



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 1 : INTRODUCTION

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.



Table of Contents

1 Uncertainty

2 Probabilistic Graphical Model

3 Applications of Probabilistic Graphical Model

4 Course Logistics

Uncertainty

- We select a course of actions among many possibilities.
- Decisions may be based on the information obtained from the environment, previous knowledge and the objectives.
- Eg: It looks cloudy. Should I carry an umbrella?
- The information and knowledge is incomplete or unreliable. So the decisions made are not certain. **We make decisions under uncertainty.**
- One of the goals of AI is to develop systems that can reason and make decisions under uncertainty.

Uncertainty

- Complexity increases
 - ▶ Each piece of knowledge may not be independently used to arrive at decisions.
 - ▶ Deduced facts are maintained along with new facts. This increases the knowledge base.
- Examples
 - ▶ A medical doctor in an emergency.
 - ▶ An autonomous vehicle that detects what might be an obstacle in its way.
 - ▶ A financial agent needs to select the best investment.

$$P(A, B) + P(A, \neg B) = P(A) \quad \text{independence} = \text{good}$$

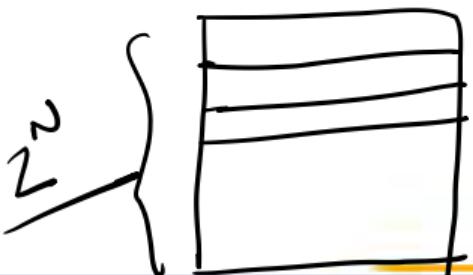
Limitations of Traditional Approach

$$P(x_1, x_2, \dots, x_n)_{2^n} = P(x_1)P(x_2/x_1)P(x_3/x_2)\dots P(x_n/x_{n-1})$$

~~correlation = bad~~

~~N variables when variables are binary.~~

- Impractical for complex problems with many variables, as the size of the Joint probability table and the direct computation of Marginal and Conditional probabilities grow exponentially with the number of variables in the model.
- Good estimates for the joint probabilities from data requires a very large database if there are many variables in the model.



$$P(Y/x_i) = \frac{P(Y \& X_i)}{P(X_i)} \quad \text{by Bayes rule}$$

marginalisation $\rightarrow P(Y \& X_i \& A_1 \& A_2 \dots)$



Table of Contents

1 Uncertainty

2 Probabilistic Graphical Model

3 Applications of Probabilistic Graphical Model

4 Course Logistics

Probabilistic Graphical Models

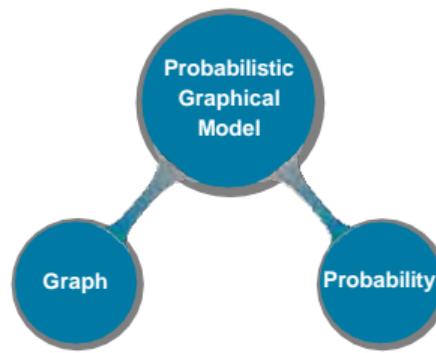
- Provide a framework for managing uncertainty based on probability theory in a computationally efficient way.
- Consider Independence relations that are valid for a certain problem, and use independence to reduce computational complexity and be memory efficient.
- Represent the dependence and independence relations between a set of variables using graphs.

Probabilistic Graphical Models

- Around 2000s
- Techniques based on probability and graphical representations were consolidated as powerful methods for representing, reasoning and making decisions under uncertainty.
- Bayesian networks, Markov networks, influence diagrams and Markov decision processes, among others.

Probabilistic Graphical Model

- Probabilistic Graphical Model combines probability theory and graph theory to deal with problems involving uncertainty and complexity and also in the design and analysis of machine learning algorithms.

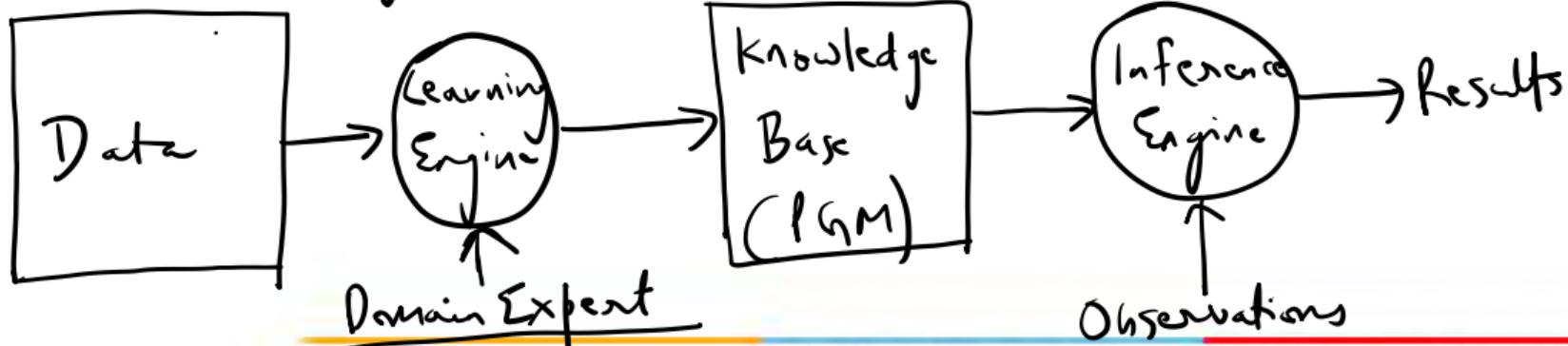


Declarative representation

Separation of knowledge and reasoning

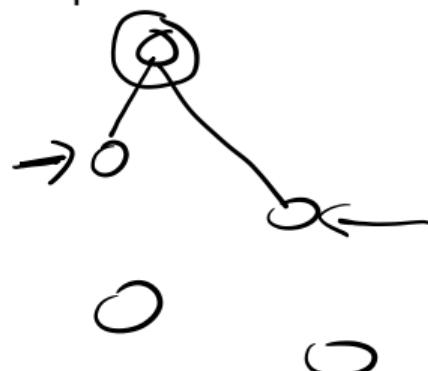
Encar, Chaffet

The representation has its own clear semantics, the reasoning algorithms are independent of this.



Need of Probabilistic Graphical Model

- Model uncertainty.
- Model complex structures with causal and spatial relationships.
- Model domain knowledge and prior knowledge.
- Draw inferences from the model.
- Learn the structure and parameters of the model.



Classification of Probabilistic Graphical Model



1 Direct or Undirected

- ▶ Directed graphs represent parent-child relations or **cause-effect** relations.
- ▶ Undirected graphs represent symmetric relations.

2 Static or Dynamic

- ▶ Static – Model represents a set of variables at a certain point in time.
- ▶ Dynamic – Model represents a set of variables across different times.

3 Probabilistic or Decisional

- ▶ Probabilistic models include random variables.
- ▶ Decisional models include random