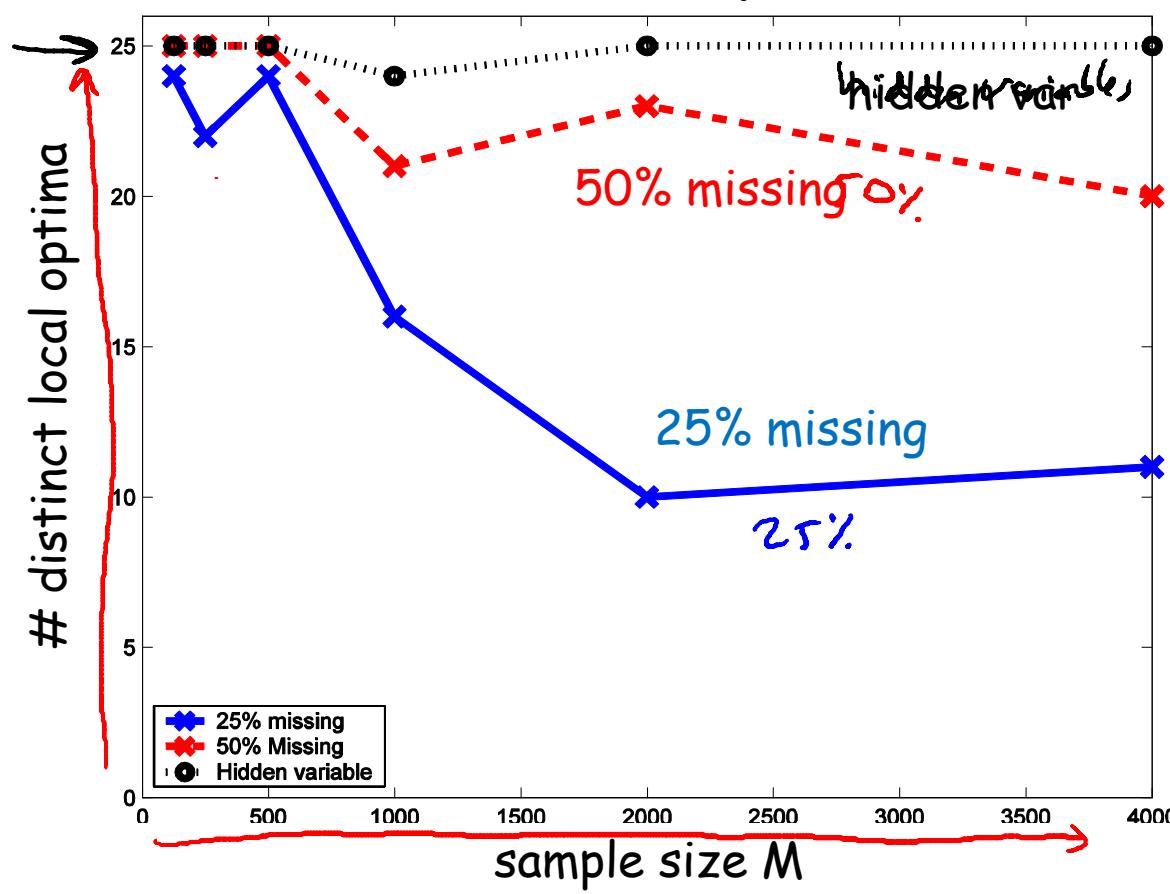


- Early stopping using cross validation
- Use MAP with parameter priors rather than MLE

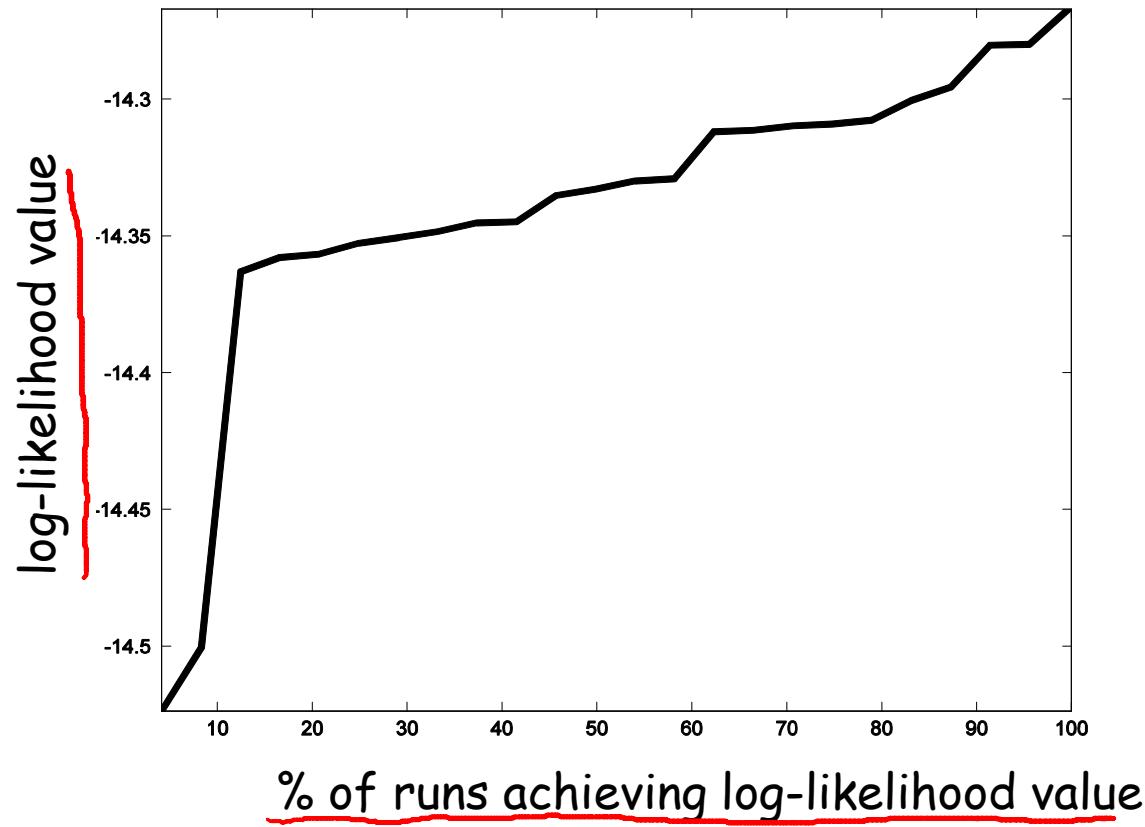
G. Elidan

Daphne Koller

Local Optima



Significance of Local Optima



G. Elidan

% of runs achieving log-likelihood value

Daphne Koller

Initialization is Critical

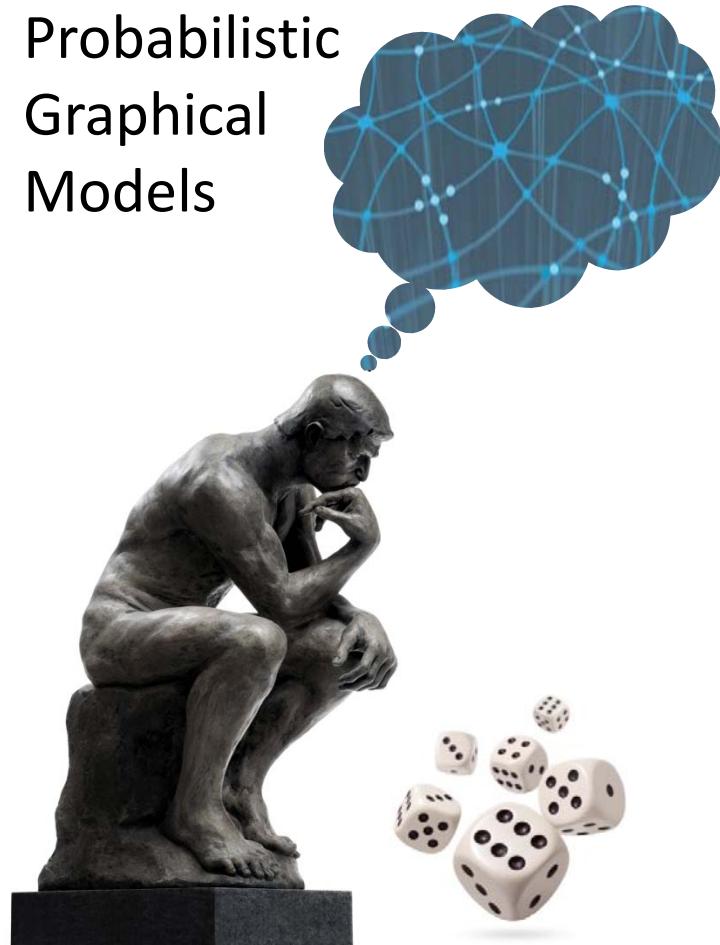
- Multiple random restarts
- From prior knowledge
- From the output of a simpler algorithm

clustering (k-means
hierarchical
agglomerative
clustering)

Summary

- Convergence of likelihood \neq convergence of parameters
- Running to convergence can lead to overfitting
- Local optima are unavoidable, and increase with the amount of missing data
- Local optima can be very different
- Initialization is critical

Probabilistic
Graphical
Models

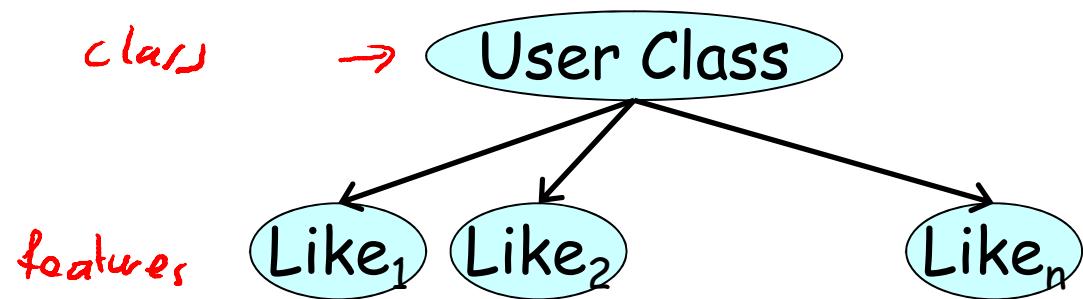


Learning

Incomplete Data

Learning with
Latent
Variables

Discovering User Clusters



J. Breese

MSNBC Story clusters

(Readers of commerce and technology stories (36%):

- E-mail delivery isn't exactly guaranteed
- Should you buy a DVD player?
- Price low, demand high for Nintendo

Sports Readers (19%):

- Umps refusing to work is the right thing
- Cowboys are reborn in win over eagles
- Did Orioles spend money wisely?

Readers of top promoted stories (29%):

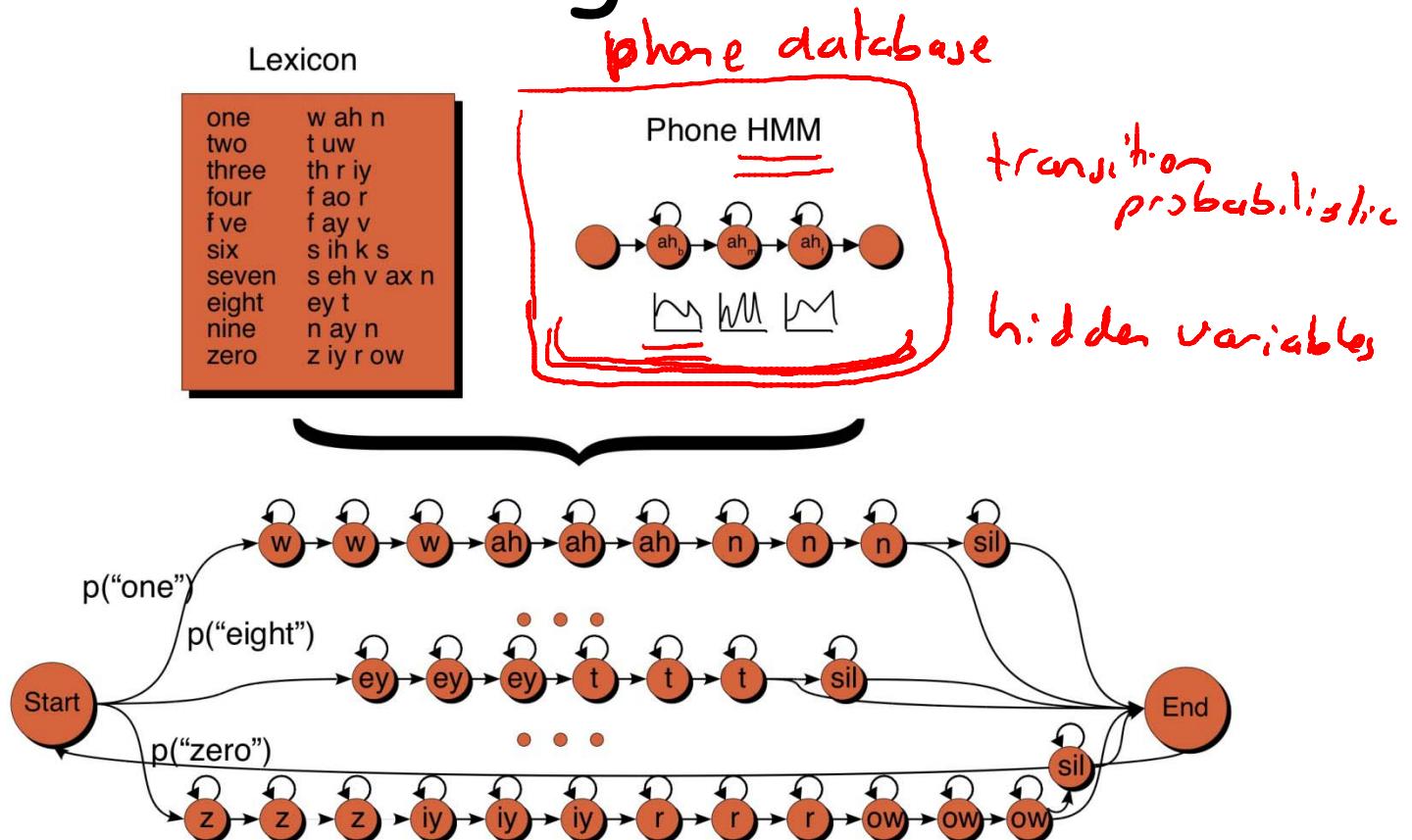
- 757 Crashes At Sea
- (Israel, Palestinians Agree To Direct Talks
- (Fuhrman Pleads Innocent To Perjury

Readers of "Softer" News (12%):

- The truth about what things cost
- Fuhrman Pleads Innocent To Perjury
- Real Astrology

Daphne Koller

Speech Recognition HMM



Dan Jurafsky, Stanford

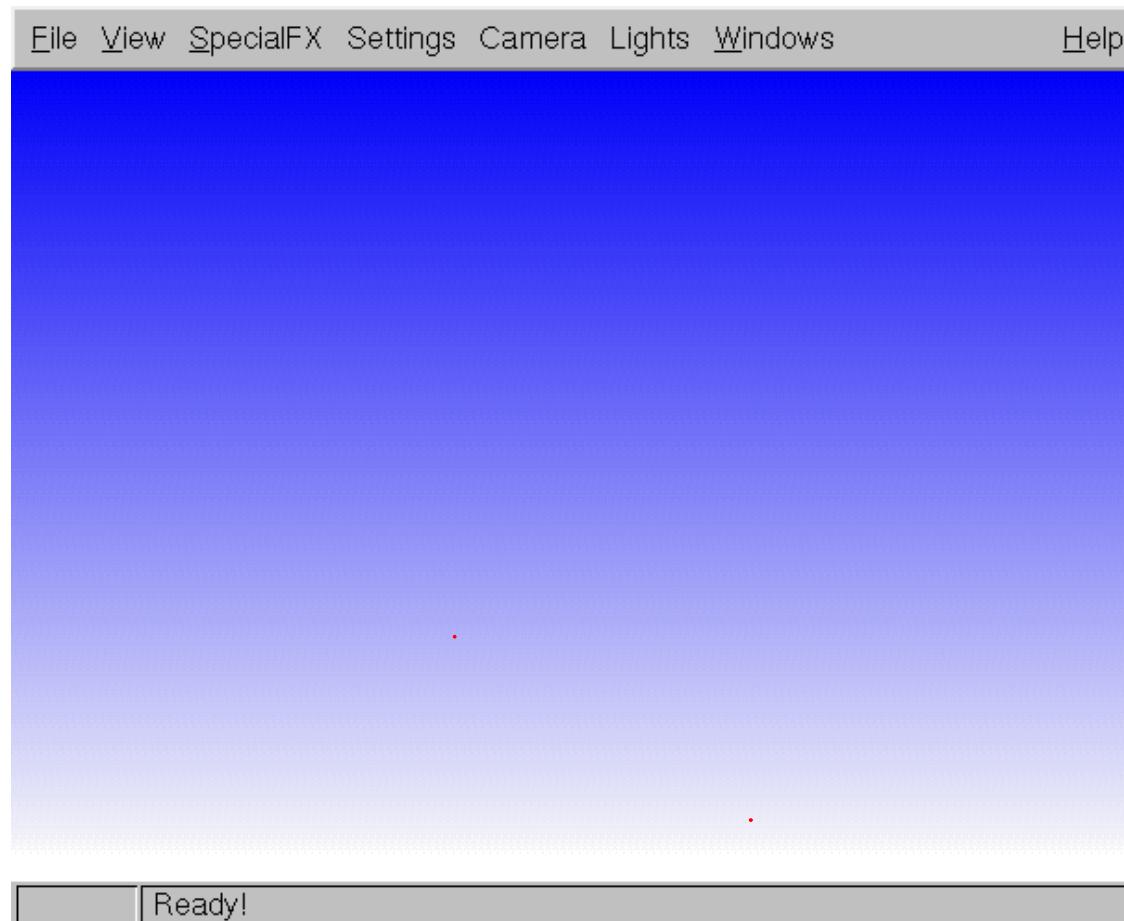
Daphne Koller

3D Robot Mapping

- Input: Point cloud from laser range finder obtained by moving robot
 - Output: 3D planar map of environment
-
- Parameters: Location & angle of walls (*planes*)
 - Latent variables: Assignment of points to walls
association

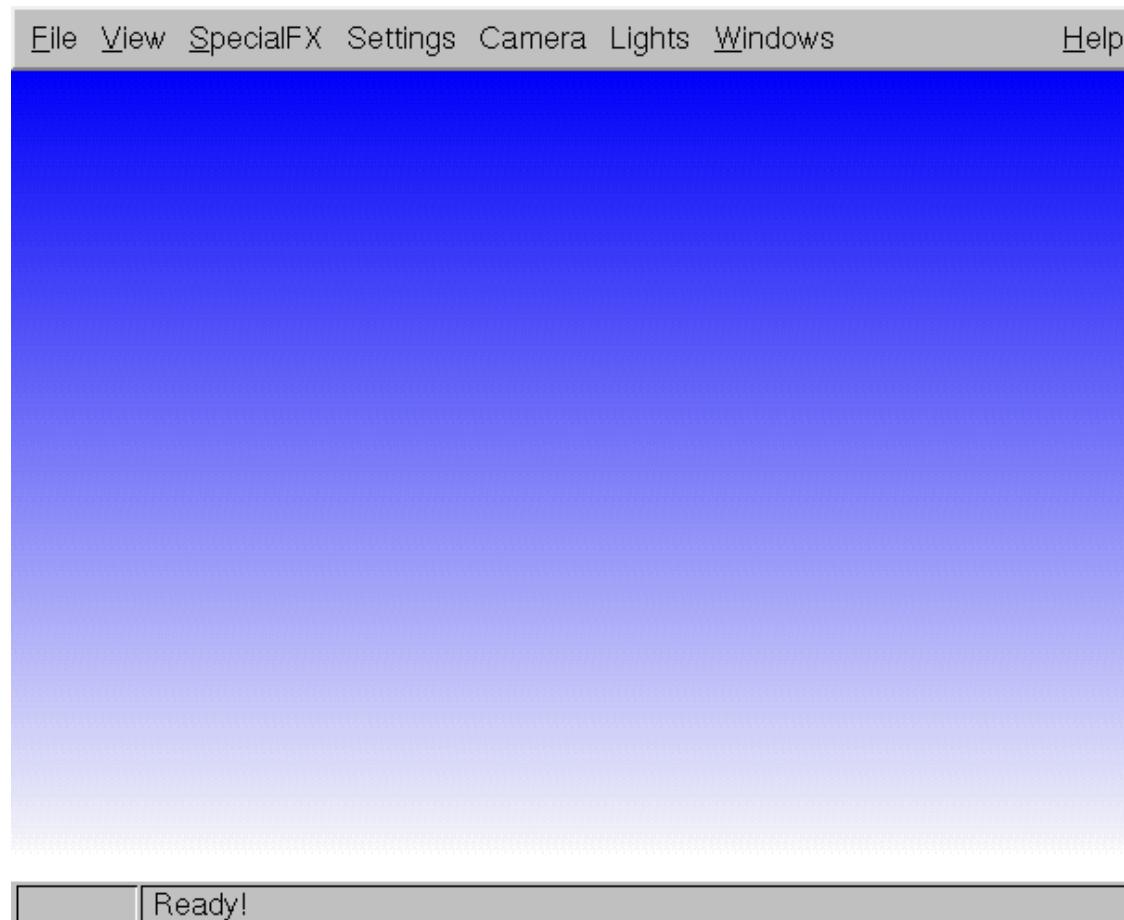
Thrun, Martin, Liu, Haehnel, Emery-Montemerlo, Chakrabarti, Burgard,
IEEE Transactions on Robotics, 2004

Daphne Koller



Thrun, Martin, Liu, Haehnel, Emery-Montemerlo, Chakrabarti, Burgard,
IEEE Transactions on Robotics, 2004

Daphne Koller



Thrun, Martin, Liu, Haehnel, Emery-Montemerlo, Chakrabarti, Burgard,
IEEE Transactions on Robotics, 2004

Daphne Koller

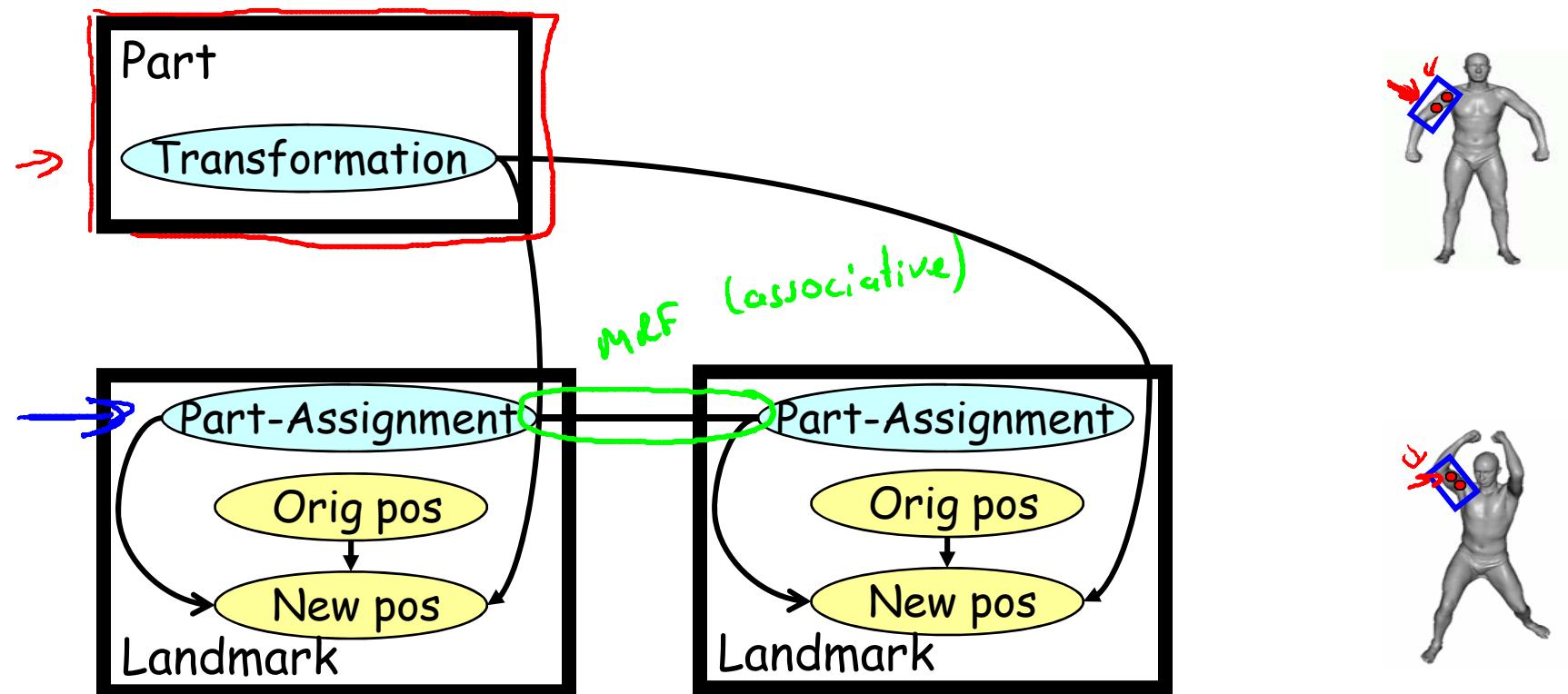
Body Parts from Point Cloud Scans



Anguelov, Koller, Pang, Srinivasan, Thrun UAI 2004

Daphne Koller

Collective Clustering Model



Anguelov, Koller, Pang, Srinivasan, Thrun UAI 2004

Daphne Koller



Anguelov, Koller, Pang, Srinivasan, Thrun UAI 2004

Daphne Koller

Helicopter Demo Alignment

- Input: Multiple sample trajectories by different pilots flying same sequence
- Output:
 - Aligned trajectories
 - Model of "template" trajectory



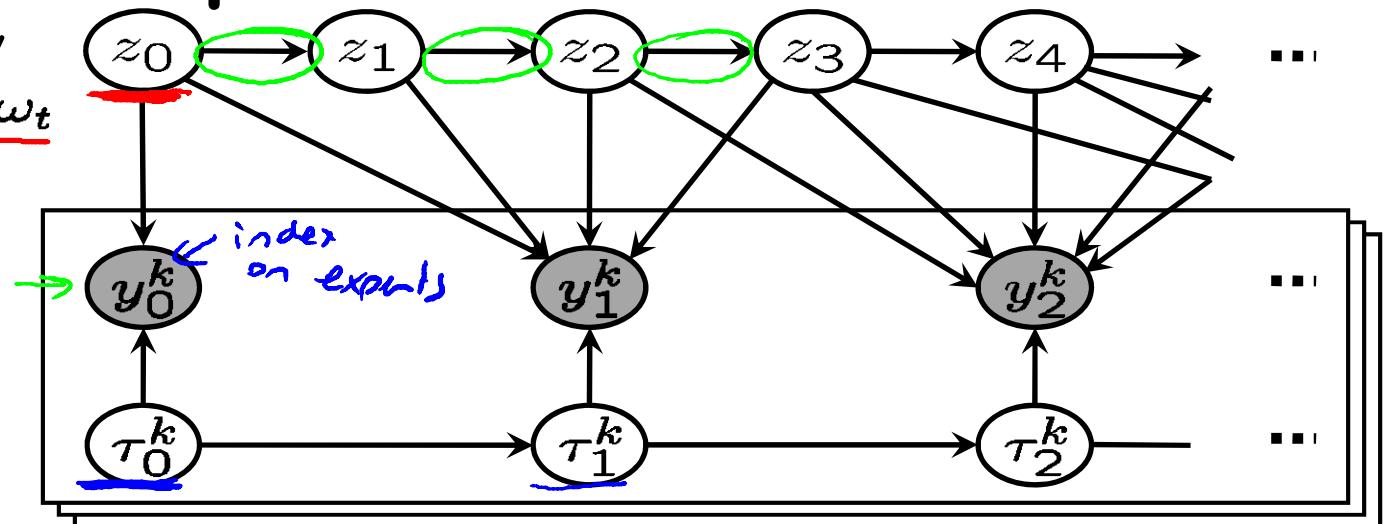
Coates, Abbeel, Ng, ICML 2008

Daphne Koller

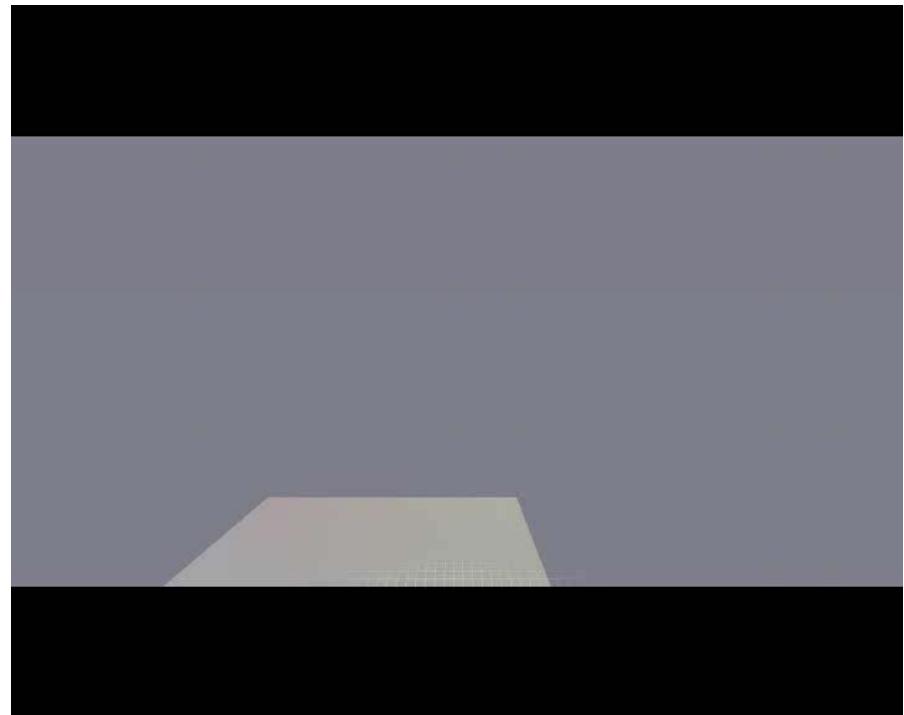
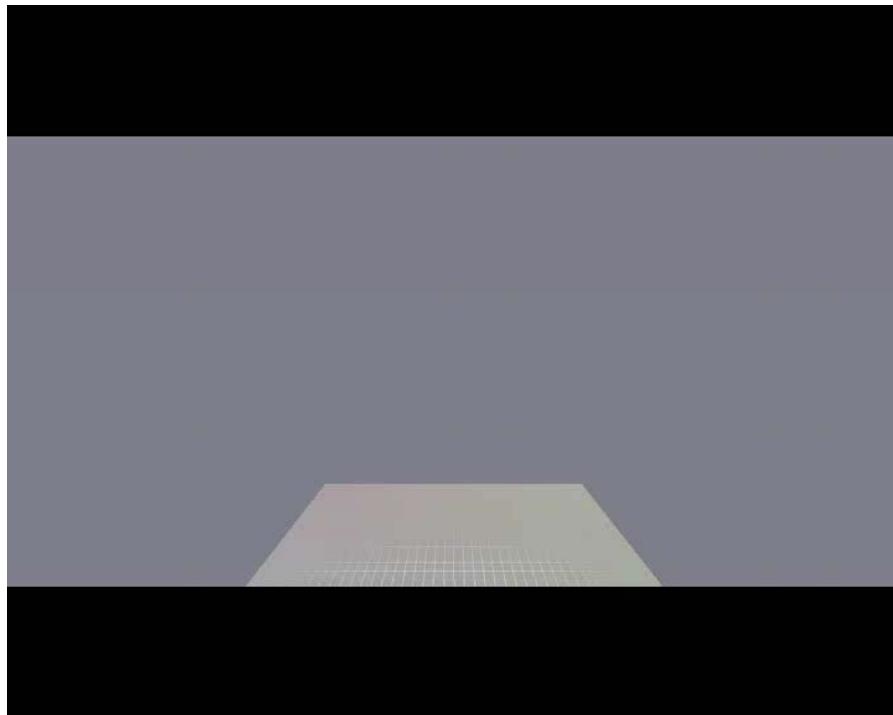
Graphical model

Intended trajectory
 $z_{t+1} = \underline{f(z_t) + \omega_t}$

Expert demonstrations
 $y_j = z_{\tau_j} + \nu_j$
Time indices



All Expert Demos



Coates, Abbeel, Ng, ICML 2008

Daphne Koller

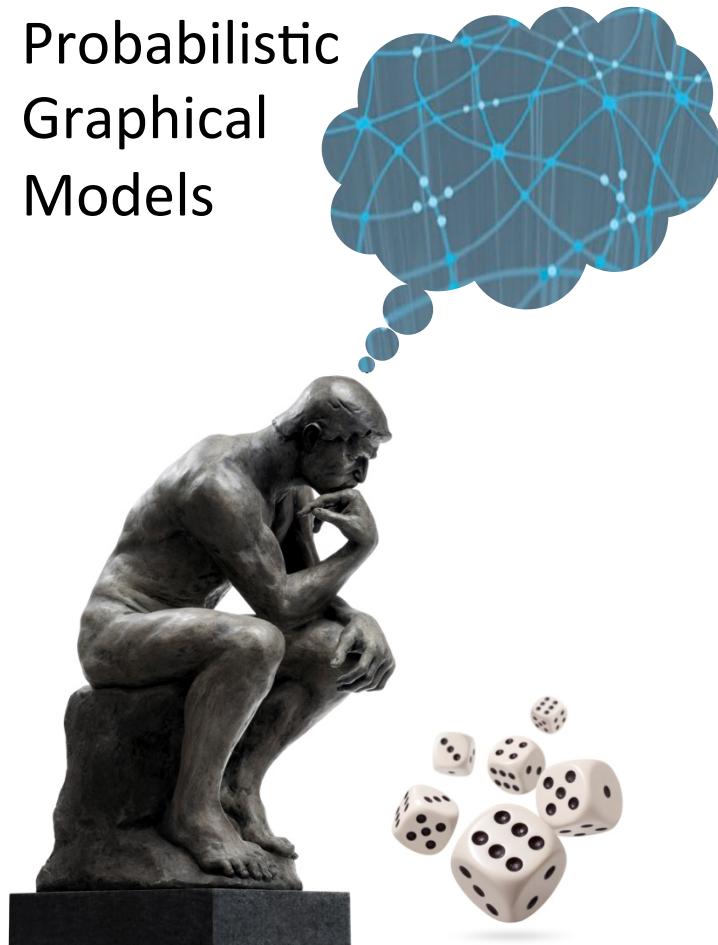
Picking Latent Variable Cardinality

- If we use likelihood for evaluation, more values is always better
- Can use score that penalizes complexity
 - BIC - tends to underfit
 - Extensions of BDe to incomplete data (approximations)
- Can use metrics of cluster coherence to decide whether to add/remove clusters
- Bayesian methods (Dirichlet processes) can average over different cardinalities
(MCMC) *distribution over cardinality*

Summary

- Latent variables are perhaps the most common scenario for incomplete data
 - often a critical component in constructing models for richly structured domains
- Latent variables satisfy MAR, so can use EM
- Serious issues with unidentifiability & multiple optima
necessitate good initialization
- Picking variable cardinality is a key question

Probabilistic
Graphical
Models



Representation

Wrapup

Knowledge
Engineering

Important Distinctions

- Template based versus specific
- Directed versus undirected
- Generative versus discriminative
- Hybrids are also common

Important Distinctions

Template-based

image segmentation

Specific

medical diagnosis

fault diagnosis

small number of variable types

features are most predictive

large number of "unique" variables

Important Distinctions

Generative

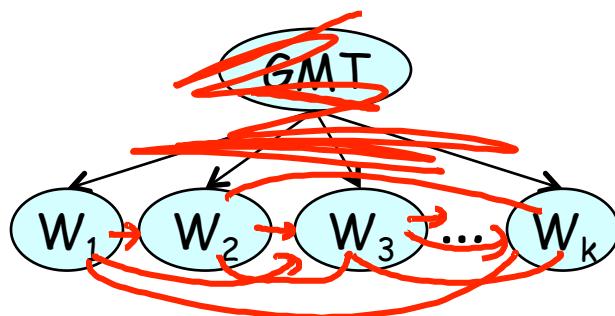
task shifts
(easier to train in
certain regimes)

Discriminative

particular prediction task
richly expressive features
(avoid dealing w. correlations)
 \Rightarrow high performance

Variable Types

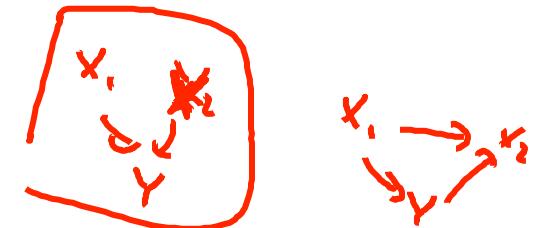
- Target ↵
- Observed ↵
 - Including complex, constructed features
- Latent — *simplify our structure*
hidden



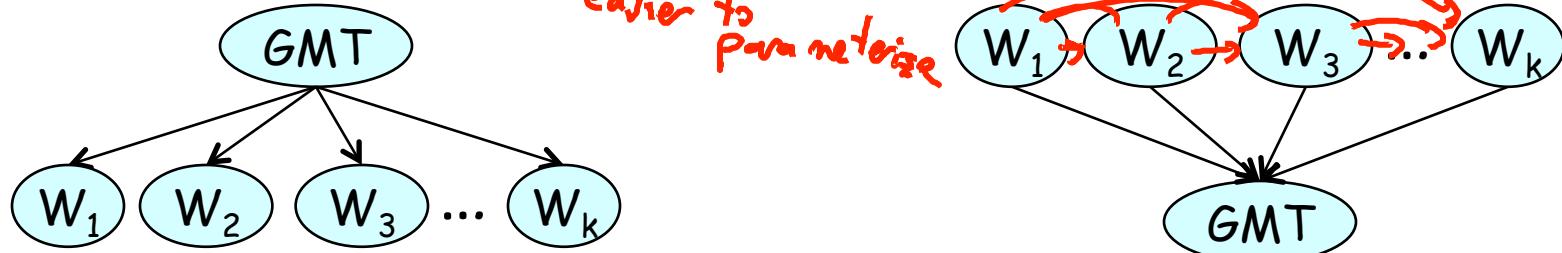
$$X \rightarrow Y$$

$$Y \rightarrow X$$

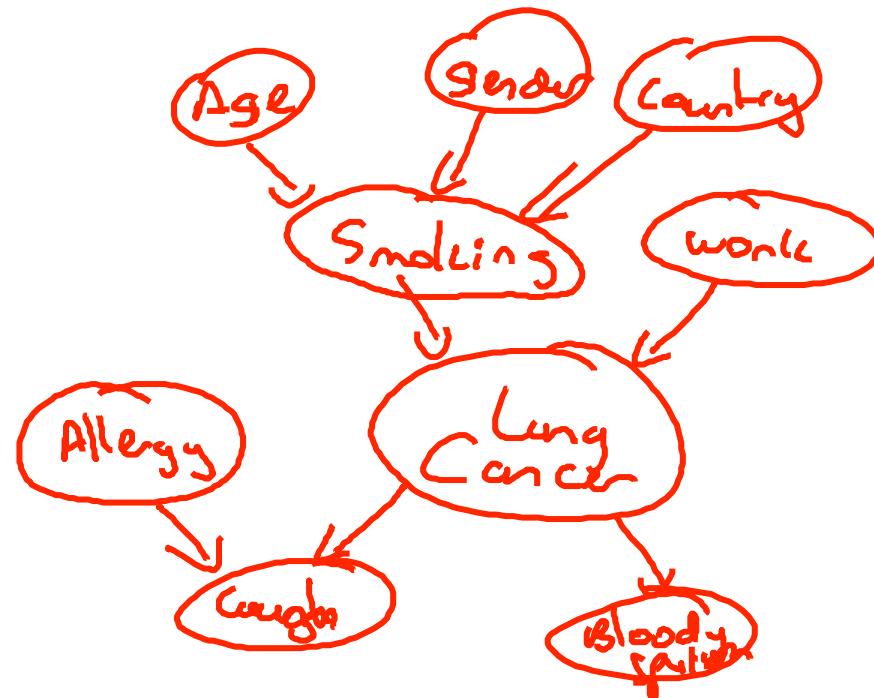
Structure



- Causal versus non-causal ordering



Extending the Conversation



Parameters: Values

- What matters:
 - Zeros
 - Orders of magnitude $\frac{1}{10}$ vs $\frac{1}{100}$
 - Relative values \leftarrow
- Structured CPDs

Parameters: Local Structure

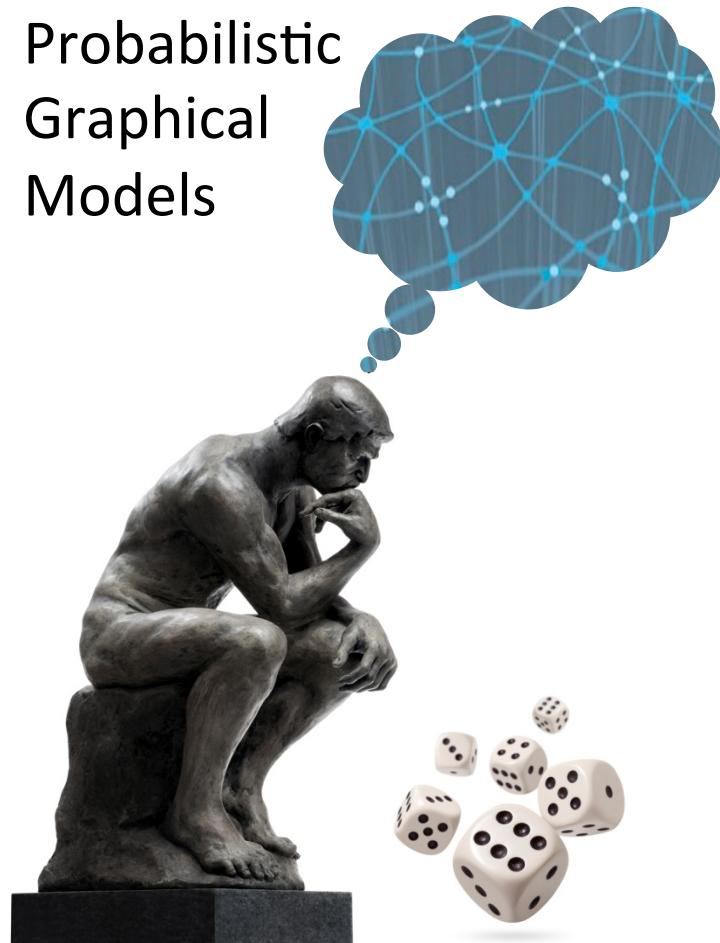
- Table CPDs are the exception

	<u>Context-specific</u>	<u>Aggregating</u>
<u>Discrete</u>	tree CPDs	Sigmoid noisy OR
<u>Continuous</u>	regression tree (thresholds) CLG	Linear Gaussian

Iterative Refinement

- Model testing
- Sensitivity analysis for parameters
- Error analysis
 - Add features
 - Add dependencies

Probabilistic
Graphical
Models



Learning

Summary

Methods, Parameters, and Evaluation

Learning from 10K Feet

- Hypothesis (model) space
- Objective function
- Optimization algorithm

Hypothesis Space

- What are we searching for
 - Parameters ←
 - Structure ←
- Imposing constraints
 - For computational efficiency ←
 - To reduce model capacity ←
 - To incorporate prior knowledge ←

Objective Function

- Penalized likelihood
 - $\ell((G, \theta_G) : D) + R(G, \theta_G)$
 - Parameter prior (MRFs - L_2 or L_1) (BNs - Dirichlet)
 - Structure complexity penalty
- Bayesian score (integrating parameters)
 - $\log P(G | D)$
 - $= \underbrace{\log P(D | G)}_{\text{marginal likelihood}} + \underbrace{\log P(G)}_{\text{graph prior}} + \text{Const}$

Optimization Algorithm

- Continuous
 - Closed form - BNs with multinomial
 - Gradient ascent ^{MRF}
missing data
 - EM - learning with missing data
- Discrete
 - Max spanning tree
 - Hill-climbing add, determine
- Discrete + continuous - computationally expensive

Hyperparameters

- Model hyperparameters
 - Equivalent sample size for parameter prior
 - Regularization strength for L1 or L2
 - Stopping criterion for EM
 - Strength of structure penalty
 - Set of features
 - # of values of latent variable
- Optimize on validation set
 - ~~training set~~
 - ~~train on training set~~
 - ~~evaluate on validation set~~
 - cross-validation

Model Evaluation Criteria

- Log-likelihood on test set
- Task-specific objective *segmentation accuracy
speech recognition WER*
- “Match” with prior knowledge

Troubleshooting: Underfitting

- Training & test performance both low
- Solutions
 - Decrease regularization
 - Reduce structure penalties
 - Add features via error analysis

Troubleshooting: Overfitting

- Training performance high, test performance low
- Solutions:
 - Increase regularization
 - Impose capacity constraints
 - Reduce feature set

Troubleshooting: Optimization

- Optimization may not be converging to good / global optimum
 - Can happen even if problem is convex
- Compare different learning rates, different random initializations

Troubleshooting: Objective Mismatch

Objective(M₁) >> Objective(M₂) \times

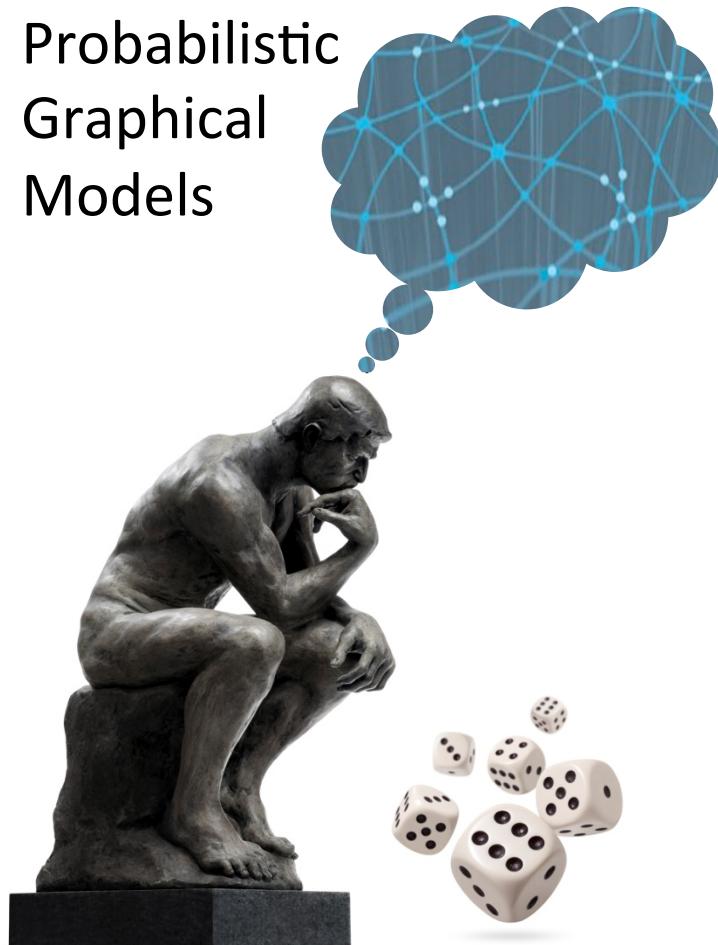
Performance(M₁) << Performance(M₂)

- Need to redesign objective to match desired performance criterion

Typical Learning Loop

- 
- Design model "template"
 - Select hyperparameters via CV on training set
 - Train on training set with chosen hyperparams
 - Evaluate performance on held-out set
 - Error analysis & model redesign
 - Report results on separate test set

Probabilistic
Graphical
Models



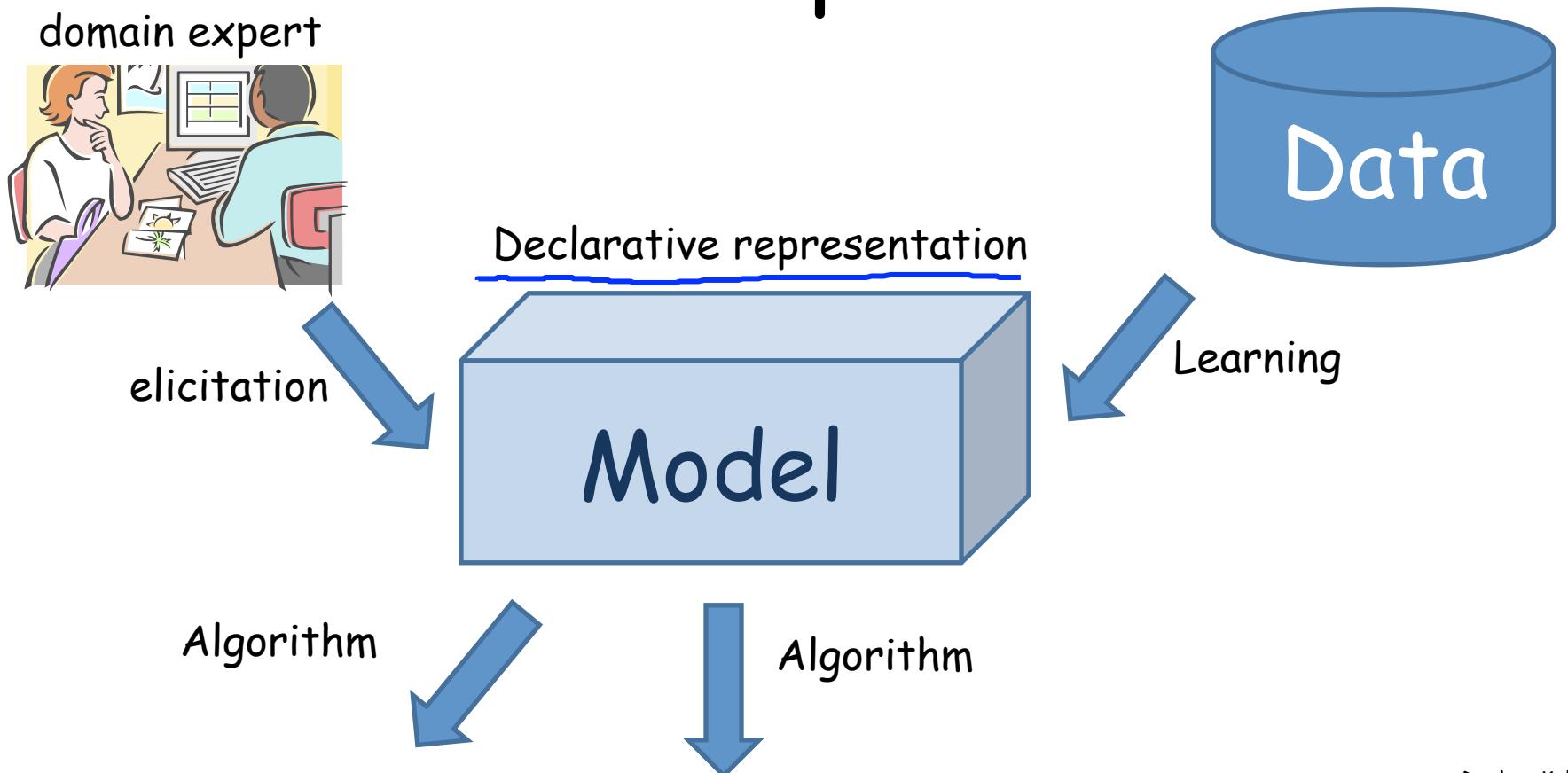
Summary

Probabilistic Graphical Models

Why PGMs?

- PGMs are the marriage of statistics and computer science
 - Statistics: Sound probabilistic foundations
 - Computer science: Data structures and algorithms for exploiting them

Declarative Representation



Daphne Koller

When PGMs?

- When we have noisy data and uncertainty
- When we have lots of prior knowledge
- When we wish to reason about multiple variables
- When we want to construct richly structured models from modular building blocks

Intertwined Design Choices

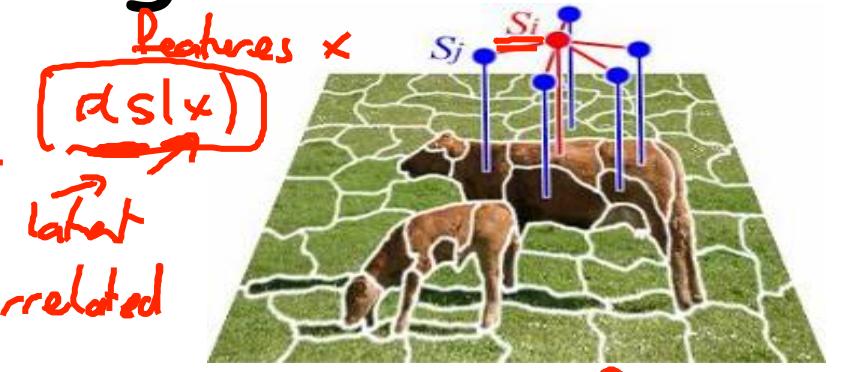
- Representation
 - affects cost of inference & learning
- Inference algorithm
 - Used as a subroutine in learning
 - Some are only usable in certain types of models
- Learning algorithm
 - Learnability imposes modeling constraints

Example: Image Segmentation

- BNs vs MRFs vs CRFs

- Naturalness of model
- Using rich features
- Inference costs
- Training cost
- Learn with missing data

✗ CRF →



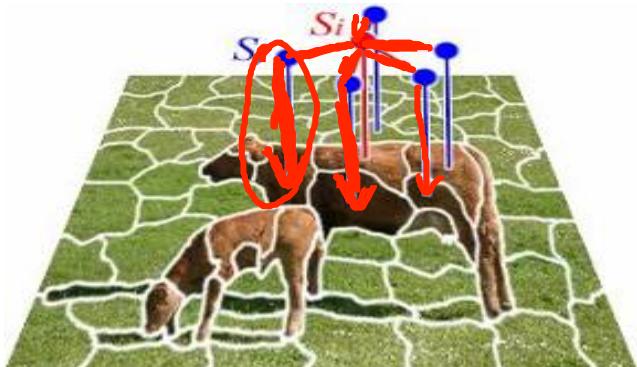
(associative, regular)

adjacent pixels have same labels
cows are on top of grass

learn to segment from unsupervised data

Mix & Match: Modeling

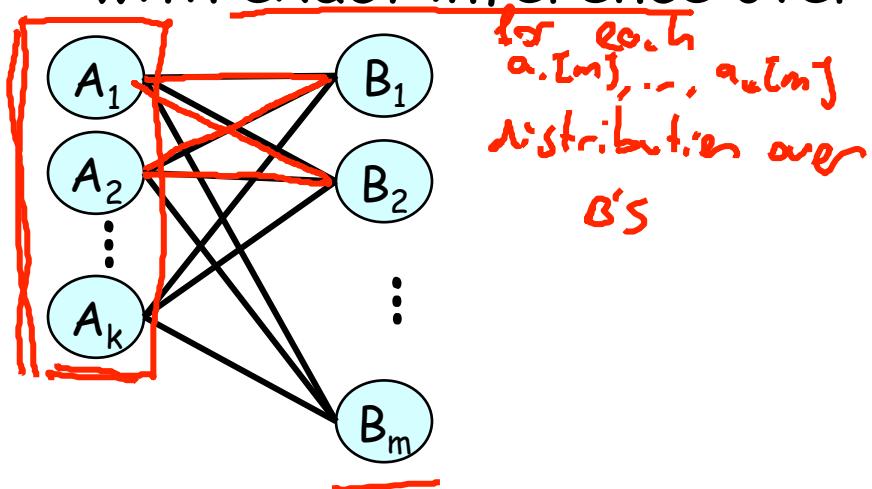
- Mix directed & undirected edges
- E.g., image segmentation from unlabeled images
 - Undirected edges over labels S - natural *(lack of)* directionality
 - Directed for $P(X_i | S_i)$ - easy learning (w/o inference)



explain image
using segment
characteristics

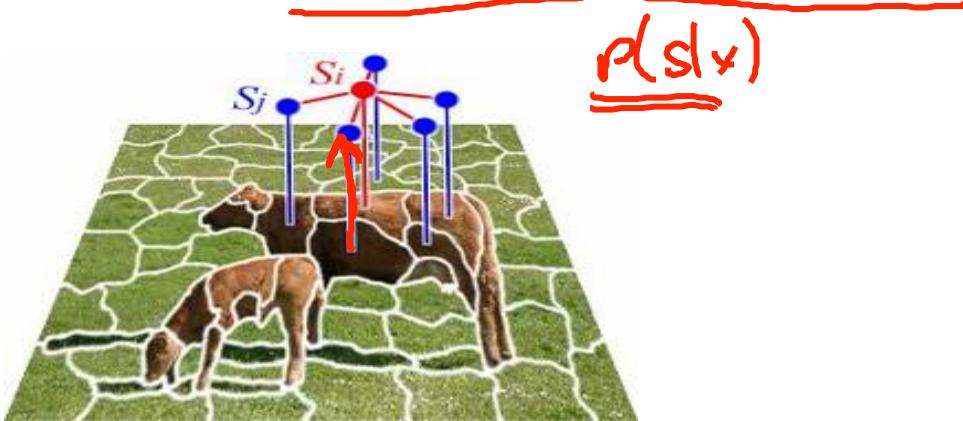
Mix & Match: Inference

- Apply different inference algorithms to different parts of model
- E.g., combine approximate inference (BP or MCMC) with exact inference over subsets of variables



Mix & Match: Learning

- Apply different learning algorithms to different parts of model
- E.g., combine high-accuracy, easily-trained model (e.g., SVM) for node potentials $P(S | X)$ with CRF learning for higher-order potentials



Daphne Koller

Summary

- Integrated framework for reasoning and learning in complex, uncertain domains
 - Large bag of tools within single framework
- Used in a huge range of applications
- Much work to be done, both on applications and on foundational methods

Lecture Slides

[Help Center](#)

Click on the lecture titles to download the annotated slides for each lecture, or click on the slides link next to each section label to download the combined slides for the whole section. For further reading, we have also provided relevant references to the [class textbook](#) next to each lecture.

Introduction and Overview ([combined slides](#))

[Welcome!](#)

[Overview and Motivation](#) Chapter 1.

[Distributions](#) Chapters 2.1.1 to 2.1.3.

[Factors](#). Chapter 4.2.1.

Bayesian Network Fundamentals ([combined slides](#))

[Semantics and Factorization](#) Chapters 3.2.1, 3.2.2. If you are unfamiliar with genetic inheritance, please watch this short [Khan Academy video](#) for some background.

[Reasoning Patterns](#). Chapter 3.2.1.

[Flow of Probabilistic Influence](#). Chapter 3.3.1.

[Conditional Independence](#). Chapters 2.1.4, 3.1.

[Independencies in Bayesian Networks](#). Chapter 3.2.2.

[Naive Bayes](#). Chapter 3.1.3.

[Application - Medical Diagnosis](#) Chapter 3.2: Box 3.D (p. 67).

Template Models ([combined slides](#))

[Overview](#). Chapter 6.1.

[Temporal Models - DBNs](#). Chapters 6.2, 6.3.

[Temporal Models - HMMs](#). Chapters 6.2, 6.3.

[Plate Models](#). Chapter 6.4.1.

Octave Tutorial

[Octave Tutorial Code](#)

Structured CPDs ([combined slides](#))

[Overview](#). Chapters 5.1, 5.2.

[Tree-Structured CPDs](#). Chapter 5.3.

[Independence of Causal Influence](#). Chapter 5.4.

[Continuous Variables](#). Chapter 5.5.

Markov Network Fundamentals ([combined slides](#))

[Pairwise Markov Networks](#). Chapter 4.1.

[General Gibbs Distribution](#). Chapter 4.2.2.

[Conditional Random Fields](#). Chapter 4.6.1.

[Independencies in Markov Networks](#). Chapter 4.3.1.

[I-Maps and Perfect Maps](#). Chapter 3.3.4.

[Log-Linear Models](#). Chapter 4.4, p. 125.

[Shared Features in Log-Linear Models](#). Chapter 4: Box 4.B (p. 112), Box 4.C (p. 126), Box 4.D (p. 127).

Representation Wrapup: Knowledge Engineering ([combined slides](#))

[Knowledge Engineering](#).

Variable Elimination ([combined slides](#))

[Conditional Probability Queries](#). Chapter 9.3.

[MAP Queries](#). Chapter 13.2.1.

[Variable Elimination Algorithm](#). Chapter 9.2.

[Variable Elimination Complexity](#). Chapter 9.4 through 9.4.2.3.

[VE - Graph Based Perspective](#). Chapter 9.4.

[Finding Elimination Orderings](#). Chapter 9.4.3.

Belief Propagation ([combined slides](#))

[Belief Propagation](#). Chapter 11.3.2

[Properties of Cluster Graphs](#). Chapter 11.3.2

Belief Propagation, Part 2 ([combined slides](#))

[Properties of Belief Propagation](#). Chapter 11.3.3

[Clique Tree Algorithm - Correctness](#). Chapter 10.2.1

[Clique Tree Algorithm - Computation](#). Chapters 10.2.2, 10.3.3.1

[Clique Trees and Independence](#). Chapter 10.1.2

[Clique Trees and VE](#). Chapter 10.4.1

[BP in Practice](#). Box 11.C

[Loopy BP and Message Decoding](#). Box 11.A

MAP Estimation Part 1 ([combined slides](#))

[MAP Exact Inference](#). Chapter 13.2.1

[Finding a MAP Assignment](#). Chapter 13.2.2

MAP Estimation Part 2 ([combined slides](#))

[Tractable MAP Problems](#). Chapter 13.6.

[Dual Decomposition - Intuition](#). Dual Decomposition is not in the textbook, but for further information you may refer to the original paper: [MRF Energy Minimization and Beyond via Dual Decomposition](#) N. Komodakis, N. Paragios and G. Tziritas

[Dual Decomposition - Algorithm](#).

Sampling Methods ([combined slides](#))

[Simple Sampling](#). Chapter 12.1.

[Markov Chain Monte Carlo](#) . Chapter 12.3 up to 12.3.2.2.

[Using a Markov Chain](#). Chapter 12.3.5.

[Gibbs Sampling](#). Review of Chapter 12.3.2 as applied to Gibbs Sampling.

[Metropolis Hastings Algorithm](#). Chapter 12.3.4.2.

Inference In Temporal Models, Summary ([combined slides](#))

[Inference in Temporal Models](#)

[Inference - Summary](#)

Decision Making ([combined slides](#))

[Maximum Expected Utility](#) Chapter 22.1.1, 23.2.104, 23.4.1-2, 23.5.1

[Utility Functions](#) Chapter 22.2.1-3, 22.3.2, 22.4.2

[Value of Perfect Information](#) Chapter 23.7.1-2

Learning: Parameter Estimation, Part 1 ([combined slides](#))

[Overview](#). Chapter 16.1 and Intro to Chapter 17

[Maximum Likelihood Estimation](#). Chapter 17.1

[Maximum Likelihood Estimation for Bayesian Networks](#). Chapter 17.2 through 17.2.1

[Bayesian Estimation](#). Chapter 17.3.2

[Bayesian Prediction](#). Chapter 17.4

[Bayesian Estimation for Bayesian Networks](#)

Learning: Parameter Estimation, Part 2 ([combined slides](#))

[Maximum Likelihood Estimation for Log-Linear Models](#). Chapter 20.1 - 20.2

[Maximum Likelihood Estimation for Conditional Random Fields](#). Chapter 20.1 - 20.2

[MAP Estimation for Markov Random Fields and Conditional Random Fields](#). Chapter 20.1 - 20.2

Structure Learning ([combined slides](#))

[Structure Learning: Overview](#) Chapter 18.1

[Likelihood Scores](#) Chapter 18.3.1

[BIC and Asymptotic Consistency](#) Chapter 18.3.5

[Bayesian Score](#) Chapter 18.3.2-18.3.4, 18.3.6, 18.3.7

[Learning Tree Structured Networks](#) Chapter 18.4.1

[Learning General Graphs: Heuristic Search](#) Chapter 18.4.3.1-2

[Learning General Graphs: Heuristic Search and Decomposability](#) Chapter 18.4.3.3

Learning With Incomplete Data ([combined slides](#))

[Learning With Incomplete Data - Overview](#) Chapter 19.1.3 and 19.1.4

[Expectation Maximization - Intro](#) Chapter 19.2.2

[Analysis of EM Algorithm](#) Chapter 19.2.2

[EM in Practice](#) Box 19.B.

[Latent Variables](#)

Learning Summary

[Learning Summary](#)

Final Summary

[Final Summary](#)

Created Sun 8 Jan 2012 12:51 AM PST

Last Modified Thu 10 May 2012 5:45 PM PDT

Feedback — Bayesian Network Fundamentals

[Help Center](#)

You submitted this quiz on **Sat 20 Apr 2013 10:10 AM PDT**. You got a score of **10.00** out of **10.00**.

Please check our grading policy under "Course Logistics" before submitting the quiz. The quiz isn't timed - you can save your answers halfway and come back again later.

Question 1

Factor product. Let X, Y be binary variables, and let Z be a variable that takes on values 1, 2, or 3.

If $\phi_1(X, Y)$ and $\phi_2(Y, Z)$ are the factors shown below, compute the selected entries (marked by a '?') in the factor $\psi(X, Y, Z) = \phi_1(X, Y) \cdot \phi_2(Y, Z)$, giving your answer according to the ordering of assignments to variables as shown below.

Separate each of the 3 entries of the factor with spaces, e.g., an answer of

0.1 0.2 0.3

means that $\psi(1, 1, 2) = 0.1$, $\psi(1, 2, 1) = 0.2$, and $\psi(2, 1, 3) = 0.3$.

X	Y	$\phi_1(X, Y)$
1	1	0.7
1	2	0.1
2	1	0.4
2	2	0.1

Y	Z	$\phi_2(Y, Z)$
1	1	0.2
1	2	0.8
1	3	0.5
2	1	0.0
2	2	0.9
2	3	0.3

X	Y	Z	$\psi(X, Y, Z)$
1	1	1	
1	1	2	?
1	1	3	
1	2	1	?
1	2	2	
1	2	3	
2	1	1	
2	1	2	
2	1	3	?
2	2	1	
2	2	2	
2	2	3	

You entered:

0.56 0.0 0.20

Your Answer

Score

Explanation

0.56 ✓ 0.33

0.0 ✓ 0.33

0.20 ✓ 0.33

Total 1.00 / 1.00

Question 2

Factor reduction. Let X, Z be binary variables, and let Y be a variable that takes on values 1, 2, or 3.

Now say we observe $Y = 3$. If $\phi(X, Y, Z)$ is the factor shown below, compute the missing entries of the reduced factor $\psi(X, Z)$ given that $Y = 3$, giving your answer according to the ordering of assignments to variables as shown below.

As before, you may separate the 4 entries of the factor by spaces.

X	Y	Z	$\phi(X, Y, Z)$
1	1	1	14
1	1	2	60
1	2	1	40
1	2	2	27
1	3	1	42
1	3	2	85
2	1	1	4
2	1	2	59
2	2	1	54
2	2	2	3
2	3	1	96
2	3	2	30

You entered:

42 85 96 30

Your Answer

Score

Explanation

42 ✓ 0.25

85 ✓ 0.25

96 ✓ 0.25

30 ✓ 0.25

Total 1.00 / 1.00

Question 3

Properties of independent variables. Assume that A and B are independent random variables. Which of the following options are always true? You may select 1 or more options (or none of them, if you think none apply).

Your Answer

Score

Explanation



✓ 0.25

This is the standard definition of independence.

$$P(A, B) = P(A) \times P(B)$$

- $P(B|A) = P(B)$ ✓ 0.25 In intuitive terms, this means that the value of B is not dependent on the value of A. We can derive this from $P(A, B) = P(A) \times P(B)$ as follows:

$$\begin{aligned} P(A, B) &= P(A) \times P(B) && \text{(by definition of independence)} \\ &= P(B|A) \times P(A) && \text{(by chain rule of probabilities)} \end{aligned}$$

therefore $P(B|A) = P(B)$.

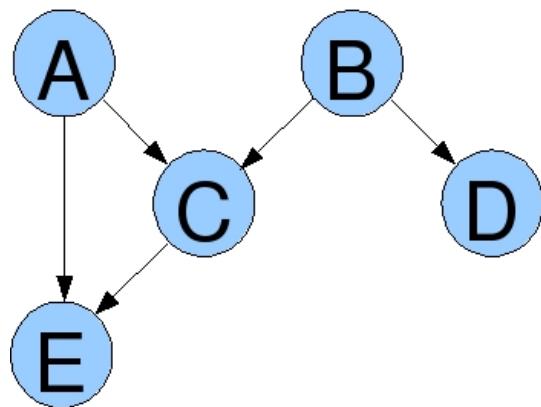
- $P(A) \neq P(B)$ ✓ 0.25

- $P(A, B) = P(A) + P(B)$ ✓ 0.25

Total 1.00 /
1.00

Question 4

Independencies in a graph. Which pairs of variables are independent in the graphical model below, given that none of them have been observed? You may select 1 or more options (or none of them, if you think none apply).

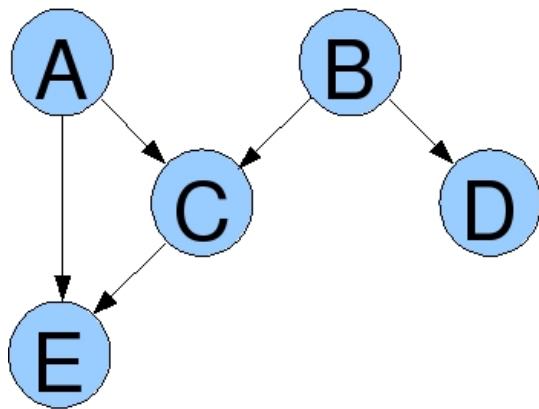


Your Answer	Score	Explanation
<input type="checkbox"/> C, D	✓ 0.20	There is an active trail connecting C and D that goes through B.
<input type="checkbox"/> B, E	✓ 0.20	There is an active trail connecting B and E that goes through C.
<input type="checkbox"/> None - there are no pairs of independent variables.	✓ 0.20	
<input type="checkbox"/> A, E	✓ 0.20	There is a directed edge from A to E.
<input checked="" type="checkbox"/> A, B	✓ 0.20	There are no active trails between A and B, so they are independent.
Total	1.00 / 1.00	

Question 5

***Independencies in a graph.** (An asterisk marks a question that is more challenging. Congratulations if you get it right!) Now

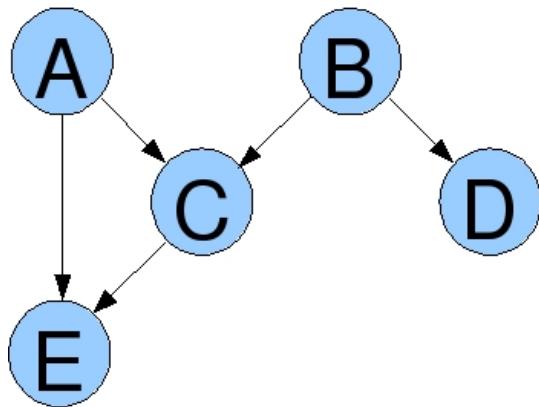
assume that the value of E is known. (E is observed. A, B, C, and D are not observed.) Which pairs of variables (not including E) are independent in the same graphical model, given E? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
<input type="checkbox"/> D, C	✓ 0.14	Influence can flow along the active trail $D \leftarrow B \rightarrow C$.
<input type="checkbox"/> B, D	✓ 0.14	There is a directed edge from B to D.
<input checked="" type="checkbox"/> None - given E, there are no pairs of variables that are independent.	✓ 0.14	Observing E activates the V-structures around C and E, giving rise to active trails between every pair of variables in the network.
<input type="checkbox"/> A, C	✓ 0.14	There is a directed edge from A to C.
<input type="checkbox"/> A, B	✓ 0.14	Observing E activates the V-structures around C and E. Hence, influence can flow from A to B through C; knowing B will now affect the probabilities of A taking on each of its values.
<input type="checkbox"/> A, D	✓ 0.14	Observing E activates the V-structures around C and E. Hence, influence can flow from A to B through C, and therefore from A to D through C and B.
<input type="checkbox"/> B, C	✓ 0.14	There is a directed edge from B to C.
Total	1.00 / 1.00	

Question 6

Factorization. Given the same model as above, which of these is an appropriate decomposition of the joint distribution $P(A, B, C, D)$?



Your Answer	Score	Explanation
-------------	-------	-------------



$P(A, B, C, D) = P(A)P(B)P(C|A)P(C|B)P(D|B)$



$P(A, B, C, D) = P(A)P(B)P(A, B|C)P(B|D)$



$P(A, B, C, D) = P(A)P(B)P(C)P(D)$



$P(A, B, C, D) = P(A)P(B)P(C|A, B)P(D|B)$



1.00 We can read off the appropriate factorization from the graph by examining the parents of each variable in the graph: A and B have no parents, while C is a child of A, B and D is a child of B . This gives us $P(A, B, C, D) = P(A)P(B)P(C|A, B)P(D|B)$.

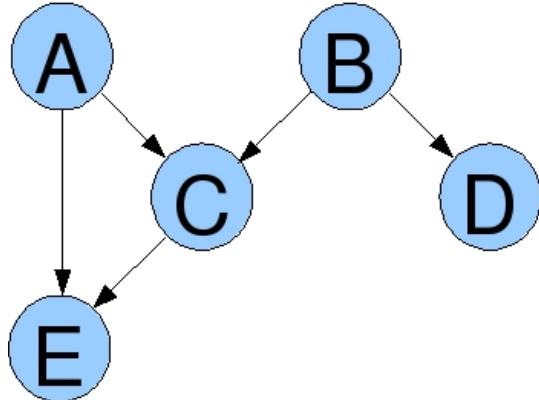
Total

1.00 /

1.00

Question 7

Independent parameters. How many independent parameters are required to uniquely define the CPD of C (the conditional probability distribution associated with the variable C) in the same graphical model as above, if A , B , and D are binary, and C and E have three values each?



If you haven't come across the term before, here's a brief explanation: A multinomial distribution over m possibilities x_1, \dots, x_m has m parameters, but $m - 1$ independent parameters, because we have the constraint that all parameters must sum to 1, so that if you specify $m - 1$ of the parameters, the final one is fixed. In a CPD $P(X|Y)$, if X has m values and Y has k values, then we have k distinct multinomial distributions, one for each value of Y , and we have $m - 1$ independent parameters in each of them, for a total of $k(m - 1)$. More generally, in a CPD $P(X|Y_1, \dots, Y_r)$, if each Y_i has k_i values, we have a total of $k_1 \times \dots \times k_r \times (m - 1)$ independent parameters.

Example: Let's say we have a graphical model that just had $X \rightarrow Y$, where both variables are binary. In this scenario, we need 1 parameter to define the CPD of X . The CPD of X contains two entries $P(X = 0)$ and $P(X = 1)$. Since the sum of these two entries has to be equal to 1, we only need one parameter to define the CPD.

Now we look at Y . The CPD for Y contains 4 entries which correspond to:

$P(Y = 0|X = 0), P(Y = 1|X = 0), P(Y = 0|X = 1), P(Y = 1|X = 1)$. Note that $P(Y = 0|X = 0)$ and $P(Y = 1|X = 0)$ should sum to one, so we need 1 independent parameter to describe those two entries; likewise, $P(Y = 0|X = 1)$ and $P(Y = 1|X = 1)$ should also sum to 1, so we need 1 independent parameter for those two entries.

Therefore, we need 1 independent parameter to define the CPD of X and 2 independent parameters to define the CPD of Y .

Your Answer	Score	Explanation
<input type="radio"/> 6		
<input type="radio"/> 4		
<input type="radio"/> 11		
<input type="radio"/> 7		
<input type="radio"/> 12		
<input type="radio"/> 3		
<input checked="" type="radio"/> 8	✓ 1.00	In a Bayesian network, the conditional probability distribution associated with a variable is the conditional probability distribution of that variable given its parents. There are 4 possibilities for the values of C's parents (A and B, which are binary). For each of these possibilities, there are 3 possible values for C, which corresponds to 2 free parameters (since the 3 numbers have to sum to 1). So there are $4 \times 2 = 8$ total free parameters.
Total	1.00 / 1.00	

Question 8

I-maps. I-maps can also be defined directly on graphs as follows. Let $I(G)$ be the set of independencies encoded by a graph G . Then G_1 is an I-map for G_2 if $I(G_1) \subseteq I(G_2)$.

Which of the following statements about I-maps are true? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> A graph K is an I-map for a graph G if and only if K encodes all of the independences that G has and more.	✓ 0.20	This is not true. K is an I-map for G if K does not make independence assumptions that are not true in G.
<input checked="" type="checkbox"/> A graph K is an I-map for a graph G if and only if all of the independencies encoded by K are also encoded by G.	✓ 0.20	K is an I-map for G if K does not make independence assumptions that are not true in G. An easy way to remember this is that the complete graph, which has no independencies, is an I-map of all distributions.
<input type="checkbox"/> I-maps are Apple's answer to Google Maps.	✓ 0.20	This is not true -- yet!
<input type="checkbox"/> The graph K that is the same as the graph G, except that all of the edges are oriented in the opposite direction as the corresponding edges in G, is always an I-map for G, regardless of the structure of G.	✓ 0.20	This is not always true; consider the V-structure $A \rightarrow B \leftarrow C$.
<input type="checkbox"/> An I-map is a function f that maps a graph G to itself, i.e., $f(G) = G$.	✓ 0.20	This is an identity function, not an I-map.

Total

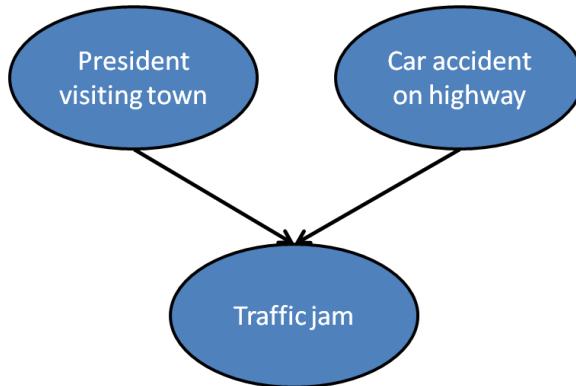
1.00 /
1.00

Question 9

***Inter-causal reasoning.** Consider the following model for traffic jams in a small town, which we assume can be caused by a car accident, or by a visit from the president (and the accompanying security motorcade).

$$P(\text{President} = 1) = 0.01$$

$$P(\text{Accident} = 1) = 0.1$$



$$P(\text{Traffic} = 1 \mid \text{President} = 0, \text{Accident} = 0) = 0.1$$

$$P(\text{Traffic} = 1 \mid \text{President} = 0, \text{Accident} = 1) = 0.5$$

$$P(\text{Traffic} = 1 \mid \text{President} = 1, \text{Accident} = 0) = 0.6$$

$$P(\text{Traffic} = 1 \mid \text{President} = 1, \text{Accident} = 1) = 0.9$$

Calculate $P(\text{Accident} = 1 \mid \text{Traffic} = 1)$ and $P(\text{Accident} = 1 \mid \text{Traffic} = 1, \text{President} = 1)$. Separate your answers with a space, e.g., an answer of

0.15 0.25

means that $P(\text{Accident} = 1 \mid \text{Traffic} = 1) = 0.15$ and $P(\text{Accident} = 1 \mid \text{Traffic} = 1, \text{President} = 1) = 0.25$. Round your answer to two decimal places.

You entered:

0.35 0.14

Your Answer

Score

Explanation

0.35



0.50

0.14



0.50

Total

1.00 / 1.00

Question Explanation

To calculate the required values, we can apply Bayes' rule. For instance,

$$\begin{aligned} P(A = 1 | T = 1, P = 1) &= \frac{P(A = 1, T = 1, P = 1)}{P(T = 1, P = 1)} \\ &= \frac{P(A = 1, T = 1, P = 1)}{P(A = 0, T = 1, P = 1) + P(A = 1, T = 1, P = 1)}. \end{aligned}$$

We can then use the chain rule of Bayesian networks to substitute the correct values in, e.g.,

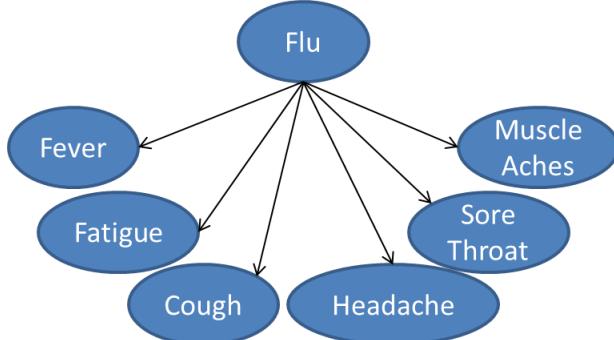
$$P(A = 1, T = 1, P = 1) = P(P = 1) \times P(A = 1) \times P(T = 1 | P = 1, A = 1)$$

This example of inter-causal reasoning meshes well with common sense: if we see a traffic jam, the probability that there was

a car accident is relatively high. However, if we also see that the president is visiting town, we can reason that the president's visit is the cause of the traffic jam; the probability that there was a car accident therefore drops correspondingly.

Question 10

***Naive Bayes.** Consider the following Naive Bayes model for flu diagnosis:



Assume a population size of 10,000. Which of the following statements are true in this model? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Say we observe that 1000 people have the flu, out of which 500 people have a headache (and possibly other symptoms) and 500 have a fever (and possibly other symptoms). We would expect that approximately 250 people with the flu also have both a headache and fever.	✓ 0.25	<p>Given that someone has the flu, whether he has a headache is independent of whether he has a fever. We can thus calculate:</p> $P(\text{Headache} = 1, \text{Fever} = 1 \text{Flu} = 1) = P(\text{Headache} = 1 \text{Flu} = 1) \times P(\text{Fever} = 1 \text{Flu} = 1)$ $\approx 0.5 * 0.5$ $= 0.25.$ <p>Since 1000 people have the flu, we can estimate that 250 of these people will have both a headache and fever.</p> <p>Note that this is only an estimate: we can assert with high confidence that $P(\text{Headache} = 1, \text{Fever} = 1 \text{Flu} = 1)$ is near to 0.25, but in general it will not be exactly 0.25. Moreover, even if it is exactly 0.25, the number of people with the flu, a headache and a fever need not be exactly 250 all the time. Think of this as analogous to flipping a fair coin: even though the probability of seeing a heads is exactly 0.5, in any given sequence of coin flips we need not see exactly half of the coins turning up heads.</p>
<input type="checkbox"/> Say we observe that 1000 people have a headache (and possibly other symptoms), out of which 500 people have the flu (and possibly other	✓ 0.25	<p>Even after observing the Headache variable, there is still an active trail from Flu to Fever. Thus, the probability of someone with a headache also having a flu is dependent on the probability of his having a fever as well. For example, if someone has a flu, he could be more likely to have a fever, irrespective of whether he has a headache or not.</p> <p>We therefore cannot estimate $P(\text{Flu} = 1, \text{Fever} = 1 \text{Headache} = 1)$ from the conditional marginal probabilities $P(\text{Flu} = 1 \text{Headache} = 1)$ and $P(\text{Fever} = 1 \text{Headache} = 1)$.</p>

symptoms), and 500 people have a fever (and possibly other symptoms). We would expect that approximately 250 people with a headache also have both the flu and a fever.

- | | |
|--|---|
| <input type="checkbox"/> Say we observe that 1000 people have the flu, out of which 500 people have a headache (and possibly other symptoms) and 500 have a fever (and possibly other symptoms). We can conclude that exactly 250 people with the flu also have both a headache and fever. | ✓ 0.25 Given that someone has the flu, whether he has a headache is independent of whether he has a fever. We can thus calculate: $\begin{aligned} P(\text{Headache} = 1, \text{Fever} = 1 \text{Flu} = 1) &= P(\text{Headache} = 1 \text{Flu} = 1) \times P(\text{Fever} = 1 \text{Flu} = 1) \\ &\approx 0.5 * 0.5 \\ &= 0.25. \end{aligned}$ Since 1000 people have the flu, we can estimate that 250 of these people will have both a headache and fever. <p>However, this is only an estimate: we can assert with high confidence that $P(\text{Headache} = 1, \text{Fever} = 1 \text{Flu} = 1)$ is near to 0.25, but in general it will not be exactly 0.25. Moreover, even if it is exactly 0.25, the number of people with the flu, a headache and a fever need not be exactly 250 all the time. Think of this as analogous to flipping a fair coin: even though the probability of seeing a heads is exactly 0.5, in any given sequence of coin flips we need not see exactly half of the coins turning up heads.</p> |
| <input type="checkbox"/> Say we observe that 500 people have a headache (and possibly other symptoms) and 500 people have a fever (and possibly other symptoms). We would expect that approximately 250 people have both a headache and fever. | ✓ 0.25 Without having observed the Flu variable, there is an active trail from Headache and Fever. Thus, the probability of someone having a headache (without observing flu status) is not independent of the probability of the same person having a fever. For example, if someone has a headache, he might be more likely to have the flu, which would correspondingly increase the probability that he has a fever as well. <p>We therefore cannot estimate $P(\text{Headache} = 1, \text{Fever} = 1)$ from the marginal probabilities $P(\text{Headache} = 1)$ and $P(\text{Fever} = 1)$.</p> |
- Total 1.00 / 1.00

Feedback — Template Models

[Help Center](#)

You submitted this quiz on **Sun 21 Apr 2013 12:13 AM PDT**. You got a score of **6.00** out of **6.00**.

Please check our grading policy under "Course Logistics" before submitting the quiz. The quiz isn't timed - you can save your answers halfway and come back again later.

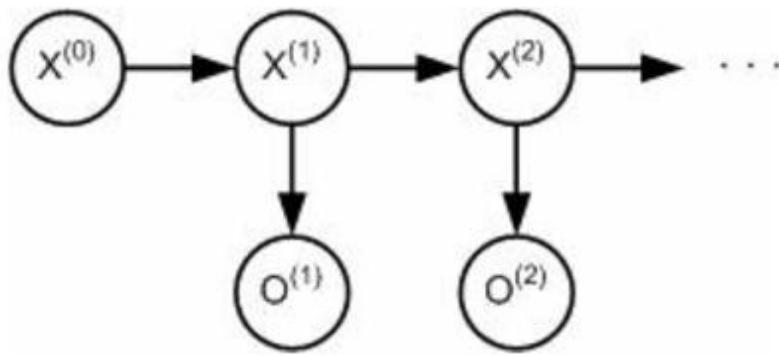
Question 1

Markov Assumption. If a dynamic system X satisfies the Markov assumption for all time $t \geq 0$, which of the following statements must be true? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> $(X^{(t+1)} \perp X^{(0:(t-1))})$	✓ 0.33	
<input type="checkbox"/> $P(X^{(t+1)}) \times P(X^{(0:(t-1))}) = P(X^{(t)})$ for all possible values of X	✓ 0.33	
<input checked="" type="checkbox"/> $(X^{(t+1)} \perp X^{(0:(t-1))} X^{(t)})$	✓ 0.33	
Total	1.00 / 1.00	

Question 2

Independencies in DBNs. In the following DBN, which of the following independence assumptions are true? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
<input checked="" type="checkbox"/> $(X^{(t-1)} \perp X^{(t+1)} X^{(t)})$	✓ 0.25	When $X^{(t)}$ is known, there is no active trail from $X^{(t-1)}$ to any other node in the network that is from a later time-point.
<input type="checkbox"/> $(O^{(t)} \perp O^{(t-1)})$	✓ 0.25	$(O^{(t)} \perp O^{(t-1)})$ is wrong because there is an active trail from $O^{(t)}$ to $O^{(t-1)}$ through $X^{(t)}$ and $X^{(t-1)}$.
<input checked="" type="checkbox"/> $(O^{(t)} \perp X^{(t+1)} X^{(t)})$	✓ 0.25	When $X^{(t)}$ is known, there is no active trail from $O^{(t)}$ to any other node in the network.
<input type="checkbox"/> $(X^{(t)} \perp X^{(t-1)})$	✓ 0.25	There is a directed edge from $X^{(t-1)}$ to $X^{(t)}$, so these variables will never be independent.
Total	1.00 / 1.00	

Question 3

Applications of DBNs. For which of the following applications might one use a DBN (i.e. the Markov assumption is satisfied)? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> Modeling time-series data, where the events at each time-point are influenced by the events at many other time-points.	✓ 0.25	This violates the Markov assumption because knowing the events at the time point right before a given time-point is not sufficient to understand the events at the given time-point.
<input checked="" type="checkbox"/> Modeling the behavior of people, where a person's behavior is	✓ 0.25	Consider each generation to be a time-slice, and this data satisfies the Markov assumption.

influenced by only the behavior of people in the same generation and the people in his/her parents' generation.

- | | |
|---|---|
| <input checked="" type="checkbox"/> Modeling data taken at different locations along a road, where the data at each location is influenced by only the data at the same location and at the location directly to the East | <input checked="" type="checkbox"/> 0.25 Consider each location to be a time slice, and order the locations from East to West. Viewed in this way, this data satisfies the Markov assumption. |
| <input type="checkbox"/> Predicting the probability that today will be a snow day (school will be closed because of the snow), when this probability depends only on whether yesterday, the day before yesterday, and 2 Mondays ago were snow days. | <input checked="" type="checkbox"/> 0.25 The probability that today will be a snow day depends on whether multiple previous days were snow days, so the Markov assumption is violated. |

Total	1.00 / 1.00
-------	----------------

Question 4

Plate Semantics. "Let A and B be random variables inside a common plate indexed by i. Which of the following statements must be true? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> There is an instance of A and an instance of B for every i.	<input checked="" type="checkbox"/> 0.25	
<input type="checkbox"/> If there is an instance of A for some i, then there is no instance of B for that i.	<input checked="" type="checkbox"/> 0.25	
<input type="checkbox"/> For each i, A(i) and B(i) are independent.	<input checked="" type="checkbox"/> 0.25	

For each i , $A(i)$ and $B(i)$ are not independent.

✓ 0.25

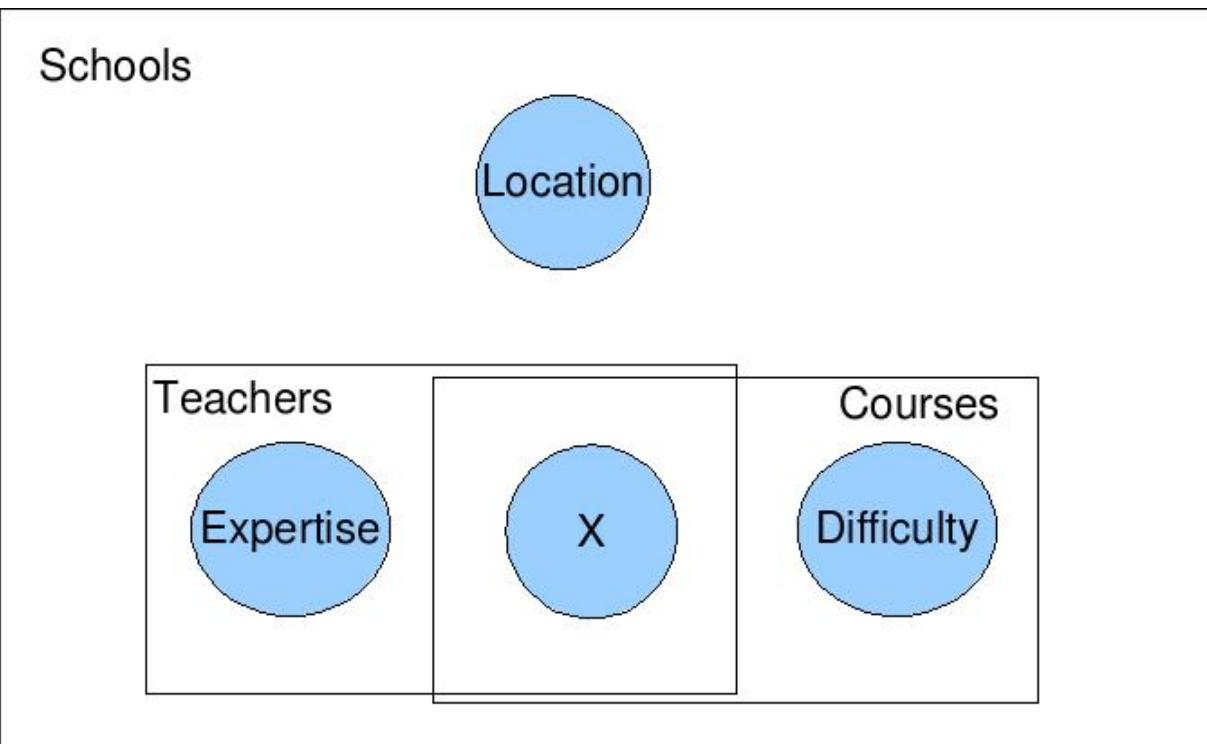
Total

1.00 /

1.00

Question 5

***Plate Interpretation.** Consider the plate model below (with edges removed). Which of the following might a given instance of X possibly represent in the grounded model? (You may select 1 or more options or none of them, if you think none apply. Keep in mind that this question addresses the variable's semantics, not its CPD.)



Your Answer

Score

Explanation

- Whether someone with expertise E taught something of difficulty D at school S ✓ 0.20 In the grounded model, there will be an instance of X for each combination of Teacher and Class, and there is a combination like this for each School. Thus, we are looking at a random variable that will say something about a specific teacher, class, and school combination, not a particular expertise, difficulty, and school combination.

- Whether a specific teacher T taught a ✓ 0.20 In the grounded model, there will be an instance of X for each combination of Teacher, Course, and School. Thus, we are looking at a random variable that will say something about a specific teacher, class, and school combination. The correct

specific course
C at school S

answer is the only one that does this.

- Whether a teacher with expertise E taught a course of difficulty D
- 0.20 In the grounded model, there will be an instance of X for each combination of Teacher and Class, and there is a combination like this for each School. Thus, we are looking at a random variable that will say something about a specific teacher and class and will also incorporate the school.

- Whether someone with expertise E taught something of difficulty D at a place in location L
- 0.20 In the grounded model, there will be an instance of X for each combination of Teacher and Class, and there is a combination like this for each School. Thus, we are looking at a random variable that will say something about a specific teacher, class, and school combination, not a particular expertise, difficulty, and location combination.

- None of these options can represent X in the grounded model
- 0.20 At least one option could represent X.

Total 1.00 /
1.00

Question 6

Grounded Plates. Using the same plate model, now assume that there are s schools, t teachers in each school, and c courses taught by each teacher. How many instances of the Expertise variable are there?

Your Answer	Score	Explanation
<input type="radio"/> stc		
<input checked="" type="radio"/> st	1.00	There is a variable for every combination of school and teacher.
<input type="radio"/> ct		
<input type="radio"/> s		

Total

1.00 /
1.00

Feedback — Structured CPDs + Week 1 Review

[Help Center](#)

You submitted this quiz on **Sun 28 Apr 2013 1:01 AM PDT**. You got a score of **9.00** out of **9.00**.

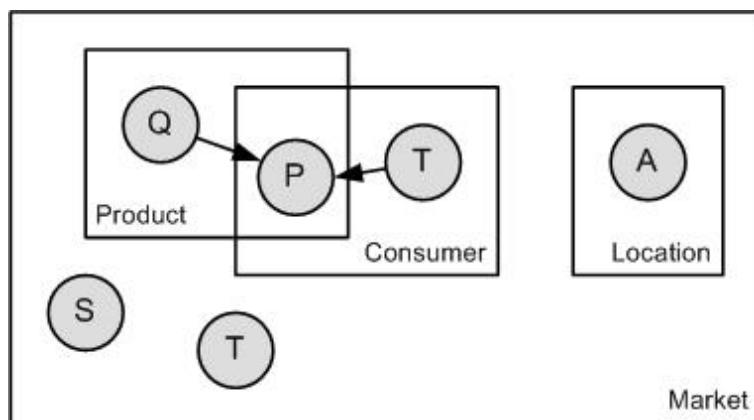
Question 1

I-maps. Suppose $(A \perp B) \in \mathcal{I}(P)$, and P is an I-map of G . Is it necessarily true that $(A \perp B) \in \mathcal{I}(G)$?

Your Answer	Score	Explanation
<input type="radio"/> No		
<input checked="" type="radio"/> Yes	✓ 1.00	Since P is an I-map of G , all independencies in P are also in G .
Total	1.00 / 1.00	

Question 2

Template Models. Consider the plate model shown below. Assume we are given K Markets, L Products, M Consumers and N Locations. What is the total number of instances of the variable P in the grounded BN?



Your Answer	Score	Explanation
<input type="radio"/> $K + L + M$		

$K \cdot L \cdot M \cdot N$

- $K \cdot L \cdot M$ ✓ 1.00 There will be one grounded instance of P for each combination of Market, Consumer, and Product. There will be $K \cdot L \cdot M$ of these combinations.

$K \cdot (N + (L \cdot M))$

Total 1.00 /
1.00

Question 3

Template Models. Consider the plate model from the previous question. What might P represent?

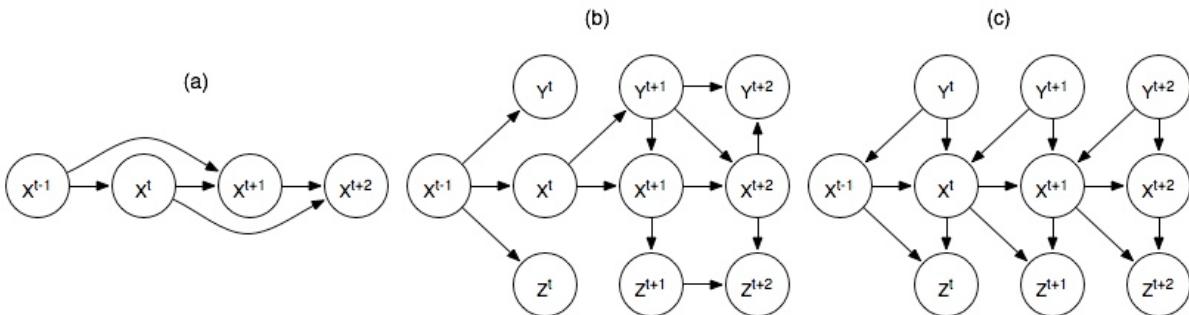
Your Answer	Score	Explanation
<input checked="" type="radio"/> Whether a specific product PROD was consumed by consumer C in market M	✓ 1.00	In the grounded model, there will be an instance of P for each combination of Product and Consumer, and there is a combination like this for each Market. Thus, we are looking at a random variable that will say something about a specific product, market, and consumer combination. The correct answer is the only one that does this.
<input type="radio"/> Whether a specific product PROD was consumed by consumer C in market M in location L		
<input type="radio"/> Whether a specific product of brand q was consumed by a consumer with age t in a market of type m that is in location a		
<input type="radio"/> Whether a		

specific product PROD was consumed by consumer C in market M that is supervised by supervisor S (assuming that there is exactly 1 unique supervisor per market) and has target audience T (assuming that there is exactly 1 unique target audience per market)

Total	1.00 /
	1.00

Question 4

Time-Series Graphs. Which of the time-series graphs satisfies the Markov assumption? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
--------------------	--------------	--------------------

- (a) 0.33 In (a), this fails because of the direct edges from nodes to nodes that are two time points away.

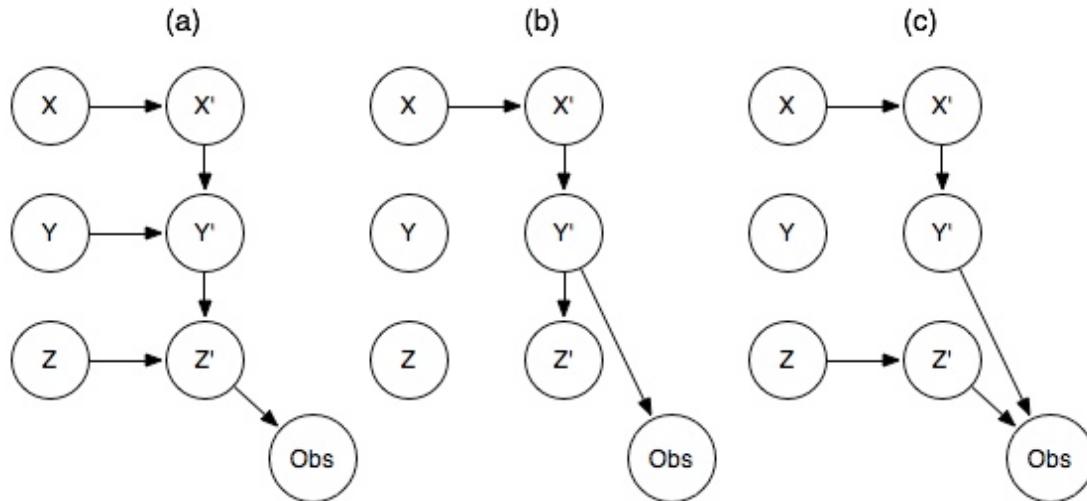
- (c) 0.33 In (c), it fails because of the backwards edges, which cause time-slices to depend on both the previous and the following time-slice.
- (b) 0.33 (b) is a time-series graph in which all variables in each time slice are independent of all variables in time slices at least 2 time slices before, given all variables in the previous time slice ($X^{(t+1)}, Y^{(t+1)}, Z^{(t+1)} \perp X^{(t-1)}, Y^{(t-1)}, Z^{(t-1)} | X^{(t)}, Y^{(t)}, Z^{(t)}$).

Total 1.00 /
1.00

Question 5

***Unrolling DBNs.** Below are 2-TBNs that could be unrolled into DBNs. Consider these unrolled DBNs (note that there are no edges within the first time-point). In which of them will $(X^{(t)} \perp Z^{(t)} | Y^{(t)})$ hold for all t , assuming $Obs^{(t)}$ is observed for all t and $X^{(t)}$ and $Z^{(t)}$ are never observed? You may select 1 or more options (or none of them, if you think none apply).

Hint: Unroll these 2-TBNs into DBNs that are at least 3 time steps long (i.e., involving variables from $t - 1, t, t + 1$).



Your Answer	Score	Explanation
-------------	-------	-------------

- (c) 0.33 (c) is incorrect because of active path $X^{(t)} \rightarrow X^{(t+1)} \rightarrow Y^{(t+1)} \rightarrow Obs^{(t+1)} \leftarrow Z^{(t+1)} \leftarrow Z^{(t)}$.
- (b) 0.33 The independence assumption holds in this network because knowing $Y^{(t)}$ blocks what was the only active trail from $X^{(t)}$ to $Z^{(t)}$.

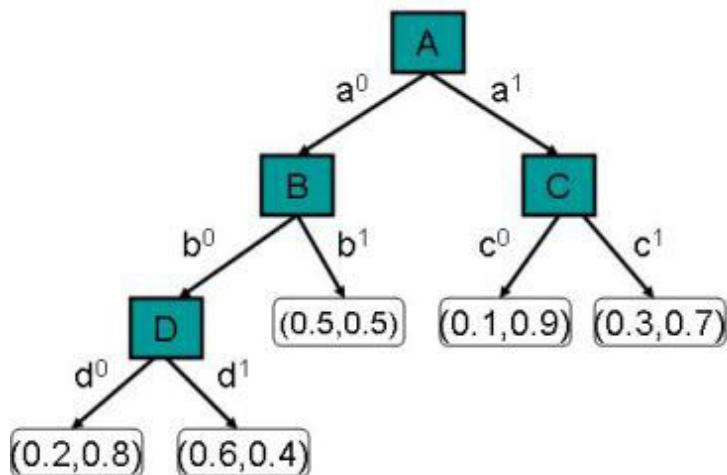
- (a) 0.33 (a) is incorrect because there is still an active path from $X^{(t)}$ to $Z^{(t)}$ through the previous time step variables ($X^{(t)} \leftarrow X^{(t-1)} \rightarrow Y^{(t-1)} \rightarrow Z^{(t-1)} \rightarrow Z$).

Total 1.00 /
1.00

Question 6

Causal Influence. Consider the CPD below. What is the probability that $E = e_0$ in the following graph, given an observation $A = a_1, B = b_0, C = c_0, D = d_1$? Note that, for the pairs of probabilities that make up the leaves, the probability on the left is the probability of e_0 , and the probability on the right is the probability of e_1 .

Tree CPD for $P(E | A, B, C, D)$



You entered:

0.1

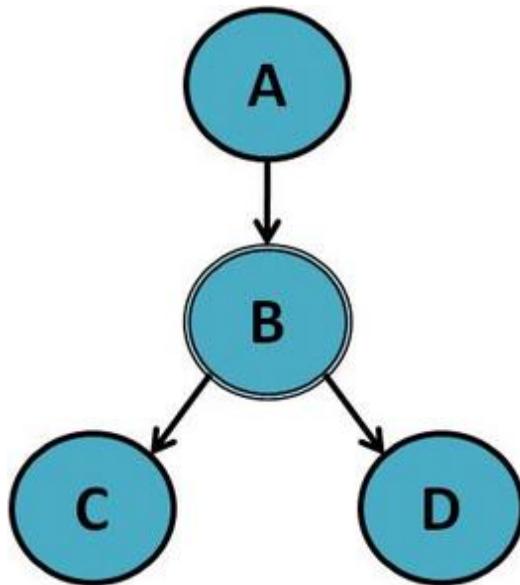
Your Answer	Score	Explanation
-------------	-------	-------------

- 0.1 1.00 This is the probability that is reached when following the tree down the appropriate branches.

Total 1.00 /
1.00

Question 7

Independencies with Deterministic Functions. In the following Bayesian network, the node B is a deterministic function of its parent A. Which of the following is an independence statement that holds in the network? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
<input type="checkbox"/> $(B \perp D C)$	✓ 0.25	B is a deterministic function of A, not C, and D is a child of B, so observing C does not make B and D independent.
<input checked="" type="checkbox"/> $(C \perp D A)$	✓ 0.25	Since B is a deterministic function of A, observing A implies that B is also observed, which d-separates C and D. Therefore, $(C \perp D A)$.
<input checked="" type="checkbox"/> $(C \perp D B)$	✓ 0.25	Since B is given and is the only parent of C and of D, C and D are independent.
<input type="checkbox"/> $(A \perp D C)$	✓ 0.25	Since A is an ancestor of both C and D, observing D does not make A and C independent.
Total	1.00 / 1.00	

Question 8

Independencies in Bayesian Networks. For the network in the previous question, let B no

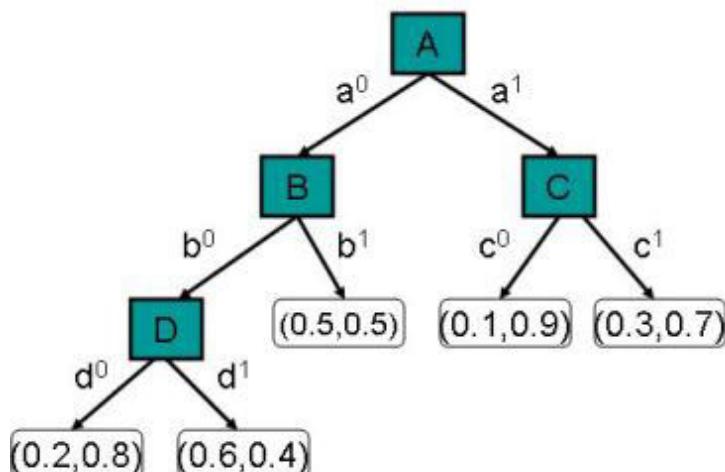
longer be a deterministic function of its parent A. Which of the following is an independence statement that holds in the modified Bayesian network? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> $(A \perp D B)$	✓ 0.25	The only active trail from A to D passes through B , and there are no V-structures between A and D , so observing B makes A and D independent.
<input type="checkbox"/> $(A \perp D C)$	✓ 0.25	Since C is not on the active trail from A to D , observing C does not make A and D independent.
<input type="checkbox"/> $(B \perp D A)$	✓ 0.25	Since B is the parent of D , B and D do not become independent when A is observed.
<input type="checkbox"/> $(B \perp D C)$	✓ 0.25	Since D is a child of B , they are not independent, even if C is observed.
Total	1.00 / 1.00	

Question 9

Context-Specific Independencies in Bayesian Networks. Which of the following are context-specific independencies that **do** exist in the tree CPD below? (Note: Only consider independencies in this CPD, ignoring other possible paths in the network that are not shown here. You may select 1 or more options (or none of them, if you think none apply)).

Tree CPD for $P(E | A, B, C, D)$



Your Answer	Score	Explanation
-------------	-------	-------------

$(E \perp_c D | a^0)$ ✓ 0.25 A variable X is independent of E given conditioning assignments \bar{z} if all paths consistent with \bar{z} traversed in the tree CPD reach a leaf without querying X . This is not true for this option because, depending on the value of B , D might be queried.

$(E \perp_c C | b^0, d^0)$ ✓ 0.25 A variable X is independent of E given conditioning assignments \bar{z} if all paths consistent with \bar{z} traversed in the tree CPD reach a leaf without querying X . This is not true for this option because C is on a separate branch from B and D , and the initial branch is not even known since it depends on A .

$(E \perp_c C | a^0, b^0)$ ✓ 0.25 A variable X is independent of E given conditioning assignments \bar{z} if all paths consistent with \bar{z} traversed in the tree CPD reach a leaf without querying X . This is true for this option.

$(E \perp_c D | b^1)$ ✓ 0.25 A variable X is independent of E given conditioning assignments \bar{z} if all paths consistent with \bar{z} traversed in the tree CPD reach a leaf without querying X . This is true for this option.

Total 1.00 /
1.00

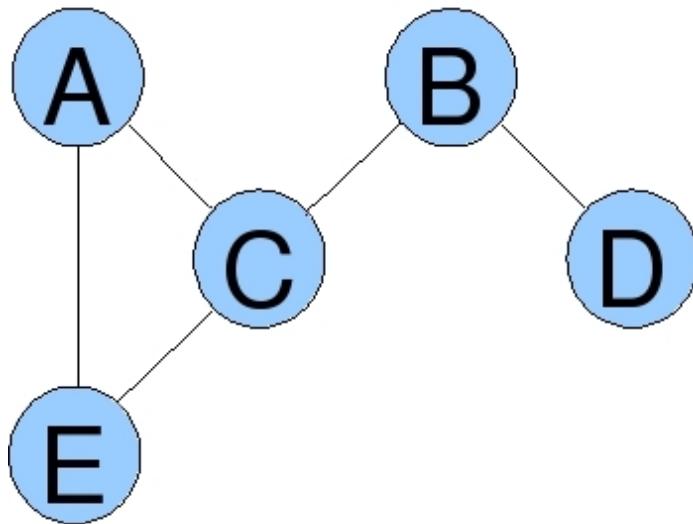
Feedback — Markov Network Fundamentals

[Help Center](#)

You submitted this quiz on **Fri 3 May 2013 12:27 PM PDT**. You got a score of **5.75** out of **7.00**. You can [attempt again](#), if you'd like.

Question 1

Independence in Markov Networks. Consider this graphical model from week 1's quizzes. This time, all of the edges are undirected (see modified graph below). Which pairs of variables are independent in this network? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
<input type="checkbox"/> D, E	✓ 0.33	There is a path connecting D and E that goes through B and C.
<input type="checkbox"/> C, D	✓ 0.33	There is a path connecting C and D that goes through B.
<input type="checkbox"/> B, E	✓ 0.33	There is a path connecting B and E that goes through C.
Total	1.00 / 1.00	

Question Explanation

There is a path from every node to every other node, so none of the nodes are independent.

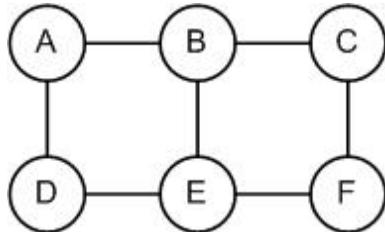
Question 2

Factor Scope. Let $\phi(a, b)$ be a factor in a graphical model, where a is a value of A and b is a value of B. What is the scope of ϕ ?

Your Answer	Score	Explanation
<input type="radio"/> {A, B, C}		
<input checked="" type="radio"/> {A, B}	✓ 1.00	
<input type="radio"/> {A}		
<input type="radio"/> {A, B, C, D, E}		
Total	1.00 / 1.00	

Question 3

Factorization. Which of the following sets of factors could factorize over the undirected graph below? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
<input checked="" type="checkbox"/> $\phi(A), \phi(B), \phi(C), \phi(D), \phi(E), \phi(F)$	✓ 0.25	The scope of each factor in the set must be a clique in the graph, which is the case here.
<input type="checkbox"/> $\phi(A, B, C), \phi(D, E, F)$	✓ 0.25	The scope of each factor in the set must be a clique in the graph, which is not the case here.
<input type="checkbox"/> $\phi(A, B, C, D), \phi(C, D, E, F)$	✓ 0.25	The scope of each factor in the set must be a clique in the graph, so factors like $\phi(A, B, C, D)$ can't be involved in a factorization over this graph.

$\phi(A, B), \phi(C, D), \phi(E, F)$

✖ 0.00

The scope of each factor in the set must be a clique in the graph, which is not the case here.

Total

0.75 /
1.00

Question 4

Factors in Markov Network. Let $\pi_1[A, B]$, $\pi_2[B, C]$, and $\pi_3[A, C]$ be all of the factors in a particular undirected graphical model. Then what is $\sum_{A, B, C} \pi_1[A, B] \times \pi_2[B, C] \times \pi_3[A, C]$? More than one answer could be correct.

Your Answer

Score

Explanation

Always greater than or equal to $\pi_1[a, b] \times \pi_2[b, c] \times \pi_3[a, c]$, where a is a value of A , b is a value of B , and c is a value of C

✓ 0.17

This is the sum over the factor products for all possible values of the variables in the factors, so it is greater than or equal to the factor product for only one combination of values.

Always equal to 1

✓ 0.17

There is no restriction that this sum over possible factor products be 1 (it is not normalized).

Always greater than or equal to 0

✓ 0.17

The factors can be any positive function.

Always less than or equal to 1

✓ 0.17

There is no restriction that the factors be less than one (it is not normalized).

Always greater than or equal to 1

✓ 0.17

There is no restriction that this sum over possible factor products be greater than 1.

Always equal to the partition function, Z

✓ 0.17

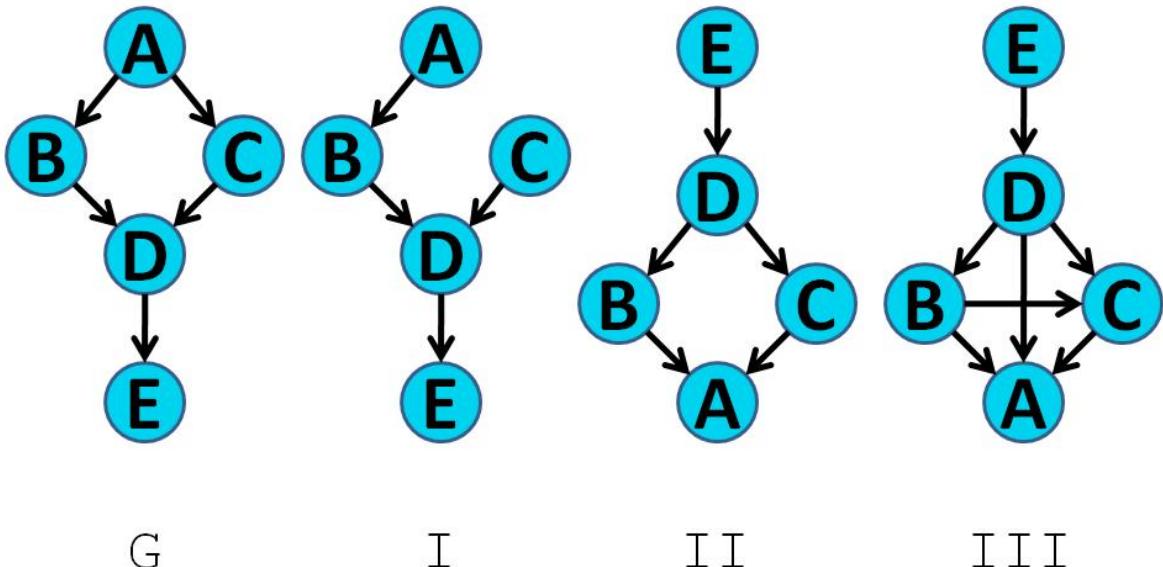
This is the formula for Z , the partition function.

Total

1.00 /
1.00

Question 5

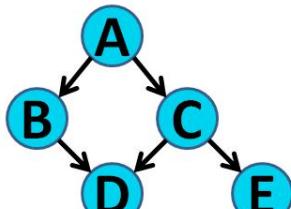
I-Maps. Graph G is a perfect I-map for distribution P , i.e. $\mathcal{I}(G) = \mathcal{I}(P)$. Which of the other graphs is a **perfect** I-map for P ?



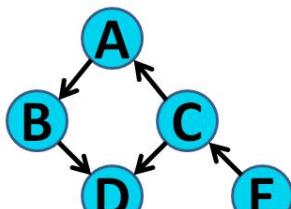
Your Answer	Score	Explanation
<input type="radio"/> I and III		
<input type="radio"/> None of the above		
<input checked="" type="radio"/> III	✗ 0.00	III does not preserve an independence relationship in G.
<input type="radio"/> II		
Total	0.00 / 1.00	

Question 6

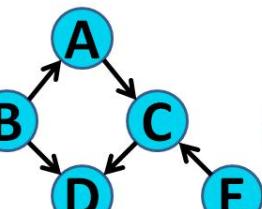
I-Equivalence. In the figure below, graph G is I-equivalent to which other graph(s)?



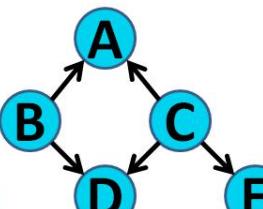
G



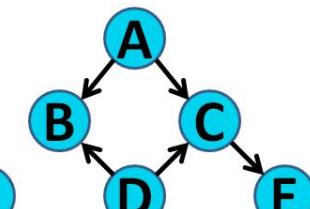
I



II



III



IV

Your Answer**Score****Explanation** I

1.00

II, III, and IV all have extra independencies.

 IV None of the above II

Total

1.00 / 1.00

Question 7

***I-Equivalence.** Let T be any directed tree (not a polytree) over n nodes, where $n \geq 1$. A directed tree is a traditional tree, where each node has at most one parent and there is only one root, i.e., all but one node has exactly one parent. (In a polytree, nodes may have multiple parents.) How many networks (including itself) are I-equivalent to T ?

Your Answer**Score****Explanation** $n!$ Depends on the specific structure of T .

n

- ✓ 1.00 The only graphs that are I-equivalent to T are directed trees with the same edges and no V-structures. Thus, making a different node the root would make an I-equivalent tree. Any of the tree nodes can be set as the root.

 2

Total 1.00 /
1.00

Feedback — Week 2 Review

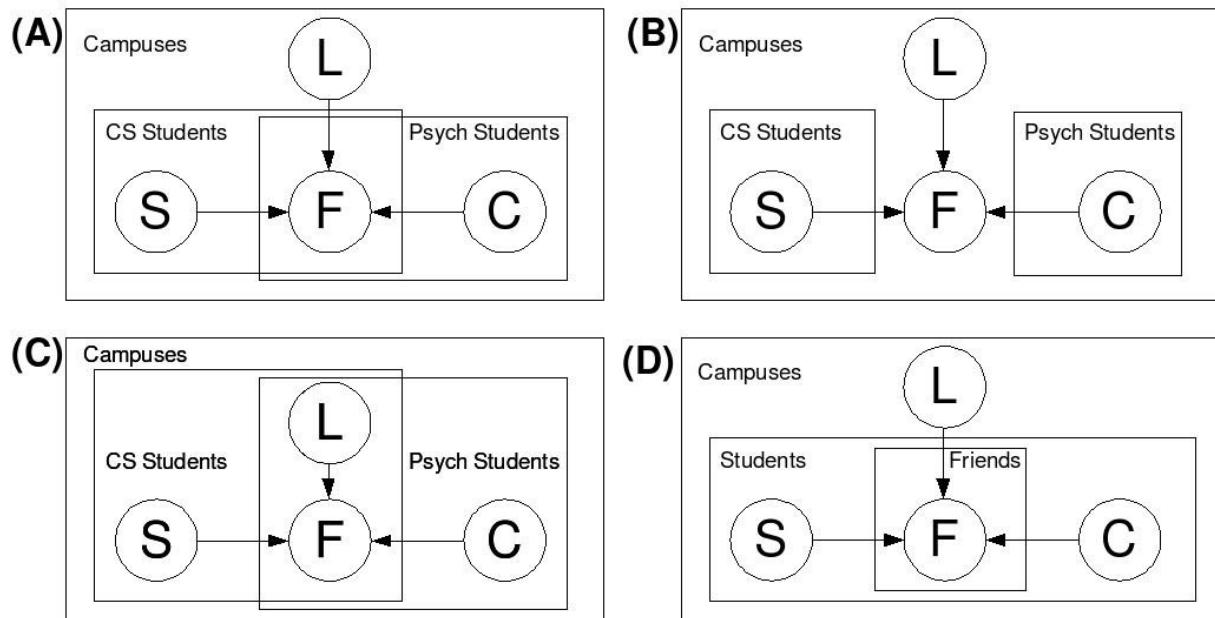
[Help Center](#)

You submitted this quiz on **Fri 3 May 2013 11:19 AM PDT**. You got a score of **3.00** out of **3.00**.

Question 1

Template Model Representation. Consider the following scenario:

On each campus there are several Computer Science students and several Psychology students (each student belongs to one xor the other group). We have a binary variable L for whether the campus is large, a binary variable S for whether the CS student is shy, a binary variable C for whether the Psychology student likes computers, and a binary variable F for whether the Computer Science student is friends with the Psychology student. Which of the following plate models can represent this scenario?



Your
Answer

Score

Explanation

(D)

None
of these
plate
models
can
represent

this
scenario

(B)

- (A) ✓ 1.00 (A) is right because there are separate plates for CS and Psych students, who do not overlap; the student plates are within the campuses plate since every student is on a campus; the location node is on the campus plate since location is only a property of campuses; the properties of types of students nodes are on those students' plates; and the friend node is on all of the plates since friendship involves both types of students and the campus.

Total 1.00 /
1.00

Question 2

***I-Equivalence.** Let Bayesian network G be a simple directed chain $X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_n$ for some number n . How many Bayesian networks are I-equivalent to G including G itself?

Your Score Explanation
Answer

$n!$

1

$2^{(n-1)}$

- n ✓ 1.00 The chain $X_1 \leftarrow \dots \leftarrow X_i \rightarrow \dots \rightarrow X_n$ is I-equivalent, where i can be 2 through n (when $i = n$, all arrows point left). Thus, there are $n - 1$ I-equivalent networks like this. Including the original network makes n .

Total 1.00 /
1.00

Question 3

Partition Function. Which of the following is a use of the partition function?

Your Answer**Score** **Explanation**

-
- The partition function describes the probability that it is possible to partition the graph into groups of connected variables, where each variable within a group has the same value.
-
- The partition function is used only in the context of Bayesian networks, not Markov networks.
-
- One can divide factor products by the partition function in order to convert them into probabilities. ✓ 1.00 This is a common use of the partition function.
-
- The partition function is the probability of each variable in the graph taking on a specific value.
-

Total

1.00 /
1.00

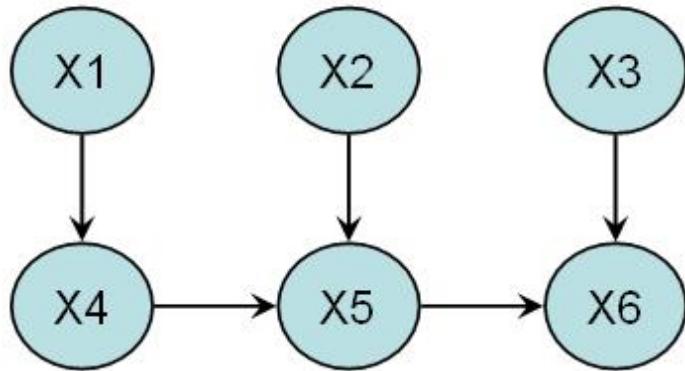
Feedback — Inference: Variable Elimination

[Help Center](#)

You submitted this quiz on **Sat 4 May 2013 12:48 PM PDT**. You got a score of **10.00** out of **10.00**.

Question 1

Intermediate Factors. Consider running variable elimination on the following Bayesian network over binary variables. Which of the nodes, if eliminated first, results in the largest intermediate factor? By largest factor we mean the factor with the largest number of entries.



Your
Answer

Score

Explanation

X_4

X_5 ✓ 1.00 Eliminating X_5 results in the intermediate factor $\tau(X_2, X_4, X_6)$, which is larger than for any of the other options.

X_6

X_2

Total 1.00 /
1.00

Question 2

Elimination Orderings. Which of the following characteristics of the variable elimination algorithm are affected by the choice of elimination ordering? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Runtime of the algorithm	✓ 0.25	The elimination ordering affects the size of the largest factor created, which determines the runtime of the algorithm.
<input checked="" type="checkbox"/> Size of the largest intermediate factor	✓ 0.25	The elimination ordering can affect the size of the largest intermediate factor.
<input checked="" type="checkbox"/> Memory usage of the algorithm	✓ 0.25	The elimination ordering affects the size of the largest factor created, which determines the memory usage of the algorithm.
<input type="checkbox"/> Which marginals can be computed correctly	✓ 0.25	The correctness of the algorithm is independent of the elimination ordering.
Total	1.00 / 1.00	

Question 3

Uses of Variable Elimination. Which of the following quantities can be computed using the sum-product variable elimination algorithm? (In the options, let X be a set of query variables, and E be a set of evidence variables in the respective networks.) You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> The most likely assignment to the variables in a Markov network.	✓ 0.25	We cannot do this with sum-product variable elimination, which is what was discussed. However, we can do this with max-product variable elimination, a modification to the algorithm which will be discussed later in the course.
<input checked="" type="checkbox"/> $P(X E = e)$ in a Markov network	✓ 0.25	This is a standard use of the variable elimination algorithm.
<input checked="" type="checkbox"/> $P(X)$ in a Bayesian network	✓ 0.25	This is a standard use of the variable elimination algorithm.

- $P(X | E = e)$ ✓ 0.25 This is a standard use of the variable elimination algorithm.
in a Bayesian
network

Total 1.00 /
1.00

Question 4

Marginalization. Suppose we run variable elimination on a Bayesian network where we eliminate all the variables in the network. What number will the algorithm produce?

You entered:

1

Your Answer	Score	Explanation
-------------	-------	-------------

1 ✓ 1.00 Bayesian networks represent valid probability distributions, and so summing up all the possible states will always return 1.

Total 1.00 /
1.00

Question 5

Marginalization. Suppose we run variable elimination on a Markov network where we eliminate all the variables in the network. What number will the algorithm produce?

Your Answer

Score Explanation

1

A positive number, not necessarily between 0 and 1, which depends on the structure of the network.

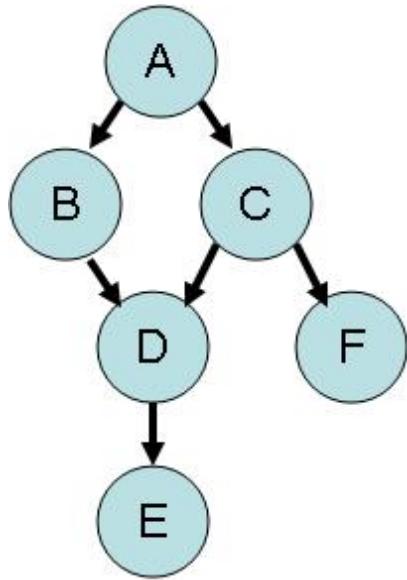
A positive number, always between 0 and 1, which depends on the structure of the network.

- Z, the partition function for the network.
- 1.00 Eliminating all the variables yields the partition function for the network.

Total 1.00 /
1.00

Question 6

Intermediate Factors. If we perform variable elimination on the graph shown below with the variable ordering B, A, C, F, E, D , what is the intermediate factor produced by the third step (just before summing out C)?



G

Your Answer	Score	Explanation
<input checked="" type="radio"/> $\psi(C, D, F)$	<input checked="" type="checkbox"/> 1.00	After eliminating B we have a new factor $\tau_1(A, D, C)$, and after eliminating A , the factor becomes $\tau_2(D, C)$, then when eliminating C , the intermediate factor is $\psi(C, D, F) = \tau_2(D, C)P(F C)$. This is because the only factors involving C at this point are $\tau_2(D, C)$ and $P(F C)$. The only other factor involving C , $P(C A)$ was already used to compute $\tau_2(D, C)$ when eliminating A , so including it again would be incorporating information from this factor twice.
<input type="radio"/> $\psi(C, D, E, F)$		
<input type="radio"/>		

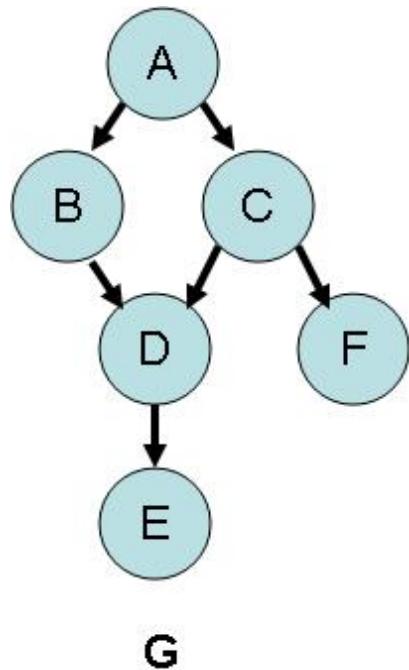
$\psi(A, B, C, D, F)$

$\psi(C, F)$

Total 1.00 /
1.00

Question 7

Induced Graphs. If we perform variable elimination on the graph shown below with the variable ordering B, A, C, F, E, D , what is the induced graph for the run?

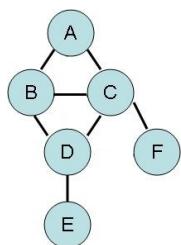
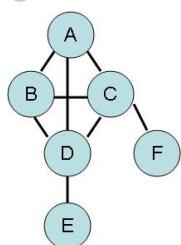


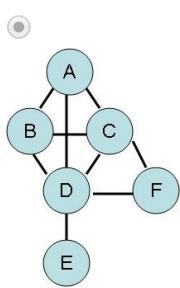
G

Your Answer

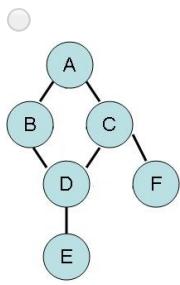
Score

Explanation





- ✓ 1.00 This is correct. There is an edge in the induced graph between every pair of variables that is present together in a factor during a run of variable elimination.



Total 1.00 /
1.00

Question 8

***Time Complexity of Variable Elimination.** Consider a Bayesian network taking the form of a chain of n variables, $X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_n$, where each of the X_i can take on k values. What is the computational cost of running variable elimination on this network if we eliminate the X_i in order (i.e., first X_1 , then X_2 and so on)?

Your Answer	Score	Explanation
-------------	-------	-------------

$O(kn^2)$

1.00 When eliminating X_1 , we sum out X_1 from $P(X_1)P(X_2|X_1)$ to obtain $\phi_1(X_2)$. For each value of X_2 , we have to do k multiplications and $k - 1$ summations, which is $O(k)$. Since X_2 can take k different values, to compute $\phi_1(X_2)$, the computational cost is $O(k^2)$ operations. The process continues for each X_i , so in total the cost is $O(nk^2)$.

$O(k^n)$

$O(nk^3)$

Total	1.00 /
	1.00

Question 9

***Time Complexity of Variable Elimination.** Now take the same chain as in the previous question, but eliminate the X_i starting from X_2 , going to X_3, \dots, X_n and then back to X_1 . What is the computational cost of running variable elimination with this ordering?

Your Answer	Score	Explanation
-------------	-------	-------------

$O(nk^3)$ 1.00 If we start by eliminating X_2 , we create an intermediate factor over X_1, X_2, X_3 , and continue from X_3 to the end and then X_1 . Since the scope of the intermediate factor involves three variables, the complexity would be $O(nk^3)$ instead of $O(nk^2)$ as in the previous question.

$O(nk^2)$

$O(nk)$

$O(kn^2)$

Total	1.00 /
	1.00

Question 10

Time Complexity of Variable Elimination. Suppose we eliminate all the variables in a Markov network using the variable elimination algorithm. Which of the following could affect the runtime of the algorithm? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
-------------	-------	-------------

Number of variables in the 0.33 The runtime is affected by the number of variables in the network, as discussed in the textbook.

network

The variable elimination ordering ✓ 0.33 This potentially affects the size of the largest factor in the network, which is a key component of the algorithm's runtime.

Number of values each variable can take ✓ 0.33 This affects the size of the largest factor in the network, which is a key component of the algorithm's runtime.

Total 1.00 /
1.00

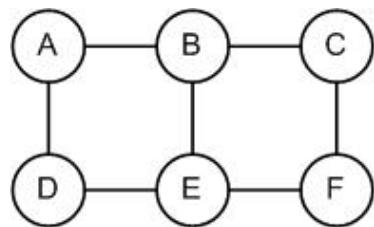
Feedback — Inference: Belief Propagation

[Help Center](#)

You submitted this quiz on **Tue 14 May 2013 9:50 PM PDT**. You got a score of **10.67** out of **11.00**. You can [attempt again](#), if you'd like.

Question 1

Cluster Graph Construction. Consider the pairwise MRF, H , shown below with potentials over $\{A,B\}$, $\{B,C\}$, $\{A,D\}$, $\{B,E\}$, $\{C,F\}$, $\{D,E\}$ and $\{E,F\}$.



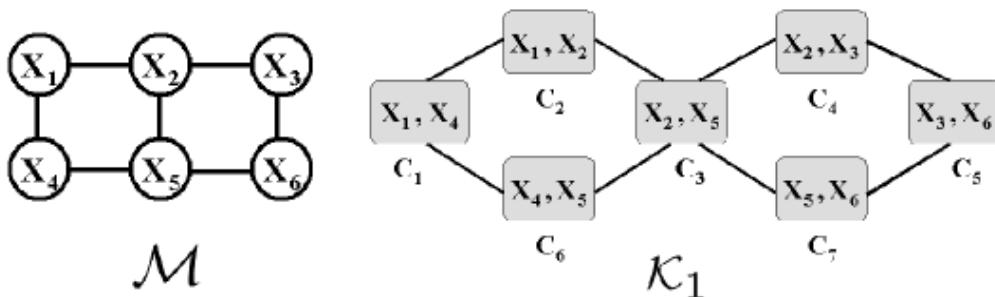
Which of the following is/are valid cluster graph(s) for H ? (A cluster graph is valid if it satisfies the running intersection property and family preservation. You may select 1 or more options, or none of them, if you think none apply.)

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> A,B,C,D,E,F	0.25	This is a valid cluster graph because a single cluster with all variables is always a valid cluster graph for any distribution. It however, does not admit efficient inference, since we would have to sum over (multiple) variables in order to extract any marginals from the cluster belief.
<input checked="" type="checkbox"/> A,B,D,E --- B,C,E,F	0.25	This graph is valid because it satisfies the running intersection property for a clique tree and family preservation.
<input type="checkbox"/> A,B,C --- D,E,F	0.25	This graph does not satisfy family preservation. For example, the potential over $\{C, F\}$ cannot be assigned to any cluster.
<input checked="" type="checkbox"/> A,B,D --- B,E --- B,C,F A,D,E --- B,E --- C,E,F	0.25	This graph is valid because it satisfies the running intersection property for cluster graphs and family preservation.
Total	1.00 / 1.00	

Question 2

Message Passing in a Cluster Graph.

Suppose we wish to perform inference over the Markov network M as shown below. Each of the variables X_i are binary, and the only potentials in the network are the pairwise potentials $\phi_{i,j}(X_i, X_j)$, with one potential for each pair of variables X_i, X_j connected by an edge in M . Which of the following expressions correctly computes the message $\delta_{3 \rightarrow 6}$ that cluster C_3 will send to cluster C_6 during belief propagation? Assume that the variables in the sepsets are equal to the intersection of the variables in the adjacent cliques.



Your Answer

Score Explanation



$$\delta_{3 \rightarrow 6}(X_5) = \sum_{X_2} \phi_{2,5}(X_2, X_5) \delta_{2 \rightarrow 3}(X_2) \delta_{4 \rightarrow 3}(X_2) \delta_{7 \rightarrow 3}(X_5) \delta_{6 \rightarrow 3}(X_2)$$

$$\text{○ } \delta_{3 \rightarrow 6}(X_5) = \sum_{X_2} \delta_{2 \rightarrow 3}(X_2) \delta_{4 \rightarrow 3}(X_2) \delta_{7 \rightarrow 3}(X_5)$$

$$\text{● } \delta_{3 \rightarrow 6}(X_5) = \sum_{X_2} \phi_{2,5}(X_2, X_5) \delta_{2 \rightarrow 3}(X_2) \delta_{4 \rightarrow 3}(X_2) \delta_{7 \rightarrow 3}(X_5)$$

✓ 1.00

This is the correct message; we first multiply in all the incoming messages from cluster 2, 4 and 7 with the initial potential $\phi_{2,5}(X_2, X_5)$ and then sum out X_2 .

$$\text{○ } \delta_{3 \rightarrow 6}(X_5) = \sum_{X_2} \phi_{2,5}(X_2, X_5)$$

Total

1.00 /
1.00

Question 3

Message Passing Computation. Consider the Markov network M from the previous question. If the initial factors in the Markov network M are of the form as shown in the table below, regardless of the specific value of i, j (we basically wish to encourage variables that are connected by an edge to share the same assignment), compute the message $\delta_{3 \rightarrow 6}$, assuming that it is the first message passed during loopy belief propagation. Assume that the messages are all initialized to the 1 message, i.e. all the entries are initially set to 1.

You may separate the entries of the message by new lines. Order the entries by lexicographic variable order: for example, if the message is over one variable X_i , then enter in $\delta_{3 \rightarrow 6}(X_i = 0)$, $\delta_{3 \rightarrow 6}(X_i = 1)$. If the message is over two variables X_i, X_j , where $i < j$, enter the answers in the order $\delta_{3 \rightarrow 6}(X_i = 0, X_j = 0)$, $\delta_{3 \rightarrow 6}(X_i = 0, X_j = 1)$, $\delta_{3 \rightarrow 6}(X_i = 1, X_j = 0)$, $\delta_{3 \rightarrow 6}(X_i = 1, X_j = 1)$.

X_i	X_j	$\phi(X_i, X_j)$
1	1	10
1	0	1
0	1	1
0	0	10

You entered:

11 11

Your Answer

11



Score

Explanation

0.50

11



0.50

Total

1.00 / 1.00

Question 4

***Extracting Marginals at Convergence.** Given that you can renormalize the messages at any point during belief propagation and still obtain correct marginals, consider the message $\delta_{3 \rightarrow 6}$ that you computed. Use this observation to compute the final and possibly approximate marginal probability $P(X_4 = 1, X_5 = 1)$ (X_4 and X_5 are the variables in the previous question) in cluster

C_6 at convergence (as extracted from the cluster beliefs), giving your answer to 2 decimal places.

You entered:

0.45

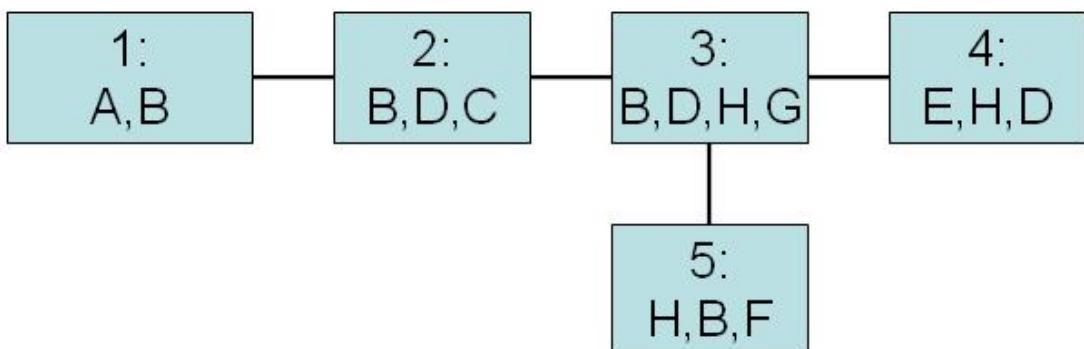
Your Answer	Score	Explanation
-------------	-------	-------------

0.45	✓ 1.00	Since $\delta_{3 \rightarrow 6}$ is proportional to a uniform factor, it gives no information; the same holds for $\delta_{1 \rightarrow 6}$. Thus the final cluster beliefs are proportional to the initial beliefs, giving us a probability of $\frac{10}{22}$.
------	--------	---

Total	1.00 / 1.00
-------	----------------

Question 5

Message Ordering. In the clique tree below which of the following starting message-passing orders is/are valid? (Note: These are not necessarily full sweeps that result in calibration. You may select 1 or more options, or none of them, if you think none apply.)



Your Answer

Score Explanation



$C_1 \rightarrow C_2, C_2 \rightarrow C_3, C_5 \rightarrow C_3, C_3 \rightarrow C_4$



0.25 This is a valid ordering because cliques only pass messages when they are ready.



$C_4 \rightarrow C_3, C_5 \rightarrow C_3, C_3 \rightarrow C_2, C_1 \rightarrow C_2$



0.25 This is a valid ordering because cliques do not pass messages until they are ready.



$C_4 \rightarrow C_3, C_3 \rightarrow C_2, C_2 \rightarrow C_1$



0.25 C_3 needs to have received a message from C_5 before it is ready to pass a message to C_2 .



0.25 C_3 needs to have received a

$$C_1 \rightarrow C_2, C_2 \rightarrow C_3, C_3 \rightarrow C_4, C_3 \rightarrow C_5$$

message from C_5 before it is ready to pass a message to C_4 .

Total	1.00 /
	1.00

Question 6

Message Passing in a Clique Tree. In the clique tree above, what is the correct form of the message from clique 3 to clique 2, $\delta_{3 \rightarrow 2}$, where $\psi_i(C_i)$ is the initial potential of clique i ?

Your Answer	Score	Explanation
<input checked="" type="radio"/> $\sum_{G,H} \psi_3(C_3) \times \delta_{4 \rightarrow 3} \times \delta_{5 \rightarrow 3}$	✓ 1.00	This is correct; to compute a message, we need to multiply the initial potential of clique 3 by all the incoming messages except the one from clique 2 and eliminate the variables that are not in the sepset.
<input type="radio"/> None of these		
<input type="radio"/> $\sum_{B,D} \psi_3(C_3)$		
<input type="radio"/> $\sum_{B,D} \psi_3(C_3) \times \delta_{4 \rightarrow 3} \times \delta_{5 \rightarrow 3}$		

Total	1.00 /
	1.00

Question 7

Family Preservation. Suppose we have a factor $P(A | C)$ that we wish to include in our sum-product message passing inference. We should:

Your Answer	Score	Explanation
<input checked="" type="radio"/> Assign the factor to one clique that contain A and C	✓ 1.00	Family Preservation explains that the proper construction of a clique tree (cluster graph) requires assigning each factor to one cluster whose scope contains the scope of the factor.
<input type="radio"/> None of these		
<input type="radio"/> Assign the factor to one clique that contain A or C		

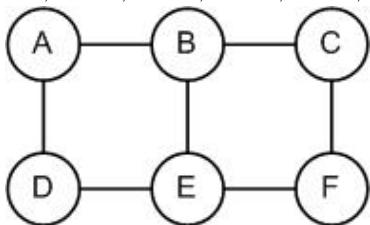
- Assign the factor to **all** cliques that contain **A and C**

Total	1.00 / 1.00
-------	----------------

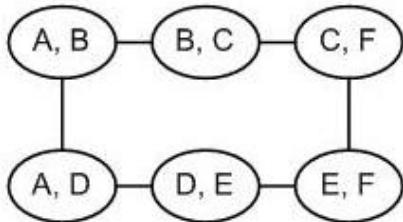
Question 8

Clique Tree Properties. Consider the following Markov Network over potentials

$\phi_{A,B}, \phi_{B,C}, \phi_{A,D}, \phi_{B,E}, \phi_{C,F}, \phi_{D,E}$, and $\phi_{E,F}$:



Which of the following properties are necessary for a valid clique tree for the above network, but are NOT satisfied by this graph:



You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> Running intersection property	✓ 0.25	The graph does satisfy the running intersection property, assuming that the sepsets are equal to the intersection of the variables in the adjacent cliques.
<input type="checkbox"/> The number of nodes in a clique tree containing a variable should be exactly the number of factors in the Markov network that contain the same variable	✓ 0.25	Multiple factors can be assigned to the same node in a clique tree, provided family preservation holds.
<input type="checkbox"/> Node degree less than or equal to 2	✓ 0.25	This is not a necessary condition for a cluster graph to be a clique tree.
<input checked="" type="checkbox"/> Family preservation	✓ 0.25	Family preservation is violated because $\phi_{B,E}$ cannot be assigned anywhere.

Total	1.00 /
	1.00

Question 9

Cluster Graphs vs. Clique Trees. Suppose that we ran sum-product message passing on a cluster graph G for a Markov network M and that the algorithm converged. Which of the following statements is true **only if** G is a clique tree and is **not** necessarily true otherwise?

Your Answer	Score	Explanation
-------------	-------	-------------

G is calibrated.

The sepsets in G are the product of the two messages passed between the clusters adjacent to the sepset.

All the options are true for cluster graphs in general.

The beliefs and sepsets of G can be used to compute the joint distribution defined by the factors of M .

If there are E edges in G , there exists a message

✓ 1.00

This is a property specific to clique trees. We can select one of the cliques to be the root clique, and pass messages away from the root clique to all other cliques. Then, we can pass messages from all other cliques towards the root clique and we are guaranteed to have calibrated the tree. In a cluster graph however, depending on

ordering that
guarantees
convergence
after passing
 $2E$
messages.

the potentials, convergence may take longer.

Total 1.00 /
1.00

Question 10

***Numerical Issues in Belief Propagation.** In practice, one of the issues that arises when we propagate messages in a *clique tree* is that when we multiply many small numbers, we quickly run into the precision limits of floating-point numbers, resulting in arithmetic underflow. One possible approach for addressing this problem is to renormalize each message, as it's passed, such that its entries sum to 1. Assume that we do not store the renormalization factor at each step. Which of the following statements describes the consequence of this approach?

Your Answer	Score	Explanation
<input checked="" type="radio"/> This does not change the results of the algorithm: when the clique tree is calibrated, we can obtain from it both the partition function and the correct marginals.	✓ 0.00	Think about how renormalizing the messages interacts with the computation of the partition function; how would we compute the partition function in the ideal case where we did not need to renormalize?
<input type="radio"/> This renormalization will give rise to incorrect marginals at calibration.		
<input type="radio"/> Calibration will not even be achieved using this scheme.		

Total 0.00 /
0.00

Question 11

***Numerical Issues in Belief Propagation.** The same numerical issues arise when we propagate messages in a *cluster graph with loops*. Which of the following statements describes the consequence of this approach? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> At convergence, the cluster graph will (in general) not satisfy the cluster graph invariant, i.e., the product of cluster beliefs divided by the product of sepset beliefs will not be equal to the original unnormalized distribution.	✗ 0.00	This is true because the statement is written as an equality to the unnormalized measure, from which we can reconstruct the partition function. In the normalized version, that partition function is lost. But a modified version of the cluster graph invariant, in which we change equality to a proportionality statement, still holds.
<input type="checkbox"/> At convergence, the marginals that we extract from each of the beliefs in the clusters will now be the marginals of the original joint distribution.	✓ 0.33	While renormalizing helps solves the numerical problems, this statement is not true because, ultimately the beliefs (when renormalized to sum to 1) will be the same as when we do not renormalize the messages. Hence, in general, we will not obtain the correct marginals since loopy belief propagation yields only approximate marginals.
<input type="checkbox"/> Convergence will never be achieved in this new scheme.	✓ 0.33	Whether convergence is achieved is dependent on the factors in the network and the network structure and does not depend on whether or not we renormalize. For instance, if the network contains deterministic potentials, the algorithm is less likely to converge.
Total	0.67 / 1.00	

Question 12

Convergence in Belief Propagation. Suppose we ran belief propagation on a cluster graph G and a clique tree T for the same Markov network that is a perfect map for a distribution P . Assume that both G and T are valid, i.e., they satisfy family preservation and the running intersection property. Which of the following statements regarding the algorithm are true? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Assuming the algorithm converges, if a variable X appears in two cliques in T , the marginals $P(X)$ computed from the the two clique beliefs must agree.	✓ 0.25	This is true due to the calibration property at convergence and the fact that the clique tree must satisfy the running intersection property (RIP). If a variable X is in two cliques, the cliques must be connected by a path for which X is always in the sepset (by the RIP), and so by calibration, the cliques must agree on the beliefs over the X .
<input checked="" type="checkbox"/> Assuming the algorithm converges, if a variable X appears in two clusters in G , the marginals $P(X)$ computed from the two cluster beliefs must agree.	✓ 0.25	This is true due to the calibration property at convergence and the fact that the cluster graph must satisfy the running intersection property (RIP). If a variable X is in two clusters, the clusters must be connected by a path for which X is always in the sepset (by the RIP), and so by calibration, the clusters must agree on the beliefs over the X .
<input type="checkbox"/> Belief propagation always converges on G .	✓ 0.25	This is not always true, for example, when there are strong opposing potentials along a loop in the cluster graph.
<input type="checkbox"/> If the algorithm converges, the final cluster beliefs in G , when renormalized to sum to 1,	✓ 0.25	This is not true, because the cluster graph may contain loops. One consequence of this is that information about a variable may be counted twice. In general, cluster graphs only return approximate marginals.

are true
marginals of
 P .

Total 1.00 /
 1.00

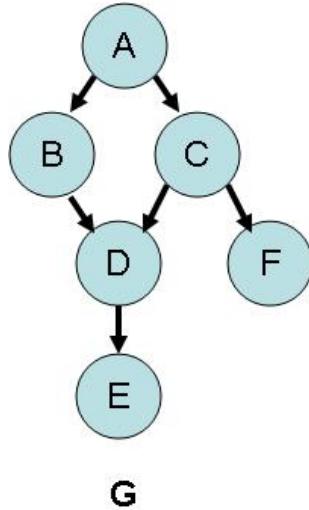
Feedback — Message Passing for MAP + Week 3 Review

[Help Center](#)

You submitted this quiz on **Tue 14 May 2013 12:46 PM PDT**. You got a score of **5.00** out of **5.00**.

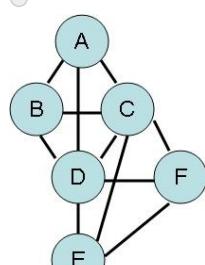
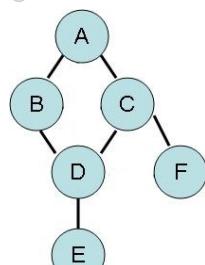
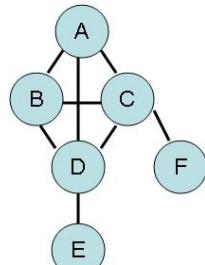
Question 1

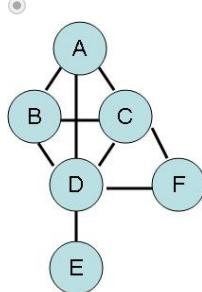
Induced Graphs. If we perform variable elimination on the graph shown below with the variable ordering B, A, C, F, E, D, G , what is the induced graph for the run?



Your Answer

Score Explanation



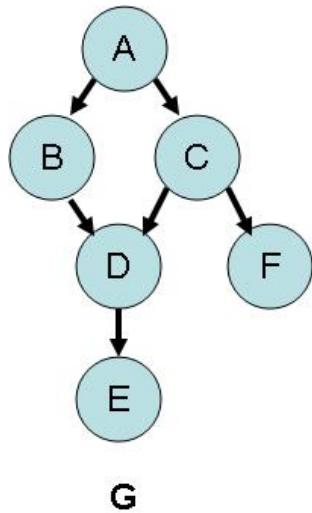


- 1.00 This is correct. There is an edge in the induced graph between every pair of variables that is present together in a factor during a run of variable elimination.

Total 1.00 /
1.00

Question 2

Intermediate Factors. If we perform variable elimination on the graph shown below with the variable ordering F, E, D, C, B, A , what is the intermediate factor produced by the third step (just before summing out D)?



Your Answer Score Explanation

$\psi(B, C)$

$\psi(B, C, D, E)$

$\psi(B, C, D, E, F)$

- $\psi(B, C, D)$ 1.00 This is correct. The factors involved in eliminating D are $\phi(B, C, D) = P(D | B, C)$ and $\tau_2(D)$ (from eliminating E), so the intermediate factor generated before eliminating D is the product of these two factors, $\psi(B, C, D) = \phi(B, C, D)\tau_2(D)$.

$\psi(A, B, C, D)$

Total 1.00 /
1.00

Question 3

Clique Tree Calibration. Which of the following is true? You may select more than one option (or none, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> If there exists a pair of adjacent cliques that are max-calibrated, then a clique tree is max-calibrated.	✓ 0.25	It is true that adjacent cliques have to be max-calibrated, but all adjacent pairs need to be max-calibrated, not just any two.
<input type="checkbox"/> If a clique tree is max-calibrated, then within each clique, all variables are max-calibrated with each other.	✓ 0.25	Calibration deals with adjacent pairs of cliques, not with the variables inside one clique alone.
<input type="checkbox"/> After we complete one upward pass of the max-sum message passing algorithm, the clique tree is max-calibrated.	✓ 0.25	The beliefs are max-calibrated only after we do a downward pass.
<input checked="" type="checkbox"/> If a clique tree is max-calibrated, then all pairs of adjacent cliques are max-calibrated.	✓ 0.25	This is the condition that makes a clique tree max-calibrated. All adjacent cliques have to agree over their sepset beliefs.
Total	1.00 / 1.00	

Question 4

Real-World Applications of MAP Estimation. Suppose that you are in charge of setting up a soccer league for a bunch of kindergarten kids, and your job is to split the N children into K teams. The parents are very controlling and also uptight about which friends their kids associate with. So some of them bribe you to set up the teams in certain ways.

The parents' bribe can take two forms: For some children i , the parent says "I will pay you A_{ij} dollars if you put my kid i on the same team as kid j "; in other cases, the parent of child i says "I will pay you B_i dollars if you put my kid on team k ." In our notation, this translates to factor $f_{i,j}(x_i, x_j) = A_{ij} \cdot \mathbf{1}\{x_i = x_j\}$ or $g_i(x_i) = B_i \cdot \mathbf{1}\{x_i = k\}$, respectively, where x_i is the assigned team of child i and $\mathbf{1}\{\cdot\}$ is the indicator function. More formally, if we define x_i to be the assigned team of child i , the amount of money you get for the first type of bribe will be $f_{i,j}(x_i, x_j)$.

Being greedy and devoid of morality, you want to make as much money as possible from these bribes. What are you trying to find?

Your Answer	Score	Explanation
<input checked="" type="radio"/> $\text{argmax}_{\bar{x}} \sum_i g_i(x_i) + \sum_{i,j} f_{i,j}(x_i, x_j)$	✓ 1.00	Correct. The total amount of money is the sum of the indicator functions, so you want to find the assignment that maximizes the sum.
<input type="radio"/> $\text{argmax}_{\bar{x}} \prod_i g_i(x_i) \cdot \prod_{i,j} f_{i,j}(x_i, x_j)$		
<input type="radio"/> $\text{argmax}_{\bar{x}} \sum_i g_i(x_i)$		
<input type="radio"/> $\text{argmax}_{\bar{x}} \prod_i g_i(x_i)$		
Total	1.00 / 1.00	

Question 5

***Decoding MAP Assignments.** You want to find the optimal solution to the above problem using a clique tree over a set of factors ϕ . How could you accomplish this such that you are guaranteed to find the optimal solution? (Ignore issues of tractability, and assume that if you specify a set of factors ϕ , you will be given a valid clique tree of minimum tree width.)

Your Answer	Score	Explanation
<input type="radio"/> The optimal solution is not guaranteed to be found in this manner using clique trees.		
<input type="radio"/> Set $\phi_{i,j} = \exp(f_{i,j})$, $\phi_i = \exp(g_i)$, get the clique tree, run sum-product message passing, and decode the marginals.		
<input checked="" type="radio"/> Set $\phi_{i,j} = \exp(f_{i,j})$, $\phi_i = \exp(g_i)$, get the clique tree over this set of factors, run max-sum message passing on this clique tree, and decode the marginals.	1.00	<p>We want to compute $\operatorname{argmax}_{\bar{x}} \sum_i g_i(x_i) + \sum_{i,j} f_{i,j}(x_i, x_j) = \operatorname{argmax}_{\bar{x}} \log \left[\prod_i \exp(g_i(x_i)) \cdot \prod_{i,j} \exp(f_{i,j}(x_i, x_j)) \right]$. Since maximizing $\log(z)$ over z is the same as maximizing z over z, we can simply compute $\operatorname{argmax}_{\bar{x}} \prod_i \exp(g_i(x_i)) \cdot \prod_{i,j} \exp(f_{i,j}(x_i, x_j))$, which is what max-sum message passing returns. So setting the potentials appropriately and running clique tree inference (which is exact) is guaranteed to get the optimal solution.</p> <p>(Remember that max-sum message passing involves taking a log-transform of the factors first, and summing up log-transformed factors is equivalent to multiplying them together; don't be tricked by the "sum"!)</p>
<input type="radio"/> Set $\phi_{i,j} = f_{i,j}$, $\phi_i = g_i$, get the clique tree over this set of factors, run max-sum message passing on this clique tree, and decode the marginals.		
<input type="radio"/> Set $\phi_{i,j} = f_{i,j}$, $\phi_i = g_i$, get the clique tree, run sum product message passing, and decode the marginals.		
Total	1.00 / 1.00	

Feedback — MAP Inference + Week 4 Review

[Help Center](#)

You submitted this quiz on **Tue 21 May 2013 10:09 PM PDT**. You got a score of **4.00** out of **4.00**.

Question 1

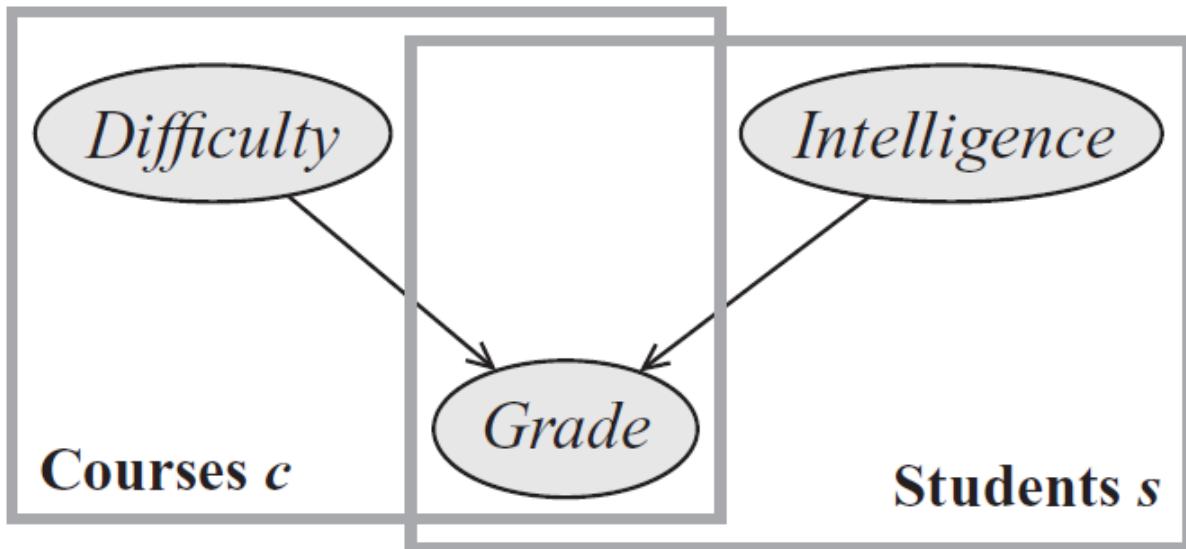
Reparameterization. Suppose we have a calibrated clique tree T and calibrated cluster graph G for the same Markov network, and have thrown away the original factors. Now we wish to reconstruct the joint distribution over all the variables in the network only from the beliefs and sepsets. Is it possible for us to do so from the beliefs and sepsets in T ? Separately, is it possible for us to do so from the beliefs and sepsets in G ?

Your Answer	Score	Explanation
<input type="radio"/> It is not possible in T or G .		
<input checked="" type="radio"/> It is possible in both T and G	1.00	Using the clique tree and cluster graph as reparameterizations, we can reconstruct the original distribution.
<input type="radio"/> It is possible in T but not in G .		
<input type="radio"/> It is possible in G but not in T .		
Total	1.00 / 1.00	

Question 2

***Markov Network Construction.** Consider the unrolled network for the plate model shown below, where we have n students and m courses. Assume that we have observed the grade of all students in all courses. In general, what does a pairwise Markov network that is a minimal I-map for the conditional distribution look like? (Hint: the factors in the network are the CPDs

reduced by the observed grades. We are interested in modeling the conditional distribution, so we do not need to explicitly include the Grade variables in this new network. Instead, we model their effect by appropriately choosing the factor values in the new network.)



Your Answer	Score	Explanation
<input type="radio"/> A fully connected graph with instantiations of the Difficulty and Intelligence variables.		
<input type="radio"/> Impossible to tell without more information on the exact grades observed.		
<input checked="" type="radio"/> A fully connected bipartite graph where instantiations of the Difficulty variables are on one side and	1.00	The factors, reduced by the evidence on the grades, have scopes over 2 variables: a Difficulty variable for a particular course and the Intelligence variable for a particular student. Hence, the variables naturally partition into two groups: all the instantiations of the Intelligence variables and all the instantiations for the Difficulty variables, which means we have a bipartite graph. It is a fully connected bipartite graph because we have a factor for each (Course, Student) assignment.

instantiations
of the
Intelligence
variables are
on the other
side.

A graph
over
instantiations
of the
Difficulty
variables
and
instantiations
of the
Intelligence
variables,
not
necessarily
bipartite;
there could
be edges
between
different
Difficulty
variables,
and there
could also
be edges
between
different
Intelligence
variables.

A bipartite
graph where
instantiations
of the
Difficulty
variables are
on one side
and
instantiations
of the
Intelligence
variables are
on the other
side. In
general, this

graph will
not be fully
connected.

Total 1.00 /
1.00

Question 3

Clique Tree Construction. We now wish to perform inference in the pairwise Markov network you came up with in the previous question. Define the size of a clique to be the number of variables in the clique. There exists a clique tree T^* for the pairwise Markov network such that the size of the largest clique in T^* is the smallest amongst all possible clique trees for this network. What is the size of the largest sepset in T^* ?

Note: if you're wondering why we would ever care about this, remember that the complexity of inference depends on the number of entries in the largest factor produced in the course of message passing, which in turn, is affected by the size of the largest clique in the network, amongst other things.

Hint: Use the relationship between sepsets and conditional independence to derive a lower bound for the size of the largest sepset, then construct a clique tree that achieves this bound.

Your Answer

Score Explanation

$m + n$

mn

$\min(m, n)$ ✓ 1.00 Given any clique tree, when you condition on the variables in one sepset, the variables on one side of the sepset in the clique tree have to be rendered independent of the variables on the other side of the clique tree (excluding the variables in the sepset, of course). From this, we can conclude that the minimum sepset size is at least $\min(m, n)$, because if you condition on a subset of variables smaller than $\min(m, n)$, the remaining variables will still be dependent. It is more straightforward to construct a clique tree that satisfies this lower bound.

$\min(m, n) + 1$

$\max(m, n) + 1$

$mn + 1$

$m + n + 1$

$\max(m, n)$

Total 1.00 /
1.00

Question 4

***Primal vs. Dual.** Suppose we have a chain MRF (with pairwise potentials), and we wish to find the MAP assignment to the variables. You decide to run max-sum inference to find the exact MAP assignment x^* . Your friend decides to use the dual decomposition algorithm to find the MAP assignment instead. Her algorithm converges and she solves the decoding problem to obtain an assignment x' such that $L(\lambda) = \text{score}(x')$ for the assignment. Which of the following relationships between $\text{score}(x^*)$ and $\text{score}(x')$ is true?

Your Answer

Score Explanation

$\text{score}(x^*) > \text{score}(x')$

None of the
relationships shown are
true.

$\text{score}(x^*) \neq \text{score}(x')$

$\text{score}(x^*) = \text{score}(x')$ ✓ 1.00 This is correct, because $L(\lambda) = \text{score}(x')$ implies that x' is also a MAP assignment, so we must have $\text{score}(x^*) = \text{score}(x')$.

Total 1.00 /
1.00

Feedback — Sampling Methods

[Help Center](#)

You submitted this quiz on **Tue 21 May 2013 12:41 PM PDT**. You got a score of **7.00** out of **7.00**.

Question 1

Forward Sampling. One strategy for obtaining an estimate to the conditional probability $P(\mathbf{y} | \mathbf{e})$ is by using forward sampling to estimate $P(\mathbf{y}, \mathbf{e})$ and $P(\mathbf{e})$ separately and then computing the ratio. We can use the Hoeffding Bound to obtain a bound on both the numerator and the denominator. Assume M is large. When does the resulting bound provide meaningful guarantees? Think about the difference between the true value and our estimate. Recall that we need $M \geq \frac{\ln(2/\delta)}{2\epsilon^2}$ to get an additive error bound ϵ that holds with probability $1 - \delta$ for our estimate.

Your Answer**Score****Explanation**

It never provides a meaningful guarantee.

It provides a meaningful guarantee, but only when ϵ is small relative to $P(\mathbf{e})$ and $P(\mathbf{y}, \mathbf{e})$

It provides a meaningful guarantee, but only when δ is small relative to $P(\mathbf{e})$ and $P(\mathbf{y}, \mathbf{e})$

It always provides meaningful guarantees.

 1.00

True. When ϵ isn't small with respect to $P(\mathbf{y}, \mathbf{e})$ and $P(\mathbf{e})$ the value of the estimated ratio $P(\mathbf{y}, \mathbf{e})/P(\mathbf{e})$ can be far from the true value of $P(\mathbf{y}|\mathbf{e})$ even if the absolute value of ϵ and hence the absolute error in estimating $P(\mathbf{e})$ and $P(\mathbf{y}, \mathbf{e})$ is small.

Total

1.00 /

1.00

Question 2

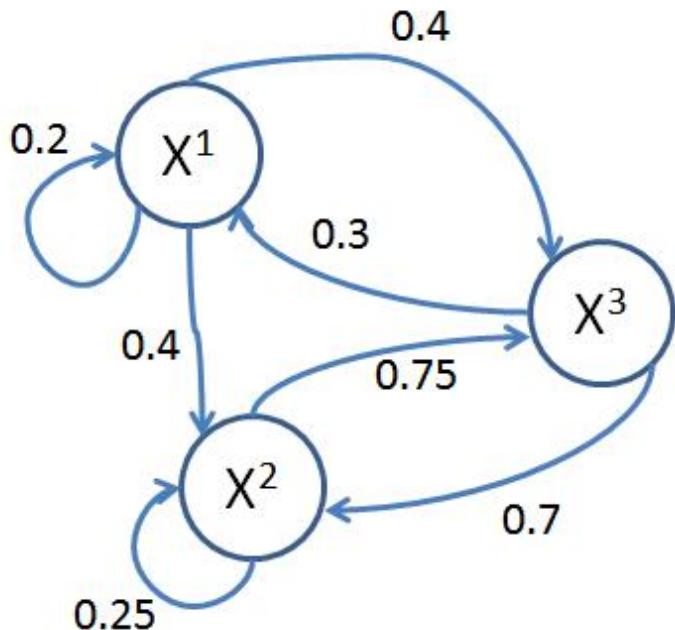
Rejecting Samples. Consider the process of rejection sampling to generate samples from the posterior distribution $P(X | e)$. If we want to obtain M samples, what is the expected number of samples that would need to be drawn from $P(X)$?

Your Answer	Score	Explanation
<input type="radio"/> $M \cdot P(X e)$		
<input type="radio"/> $M \cdot P(e)$		
<input type="radio"/> $M/(1 - P(e))$		
<input type="radio"/> $M \cdot (1 - P(e))$		
<input type="radio"/>		
<input type="radio"/> $M \cdot (1 - P(X e))$		
<input checked="" type="radio"/> $M/P(e)$	✓ 1.00	This is correct because it accounts for the samples we will reject if they don't agree with the evidence and end up with keeping M samples. Let A be the total number of samples. Then probability of keeping each sample is $P(e)$. Therefore, $M = P(e) * A$.

Total 1.00 /
1.00

Question 3

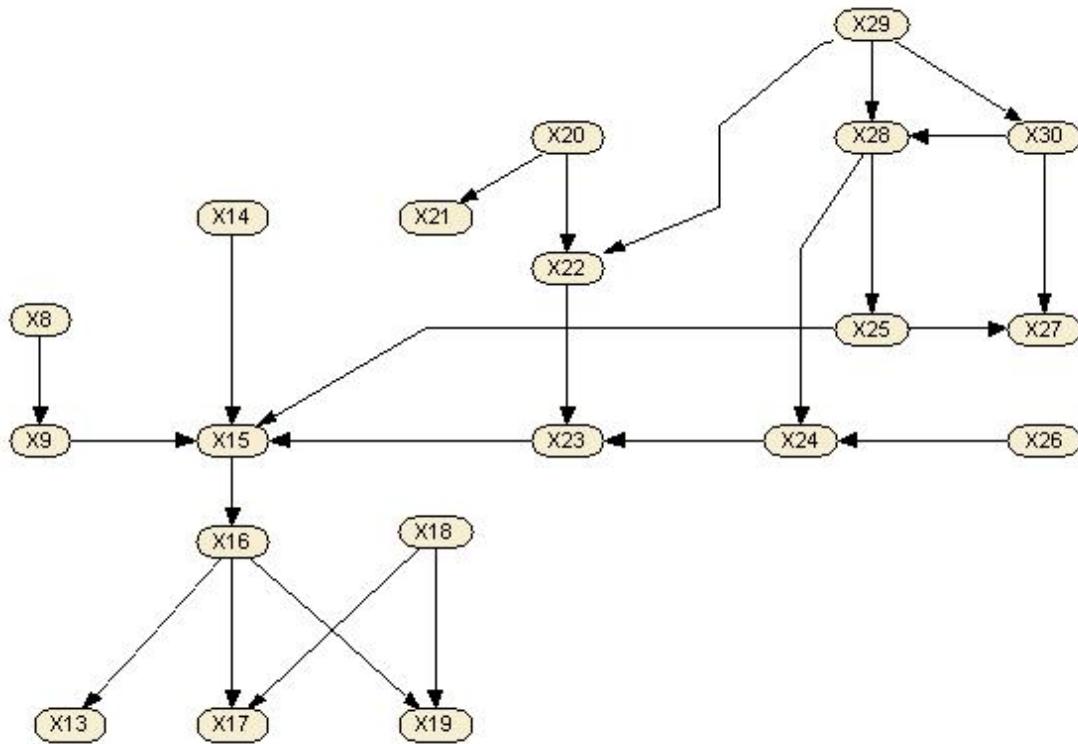
Stationary Distributions. Consider the simple Markov chain shown in the figure below. By definition, a stationary distribution π for this chain must satisfy which of the following properties? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
<input checked="" type="checkbox"/> $\pi(x_1) = 0.2\pi(x_1) + 0.3\pi(x_3)$	✓ 0.17	
<input type="checkbox"/> $\pi(x_3) = 0.3\pi(x_1) + 0.7\pi(x_3)$	✓ 0.17	
<input type="checkbox"/> $\pi(x_1) = \pi(x_2) = \pi(x_3)$	✓ 0.17	
<input type="checkbox"/> $\pi(x_3) = 0.4\pi(x_1) + 0.5\pi(x_2)$	✓ 0.17	
<input type="checkbox"/> $\pi(x_1) = 0.2\pi(x_1) + 0.4\pi(x_2) + 0.4\pi(x_3)$	✓ 0.17	
<input checked="" type="checkbox"/> $\pi(x_1) + \pi(x_2) + \pi(x_3) = 1$	✓ 0.17	
Total	1.00 / 1.00	

Question 4

***Gibbs Sampling in a Bayesian Network.** Suppose we have the Bayesian network shown in the image below. If we are sampling the variable X_{23} as a substep of Gibbs sampling, what is the closed form equation for the distribution we should use over the value x'_{23} ? By closed form, we mean that all computation such as summations are tractable and that we have access to all terms without requiring extra computation.

**Your Answer****Score** **Explanation**

$P(x'_{23} | x_{-23})$ where x_{-23} is all variables except x_{23}

$P(x'_{23} | x_{22}, x_{24})P(x_{15} | x'_{23}, x_{14}, x_9, x_{25})$

$$\frac{P(x'_{23}|x_{22},x_{24})P(x_{15}|x'_{23},x_{14},x_9,x_{25})}{\sum_{x''_{23}} P(x''_{23}|x_{22},x_{24})P(x_{15}|x''_{23},x_{14},x_9,x_{25})}$$



1.00

This is correct. This distribution can be computed based only on the Markov blanket of x_{23} .



$$\frac{P(x'_{23}|x_{22},x_{24})P(x_{15}|x'_{23},x_{14},x_9,x_{25})}{\sum_{x''_9,x''_{14},x''_{22},x''_{24},x''_{25}} P(x''_{23}|x''_{22},x''_{24})P(x''_{15}|x''_{23},x''_{14},x''_9,x''_{25})}$$

None of these; they are all either incorrect or not in closed form

$P(x'_{23} | x_{22}, x_{24})$

Total

1.00 /

1.00

Question Explanation

$P(x'_{23} | x_{-23})$ is correct but not in closed form because we don't have direct access to this

term. To get it, we have to expand it, and since all factors involving variables not in the Markov blanket cancel out (see equation 12.23 in Chapter 12.3.3), we get the correct answer. (K & F, pg 513)

Question 5

Gibbs Sampling. Suppose we are running the Gibbs sampling algorithm on the Bayesian network $X \rightarrow Y \rightarrow Z$. If the current sample is $\langle x_0, y_0, z_0 \rangle$ and we sample y as the first substep of the Gibbs sampling process, with what probability will the next sample be $\langle x_0, y_1, z_0 \rangle$ in the first substep?

Your Answer	Score	Explanation
<input type="radio"/>		$P(x_0, y_1, z_0)$
<input checked="" type="radio"/>	1.00	For Gibbs Sampling, we select one variable and keep others constant to compute the conditional probability of that variable being sampled given all the other variables.
<input type="radio"/>		$P(y_1 x_0)$
<input type="radio"/>		$P(x_0, z_0 y_1)$
Total	1.00 / 1.00	

Question 6

Collecting Samples. Assume we have a Markov chain that we have run for a sufficient burn-in time, and now wish to collect samples and use them to estimate the probability that $X_i = 1$. Can we collect and use every sample from the Markov chain after the burn-in?

Your Answer	Score	Explanation
<input checked="" type="radio"/> Yes, that would give a correct estimate of the probability. However, we cannot apply the Hoeffding bound to estimate	1.00	This is correct because after the burn-in time the collected samples are all samples from the stationary (posterior) distribution. The Hoeffding bound cannot be used, because consecutive samples from the chain are not independent.

the error in our estimate.

Yes, and if we collect m consecutive samples, we can use the Hoeffding bound to provide (high-probability) bounds on the error in our estimated probability.

No, Markov chains are only good for one sample; we have to restart the chain (and burn-in) before we can collect another sample.

No, once we collect one sample, we have to continue running the chain in order to "re-mix" it before we get another sample.

Total 1.00 /
1.00

Question 7

Markov Chain Mixing. Which of the following classes of chains would you expect to have the shortest mixing time in general?

Your Answer

Score **Explanation**

- Markov chains where state spaces are well connected and transitions between states have high probabilities. ✓ 1.00 This is correct because if you are able to move around the state space, you are more likely to mix in quickly.
- Markov chains with many

distinct and peaked probability modes.

Markov chains for networks with nearly deterministic potentials.

Markov chains with distinct regions in the state space that are connected by low probability transitions.

Total 1.00 /
1.00

Feedback — Inference in Temporal Models

[Help Center](#)

You submitted this quiz on **Mon 27 May 2013 12:04 PM PDT**. You got a score of **2.75** out of **3.00**. You can [attempt again](#), if you'd like.

Question 1

Unrolling DBNs. Which independencies hold in the unrolled network for the following 2-TBN for all t ?



There are four variables in a time slice plate indexed by t . The four variables are: weather, velocity, location, and failure. To the right of the plate there is another time slice plate indexed by $t+1$. The second plate has the same four variables, but they are denoted by the variable name followed by a prime symbol, for example weather prime. There are arrows connecting the variables in the first plate to the second plate: weather to weather prime, weather to velocity prime, weather to failure prime; velocity to velocity prime, velocity to location prime; location to location prime; failure to failure prime. In addition to the four variables in the second plate, there is another variable obs prime inside the second plate, and there are two arrows from location prime and failure prime, respectively, to it.

(Hint: it may be helpful to draw the unrolled DBN for several slices)

Your Answer	Score	Explanation
<input type="checkbox"/> $(Failure^t \perp Location^t \mid Obs^{1 \dots t})$	✓ 0.17	One can trace an active path between $Failure^t$ and $Location^t$ due to the active v-structure given by Obs^t
<input type="checkbox"/> None of these	✓ 0.17	Some of the independencies do hold. Perhaps you could try to draw the unrolled DBN and see whether active paths exist between the variables in question.
<input type="checkbox"/> $(Failure^t \perp Velocity^t \mid Obs^{1 \dots t})$	✓ 0.17	One can trace an active path between $Failure^t$ and $Velocity^t$ in the unrolled DBN.

<input type="checkbox"/> $(Weather^t \perp Velocity^t Obs^{1\dots t})$	✓ 0.17	One can trace an active path between $Weather^t$ and $Velocity^t$ in the unrolled DBN.
<input checked="" type="checkbox"/> $(Weather^t \perp Velocity^t Weather^{(t-1)}, Obs^{1\dots t})$	✓ 0.17	$Weather^t$ is blocked by $Weather^{t-1}$ for all t .
<input type="checkbox"/> $(Weather^t \perp Location^t Velocity^t, Obs^{1\dots t})$	✓ 0.17	One can trace an active path between $Weather^t$ and $Location^t$ in the unrolled DBN.

Total 1.00 / 1.00

Question 2

***Limitations of Inference in DBNs.** What makes inference in DBNs difficult?

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> As t grows large, we generally lose independencies of the form $(X^{(t)} \perp Y^{(t)} Z^{(t)})$	✓ 0.25	This is true, and this phenomenon is known as entanglement.
<input type="checkbox"/> Standard clique tree inference cannot be applied to a DBN	✓ 0.25	We can apply clique tree inference to a DBN; it just might be slow and undesirable in certain cases.
<input checked="" type="checkbox"/> In many networks, maintaining an exact belief state over the variables requires a full joint distribution over all variables in each time slice	✓ 0.25	This is true because we generally lose independencies relating variables in the belief state due to entanglement. Hence, the only way to maintain an exact belief state often requires a full joint distribution.
<input type="checkbox"/> As t grows large, we generally lose all independencies in the ground network	✓ 0.25	We do indeed lose some independencies, but do we lose all independencies? For instance, consider whether variables in time step $t+1$ are independent of variables in time step $t-1$ given those in time step t .

Total	1.00 /
	1.00

Question 3

Entanglement in DBNs. Which of the following are consequences of entanglement in Dynamic Bayesian Networks over discrete variables?

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> All variables in the unrolled DBN become correlated.	✗ 0.00	This is not true; only variables in the belief state become correlated.
<input type="checkbox"/> The belief state factorizes in the unrolled DBN if the belief state factorizes in the 2-TBN for the DBN.	✓ 0.25	This is not a consequence of entanglement. In fact, even if the belief state factorizes in the 2-TBN for the DBN, it is unlikely to factorize in the unrolled network due to entanglement.
<input type="checkbox"/> The size of an exact representation of the belief state is quadratic in the number of variables.	✓ 0.25	This is generally not true, unless we represent the belief state using a Gaussian distribution; a Gaussian distribution is completely determined by the mean vector and covariance matrix, the latter having a size that is quadratic in the number of variables.
<input checked="" type="checkbox"/> The size of an exact representation of the belief state is exponentially large in the number of variables.	✓ 0.25	This is true, since the only way to represent the belief state exactly is to maintain a full joint distribution.

Total	0.75 /
	1.00

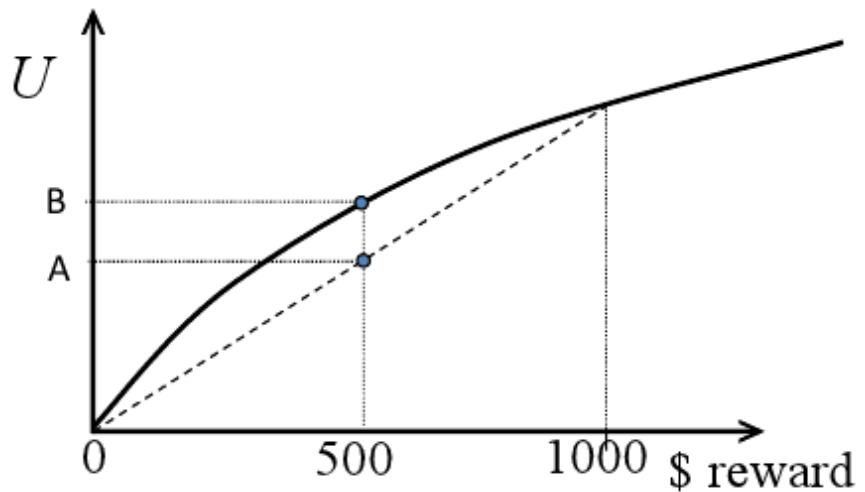
Feedback — Decision Theory

[Help Center](#)

You submitted this quiz on **Fri 24 May 2013 8:53 AM PDT**. You got a score of **4.00** out of **4.00**.

Question 1

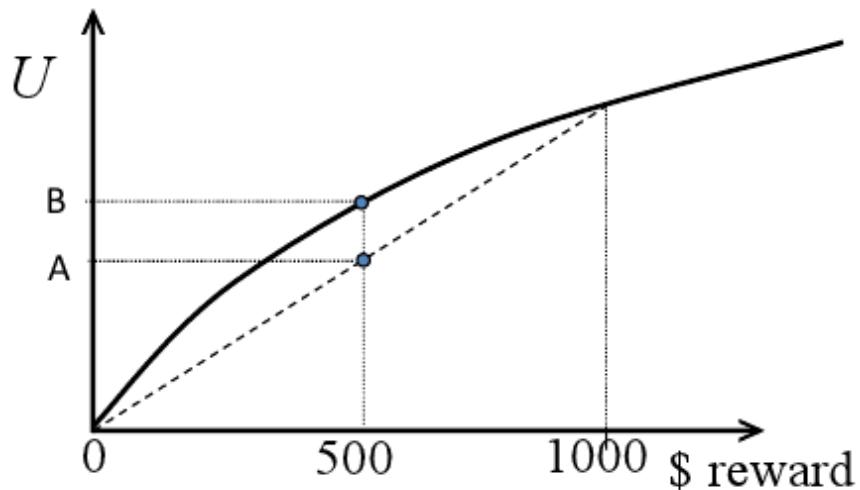
Utility Curves. What does the point marked *A* on the *Y* axis correspond to? (Mark all that apply.)



Your Answer	Score	Explanation
<input type="checkbox"/> \$500	✓ 0.25	Think about what the plot is showing.
<input type="checkbox"/> $U(\$500)$	✓ 0.25	A is not on the utility curve.
<input checked="" type="checkbox"/> $U(\ell)$ where ℓ is a lottery that pays \$0 with probability 0.5 and \$1000 with probability 0.5.	✓ 0.25	Yes, this is correct, since the value of the lottery is equivalent to $0.5U(\$0) + 0.5U(\$1000)$.
<input checked="" type="checkbox"/> $0.5U(\$0) + 0.5U(\$1000)$	✓ 0.25	This is correct, as you can observe from the geometry of the triangles in the figure.
Total	1.00 / 1.00	

Question 2

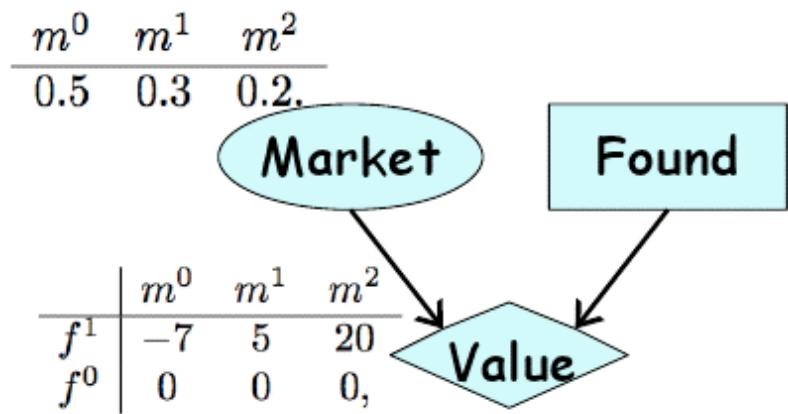
Utility Curves. What does the point marked B on the Y axis correspond to? (Mark all that apply.)



Your Answer	Score	Explanation
<input type="checkbox"/> $0.5U(\$0) + 0.5U(\$1000)$	✓ 0.25	Think about the fact that B lies on the curve.
<input checked="" type="checkbox"/> $U(\$500)$	✓ 0.25	Yes, this is correct, since point B is on the curve, it represents $U(\$500)$.
<input type="checkbox"/> $U(\ell)$ where ℓ is a lottery that pays \$0 with probability 0.5 and \$1000 with probability 0.5.	✓ 0.25	Think about the fact that B lies on the curve.
<input type="checkbox"/> \$500	✓ 0.25	Think about the fact that B lies on the curve.
Total	1.00 / 1.00	

Question 3

Expected Utility. In the simple influence diagram on the right, with the CPD for M and the utility function V , what is the expected utility of the action f^1 ?



Your
Answer

0

20

2

1.00

This is correct. The expected utility is given by $0.5*(-7) + 0.3*5 + 0.2*20 = 2$.

5

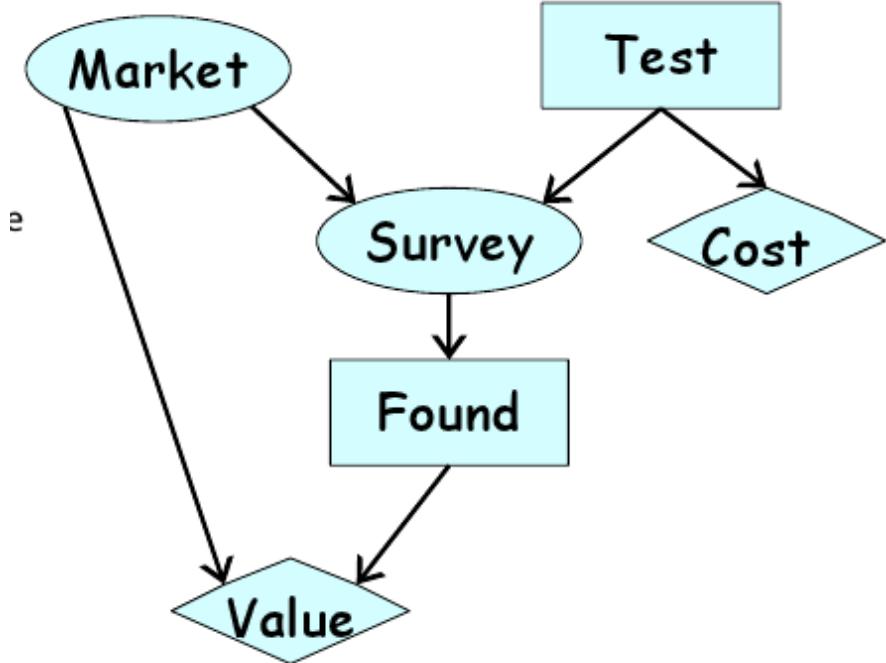
Total

1.00 /

1.00

Question 4

***Uninformative Variables.** In the influence diagram on the right, what is an appropriate way to have the model account for the fact that if the Test wasn't performed (t^0), then the survey is uninformative?

**Your Answer****Score** **Explanation**

Set $P(S|M, t^0)$ to be uniform.

Set $P(S|M, t^0) = P(S|M, t^1)$.

Set $P(S|M, t^0)$ so that S takes some new value “not performed” with probability 1. ✓ 1.00 This is the appropriate action. Assigning S to any other value would not be desirable, as these other values may represent survey results, but we have not actually conducted the survey.

Set $P(S|M, t^0)$ so that S takes the value s^0 with probability 1.

Total 1.00 /
1.00

Feedback — Learning in Parametric Models

[Help Center](#)

You submitted this quiz on **Tue 4 Jun 2013 11:14 AM PDT**. You got a score of **7.00** out of **7.00**.

Question 1

Computing Sufficient Statistics. Suppose that you are playing Dungeons & Dragons, and you suspect that the 4-sided die that your Dungeon Master is using is biased. In the past 60 times that you have attacked with your dagger and the 4-sided die was rolled to calculate how many hit points of damage you inflicted, 20 times it has come up 1, 15 times it has come up 2, 15 times it has come up 3, and 10 times it has come up 4. Let θ_1 be the true probability of the die landing on 1, and similarly for θ_2 , θ_3 , and θ_4 . You want to estimate these parameters from the past 60 rolls that you observed using a simple multinomial model. Which of the following is a sufficient statistic for this data?

Your Answer**Score****Explanation**

The total number of times you viciously attacked a monster with your dagger (i.e., the total number of times that the dice was rolled).

A vector with four components, with the i^{th} component being the number of times you dealt i hit points worth of damage.

✓ 1.00

A sufficient statistic is a function of the data that summarizes the relevant information for computing the likelihood. The sufficient statistics for a multinomial model are the "counts" of each possible result. The number of times each digit was rolled allows us to compute the likelihood function.

The total amount of damage you inflicted with your trusty dagger

(i.e., the sum of all die rolls).

- None of these are sufficient statistics.

Total 1.00 /
1.00

Question 2

MLE Parameter Estimation. In the context of the previous question, what is the unique Maximum Likelihood Estimate (MLE) of the parameters θ ? Enter θ_1 on the first line, θ_2 on the second, and so forth (til θ_4). Give your answers rounded to the nearest ten-thousandth (i.e. 1/3 should be 0.3333).

You entered:

0.3333 0.2500 0.2500 0.1667

Your Answer	Score	Explanation
0.3333	✓ 0.25	
0.2500	✓ 0.25	
0.2500	✓ 0.25	
0.1667	✓ 0.25	
Total	1.00 / 1.00	

Question 3

Likelihood Functions. For a Naive Bayes network with one parent node, X , and 3 children nodes, Y_1, Y_2, Y_3 , which of the expressions below would be a correct expression for the likelihood, decomposed in terms of the local likelihood functions?

Your Answer	Score	Explanation
<input type="radio"/> $L(\theta : D) = \prod_{m=1}^M P(y_1[m], y_2[m], y_3[m] : \theta)P(x[m] y_1[m], y_2[m], y_3[m])$		



$$L(\theta : D) = (\prod_{m=1}^M P(x[m] : \theta_X))(\prod_{m=1}^M P(y_1[m]|x[m] : \theta_{Y_1|X}))(\prod_{m=1}^M$$

✓ 1.00

The formulation for a likelihood function decomposed into local likelihood functions follows directly from the lecture videos and takes this form.

$L(\theta : D) = \prod_{m=1}^M P(x[m], y_1[m], y_2[m], y_3[m] : \theta)$

$L(\theta : D) = \prod_{m=1}^M P(y_1[m]|x[m] : \theta_{Y_1|X})P(y_2[m]|x[m] : \theta_{Y_2|X})P(y_3[m]$

Total	1.00 /
	1.00

Question 4

MLE for Naive Bayes. Using a Naive Bayes model for spam classification with the vocabulary $V = \{"SECRET", "OFFER", "LOW", "PRICE", "VALUED", "CUSTOMER", "TODAY", "DOLLAR", "MILLION", "SPORTS", "IS", "FOR", "PLAY", "HEALTHY", "PIZZA"\}$. We have the following example spam messages $SPAM = \{"MILLION DOLLAR OFFER", "SECRET OFFER TODAY", "SECRET IS SECRET"\}$ and normal messages, $NON-SPAM = \{"LOW PRICE FOR VALUED CUSTOMER", "PLAY SECRET SPORTS TODAY", "SPORTS IS HEALTHY", "LOW PRICE PIZZA"\}$.

We create a multinomial naive Bayes model for the data given above. This can be modeled as a parent node taking values SPAM and NON-SPAM and a child node for each word in the vocabulary. The θ values are estimated based on the number of times that a word appears in the vocabulary. Give the MLEs for θ_{SPAM} , $\theta_{SECRET|SPAM}$, $\theta_{SECRET|NON-SPAM}$, $\theta_{SPORTS|NON-SPAM}$, $\theta_{DOLLAR|SPAM}$ respectively. Separate each with new lines, in the order listed above. Enter the value as a decimal rounded to the nearest ten-thousandth (0.xxxx).

You entered:

0.4286 0.3333 0.0667 0.1333 0.1111

Your Answer	Score	Explanation
0.4286	✓ 0.20	Recall for naive bayes models, our theta parameter is just the proportion of samples with the specified characteristic out of all samples that match the given characteristics. ie. $\theta_{A B}$ is the proportion of samples of type A out of those matching description B .
0.3333	✓ 0.20	Recall for naive bayes models, our theta parameter is just the proportion of samples with the specified characteristic out of all samples that match the given characteristics. ie. $\theta_{A B}$ is the proportion of samples of type A out of those matching description B .
0.0667	✓ 0.20	Recall for naive bayes models, our theta parameter is just the proportion of samples with the specified characteristic out of all samples that match the given characteristics. ie. $\theta_{A B}$ is the proportion of samples of type A out of those matching description B .
0.1333	✓ 0.20	Recall for naive bayes models, our theta parameter is just the proportion of samples with the specified characteristic out of all samples that match the given characteristics. ie. $\theta_{A B}$ is the proportion of samples of type A out of those matching description B .
0.1111	✓ 0.20	Recall for naive bayes models, our theta parameter is just the proportion of samples with the specified characteristic out of all samples that match the given characteristics. ie. $\theta_{A B}$ is the proportion of samples of type A out of those matching description B .
Total	1.00 / 1.00	

Question 5

Learning Setups. Consider the following scenario: You have been given a dataset that contains patients and their gene expression data for 10 genes. You are also given a 0/1 label where 1 means that patient has disease A and 0 means the patient does not. Your goal is to learn a classification algorithm that could predict these labels with high accuracy. You split the data into three sets:

- 1: Set of patients used for fitting the classifier parameters (e.g., the weights and bias of a logistic regression classifier).
- 2: Set of patients used for tuning the hyperparameters of the classifier (e.g., how much regularization to apply).

3: Set of patients used to assess the performance of the classifier.

What are these sets called?

Your Answer	Score	Explanation
<input checked="" type="radio"/> 1: Training Set, 2: Validation Set, 3: Test Set	✓ 1.00	We fit parameters on training set, tune on validation set and assess performance on test set.
<input type="radio"/> 1: Validation Set, 2: Test Set, 3: Training Set.		
<input type="radio"/> 1 & 2: Training Set, 3: Validation Set.		
<input type="radio"/> 1: Training Set, 2: Test Set, 3: Validation Set		
Total	1.00 / 1.00	

Question 6

Constructing CPDs. Assume that we are trying to construct a CPD for a random variable whose value labels a document (e.g., an email) as belonging to one of two categories (e.g., spam or non-spam). We have identified K words whose presence (or absence) in the document each changes the distribution over labels (e.g., the presence of the word "free" is more likely to indicate that the email is spam). Assume that we have M labeled documents that we use to estimate the parameters for the CPD of the label given indicator variables representing the appearance of words in the document. We plan to use maximum likelihood estimation to select the parameters of this CPD.

If $M = 1,000$ and $K = 30$, which of the following CPD types are most likely to provide the best generalization performance to unseen data? Mark all that apply.

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> A sigmoid CPD	✓ 0.25	With a sigmoid CPD the number of parameters that will need to be learned is $K = 30$ (plus 1 for the bias term) and thus $M = 1000$ instances are sufficient to get a reasonable maximum likelihood estimation of the parameters and hence the distribution.
<input type="checkbox"/> A table CPD	✓ 0.25	A table CPD has $(2^{30} - 1)$ free parameters and hence we do not have enough instances to get a reasonable estimate of the

CPD

distribution for this type of CPD (the variance of our estimator will be high).

None of these CPDs would work.

0.25

One of these CPDs would likely provide better generalization performance than the others.

A linear Gaussian CPD

0.25

A linear Gaussian CPD is inappropriate because the label variable is discrete (and in fact, binary) as it would take on the values "present" and "not present".

Total

1.00 /

1.00

Question 7

Constructing CPDs. For the same scenario as described in the previous question, if $M = 100,000$ and $K = 3$, which of the following CPD types is most likely to provide the best generalization performance to unseen data?

Your Answer

Score

Explanation

A table CPD

1.00

In this scenario, a table CPD has $(2^3 - 1)$ free parameters, so we have enough instances to get a good estimate of the distribution for this type of CPD.

A linear Gaussian CPD

A sigmoid CPD

A tree CPD with $K = 3$ leaves

Total

1.00 /

1.00

Feedback — Bayesian Priors for BNs

[Help Center](#)

You submitted this quiz on **Tue 4 Jun 2013 12:55 PM PDT**. You got a score of **4.00** out of **4.00**.

Question 1

BDe Priors. The following is a common approach for defining a parameter prior for a Bayesian network, and is referred to as the BDe prior. Let P_0 be some distribution over possible assignments x_1, \dots, x_n , and select some fixed α . For a node X with parents \mathbf{U} we define $\alpha_{x|\mathbf{u}} = \alpha P_0(x, \mathbf{u})$.

For this question, assume X takes one of m values and that X has k parents, each of which takes d values. If we choose P_0 to be the uniform distribution, then what is the value of $\alpha_{x|\mathbf{u}}$?

Your Answer**Score****Explanation** α $\alpha/(md^k)$ 

1.00

For any joint distribution it must hold that

$$\sum_x \sum_{\mathbf{u}} P_0(x, \mathbf{u}) = 1; \text{ and for a uniform distribution all } md^k \text{ terms in the sum are constant and hence must equal to } \frac{1}{md^k}.$$
 $\alpha/((m+k)d)$ $\alpha/(mk^d)$

Total

1.00 /

1.00

Question 2

Learning with a Dirichlet Prior. Suppose we are interested in estimating the distribution over the English letters. We assume an alphabet that consists of 26 letters and the space symbol, and we ignore all other punctuation and the upper/lower case distinction. We model the distribution over the 27 symbols as a multinomial parametrized by $\theta = (\theta_1, \dots, \theta_{27})$ where $\sum_i \theta_i = 1$ and

all $\theta_i \geq 0$. Now we go to Stanford's Green library and repeat the following experiment: randomly pick up a book, open a page, pick a spot on the page, and write down the nearest symbol that is in our alphabet. We use $X[m]$ to denote the letter we obtain in the m th experiment. In the end, we have collected a dataset $D = \{x[1], \dots, x[2000]\}$ consisting of 2000 symbols, among which "e" appears 260 times. We use a Dirichlet prior over θ , i.e. $P(\theta) = Dirichlet(\alpha_1, \dots, \alpha_{27})$ where each $\alpha_i = 10$. What is the predictive probability that letter "e" occurs with this prior? (i.e., what is $P(X[2001] = "e" | D)$)? Write your answer as a decimal rounded to the nearest **ten thousandth** (0.xxxx).

You entered:

0.1189

Your Answer

0.1189



Score

1.00

Explanation

Total

1.00 / 1.00

Question 3

Learning with a Dirichlet Prior. In the setting of the previous question, suppose we had collected $M = 2000$ symbols, and the number of times "a" appeared was 100, while the number of times "p" appeared was 87. Now suppose we draw 2 more samples, $X[2001]$ and $X[2002]$. If we use $\alpha_i = 10$ for all i , what is the probability of $P(X[2001] = "p", X[2002] = "a" | D)$? (round your answer to the nearest **millionth**, 0.xxxxxx)

You entered:

0.002070

Your Answer

0.002070

Score



Explanation

Using the chain rule, this breaks down to

$P(X[2001] = "p" | D) \cdot P(X[2002] = "a" | X[2001] = "p", D)$.

Using this formation and using the updated estimates and total count in $P(X[2002] = "a" | X[2001] = "p", D)$, we get the correct

option as $\frac{97}{2270} \frac{110}{2271}$ which rounds to .002070

Total

1.00 /

1.00

Question 4

***Learning with a Dirichlet Prior.** In the setting of previous two questions, suppose we have collected M symbols, and let $\alpha = \sum_i \alpha_i$ (we no longer assume that each $\alpha_i = 10$). In which situation(s) does the Bayesian predictive probability using the Dirichlet prior (i.e., $P(X[M+1] | D)$) converge to the MLE estimation for any distribution over M ? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> $\alpha \rightarrow 0$ and M is fixed	✓ 0.20	The Dirichlet prior is a weighted average of the prior mean and the MLE estimate. Thus, if $\alpha \rightarrow 0$ for a fixed value of M then the probability will be dominated by the actual counts as the influence of our prior vanishes and will converge to MLE estimation.
<input type="checkbox"/> None of the above	✓ 0.20	There is at least one answer here that converges to the MLE estimation. Convergence does not mean that for any given value of α, M the estimate will be exactly the MLE estimate, but only that it becomes infinitesimally close to the MLE estimate.
<input checked="" type="checkbox"/> $\frac{\alpha}{M} \rightarrow 0$	✓ 0.20	The Dirichlet prior is a weighted average of the prior mean and the MLE estimate. Thus, in this case the probability will be dominated by the actual counts as the influence of our prior vanishes and will converge to MLE estimation.
<input type="checkbox"/> $\frac{\alpha}{M} \rightarrow \infty$	✓ 0.20	The Dirichlet prior is a weighted average of the prior mean and the MLE estimate. Thus, in this case the probability will be dominated by the prior as the relative influence of our actual counts vanish and will not converge to MLE estimation.
<input type="checkbox"/> Both α and M are fixed and non-zero for some fixed distribution over α	✓ 0.20	The Dirichlet prior is a weighted average of the prior mean and the MLE estimate. In this case, for some fixed α we can find a distribution over M that is not the same as the distribution over α . Thus, our prior will keep us some constant away from the MLE distribution.
Total	1.00 / 1.00	

Feedback — Parameter Learning in MNs

[Help Center](#)

You submitted this quiz on **Tue 4 Jun 2013 1:39 PM PDT**. You got a score of **3.00** out of **3.00**.

Question 1

***MRF Parameter Learning.** Consider the process of gradient-ascent training for a log-linear model with k features, given a data set D with M instances. Assume for simplicity that the cost of computing a single feature over a single instance in our data set is constant, as is the cost of computing the expected value of each feature once we compute a marginal over the variables in its scope. Also assume that we can compute each required marginal in constant time after we have a calibrated clique tree.

Assume that we use clique tree calibration to compute the expected sufficient statistics in this model and that the cost of doing this is c . Also, assume that we need r iterations for the gradient process to converge.

What is the cost of this procedure? Recall that in big-O notation, same or lower-complexity terms are collapsed.

Your Answer**Score****Explanation** $O(Mk + Mrc)$  $O(Mk + r(c + k))$ 

1.00

Before we start the gradient ascent process, we compute the empirical expectation for each of the k features by summing over their values for each of the M instances in our data set D ; the cost of this is Mk . Then, at each iteration, we use clique tree calibration at a cost c and extract the expected sufficient statistics from calibrated beliefs and update each of the k parameters θ_i . Thus, the cost per iteration is $c + k$, and the total cost for r iterations is $r(c + k)$. Together with the initial computation of empirical expectations, we get a total cost of $O(Mk + r(c + k))$.

 $O(r(Mc + k))$ $O(r(Mk + c))$

Total

1.00 /

1.00

Question 2

***CRF Parameter Learning.** Consider the process of gradient-ascent training for a CRF log-linear model with k features, given a data set D with M instances. Assume for simplicity that the cost of computing a single feature over a single instance in our data set is constant, as is the cost of computing the expected value of each feature once we compute a marginal over the variables in its scope. Also assume that we can compute each required marginal in constant time after we have a calibrated clique tree.

Assume that we use clique tree calibration to compute the expected sufficient statistics in this model, and that the cost of running clique tree calibration is c . Assume that we need r iterations for the gradient process to converge.

What is the cost of this procedure? Recall that in big-O notation, same or lower-complexity terms are collapsed.

Your Answer**Score****Explanation**

$O(Mk + Mrc)$

$O(rMc + rk)$

✓ 1.00

When training the CRF, at each iteration we need to perform clique tree calibration and compute the expected value of each of the k features M times; thus, the computation at each iteration required $M(c + k)$ operations. Note, however, that we can actually do better: if we aggregate the probabilities from these M clique trees into a single clique tree and then compute the feature value using the aggregated clique tree, we get $Mc + k$ operations per iteration; the procedure is correct due to linearity of expectations.

$O(Mk + rc)$

$O(Mk + r(c + k))$

Total

1.00 /

1.00

Question 3

Parameter Learning in MNs vs BNs. Compared to learning parameters in Bayesian networks, learning in Markov networks is generally...

Your Answer	Score	Explanation
<input type="radio"/> equally difficult, as both require an inference step at each iteration.		
<input checked="" type="radio"/> more difficult because we cannot use parallel optimization of subparts of our likelihood as we often can in BN learning.	✓ 1.00	Correct. One trick that often makes Bayes Net learning more efficient is our ability to optimize each CPD independently after we have obtained our expected counts. Markov Net learning cannot be decoupled, as the partition function couples all parameters in Markov Nets.
<input type="radio"/> less difficult as we do not need to account for the directed nature of factors as we do in a Bayes Net.		
<input type="radio"/> less difficult because we must separately optimize decoupled portions of the likelihood function in a Bayes Net, while we can optimize portions together in a Markov network.		
Total	1.00 / 1.00	

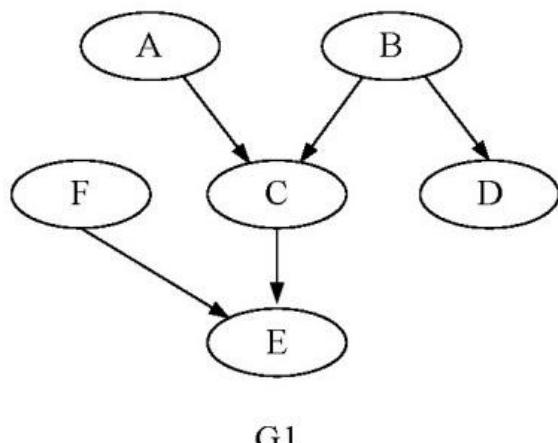
Feedback — Structure Scores

[Help Center](#)

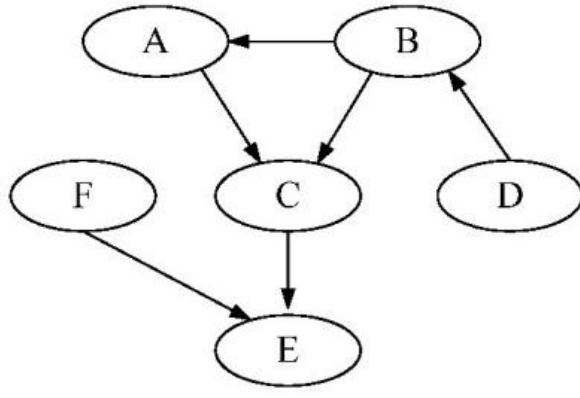
You submitted this quiz on **Tue 11 Jun 2013 1:44 PM PDT**. You got a score of **4.00** out of **5.00**. You can [attempt again](#), if you'd like.

Question 1

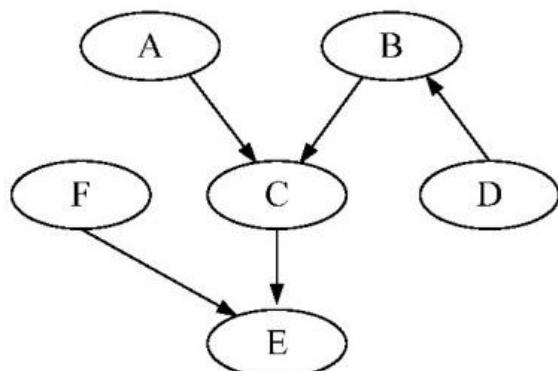
Likelihood Scores. Consider the following 4 graphs:



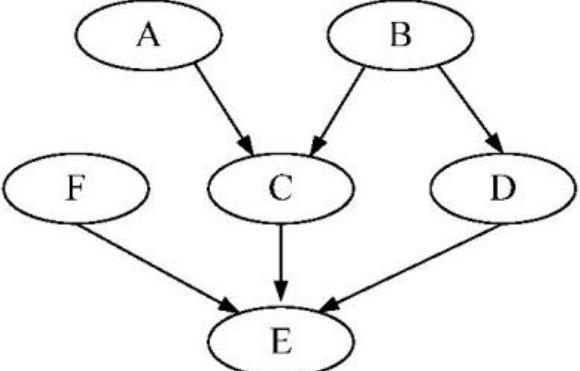
G1



G2



G3



G4

Which of the following statements about the likelihood scores of the different graphs is/are true?

You may choose more than 1 option (or none, if you think none are true).

Your Answer



$\text{Score}_L(G2 : D) \geq \text{Score}_L(G4 : D)$ for every dataset D

Score Explanation



0.25

$G2$ and $G4$ encode different sets of independence relations, neither of which is contained in the other. For example, $G2$ satisfies $D \perp E | B$, while $G4$ satisfies $A \perp B$. Hence,

there will be some datasets for which $\text{Score}_L(G2 : D)$ is larger, and others for which $\text{Score}_L(G4 : D)$ is larger.

- | | | |
|--|---|--|
| <input type="checkbox"/>
$\text{Score}_L(G1 : D) \geq \text{Score}_L(G4 : D)$
for every dataset D | ✓ 0.25 | $G4$ is an I-map of $G1$, that is, every independence relation that is in $G4$ is also in $G1$. Hence, $G4$ can represent all distributions that $G1$ can, and so its likelihood score will not be lower than that of $G1$. |
| <input checked="" type="checkbox"/>
$\text{Score}_L(G4 : D) \geq \text{Score}_L(G3 : D)$
for every dataset D | ✓ 0.25 | $G4$ is an I-map of $G3$, that is, every independence relation that is in $G4$ is also in $G3$. Hence, $G4$ can represent all distributions that $G3$ can, and so its likelihood score will not be lower than that of $G3$. |
| <input checked="" type="checkbox"/>
$\text{Score}_L(G2 : D) \geq \text{Score}_L(G1 : D)$
for every dataset D | ✓ 0.25 | $G2$ is an I-map of $G1$, that is, every independence relation that is in $G2$ is also in $G1$. Hence, $G2$ can represent all distributions that $G1$ can, and so its likelihood score will not be lower than that of $G1$. |

Total 1.00 /
1.00

Question 2

BIC Scores. Consider the same 4 graphs as in the previous question, but now think about the BIC score. Which of the following statements is/are true?

Your Answer	Score	Explanation
<input type="checkbox"/> $\text{Score}_{BIC}(G1 : D) \neq \text{Score}_{BIC}(G3 : D)$ for every dataset D	✓ 0.25	I-equivalent graphs have the same likelihood score and the same complexity (in terms of the number of independent parameters). Hence, they have the same BIC score.
<input type="checkbox"/> $\text{Score}_{BIC}(G4 : D) \geq \text{Score}_{BIC}(G1 : D)$ for every dataset D	✓ 0.25	While the likelihood score of $G4$ is always greater than or equals to that of $G1$, $G4$ is also a more complex graph. As the BIC score

is essentially the likelihood score minus a penalty for more complex models, these two components are in opposition. For large datasets in which G_4 is a much better model than G_1 , the likelihood score could dominate; conversely, for small datasets or for datasets that are generated by a distribution that G_1 can encode, the BIC score would be in G_1 's favor.



$\text{Score}_{BIC}(G_2 : D) \geq \text{Score}_{BIC}(G_1 : D)$
for every dataset D



0.25

While the likelihood score of G_2 is always greater than or equals to that of G_1 , G_2 is also a more complex graph. As the BIC score is essentially the likelihood score minus a penalty for more complex models, these two components are in opposition. For large datasets in which G_2 is a much better model than G_1 , the likelihood score could dominate; conversely, for small datasets or for datasets that are generated by a distribution that G_1 can encode, the BIC score would be in G_1 's favor.



$\text{Score}_{BIC}(G_1 : D) = \text{Score}_{BIC}(G_3 : D)$
for every dataset D



0.25

I-equivalent graphs have the same likelihood score, and have the same complexity (in terms of the number of independent parameters). Hence, they have the same BIC score.

Total

1.00 /

1.00

Question 3

Likelihood Guarantees. Consider graphs G_2 and G_3 . We have a dataset D generated from some probability distribution P , and the likelihood scores for G_2 and G_3 are $\text{Score}_L(G_2 : D)$ and $\text{Score}_L(G_3 : D)$, respectively. Let $\theta_{D,2}^*$ and $\theta_{D,3}^*$ be the maximum likelihood parameters

for each network, taken with respect to the dataset D . Now let $L(X : G, \theta)$ represent the likelihood of dataset X given the graph G and parameters θ , so

$$\text{Score}_L(G2 : D) = L(D : G2, \theta_{D,2}^*) \text{ and } \text{Score}_L(G3 : D) = L(D : G3, \theta_{D,3}^*).$$

Suppose that $L(D : G2, \theta_{D,2}^*) > L(D : G3, \theta_{D,3}^*)$. If we draw a new dataset E from the distribution P , which of the following statements can we guarantee? If more than one statement holds, choose the more general statement.

Your Answer

Score Explanation

None of the others

✓ 1.00

$\theta_{D,2}^*$ and $\theta_{D,3}^*$ correspond to the ML estimation from dataset D . Given the new dataset E , they might not be the ML estimation parameters any longer. Since the dataset D and E might not be sufficiently large enough to accurately characterize P , there is no guarantee on the relation of likelihood scores of the new dataset.

$L(E : G2, \theta_{D,2}^*) < L(E : G3, \theta_{D,3}^*)$

$L(E : G2, \theta_{D,2}^*) \neq L(E : G3, \theta_{D,3}^*)$

$L(E : G2, \theta_{D,2}^*) = L(E : G3, \theta_{D,3}^*)$

$L(E : G2, \theta_{D,2}^*) > L(E : G3, \theta_{D,3}^*)$

Total	1.00 /
	1.00

Question 4

Hidden Variables. Consider the case where the generating distribution has a naive Bayes structure, with an unobserved class variable C and its binary-valued children X_1, \dots, X_{100} . Assume that C is strongly correlated with each of its children (that is, distinct classes are associated with fairly different distributions over each X_i). Now suppose we try to learn a network structure directly on X_1, \dots, X_{100} , **without including C in the network**. What network structure are we likely to learn if we have 10,000 data instances, and we are using table

CPDs with the **likelihood** score as the structure learning criterion?

Your Answer	Score	Explanation
<input type="radio"/> Some connected network over X_1, \dots, X_{100} that is not fully connected nor empty.		
<input checked="" type="radio"/> A fully connected network, i.e., one with an edge between every pair of nodes.	1.00	In the generating distribution, for any pair of variables X_i, X_j , the trail $X_i \leftarrow C \rightarrow X_j$ is active. Thus, there are no independence relations of the form $X_i \perp X_j$. This means that when we try to use the likelihood score to learn a network structure over only the X_i 's, we will end up with a fully connected network.
<input type="radio"/> The empty network, i.e., a network consisting of only the variables but no edges between them.		
Total	1.00 / 1.00	

Question 5

Hidden Variables. Now suppose that we use the BIC score instead of the likelihood score in the previous question. What network structure are we likely to learn with the same 10,000 data instances?

Your Answer	Score	Explanation
<input type="radio"/> Some connected network over X_1, \dots, X_{100} that is not fully connected nor empty.		

empty.

The empty network, i.e., a network consisting of only the variables but no edges between them.

A fully connected network, i.e., one with an edge between every pair of nodes. ✖ 0.00 Even though a fully connected network may be the best representation for the true underlying distribution, we don't have enough data to learn it, and the BIC structure penalty will not allow the learning of a network with such high complexity, given only 10,000 instances.

Total 0.00 /
1.00

Feedback — Tree Learning and Hill Climbing

[Help Center](#)

You submitted this quiz on **Tue 11 Jun 2013 12:02 PM PDT**. You got a score of **4.00** out of **4.00**.

(The title of this quiz would fit right into a wilderness course.)

Question 1

Detective! You are a **detective** tracking down a serial robber who has already stolen from 1,000 victims, thereby giving you a large enough training set. No one else has been able to catch him (or her), but you are certain that there is a method (specifically, a Bayesian network) in this madness. You decide to model the robber's activity with a **tree-structured network** (meaning that each node has **at most** one parent): this network has (observed) variables such as the location of the previous crime, the gender and occupation of the previous victim, the current day of the week, etc., and a single unobserved variable, which is the location of the next robbery. The aim is to predict the location of the next robbery.

Unfortunately, you have forgotten all of your classical graph algorithms, but fortunately, you have a copy of *The Art of Computer Programming* (volume 4B) next to you. Which graph algorithm do you look up to help you find the optimal tree-structured network? Assume that the structure score we are using satisfies score decomposability and score equivalence.

Your Answer	Score	Explanation
<input type="radio"/> Finding an undirected spanning forest with the largest diameter (i.e., the longest distance between any pair of nodes).		
<input type="radio"/> Finding the shortest path between all pairs of points.		
<input type="radio"/> Finding the maximum flow through the graph, using any pair of observed nodes as source and sink.		
<input checked="" type="radio"/> Finding the maximum-weight undirected spanning forest (i.e., a set of undirected edges such that there is at most one path between any pair of nodes).	1.00	The tree-structured Bayesian network that we eventually want to construct is directed. However, if we have score equivalence, finding the maximum-weight undirected spanning forest is equivalent to finding the maximum-weight directed spanning forest and is easier to implement.
<input type="radio"/> Finding a directed spanning forest with the largest diameter (i.e., the longest distance between any pair of nodes).		

Total	1.00 /
	1.00

Question 2

***Recovering Directionality.** Once again, assume that our structure score satisfies score decomposability and score equivalence. After we find the optimal undirected spanning forest (containing n nodes), how can we recover the optimal directed spanning forest (and catch the robber)?

If more than one option is correct, pick the faster option; if the options take the same amount of time, pick the more general option.

Your Answer	Score	Explanation
<input checked="" type="radio"/> Pick any arbitrary root, and direct all edges away from it. This takes $O(n)$ time.	✓ 1.00	No matter which root we pick, the resulting trees are in the same I-equivalence class; in fact, there are no valid directed trees that cannot be obtained with this procedure. Because of score equivalence, it does not matter which root we pick.
<input type="radio"/> Evaluate all possible directions for the edges by iterating over them. This takes $O(2^n)$ time, since there are at most 2^n possible sets of edge directions in the spanning forest.		
<input type="radio"/> Pick any arbitrary direction for each edge, which takes $O(n)$ time. Because of score equivalence, all possible directed versions of the optimal undirected spanning forest have the same score, so this is valid.		
<input type="radio"/> Evaluate all possible directions for the edges. While there are at most 2^n possible sets of edge directions, we can exploit score decomposability to find the best directed spanning forest in $O(n)$ time.		
Total	1.00 /	
	1.00	

Question 3

***Augmenting Trees.** It turns out that the tree-structured network we learnt in the preceding questions was not sufficient to

apprehend the robber, allowing him to claim his 1001th victim. Not one to be discouraged, you decide to increase the expressiveness of your network.

Assume that we now want to learn a hybrid naive-Bayes/tree-structured network, where we have a single class variable C as well as the variables X_1, \dots, X_n . In this model, each X_i has C as a parent, and there is also a tree connecting the X_i 's; that is, each X_i , in addition to C , may also have up to one other parent X_j . For our baseline network G_0 , we are going to use the naive Bayes network, in which each X_i has only C as a parent. We are thus aiming to optimize the difference in likelihood scores $\text{Score}_L(G : \mathcal{D}) - \text{Score}_L(G_0 : \mathcal{D})$, where D is the training dataset.

If we use the appropriate spanning tree algorithm to find the optimal forest structure, what is the correct edge weight to use for $w_{j \rightarrow i}$? In these options, $M = 1001$ is the size of our training dataset, and $I_{\hat{P}}(\mathbf{A}, \mathbf{B})$ is the mutual information in the empirical distribution of the variables in set \mathbf{A} with the variables in set \mathbf{B} .

Your Answer	Score	Explanation
<input type="radio"/>		$M \cdot (I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; C)) - H_{\hat{P}}(X_i)$
<input type="radio"/>		$M \cdot I_{\hat{P}}(X_i; X_j, C)$
<input type="radio"/>		$M \cdot I_{\hat{P}}(X_i; X_j, C) - H_{\hat{P}}(X_i)$
<input checked="" type="radio"/>	1.00	$w_{j \rightarrow i} = \text{FamScore}_L(X_i X_j, C : D) - \text{FamScore}_L(X_i C : D)$. In the case of the likelihood score, this gives us $M(I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; C))$, since the entropy terms only depend on X_i and cancel each other out.
<input type="radio"/>		$M \cdot (I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; X_j))$
Total	1.00 / 1.00	

Question 4

Trees vs. Forests. Congratulations! Your hybrid naive-Bayes/tree-structured network managed to correctly predict where the criminal would be next, allowing the police to catch him (or her) before the 1002th victim got robbed. The grateful populace beg you to return to studying probabilistic graphical models.

While re-watching the video lectures, you begin to wonder if the algorithm we have been using to learn tree-structured networks can produce a forest, rather than a single tree. Assume that we use the likelihood score, and also assume that the maximum spanning forest algorithm breaks ties (between equal-scoring trees) arbitrarily. Which of the following is true? In this question, interpret "forest" to mean a set of two or more disconnected trees.

Your Answer	Score	Explanation
<input checked="" type="radio"/> It's theoretically possible for the algorithm to produce a forest. However, this will only occur in very contrived and	1.00	A forest will be produced only if we can partition the variables into two disjoint sets A and B , such that all edges $X_j \rightarrow X_i$ with either $X_i \in A, X_j \in B$ or $X_i \in B, X_j \in A$ have weight 0. This will be the case only if all variables in A are independent of the variables in B in the empirical distribution. While this is not impossible, it is very unlikely to happen in practice.

unrealistic circumstances, not in practice.

This algorithm will never produce a forest, since there will always be a tree that has strictly higher score.

It's possible for the algorithm to produce a forest, since there are cases in which a forest will have a higher score than any tree.

It's possible for the algorithm to produce a forest even though trees will always score more highly, since the algorithm need not find the structure that is globally optimal (relative to the likelihood score).

Total	1.00 / 1.00
-------	----------------

Feedback — Learning with Incomplete Data

[Help Center](#)

You submitted this quiz on **Mon 17 Jun 2013 12:17 PM PDT**. You got a score of **4.00** out of **4.00**.

Question 1

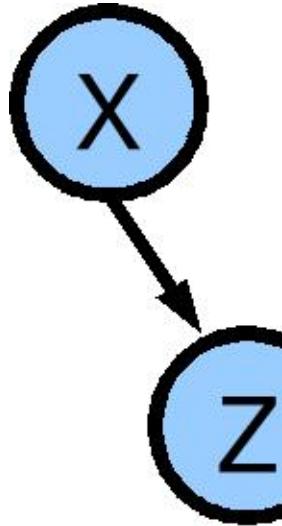
Missing At Random. Suppose we are conducting a survey of job offers and salaries for Stanford graduates. We already have the major of each of these students recorded, so in the survey form, each graduating student is only asked to list up to two job offers and salaries he/she received. Which of the following scenarios is/are missing at random (MAR)?

Your Answer	Score	Explanation
<input type="checkbox"/> One of the students submitted his name as "ROBERT"); foreach student in SURVEYS{ if(student.salary>80000) REMOVE student;" causing all students who submitted surveys with a salary entry over 80,000 to be removed from the results.	✓ 0.25	This is not MAR because whether the data is missing depends on the missing value (the salary).
<input checked="" type="checkbox"/> The person recording the information accidentally lost some of the completed survey forms.	✓ 0.25	We say data is MAR if whether the data is missing is independent of the missing values themselves given the observed values. This is MAR because whether the data is missing depends on random loss that does not correspond to salary. In fact, this is MCAR.
<input type="checkbox"/> Students who accepted a low-salaried job offer tended not to reveal it.	✓ 0.25	This is not MAR because whether the data is missing depends on the missing value (the salary).
<input checked="" type="checkbox"/> The person recording the information didn't care about humanities students and neglected to record their salaries.	✓ 0.25	We say data is MAR if whether the data is missing is independent of the missing values themselves given the observed values. Whether the data is missing was determined by major, which is observed.
Total	1.00 /	

1.00

Question 2

Computing Sufficient Statistics. Given the network and data instances shown below, how do we compute the expected sufficient statistics for a particular value of the parameters?



Data Instances

$(x_0, ?, ?)$

$(x_0, y_0, ?)$

$(x_1, y_1, ?)$

$(x_0, y_1, ?)$

$(?, y_0, ?)$

Your Answer

Score Explanation



$\bar{M}[x_0, y_0, z_0] = P(y_0, z_0 | x_0, \theta) + P(z_0 | x_0, y_0, \theta) + P(z_0 | x_1, y_1, \theta) +$

$\bar{M}[x_0, y_0, z_0] = P(z_0 | x_0, \theta) + P(z_0 | x_0, y_0, \theta) + P(z_0 | y_0, \theta)$



$\bar{M}[x_0, y_0, z_0] = P(y_0, z_0 | x_0, \theta) + P(z_0 | x_0, y_0, \theta) + P(x_0, z_0 | y_0, \theta)$

✓ 1.00

The expected sufficient statistics for the assignment (x_0, y_0, z_0) are the sum of the probability that each instance is consistent with that assignment (which is 0 for all inconsistent instances).

$\bar{M}[x_0, y_0, z_0] = 3$

Total

1.00 /

1.00

Question 3

Likelihood of Observed Data. In a Bayesian Network with partially observed training data, computing the likelihood of observed data for a given set of parameters...

Your Answer

Score

Explanation

requires probabilistic inference, AS IN the case of fully observed data.

cannot be achieved by probabilistic inference, while it CAN in the case of fully observed data.

requires probabilistic inference, while it DOES NOT in the case of fully observed data.

 1.00

With missing data, inference is required to complete the expected sufficient statistics (ESS) for the expected likelihood function. Thus, inference is not needed to compute the ESS in the case of fully observed data.

Total

1.00 /

1.00

Question 4

PGM with latent variables. Adding hidden variables to a model can significantly increase the expressiveness of a model. However, there are also some issues that arise when we try to add hidden variables.

For which of these problems can we learn a reasonable model by simply choosing the parameters that maximize training likelihood? Assume that all variables, hidden or (partially) observed, are discrete and follow a table CPD. You may choose more than one option (or none, if

you think none apply).

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Given a fixed set of edges, learning the parameters in the table CPDs of observed nodes that have hidden nodes as parents.	✓ 0.25	This is a standard parameter estimation with missing data problem that we can solve with methods such as the EM algorithm. While using only training likelihood runs the risk of overfitting, given a large enough training set, the parameters found are still likely to perform reasonably well.
<input checked="" type="checkbox"/> Given a fixed set of edges, learning the parameters in the table CPDs of each hidden node.	✓ 0.25	This is a standard parameter estimation with missing data problem that we can solve with methods such as the EM algorithm. While using only training likelihood runs the risk of overfitting, given a large enough training set, the parameters found are still likely to perform reasonably well.
<input type="checkbox"/> Choosing the number of hidden variables to add to the graphical model.	✓ 0.25	Training set likelihood will always increase with the number of hidden variables we add, so we will end up with infinitely many hidden variables. This is because for any two integers $N > n$, the set of graphs with N hidden nodes is more expressive; it can represent whatever probability distributions the set of graphs with n hidden nodes can, and possibly more.
<input type="checkbox"/> Choosing which edges involving only the observed nodes to add to the graph.	✓ 0.25	Training set likelihood will always increase with every edge we add, so we will end up with a complete graph over all variables. This is because the addition of an edge always increases the expressiveness of a graph, in terms of the probability distributions that it can possibly represent.
Total	1.00 / 1.00	

Feedback — Expectation Maximization

[Help Center](#)

You submitted this quiz on **Tue 18 Jun 2013 11:37 AM PDT**. You got a score of **7.00** out of **7.00**.

Question 1

Bayesian Clustering using Normal Distributions. Suppose we are doing Bayesian clustering with K classes, and multivariate normal distributions as our class-conditional distributions. Let $\mathbf{X} \in \mathbf{R}^n$ represent a single data point, and $C \in \{1, 2, \dots, K\}$ its unobserved class. Which of the following statement(s) is/are always true in the general case?

Your Answer

Score Explanation



$P(\mathbf{X}|C = c) \sim N(0, I_n) \quad \forall c \in \{1, 2, \dots, K\}$,
where 0 is the all-zero vector and I_n is the $n \times n$ identity matrix.



$P(\mathbf{X}) \sim N(\mu, \Sigma)$, for some parameters μ and Σ that represent the overall distribution of the data.



$P(\mathbf{X}|C = c) \sim N(\mu_c, \Sigma_c) \quad \forall c \in \{1, 2, \dots, K\}$,
for some class-specific parameters μ_c and Σ_c that represent the distribution of data coming from the class c .



1.00

This is the definition of having a multivariate normal distribution as the class-conditional distribution: given the class from which the data point came from, the distribution of the data point follows a multivariate normal distribution with mean and covariance parameters specific to its particular class.

$X_i \perp X_j \quad \forall i, j \in \{1, 2, \dots, n\}, i \neq j$, i.e., for any given data point, knowing one coordinate does not give us any information about another coordinate.

$X_i \perp X_j | C \quad \forall i, j \in \{1, 2, \dots, n\}, i \neq j$, i.e., for any given data point, if we know which class the data point comes from, then knowing one

coordinate does not give us any information about another coordinate.

Total	1.00 /
	1.00

Question 2

Hard Assignment EM. Continuing from the previous question, let us now fix each class-conditional distribution to have the identity matrix as its covariance matrix. If we use hard-assignment EM to estimate the class-dependent mean vectors, which of the following can we say about the resulting algorithm?

Your Answer	Score	Explanation
<input checked="" type="radio"/> It is equivalent to running standard k-means clustering with K clusters.	✓ 1.00	You will always assign vertices to their closest (in Euclidean distance) cluster centroid, just as in k -means.
<input type="radio"/> It is an an algorithm that cannot be viewed as an instance of k-means.		
<input type="radio"/> It is an instance of k-means, but using a different distance metric rather than standard Euclidean distance.		

Total	1.00 /
	1.00

Question 3

***Hard Assignment EM.** Now suppose that we fix each class-conditional distribution to have the same diagonal matrix D as its covariance matrix, where D is **not** the identity matrix. If we use hard-assignment EM to estimate the class-dependent mean vectors, which of the following can we say about the resulting algorithm?

Your Answer	Score	Explanation
-------------	-------	-------------

It is an algorithm that cannot be viewed as an instance of k-means.

It is equivalent to running standard k-means clustering with K clusters.

It is an instance of k-means, but using a different distance metric rather than standard Euclidean distance.

✓ 1.00 You will always assign vertices to their closest cluster centroid, just as in k -means. But here the definition of ``closest'' is skewed by the covariance matrix so that it does not equally depend on each dimension and is thus not a Euclidean distance.

Total	1.00 /
	1.00

Question 4

EM Running Time. Assume that we are trying to estimate parameters for a Bayesian network structured as a binary (directed) tree (**not** a polytree) with n variables, where each node has at most one parent. We parameterize the network using table CPDs. Assume that each variable has d values. We have a data set with M instances, where some observations in each instances are missing. What is the tightest asymptotic bound you can give for the worst case running time of EM on this data set for K iterations? In this and following questions, concentrate on the EM part of the learning algorithm only. You don't need to consider the running time of additional steps if the full learning algorithm needs any.

Your Answer

Score **Explanation**

Can't tell using only the information given

$O(KMn^2d^2)$

$O(KMn^2d)$

- 1.00 At each iteration and for every instance, it is required to run exact inference over the given network. Using clique-tree calibration, the cost of inference is the number of cliques (n) times the size of the clique potential which is d^2 (due to the tree-structure of the network each clique can have only 2 variables in its scope).

Total 1.00 /
1.00

Question 5

EM Running Time. Use the setting of question 4, but now we assume that the network is a [polytree](#), in which some variables have several parents. What is the cost of running EM on this data set for K iterations?

- | Your Answer | Score | Explanation |
|---|----------------|---|
| <input type="radio"/> $O(KMnd^2)$ | | |
| <input type="radio"/> $O(KMn^2d^2)$ | | |
| <input checked="" type="radio"/> Can't tell using only the information given. | 1.00 | We cannot tell because now the factors in the clique tree can be considerably larger than d^2 (but we do not know how much larger they might be). |
| <input type="radio"/> $O(KMn^2d)$ | | |
| Total | 1.00 /
1.00 | |

Question 6

***Optimizing EM.** Now, going back to the setting of the question 4 (each node has at most one parent), assume that we are in a situation where at most 2 variables in each data instance are unobserved (not necessarily the same 2 in each instance). Can we implement EM more efficiently? If so, which of the following reduced complexities can you achieve?

- | Your Answer | Score | Explanation |
|----------------------------------|-------|-------------|
| <input type="radio"/> $O(KMd^2)$ | | |

No
computational savings can be achieved.

$O(KMnd^2)$

- $O(K(M+n)d^2)$ ✓ 1.00 In this case, the cost of the E-step is Md^2 since we can easily compute the probabilities of each possible completion of instances when only up to 2 variables are missing. We can use this to compute the expected sufficient statistics (where we will be summing over the M instances and up to d^2 possible completed instances). The cost of the M-step will be nd^2 (it is equal to the number of parameter values computed).

Total	1.00 /
	1.00

Question 7

*Optimizing EM. Still in the setting of the question 4, now assume that we are in a situation where at most 2 variables in each data instance are unobserved, but it's the same 2 each instance. Can we implement EM more efficiently? If so, which of the following reduced complexities can you achieve?

Your Answer

Score

Explanation

$O(KMnd^2)$

$O(K(M+n)d^2)$

No
computational savings can be achieved.

- $O(KMd^2)$ ✓ 1.00 In this case, most of the graph is conditionally independent of the unobserved variables, so we can restrict our EM process to the sub-graph consisting of the unobserved variables and their Markov blankets, and fix parameters for the rest of the network once at the beginning, using standard MLE. Thus, the cost of updating a small subset of the parameters at each M-step will be no more than $O(d^2)$.

In the E-step, for each instance, we will run inference over a small subset of variables at a cost of d^2 per instance.

Accordingly, the cost of the E-step will be Md^2 .

Total	1.00 /
	1.00

Feedback — Final Exam - DO NOT CLICK UNTIL READY

You submitted this quiz on **Thu 20 Jun 2013 1:55 PM PDT**. You got a score of **15.43** out of **20.00**.

[Help Center](#)

Question 1

Factor marginalization. Let X, Z be binary variables, and let Y be a variable that takes on values 1, 2, or 3.

If $\phi(X, Y, Z)$ is the factor shown below, compute the entries of the factor $\psi(Y, Z) = \sum_X \phi(X, Y, Z)$, giving your answer according to the ordering of assignments to variables as shown below.

As before, you may separate the 4 entries of the factor by new lines.

X	Y	Z	$\phi(X, Y, Z)$
1	1	1	68
1	1	2	95
1	2	1	65
1	2	2	63
1	3	1	57
1	3	2	5
2	1	1	40
2	1	2	40
2	2	1	14
2	2	2	78
2	3	1	16
2	3	2	89

Y	Z	$\psi(Y, Z)$
1	1	?
1	2	?
2	1	?
2	2	?
3	1	
3	2	

You entered:

108
135

Your Answer

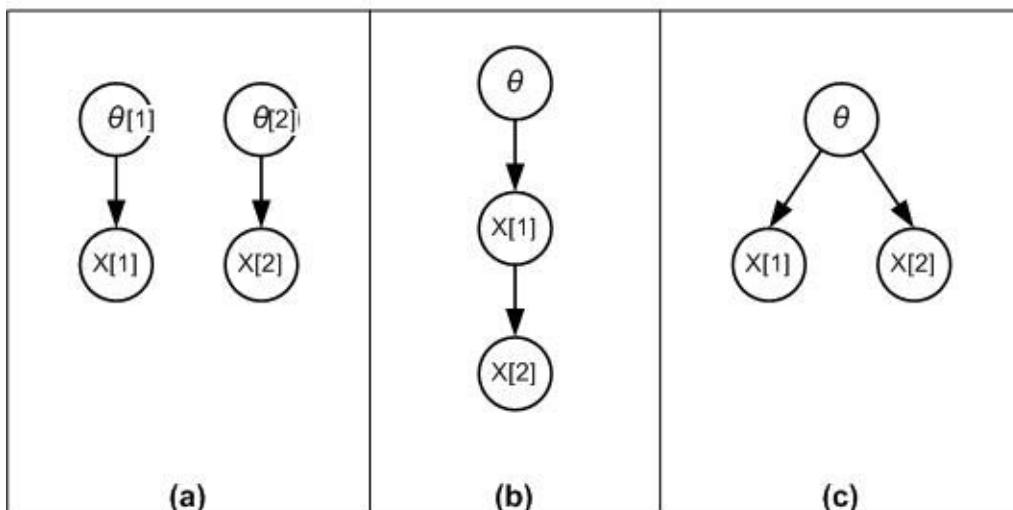
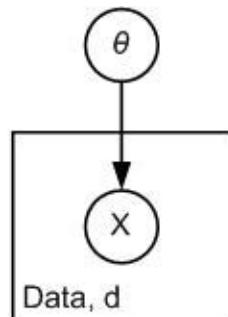
Score

Explanation

108	✓	0.25
135	✓	0.25
79	✓	0.25
141	✓	0.25
Total	1.00 / 1.00	

Question 2

Grounded Plates. Which of the following is a valid grounded model for the plate shown? You may select 1 or more options (or none of them, if you think none apply).



Your Answer	Score	Explanation
<input checked="" type="checkbox"/> (a) -- watch out, options are not in order	✗ 0.00	(a) is incorrect because only the variables in the plate get replicated when the plate model is grounded, and θ is not in the plate.
<input checked="" type="checkbox"/> (c) -- watch	✓ 0.33	(c) is correct because θ is outside the plate and has edges

out, options are not in order		from it to the nodes in the plate, and none of the nodes within the plate share edges with each other.
<input type="checkbox"/> (b) -- watch out, options are not in order	✓ 0.33	(b) is incorrect because there are no arrows connecting nodes within the plate.
Total	0.67 / 1.00	

Question Explanation

(c) is correct because θ is outside the plate and has edges from it to the nodes in the plate, and none of the nodes within the plate share edges with each other.

Question 3

***Dual Decomposition Slaves.** Suppose you wish to perform MAP inference on a pairwise MRF that is an $n \times n$ grid. Which of the following decompositions of the MAP problem into slaves could you use with the dual-decomposition algorithm? You may select 1 or more options (or none of them, if you think none apply).

Hint: You need to make sure that all factors are accounted for at least once in the slaves, and that each slave can be solved efficiently.

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Slaves that each consist of a single factor in the network.	✓ 0.17	This decomposition is fine.
<input checked="" type="checkbox"/> 4 slaves where each slave is an $n/2 \times n/2$ section of the grid, such that the 4 sections when put together, cover the grid completely.	✗ 0.00	You wouldn't use this decomposition because the slaves can't be efficiently maximized locally.
<input checked="" type="checkbox"/> Slaves that are spanning trees of the grid; the trees, in combination, include all the edges in the grid.	✓ 0.17	This decomposition is fine; running MAP on trees is fast, because the size of each clique is small.
<input type="checkbox"/> Slaves that consist of disjoint 2x2 squares.	✓ 0.17	You wouldn't use this decomposition because all factors need to be accounted for.
<input type="checkbox"/> One slave for each row and column of the grid.	✗ 0.00	This decomposition is fine.
<input type="checkbox"/> Slaves that are spanning trees of the grid; the trees, in combination,	✓ 0.17	You wouldn't use this decomposition because all factors need to be accounted

do not include all the edges in the grid.

Total	0.67 /
	1.00

Question 4

Metropolis-Hastings Algorithm. Assume we have an $n \times n$ grid-structured MRF over the variables $X_{i,j}$. Let $\mathbf{X}_i = \{\mathbf{X}_{i,1}, \dots, \mathbf{X}_{i,n}\}$ and $\mathbf{X}_{-i} = \mathcal{X} - \mathbf{X}_i$. Consider the following instance of the Metropolis-Hastings algorithm: at each step, we take our current assignment \mathbf{x}_{-i} and use exact inference to compute the conditional probability $P(\mathbf{X}_i | \mathbf{x}_{-i})$. We then sample \mathbf{x}'_i from this posterior distribution, and use that as our proposal. What is the correct acceptance probability for this proposal?

Hint: what is the relationship between this and Gibbs sampling?

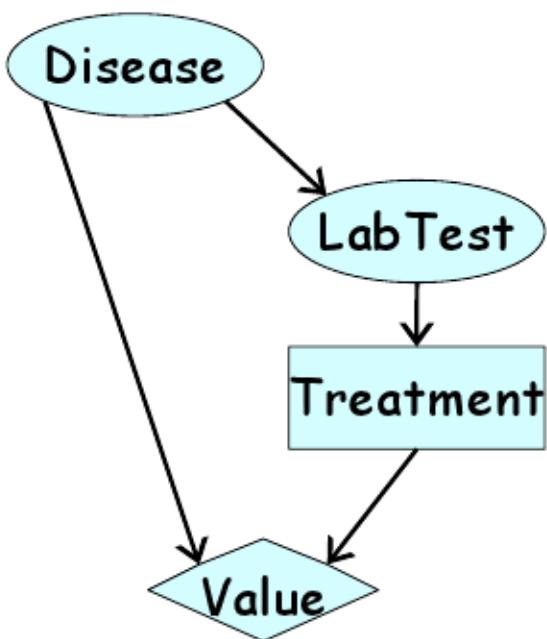
Your Answer	Score	Explanation
<input type="radio"/> $P(\mathbf{x}_i \mathbf{x}_{-i})/P(\mathbf{x}'_i \mathbf{x}_{-i})$		
<input type="radio"/> $P(\mathbf{x}'_i, \mathbf{x}_{-i})/P(\mathbf{x}_i, \mathbf{x}_{-i})$		
<input type="radio"/> $P(\mathbf{x}'_i \mathbf{x}_{-i})/P(\mathbf{x}_i \mathbf{x}_{-i})$		
<input checked="" type="radio"/> 1	✓ 1.00	
<input type="radio"/> $P(\mathbf{x}_i, \mathbf{x}_{-i})/P(\mathbf{x}'_i, \mathbf{x}_{-i})$		
Total	1.00 / 1.00	

Question Explanation

This is (block) Gibbs sampling, where we sample multiple variables simultaneously from their conditional distribution given all other variables. Gibbs sampling is an instance of MH that has an acceptance probability of 1.

Question 5

***Value of Information.** In the influence diagram on the right, when does performing LabTest have value? That is, when would you want to observe the LabTest variable?



Hint: Think about when information is valuable in making a decision.

Your Answer	Score	Explanation
<input type="radio"/> When there is some disease d such that $\text{argmax}_t V(d, t) \neq \text{argmax}_t \sum_d P(d)V(d, t)$		
<input type="radio"/> When there is some treatment t such that $V(D, t)$ is different for different diseases D .		
<input type="radio"/> When $P(D L)$ is different from $P(D)$.		
<input checked="" type="radio"/> When there is some lab value l such that $\text{argmax}_t \sum_d P(d l)V(d, t) \neq \text{argmax}_t \sum_d P(d)V(d, t)$	✓ 1.00	
Total	1.00 / 1.00	

Question Explanation

There is no value in information (observing LabTest) unless the information changes a decision (of Treatment in this case).

Question 6

***Multiplexer CPDs.** What is the form of the independence that is implied by the multiplexer CPD and that we used in our derivation of the posterior over the parameters of the simple Bayesian Network $x \rightarrow y$? (i.e. the factorization of $P(\theta_x, \theta_{Y|x^1}, \theta_{Y|x^0} | \mathbf{D})$). Recall that a CPD is defined as a multiplexer if it has the structure $P(Y|A, Z_1, \dots, Z_k) = \mathbf{I}\{Y = Z_a\}$ where the values of A are the natural numbers 1 through k . **Also note that the answer is specific to multiplexer**

CPDs and is not implied by the graph structure alone.

Your Answer	Score	Explanation
<input type="radio"/> $\theta_{Y x^0} \perp \theta_{Y x^1} X$		
<input type="radio"/> $\theta_{Y X} \perp X \theta_X$		
<input type="radio"/> $\theta_{Y x^0}, \theta_{Y x^1} \perp X$		
<input checked="" type="radio"/> $\theta_{Y X} \perp \theta_X$	✖	0.00
<input type="radio"/> $\theta_{Y x^0} \perp \theta_{Y x^1} X, Y$		
Total	0.00 / 1.00	

Question Explanation

The answer is $\theta_{Y|x^0} \perp \theta_{Y|x^1} | X, Y$. This solution is implied by the multiplexer CPD but not by the graph structure as it relies on the CPD being independent based on which x value is active (for those with the Probabilistic Graphical Models textbook, this is similar to example 5.19 on page 174). The other options may hold, but they are implied by the graph structure and not by the specific CPD.

Question 7

***Score Consistency.** Assume that the dataset D has m examples, each drawn independently from the distribution P^* , for which the graph G^* is a perfect map. What do we mean when we say that the BIC score $\text{Score}_{BIC}(G : D)$, measured with respect to D , is **consistent**?

Hint: We are looking for a definition that will always be true, not just probably be true.

Your Answer	Score	Explanation
<input type="radio"/> As $m \rightarrow \infty$, with probability 1 we will draw a dataset D from P^* such that the inequality $\text{Score}_{BIC}(G^* : D) > \text{Score}_{BIC}(G : D)$ holds for all other graphs $G \neq G^*$.		
<input type="radio"/> As $m \rightarrow \infty$, no matter which examples were drawn from P^* into the dataset D , the inequality $\text{Score}_{BIC}(G^* : D) > \text{Score}_{BIC}(G : D)$ will always be true for all other graphs $G \neq G^*$.		
<input checked="" type="radio"/> As $m \rightarrow \infty$, with probability 1 we will draw a dataset D from P^* such that the inequality $\text{Score}_{BIC}(G^* : D) > \text{Score}_{BIC}(G : D)$ holds for all other graphs G which are not I-equivalent to G^* .	✓	1.00

- As $m \rightarrow \infty$, no matter which examples were drawn from P^* into the dataset D , the inequality $\text{Score}_{BIC}(G^* : D) > \text{Score}_{BIC}(G : D)$ will always be true for all graphs G which are not I-equivalent to G^* .

Total	1.00 / 1.00
-------	----------------

Question Explanation

It is still possible, though extremely unlikely for large m , to draw a pathological dataset D that does not reflect the underlying distribution P^* at all. Hence, any statement we make about consistency can only be about the probability of obtaining a dataset D for which the desired consistency conditions hold.

Now, remember that consistent scoring metrics like the BIC score allow us to uniquely identify the correct I-equivalence class (in this case, that of G^*) as the number of data points grows large. For this, we need strict inequality in the BIC scores of G^* versus other graphs not in its I-equivalence class.

Question 8

EM and Convergence. When checking for the convergence of the EM algorithm, we can choose to measure changes in either the log-likelihood function or in the parameters. For a generic application, we typically prefer to check for convergence using the log-likelihood function. However, this is not always the case, especially when the values of the parameters are important in and of themselves. In which situations would we also be concerned about reaching convergence in terms of the parameters? Do not worry about the implementation details in the following models.

Your Answer	Score	Explanation
<input type="checkbox"/> We are trying to transcribe human speech by building a Hidden Markov Model (HMM) and learning its parameters with the EM algorithm. The end-goal is correctly transcribing raw audio input into words.	✓ 0.14	In this application, we are learning parameters to improve performance (i.e., transcription accuracy) and are not as interested in the parameters in and of themselves.
<input type="checkbox"/> We are building a graphical model for medical diagnosis, where nodes can represent symptoms, diseases,	✓ 0.14	In this application, we are learning parameters to improve performance (i.e., medical diagnosis), and are not as interested in the parameters in and of themselves.

predisposing factors, and so on. Our only aim is to maximize our chances of correctly predicting diseases that patients are suffering from.

- | | | |
|--|---|--|
| <input checked="" type="checkbox"/> We have a graphical model in which each node is a superpixel, and we are using EM to learn the parameters that specify the relations between superpixels. Our end-goal is to build an image segmentation pipeline that is highly accurate. | ✗ 0.00 | In this application, we are learning parameters to improve performance (i.e., image segmentation accuracy) and are not as interested in the parameters in and of themselves. |
| <input type="checkbox"/> We are building a graphical model for medical diagnosis, where nodes can represent symptoms, diseases, predisposing factors, and so on. This system will not be deployed in the clinic; our only aim is to understand how various predisposing factors can interact with each other in increasing disease risk. | ✗ 0.00 | In this application, we are interested in the actual value of the parameters, and we are using the data likelihood as a means of estimating these parameters accurately. |
| <input type="checkbox"/> We have a graphical model in which each node represents an object part, and we are using EM to learn the parameters that specify the relations between object parts. Our end-goal is to build an image classification system that can accurately recognize the image as one of several known objects. | ✓ 0.14 | In this application, we are learning parameters to improve performance (i.e., object recognition accuracy) and are not as interested in the parameters in and of themselves. |
| <input type="checkbox"/> We are building a graphical model to represent a biological network, where each | ✗ 0.00 | In this application, we are interested in the actual value of the parameters, and we are using the data likelihood as a means of estimating these parameters accurately. |

node corresponds to a gene. We want to learn the interactions between genes by finding the parameters that maximize the likelihood of a given training dataset of gene expression measurements. The interactions we find will then be further studied by biologists.

- | | | |
|---|---|--|
| <input type="checkbox"/> We are trying to better understand high-energy physics by using a graphical model to analyze time-series data from particle accelerators. The hope is to elucidate the types of interactions between different particle types. | ✖ 0.00 | In this application, we are interested in the actual value of the parameters, and we are using the data likelihood as a means of estimating these parameters accurately. |
|---|---|--|

Total	0.43 / 1.00
-------	----------------

Question 9

Parameter Estimation with Missing Data. The process of learning Bayesian Network parameters with missing data (partially observed instances) is more difficult than learning with complete data for which of the following reasons? You may select one or more options, or none if you think none apply.

Your Answer	Score	Explanation
<input type="checkbox"/> We require more training data, because we must throw out all incomplete instances.	✓ 0.25	We do not need to throw out incomplete instances; we can use procedures with inference to leverage our incomplete instances in estimating the parameters.
<input checked="" type="checkbox"/> Because there can be multiple optimal values, we	✖ 0.00	While there may be cases where we want to run from different initializations in order to find a "good" set of parameters, it is generally not our goal to find all optima

must always run our learning algorithm multiple times from different initializations to make sure we find ALL of them.

nor can we ensure that our parameters are globally optimal.

- While there is still always a single optimal value for the parameters, it can only be found using an iterative method. ✓ 0.25 There can be more than a single optimal value for the parameters.

- We lose local decomposition, whereby each CPD can be estimated independently. ✓ 0.25 When all values are observed, we can ignore the values of nodes that are not directly connected, but when there are unobserved values, the CPD of parent nodes can affect the optimal parameters in child nodes.

Total 0.75 /
1.00

Question 10

Optimality of Hill Climbing. Jack and Jill come up to you one day with a worried look on their face. "All this while we've been climbing hills, trying to improve upon our graph structure," they say. "We've been considering edge deletions, reversals, and additions at each step. Today, we found that no single edge deletion, reversal, or addition could give us a higher-scoring structure. Are we guaranteed that our current graph is the best graph structure?" What should you tell them? You may assume that their dataset is sufficiently large, and that your answer should hold for a general graph.

Your Answer	Score	Explanation
<input type="radio"/> Yes - greedy hill-climbing provably finds the true graph structure, provided our dataset is large enough.		
<input checked="" type="radio"/> No - greedy hill-climbing will find only local maxima of the scoring function with	✓ 1.00	During greedy hill-climbing, we only make moves that will improve our current structure score. This gets us to a local maximum of the scoring function, but need not necessarily get us to the global optimum.

respect to our available moves. While it might find the true graph structure on occasion, we cannot guarantee this.

Yes, but only if we use random restarts and tabu search.

No - greedy hill-climbing can never find the true graph structure, only local maxima of the scoring function with respect to our available moves.

No - greedy hill-climbing will only find the true graph structure if we restrict the number of parents for each node to at most 2.

Yes, but only if we extend our range of available moves to allow for pairs of edges to be changed simultaneously.

Total 1.00 /
1.00

Question Explanation

During greedy hill-climbing, we only make moves that will improve our current structure score. This gets us to a local maximum of the scoring function, but need not necessarily get us to the global optimum.

Question 11

***Latent Variable Cardinality.** Assume that we are doing Bayesian clustering, and want to select the cardinality of the hidden class variable. Which of these methods can we use? Assume that the structure of the graph has already been fixed. You may choose more than one option (or none, if

you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> Training several models, each with a different cardinality for that hidden variable. For each model, we choose the (table CPD) parameters that maximize the likelihood on the training set . We then pick the model with the highest training set likelihood.	✓ 0.20	Training set likelihood will always increase with the cardinality of each hidden variable, so we will always end selecting the model which has the largest cardinality.
<input checked="" type="checkbox"/> Training several models, each with a different cardinality for that hidden variable. For each model, we choose the (table CPD) parameters that maximize the likelihood on the training set . We then pick the model with the highest likelihood on a held-out validation set .	✓ 0.20	This is the closest we can get to measuring the performance of each model on the test set (in the sense of maximizing data likelihood), without actually using the test set.
<input checked="" type="checkbox"/> Training several models, each with a different cardinality for that hidden variable. For each model, we choose the (table CPD) parameters that maximize the	✓ 0.20	If the purpose of the model is to eventually perform this external task, then measuring its performance on the true task is arguably more effective than simply measuring data likelihood. Since the external dataset was previously unseen, we will not run into the problem of always picking the model with the highest cardinality.

likelihood on the **training set**. We then pick the model that performs the best on some external evaluation task, using a **held-out validation set**.

For example, say we are using Bayesian clustering to classify customers visiting an online store, with the aim of giving class-specific product recommendations.

We could run each model in an alpha-beta testing framework (where different customers may see the result of different models), and measure the percentage of customers that end up purchasing what each model recommends.

-
- If we have relevant prior knowledge, we can simply use this to set the cardinality by hand.
- ✓ 0.20 In some cases (e.g., we are modeling clothing preference in a population, and we introduce a hidden variable that we hope will pick up the differences between genders), we have sufficient prior knowledge to choose a reasonable value for the cardinality of our hidden variable.

-
- Training several models, each with a different cardinality for that hidden variable. For each model, we choose the (table CPD) parameters that
- ✓ 0.20 We never use the test set for model selection, only for evaluation after the model and its parameters have been picked.

maximize the likelihood on the **training set**. We then pick the model with the highest **test set** likelihood.

Total	1.00 /
	1.00

Question 12

EM Stopping Criterion. When learning the parameters $\theta \in \mathbf{R}^n$ of a graphical model using the EM algorithm, an important design decision is choosing when to stop training. Let $\ell_{\text{Train}}(\theta)$, $\ell_{\text{Valid}}(\theta)$, and $\ell_{\text{Test}}(\theta)$ be the log-likelihood of the parameters θ on the training set, a held-out validation set, and the test set, respectively. Let θ^t be the parameters at the t -th iteration of the EM algorithm. We can denote the change in the dataset log-likelihoods at each iteration with $\Delta\ell_{\text{Train}}^t = \ell_{\text{Train}}(\theta^t) - \ell_{\text{Train}}(\theta^{t-1})$ and the corresponding analogues for the validation set and the test set. Likewise, let $\Delta\theta^t = \theta^t - \theta^{t-1}$ be the vector of changes in the parameters at time step t .

Which of the following would be reasonable conditions for stopping training at iteration t ? You may choose more than one option.

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> $\ \Delta\theta^t\ _2^2$ becomes small, i.e., it falls below a certain tolerance $\epsilon > 0$. Note: The ℓ_2 norm, also known as the Euclidean norm, is defined for any vector $x \in \mathbf{R}^n$ as $\ x\ _2^2 = \sum_{i=1}^n x_i^2$.	✓ 0.25	This is likely to return parameters that lie near to a local maximum of the log-likelihood function on the training set. (In practice, this quantity is very rarely exactly 0 at the convergence point, due to issues with floating point inaccuracies and a potentially infinite number of steps required to converge to the exact value of the local maximum.)
<input type="checkbox"/> $\Delta\ell_{\text{Test}}$ becomes small, i.e., it falls below a certain tolerance $\epsilon > 0$	✓ 0.25	We never use the test set for parameter learning / selection. Instead, the test set should only be used for evaluation (after we've selected our parameters based on the training and/or validation sets).

<input checked="" type="checkbox"/> $\Delta\ell_{\text{Valid}}^t$ becomes negative.	✓ 0.25	Stopping when the log-likelihood starts to decrease on a held-out validation set is a good way to alleviate the problem of overfitting parameters to the training set.
<input type="checkbox"/> $\Delta\ell_{\text{Train}}$ becomes negative.	✓ 0.25	This condition will never be met: the EM algorithm is guaranteed to monotonically increase log-likelihood on the training set, until it reaches a local maximum. When it reaches a local maximum, the EM algorithm will not make further changes to the parameters, so $\Delta\ell_{\text{Train}} = 0$. (In practice, $\Delta\ell_{\text{Train}}$ is very rarely exactly 0 at the convergence point, due to issues with floating point inaccuracies and a potentially infinite number of steps required to converge to the exact value of the local maximum.)
Total	1.00 / 1.00	

Question 13

EM Parameter Selection. Once again, we are using EM to estimate parameters of a graphical model. We use n random starting points $\{\theta_i^0\}_{i=1,2,\dots,n}$, and run EM to convergence from each of them to obtain a set of candidate parameters $\{\theta_i\}_{i=1,2,\dots,n}$. We wish to select one of these candidate parameters for use. As in the previous question, let $\ell_{\text{Train}}(\theta)$, $\ell_{\text{Valid}}(\theta)$, and $\ell_{\text{Test}}(\theta)$ be the log-likelihood of the parameters θ on the training set, a held-out validation set, and the test set, respectively.

Which of the following methods of selecting final parameters θ would be a reasonable choice?

You may pick more than one option.

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Pick $\theta = \operatorname{argmax}_{i=1,2,\dots,n} \ell_{\text{Valid}}(\theta_i)$.	✓ 0.25	Given enough data to create a validation set that is separate from both the training and test set, this could give better results than (a), as the problem of overfitting to the training set would be alleviated. It is not guaranteed that creating a held-out validation set (as opposed to adding the data to the training set) will always improve performance: if there is insufficient data, using all the available data for training could give better results.
<input type="checkbox"/> Any one; the θ_i are all	✓ 0.25	While all θ_i are indeed local maxima of the

equivalent, since all of them are local maxima of the log-likelihood function.

log-likelihood function with respect to the training data, each of these local maxima corresponds to a different value of the log-likelihood function. (It is important to remember that these are not **global** maxima.)

Pick

$$\theta = \operatorname{argmax}_{i=1,2,\dots,n} \ell_{\text{Test}}(\theta_i).$$

✓ 0.25

We never use the test set for selecting any parameters; instead, we use it only for evaluating performance. This is because test set performance is meant as a measure of generalization ability, and we do not want to fit our parameters to the test set.

Pick

$$\theta = \operatorname{argmax}_{i=1,2,\dots,n} \ell_{\text{Train}}(\theta_i).$$

✗ 0.00

EM finds only local optima, so it is a good idea to find multiple local optima and pick the highest. This is to avoid ending up with poor local maxima of the log-likelihood function, i.e., those that have a relatively low log-likelihood with respect to other possible parameter values. Here, we are attempting to address the fact that we can't always optimize the function well, and not the problem of the optimum not generalizing well to the test data.

Total

0.75 /

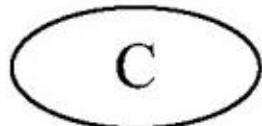
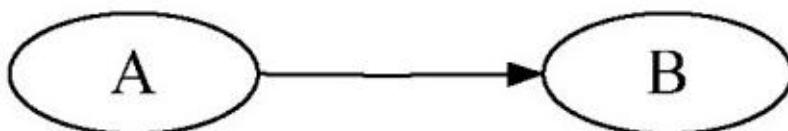
1.00

Question 14

Greedy Hill-Climbing. Your friend is performing greedy hill-climbing structure search over a network with three variables using three possible operations and the BIC score with dataset \mathcal{D} :

- Add an edge
- Delete an edge
- Reverse an edge

She tells you that after examining \mathcal{D} , she took a single step and got the following graph:



G

She also tells you that for the next step she has determined that there is a **unique** optimal greedy operation o to take. Which of the following steps could o be?

Hint: The fact that it is unique eliminates some possibilities for o .

Your Answer	Score	Explanation
<input type="checkbox"/> Add edge $A \rightarrow C$	✓ 0.17	Is there any other operation that would have the same improvement?
<input type="checkbox"/> Reverse edge $A \rightarrow B$	✓ 0.17	You cannot reverse an edge because the score will stay the same as BIC is score-equivalent.
<input checked="" type="checkbox"/> Delete edge $A \rightarrow B$	✗ 0.00	You cannot delete an edge because you wouldn't have added that edge in the first place in the greedy search.
<input type="checkbox"/> Add edge $C \rightarrow B$	✗ 0.00	Given the graph, this would be an unique operation to improve the score.
<input type="checkbox"/> Add edge $B \rightarrow C$	✗ 0.00	Given the graph, this would be an unique operation to improve the score.
<input type="checkbox"/> Add edge $C \rightarrow A$	✓ 0.17	Is there any other operation that would have the same improvement?
Total	0.50 / 1.00	

Question 15

***Loopy Belief Propagation.** Say you had a probability distribution P_{Φ} encoded in a set of factors Φ , and that you constructed a loopy cluster graph C to do inference in it. While you were performing loopy belief propagation on this graph, lightning struck and your computer shut down;

to your horror, when you booted it back up, the only information you could recover were the graph structure C and the cluster beliefs at the current iteration. (For each cluster, the cluster belief is its initial potential multiplied by all incoming messages. You don't have access to the sepset beliefs, the messages, or the original factors Φ .) Assume the lightning struck before you had finished, i.e., the graph is **not yet calibrated**. Can you still recover the original distribution P_Φ from this? Why?

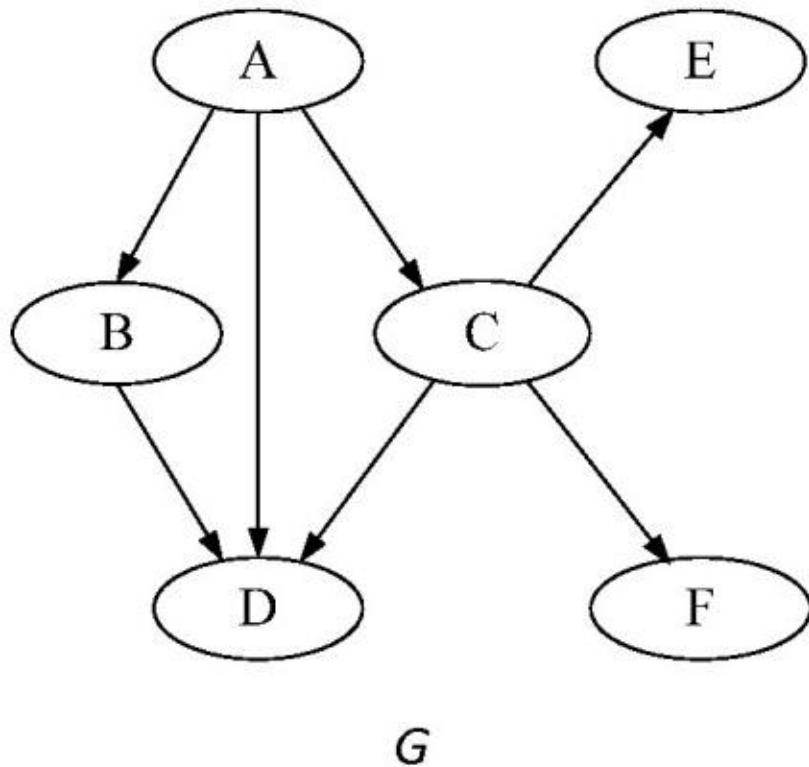
Your Answer	Score	Explanation
<input type="checkbox"/> We can reconstruct the original distribution by taking the product of cluster beliefs and normalizing it.	✓ 0.25	We need to take the ratio of the product of cluster beliefs to the sepset beliefs to reconstruct the original distribution.
<input type="checkbox"/> We can't reconstruct the original distribution because we were performing loopy belief propagation, and the reparameterization property doesn't hold when it's loopy.	✓ 0.25	Loopy or not, cluster graph beliefs are an alternative, calibrated parameterization of the original unnormalized density.
<input type="checkbox"/> We can't reconstruct the (unnormalized) original distribution because we don't have the sepset beliefs to compute the ratio of the product of cluster beliefs to sepset beliefs.	✗ 0.00	If the graph is calibrated, you can get the sepset beliefs by marginalizing the cluster beliefs, so it's ok if we lost them. But if the graph isn't calibrated, we can't get the sepset beliefs (you'll need the previous messages to get them), so we won't be able to recover the original distribution.
<input checked="" type="checkbox"/> We can reconstruct the (unnormalized) original distribution by taking the ratio of the product of cluster beliefs to sepset beliefs, and the sepset beliefs	✗ 0.00	If the graph is not calibrated, adjacent clusters won't agree on their sepsets, so we can't get the sepset beliefs (you'll need the previous messages to get them).

can be obtained
by marginalizing
the cluster beliefs.

Total	0.50 /
	1.00

Question 16

Graph Structure Search. Consider performing graph structure search using a decomposable score. Suppose our current candidate is graph G below.



We want to compute the changes of scores associated with applying three different operations:

- Delete the edge $A \rightarrow D$
- Reverse the edge $C \rightarrow E$
- Add the edge $F \rightarrow E$

Let $\delta(G : o_1), \delta(G : o_2), \delta(G : o_3)$ denote the score changes associated with each of these three operations, respectively. Which of the following equations is/are true for all datasets \mathcal{D} ?

Your Answer

$\delta(G : o_3) = \text{FamScore}(C, \{A, E\} : \mathcal{D})$

✓ 0.17

$\delta(G : o_3) = \text{FamScore}(E, \{C, F\} : \mathcal{D}) - \text{FamScore}(E, \{C\} : \mathcal{D})$ ✓ 0.17

$\delta(G : o_1) = \text{FamScore}(D, \{B, C\} : \mathcal{D}) - \text{FamScore}(D, \{A, B, C\} : \mathcal{D})$ ✓ 0.17

$\delta(G : o_1) = \text{FamScore}(D, \{B, C\} : \mathcal{D})$ ✓ 0.17

$\delta(G : o_2) = \text{FamScore}(C, \{A, E\} : \mathcal{D}) + \text{FamScore}(E, \emptyset : \mathcal{D}) - \text{FamScore}(C, \{A\} : \mathcal{D}) - \text{FamScore}(E, \{C\} : \mathcal{D})$ ✗ 0.00

$\delta(G : o_2) = \text{FamScore}(C, \{A, E\} : \mathcal{D}) - \text{FamScore}(C, \{A\} : \mathcal{D}) - \text{FamScore}(E, \{C\} : \mathcal{D})$ ✗ 0.00

Total 0.67 / 1.00

Question 17

Factorization of Probability Distributions. Consider a directed graph G . We construct a new graph G' by removing one edge from G . Which of the following is always true? You may select 1 or more options (or none of them, if you think none apply).

Your Answer	Score	Explanation
<input type="checkbox"/> No probability distribution P that factorizes over G also factorizes over G' .	✓ 0.25	A probability distribution where all variables are independent factorizes over any graph, including both G and G' .
<input type="checkbox"/> Any probability distribution P that factorizes over G' also factorizes over G .	✗ 0.00	Removing an edge can only add independencies (in both directed and undirected case) since it can't create new active paths. So $\mathcal{I}(G)$ is a subset of $\mathcal{I}(G')$. Any P that factorizes over G' would satisfy $\mathcal{I}(G')$ and therefore $\mathcal{I}(G)$, and therefore also factorizes over G too. The opposite isn't always true.
<input checked="" type="checkbox"/> Any probability distribution P	✗ 0.00	Removing an edge can only add independencies (in both directed and undirected case) since it can't create new active paths. So $\mathcal{I}(G)$ is a subset of $\mathcal{I}(G')$. Any P that factorizes

that factorizes over G also factorizes over G' .

over G' would satisfy $\mathcal{I}(G')$ and therefore $\mathcal{I}(G)$, and therefore also factorizes over G too. The opposite isn't always true.

If G and G' were undirected graphs, the answers to the other options would not change.

✓ 0.25

The answers are the same for both directed and undirected graphs.

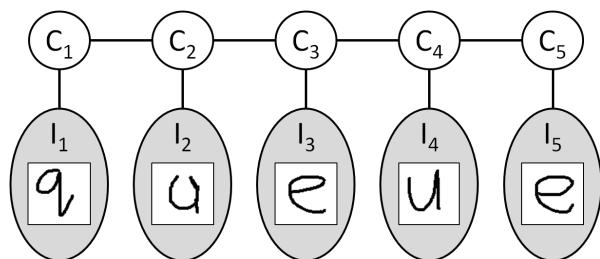
Total

0.50 /

1.00

Question 18

Template Model in CRF. The CRF model for OCR with only singleton and pairwise potentials that you played around with in PA3 and PA7 is an instance of a template model, with variables C_1, \dots, C_n over the characters and observed images I_1, \dots, I_n . The model we used is a template model in that the singleton potentials are replicated across different C_i variables, and the pairwise potentials are replicated across character pairs. The structure of the model is shown below:



Now consider the advantages of this particular template model for the OCR task, as compared to a non-template model that has the same structure, but where there are distinct singleton potentials for each C_i variable, and distinct potentials for each pair of characters. Which of the following about the advantage of using a template model is true? You may select 1 or more options (or none of them, if you think none apply).

Your Answer

Score Explanation

The template model can incorporate position-specific features, e.g. q-u occurs more frequently at the beginning of

✓ 0.25

In general, we cannot have features that depend on the identity of the variable in the template model.

a word, while a non-template model cannot.

The inference is significantly faster with the template model. ✓ 0.25 Faster inference is not an advantage. The template model requires calibrating an identical clique tree (after unrolling the template).

The same template model can be used for words of different lengths. ✓ 0.25 This is true, since for a non-template model we would need to have distinct singleton potentials for each C_i .

Parameter sharing could make the model less susceptible to over-fitting when there is less training data. ✓ 0.25 This is true, since there are less parameters in the template model.

Total	1.00 /
	1.00

Question 19

Structure Learning with Incomplete Data. After implementing the pose clustering algorithm in PA9, your friend tries to pick the number of pose clusters K for her data by running EM and evaluating the log-likelihood of her data for different values of K . What happens to her log-likelihood as she varies K ?

Your Answer	Score	Explanation
<input type="radio"/> The log-likelihood (almost) always decreases as K increases.		
<input type="radio"/> Impossible to say - depends on the data and on what K is.		
<input type="radio"/> The log-likelihood remains the same regardless of K .		
<input checked="" type="radio"/> The log-likelihood (almost) always increases as K increases.	✓ 1.00	

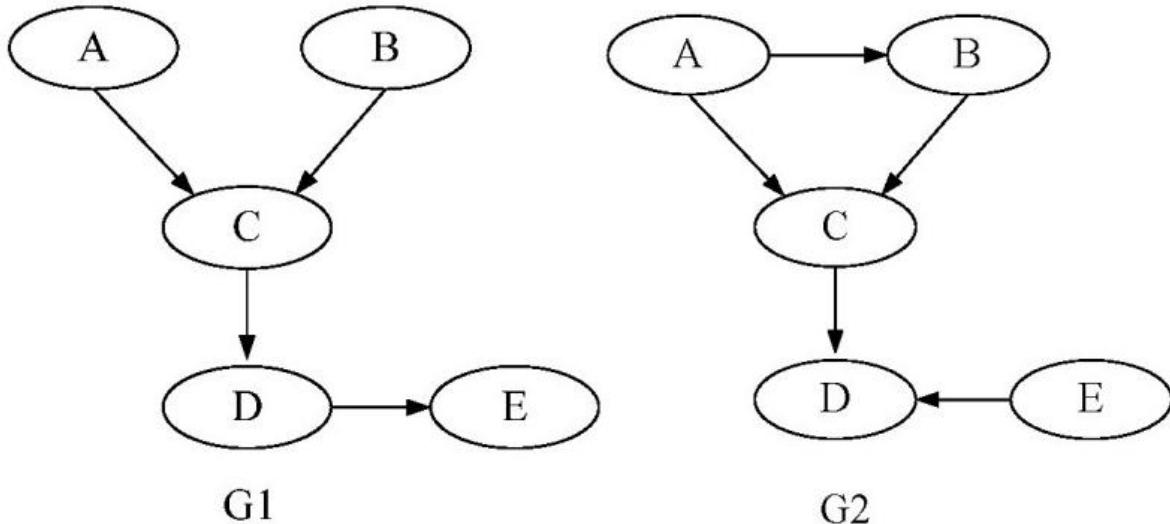
Total	1.00 /
	1.00

Question Explanation

The log-likelihood almost always increases as K increases, because of local optima. The log-likelihood of the data increases when you increase the number of clusters. In fact, in the limit where each data instance is assigned to its own cluster, the log likelihood will become infinity. Note that it does not strictly increase because EM can yield local optima.

Question 20

Calculating Likelihood Differences. While doing a hill-climbing search, you run into the following two graphs, and need to choose between them using the likelihood score.



What is the difference in likelihood scores, $\text{score}_L(G_1 : \mathbf{D}) - \text{score}_L(G_2 : \mathbf{D})$, given a dataset \mathbf{D} of size M ? Give your answer in terms of the entropy H and mutual information I . The subscripts below denote empirical values according to \mathbf{D} : for example, $H_{\mathbf{D}}(X)$ is the empirical entropy of the variable X in the dataset \mathbf{D} .

Your Answer
Score **Explanation**

- $M \times [I_{\mathbf{D}}(A; B) - H_{\mathbf{D}}(A, B)]$
- $M \times [I_{\mathbf{D}}(D; C) + I_{\mathbf{D}}(E; D) - I_{\mathbf{D}}(B; A) - I_{\mathbf{D}}(D; C, E)]$ ✓ 1.00
-
- $M \times [I_{\mathbf{D}}(C; A, B) + I_{\mathbf{D}}(D; C) + I_{\mathbf{D}}(E; D) - I_{\mathbf{D}}(B; A) - I_{\mathbf{D}}(D; C, E)]$
-
- $M \times [I_{\mathbf{D}}(D; C) + I_{\mathbf{D}}(E; D) - I_{\mathbf{D}}(A; B) - I_{\mathbf{D}}(D; C, E) - H_{\mathbf{D}}(A, B, C)]$
- $M \times I_{\mathbf{D}}(A; B)$

Total

1.00 /
1.00

Feedback — PA2 Quiz

[Help Center](#)

You submitted this homework on **Tue 30 Apr 2013 11:43 PM PDT**. You got a score of **12.00** out of **22.00**. You can [attempt again](#), if you'd like.

Question 1

James and Rene come to a genetic counselor because they are deciding whether to have another child or adopt. They want to know the probability that their un-born child will have cystic fibrosis.

Consider the Bayesian network for cystic fibrosis. We consider a person's phenotype variable to be "observed" if the person's phenotype is known. Order the probabilities of their un-born child having cystic fibrosis in the following situations from smallest to largest: (1) No phenotypes are observed (nothing clicked), (2) Jason has cystic fibrosis, (3) Sandra has cystic fibrosis.

Your Answer	Score	Explanation
-------------	-------	-------------

(2),
(1), (3)

(1),  2.00 Since Benjamin's phenotype and genotype are not observed in all of these situations, the probability that he will have cystic fibrosis (CF) is equivalent to the probability that James and Rene's unborn child will have CF. Observing that Benjamin's cousin has CF makes Benjamin more likely to have CF because CF is a genetic disease. Observing that Benjamin's brother has CF makes Benjamin more likely to have CF than when observing that Benjamin's cousin has CF because Benjamin's brother is a more closely-related relative than his cousin is.

(3),
(2), (1)

(3),
(1), (2)

(2),
(3), (1)

Total 2.00 /

2.00

Question 2

James never knew his father Ira because Ira passed away in an accident when James was a few months old. Now James comes to the genetic counselor wanting to know if Ira had cystic fibrosis. The genetic counselor wants your help in determining the probability that Ira had cystic fibrosis. Consider the Bayesian network for cystic fibrosis. We consider a person's phenotype variable to be "observed" if the person's phenotype is known. Order the probabilities of Ira having had cystic fibrosis in the following situations from smallest to largest: (1) No phenotypes are observed (nothing clicked), (2) Benjamin has cystic fibrosis, (3) Benjamin and Robin have cystic fibrosis.

Your Answer	Score	Explanation
-------------	-------	-------------

(3),
(2), (1)

(3), ✗ 0.00 Think about how observing the phenotypes of relatives might affect the probability that a person has a disease.
(1), (2)

(1),
(3), (2)

(2),
(1), (3)

(2),
(3), (1)

Total	0.00 / 2.00
-------	----------------

Question 3

Recall that, for a trait with 2 alleles, the CPD for genotype given parents' genotypes has 27 entries, and 18 parameters were needed to specify the distribution. How many parameters would be needed if the trait had 3 alleles instead of 2?

You entered:

180

Your Answer Score Explanation

180 ✓ 2.00 There are 6 possible genotypes for each parent and for the child, so the size of the CPD is $6 \times 6 \times 6 = 216$. Since the probability of having a genotype is fully defined if the probabilities for having the other genotypes are known, there are $216 - (6 \times 6) = 180$ parameters.

Total 2.00 /
 2.00

Question 4

You will now gain some intuition for why decoupling a Bayesian network can be worthwhile.

Consider a **non-decoupled** Bayesian network for cystic fibrosis with **3 alleles** over the pedigree that was used in section 2.4 and 3.3. How many parameters are needed to specify all probability distributions across the entire network?

You entered:

190

Your Answer Score Explanation

190 ✗ 0.00

Total 0.00 / 2.00

Question 5

Now consider the **decoupled** Bayesian network for cystic fibrosis with **3 alleles** over the pedigree that was used in section 2.4 and 3.3. How many parameters are needed to specify all of the probability distributions across the entire network?

Hint: A child cannot inherit an allele that is not present in either parent, so there aren't as many

degrees of freedom here as there might be without that context-specific information.

You entered:

29

Your Answer	Score	Explanation
29	✖	0.00
Total	0.00 / 2.00	

Question 6

Consider the **decoupled** Bayesian network for cystic fibrosis with three alleles that you constructed in section 3.3. We consider a person's gene copy variable to be "observed" if the person's allele for that copy of the gene is known.

James and Rene are debating whether to have another child or adopt a child. They are concerned that, if they have a child, the child will have cystic fibrosis because both of them have one F allele observed (their other gene copy is not observed), even though neither of them have cystic fibrosis. You want to give them advice, but they refuse to tell you whether anyone else in their family has cystic fibrosis. What is the **probability** (NOT a percentage) that their unborn child will have cystic fibrosis?

You entered:

0.005

Your Answer	Score	Explanation
0.005	✖	0.00
Total	0.00 / 2.00	

Question 7

Consider a Bayesian network for spinal muscular atrophy (SMA), in which there are multiple genes and 2 phenotypes.

Let n be the number of genes involved in SMA and m be the maximum number of alleles per gene. How many parameters are necessary if we use a table CPD for the probabilities for phenotype given copies of the genes from both parents?

Your Answer**Score****Explanation**

$O(2^n)$

Depends
on the
phenotype

$O(4^n)$

$O(m^{2n})$ ✓ 2.00 There are two alleles per gene, so there are $O(m^2)$ allele combinations per gene. Therefore, there are $O(m^{2n})$ parameters for n genes.

Total

2.00 /

2.00

Question 8

Consider the Bayesian network for spinal muscular atrophy (SMA), in which there are multiple genes and two phenotypes.

Let n be the number of genes involved in SMA and m be the maximum number of alleles per gene. How many parameters are necessary if we use a sigmoid CPD for the probabilities for phenotype given copies of the genes from both parents?

Your Answer**Score****Explanation**

$O(\max(m, n))$

Depends on
the phenotype

$O(mn)$

✓ 2.00

Each gene has up to m alleles, and there is an indicator for each allele for each copy of the gene. Therefore, if there were one gene, there would be $O(2m) = O(m)$ parameters. Since there are n genes, there are $O(mn)$ possible parameters.

$O(m^2n)$

Total	2.00 /
	2.00

Question 9

Consider genes A and B that might be involved in spinal muscular atrophy. Assume that A has 2 alleles A_1 and A_2 , and B has 2 alleles, B_1 and B_2 . Which of the following relationships between A and B can a sigmoid CPD capture?

Your Answer	Score	Explanation
<input checked="" type="checkbox"/> Gene A contributes to SMA, but gene B does not contribute to SMA and thus does not affect the effects of gene A on SMA.	✓ 0.29	A sigmoid CPD can capture this by giving the alleles for copies of gene A positive weights and the alleles for copies of gene B zero weights.
<input checked="" type="checkbox"/> Alleles A_1 and B_1 each independently make a person likely to have SMA.	✓ 0.29	Since their contributions are independent, a sigmoid CPD that weights the alleles for each gene based on the extent of their contribution would capture this perfectly.
<input type="checkbox"/> Allele A_1 and allele B_1 make a person more likely to be have SMA when both of these alleles are present, but neither affect SMA otherwise.	✓ 0.29	This AND relationship cannot be captured by a sigmoid CPD because interaction terms between the alleles are not present.
<input type="checkbox"/> When the alleles are A_1 and B_2 or A_2 and B_1 the person has SMA; otherwise the person does not have SMA.	✓ 0.29	This XOR relationship means that the effect of the allele for gene A depends on which allele for gene B is present; since the sigmoid CPD does not have interactive terms, it will not be able to capture this.
<input checked="" type="checkbox"/> Neither gene A nor gene B contribute to SMA.	✓ 0.29	A sigmoid CPD can capture this by giving alleles for copies of gene A as well as alleles for copies of gene B weights with value zero.
<input type="checkbox"/> Allele A_1 and allele B_1 make a person	✓ 0.29	This OR relationship cannot be captured by a sigmoid CPD because interaction terms between the alleles are

equally more likely to have SMA, but when both are present the effect on SMA is the same as when only one is present.

not present.

- Allele A_1 makes a person more likely to have SMA, while allele B_1 independently makes a person less likely to have SMA.

✓ 0.29

A sigmoid CPD can capture this by making the weights for the indicators for allele A_1 positive while making the weights for the indicators for allele B_1 negative.

Total	2.00 /
	2.00

Question 10

Consider the Bayesian network for spinal muscular atrophy that we provided in spinalMuscularAtrophyBayesNet.net. We consider a person's gene copy variable to be "observed" if the person's allele for that copy of that gene is known.

Now say that Ira and Robin come to the genetic counselor because they are debating whether to have a biological child or adopt and are concerned that their child might have spinal muscular atrophy. They have some genetic information, but because sequencing is still far too expensive to be affordable for everyone, their information is limited to only a few genes and to only 1 chromosome in each pair of chromosomes.

Order the probabilities of their un-born child having spinal muscular atrophy in the following situations from smallest to largest: (1) No genetic information or phenotypes are observed (nothing clicked), (2) Ira and Robin each have at least 1 M allele, (3) Ira and Robin each have at least 1 M allele and at least 1 B allele.

Your Answer	Score	Explanation
-------------	-------	-------------

(3),
(1), (2)

(2),
(3), (1)

(1), ✓ 2.00 Since James is unobserved, the probability that he will have spinal

(2), (3)

muscular atrophy (SMA) is equivalent to the probability that Ira and Robin's unborn child will have SMA. Observing that Ira and Robin each have an allele that is involved in causing SMA makes James more likely to have SMA than if no variables were observed. Observing that Ira and Robin each have alleles for 2 genes that are involved in causing SMA makes James even more likely to have SMA than if only 1 allele for 1 gene were observed.

(2),
(1), (3)

(3),
(2), (1)

Total 2.00 /
 2.00

Question 11

Consider the Bayesian network for spinal muscular atrophy that we provided in [spinalMuscularAtrophyBayesNet.net](#).

No longer interested in finding out whether his father had cystic fibrosis, James comes to the genetic counselor with another question: Did his father have spinal muscular atrophy? The genetic counselor now wants your help in figuring this out. This time, however, James has other information for you: both he and Robin have spinal muscular atrophy.

What is the **probability** (NOT a percentage) that Ira had spinal muscular atrophy?

You entered:

0.032

Your Answer

0.032



Score

0.00

Explanation

Total

0.00 / 2.00

Feedback — PA5 Quiz

[Help Center](#)

You submitted this homework on **Tue 21 May 2013 11:39 AM PDT**. You got a score of **31.00** out of **32.00**. You can [attempt again](#), if you'd like.

Question 1

Recall our function **CheckConvergence.m** which uses the difference between a message before and after it is updated (called the ``residual'') as a criterion for convergence. While running LBP with our naive message ordering on the network created by **ConstructRandNetwork** with on-diag weight .3 and off-diag weight .7, print out and plot the residuals of the message $19 \rightarrow 3, 15 \rightarrow 40$, and $17 \rightarrow 2$, with the iteration number on the x-axis (you may want to change the range of the y-axis). Do these messages converge at the same rate? Which converges fastest? (Note, it will be easiest do this assessment within **ClusterGraphCalibrate** and use the helper function **MessageDelta** within that file).

Your Answer	Score	Explanation
<input checked="" type="radio"/> Message $19 \rightarrow 3$ converges quickly, followed by $17 \rightarrow 2$ and finally $15 \rightarrow 40$	✓ 5.00	
<input type="radio"/> Message $15 \rightarrow 40$ converges faster than the others, which both take significantly longer.		
<input type="radio"/> All messages converge at the same rate, though this behavior is not guaranteed in LBP.		
<input type="radio"/> All messages converge at the same rate, as is always the case in LBP.		
Total	5.00 / 5.00	

Question 2

Which of the following are true about the effects the message passing order can have cluster graph calibration?

Your Answer	Score	Explanation
<input type="checkbox"/> On the same graph, one ordering may converge while another never does.	✗ 0.00	In some cases, bad message orderings can cause infinite loops in message changes that could be avoided with a different ordering.
<input type="checkbox"/> The value of the final marginals in a graph with no loops can depend on the message passing order.	✓ 1.00	Without loops, LBP is equivalent to clique tree inference which is exact and gives the same solution regardless of order.
<input checked="" type="checkbox"/> It can affect how long it takes for LBP to reach convergence.	✓ 1.00	Consider our naive ordering, it keeps all passing messages even after some have converged. Using a smarter scheme could thus avoid extra work and speed up convergence.
<input type="checkbox"/> Different message passing orders can lead to differences in the final full joint distribution given by cluster potentials.	✓ 1.00	The joint distribution will never change, this is the cluster graph invariant.
Total	3.00 / 4.00	

Question 3

Now, consider the toy image network constructed in `ConstructToyNetwork.m`. Change the values of the on- and off-diagonal weights of the pairwise factors in this network to different values (which can be done by changing the values passed to this function). First try making the the weights on the diagonal much larger than the off-diagonal weights (1 and .2 respectively), then try the opposite where the off-diagonal weights are much larger (.2 and 1), and then finally try the case where the weights are roughly equal (.5 and .5). For each such model, run LBP and exact inference (using your code from PA4). Which of the following occur in this setup? Why? (NOTE: if LBP does not converge within 100,000 iterations it is okay to truncate the run and report on the pseudo-marginals given at that point)

Your Answer	Score	Explanation
<input type="checkbox"/> All runs instances converge quickly to approximately the correct marginals due to LBP's	✓ 1.25	All statements are false

ability to overcome local maxima.

- | | |
|--|---|
| <input checked="" type="checkbox"/> The case of low on diagonal weights with high off diagonal weights has poor convergence because of anti-correlation in the variables, which causes LBP to oscillate. | ✓ 1.25 |
| <input type="checkbox"/> The case of high on diagonal weights with low off diagonal weights converges quickly as compared to the others because the strong correlation causes us to quickly enter a strong local optima. | ✓ 1.25 This will cause positive feedback cycles because of the graph's short loops, preventing hasty convergence. |
| <input checked="" type="checkbox"/> The case of high on diagonal weights with low off diagonal weights has poor convergence because of high variable correlation coupled with short loops in the network, causing positive feedback loops. | ✓ 1.25 |

Total	5.00 /
	5.00

Question Explanation

Question explanation

Question 4

Let's run an experiment using our Gibbs sampling method. As before, use the toy image network and set the on-diagonal weight of the pairwise factor (in `ConstructToyNetwork.m`) to be 1.0 and the off-diagonal weight to be 0.1. Now run Gibbs sampling a few times, first initializing the state to be all 1's and then initializing the state to be all 2's. What effect does the initial assignment have on the accuracy of Gibbs sampling? Why does this effect occur?

Your Answer

Score

Explanation

- The initial state has a significant impact on the result of our sampling as Gibbs will never switch variables because the pairwise potentials enforce strong agreement so we are in a local optima.

- The initial state has a significant impact on the result of our sampling, which makes sense as strong correlation makes mixing time long and we remain close to the initial assignment for a long time.

✓ 5.00

- The initial state is not an important factor in our result as Gibbs can make large moves of multiple variables to quickly escape this bad state.

- The initial state has a significant impact on the result as, though our chain mixes quickly, it will mix to a distribution far from the actual distribution and close to the initial assignment.

Total

5.00 /

5.00

Question Explanation

Question explanation

Question 5

Set the on-diagonal weight of our toy image network to 1 and off-diagonal weight to .2. Now visualize multiple runs with each of Gibbs, MHUniform, Swendsen-Wang variant 1, and Swendsen-Wang variant 2 using VisualizeMCMCMarginals.m (see TestToy.m for how to do this). How do the mixing times of these chains compare? How do the final marginals compare to the exact marginals? Why?

Your Answer

Score

Explanation

- Having strong pairwise potentials enforcing agreement is not a problem for any of these sampling methods and all perform equally well -- mixing quickly and ending up close to the final marginals.

- Gibbs outperforms the other variants in this instance. Gibbs has some issues with strong pairwise potentials, but is not nearly as bad as MH where blocks end up stuck with the same level so we cannot mix appropriately.

- The Swendsen-Wang variants outperform the other approaches, with faster mixing and better final marginals. This is likely due to the block-flipping nature of Swendsen-Wang which allows us to flip blocks and quickly mix in environments with strong agreeing potentials.

✓ 5.00

- All variants perform poorly in the case of strong pairwise potentials. All algorithms are subject to positive feedback loops with the tight loops in our grid and strong pairwise agreement potentials, preventing appropriate mixing.

Total	5.00 /
	5.00

Question Explanation

Question explanation

Question 6

Set the on-diagonal weight of our toy image network to .5 and off- diagonal weight to .5. Now visualize multiple runs with each of Gibbs, MHUniform, Swendsen-Wang variant 1, and Swendsen-Wang variant 2 using VisualizeMCMCMarginals.m (see TestToy.m for how to do this). How do the mixing times of these chains compare? How do the final marginals compare to the exact marginals? Why?

Your Answer

Score Explanation

- Swendsen-Wang outperforms the other variants, though all perform relatively well. SW is better because its larger block moves allow for faster mixing and mean it reaches marginal estimates closer to the true marginals faster.

- Gibbs and MHUniform perform very well and are somewhat better than the Swendsen-Wang variants. This is because the first two variants use local moves so the local marginals remained consistently close to the true marginals, while SW allows big swings over multiple variables that perturb the distribution.

- Gibbs performs poorly relative to the other variants -- exhibiting slower mixing time and marginals further from the exact ones. This difference is likely due to the Gibbs strong global dependence that prevents it from acting appropriately unless all variables are relatively well synced to their true marginals.

- All variants perform equally well. They all mix quickly and have very low variance throughout their runs -- remaining close to the true marginals. This is because the pairwise marginals do not force us into preferring agreement when we should not.

Total

5.00 /

5.00

Question Explanation

Question explanation

Question 7

When creating our proposal distribution for Swendsen-Wang, if you set all the $q_{i,j}$'s to zero, what does Swendsen-Wang reduce to?

Your Answer**Score** **Explanation**

Switching $q_{i,j}$ to 0 is equivalent to MH-Uniform.

Switching $q_{i,j}$ to 0 is equivalent to the first variant of Swendsen-Wang.

Switching $q_{i,j}$ to 0 is equivalent to a randomized variant of Gibbs sampling where we are allowed to take a random, rather than fixed, order.

✓ 3.00

Compare the resulting proposal distribution to our Gibbs proposal distribution to see that the two agree.

Switching $q_{i,j}$ to 0 leaves us without a valid proposal distribution and is not a feasible sampling algorithm.

Total

3.00 /

3.00

Feedback — PA6 Quiz

[Help Center](#)

You submitted this homework on **Sun 26 May 2013 11:04 AM PDT**. You got a score of **32.00** out of **35.00**. You can [attempt again](#), if you'd like.

Question 1

We have provided an instantiated influence diagram Fulll (complete with a decision rule for D) in the file Fulll.mat. What is the expected utility for this influence diagram? Please round to the nearest tenth (i.e., 1 decimal place), do not include commas, and do not write the number in scientific notation.

You entered:

-686.0

Your Answer	Score	Explanation
-686.0	✓ 2.00	This is the output of SimpleCalcExpectedUtility(Fulll).
Total	2.00 / 2.00	

Question 2

Run ObserveEvidence.m on Fulll to account for the following: We have been informed that variable 3 in the model, which models an overall genetic risk for ARVD, has value 2 (indicating the presence of genetic risk factors). Then run SimpleCalcExpectedUtility on the modified influence diagram. What happened to the expected utility? (Hint -- ObserveEvidence does not re-normalize the factors so that they are again valid CPDs unless the normalize flag is set to 1. -- If you do not use the normalize flag, you can use NormalizeCPDFactors.m to do the normalization.)

Your Answer	Score	Explanation
<input type="radio"/> The expected utility might or might not change because there is some randomness in the process for determining the expected utility.		

- It substantially decreased.
- It substantially increased.
- It did not change.

✓ 3.00 It decreased from -685.9 to -729.2

Total 3.00 /
3.00

Question 3

Why can we explicitly enumerate all the possible decision rules while we often cannot enumerate over all possible CPDs?

- | Your Answer | Score | Explanation |
|--|--------|--|
| <input type="radio"/> We can actually always enumerate over all possible CPDs. | | |
| <input type="radio"/> In an influence diagram, each decision node cannot have more than 1 parent, while in a general Bayes net, a node can have many parents. | | |
| <input checked="" type="radio"/> All choices have a probability of either 0 or 1, where in a general CPD, choices could take on any value in [0, 1]. | ✓ 3.00 | Because each choice is restricted to a finite set of probabilities, there is a finite number of possible decision rules, so we can enumerate over all of them. |
| <input type="radio"/> If there is one choice in a decision rule, at least one choice must have a 0 probability, where in a general CPD, no entries are restricted to having 0 probabilities. | | |

Total 3.00 /
3.00

Question 4

Let a decision node D take on d possible values. Let it have m parents that can each take on n possible values. How many possible decision rules δ_D are there?

Your Answer	Score	Explanation
-------------	-------	-------------

$d^{(2n^m)}$

$d(m^n)$

$d^{(n^m)}$

$2d(n^m)$ ✗ 0.00 Think about how every possible combination of decisions over all joint assignments of parents to values is a decision rule. Also, note that there is only one possible decision.

$d^{(m^n)}$

d^{nm}

$d(n^m)$

Total 0.00 /
 3.00

Question 5

Consider an influence diagram with 1 decision node D that can take on d values. Let D have m parents that can each take on n values. Assume that running sum-product inference takes $O(S)$ time. What is the run-time complexity of running OptimizeMEU on this influence diagram?

Your Answer	Score	Explanation
-------------	-------	-------------

$O(Sn^m)$

$O(S + n^m)$

$O(S + dnm)$ $O(d^{(n^m)})$ $O(Sdn^m)$ $O(Sdnm)$ $O(S + dn^m)$

4.00 Sum-product inference can be run only once, and it requires $O(S)$ time. The results of sum-product inference can be used to construct a table that is the number of instantiations to the parents, which is n^m , by the number of decision values, which is d . To choose the optimal decision, it is sufficient to scan through the table once. Thus, the total run-time is $O(S + dn^m)$.

Total	4.00 /
	4.00

Question 6

In which of the following situations does it make sense to use `OptimizeWithJointUtility` instead of `OptimizeLinearExpectations`?

Your Answer**Score****Explanation** When the scopes of the utility factors are large compared to the scopes of the other (random variable) factors. When every random variable in the network is a parent of at least one other utility factor.

When the bottleneck in inference is in enumerating the large number of possible assignments to the parents of the utility variables, and each utility variable has a disjoint set of parents.

- When there are large factors in the random-variables part of the influence diagram, making inference over the network slow, and there are only a few utility factors, each involving a small number of variables.
- 5.00 If there are only a few utility factors, each involving a small number of variables, combining them into a joint factor won't result in too big a factor. Since the size of the utility factor is not in running inference, and since inference is expensive and we don't want to run it once for each utility factor, it makes sense to create a joint utility factor.

Total 5.00 /
5.00

Question 7

In the field below, enter the dollar value of each of the tests, starting with T1, then T2 and T3, separated by commas and rounded to the nearest cent (e.g., "1.23, 4.54, 3.21" means that you would pay \$1.23 for the first test; any more than that, and your net utility will be lower than if you didn't perform any test). Do not precede with the amounts with dollar signs.

You entered:

155.97, 2.82, 846.15

Your Answer	Score	Explanation
155.97	✓	5.00
2.82	✓	5.00
846.15	✓	5.00
Total	15.00 / 15.00	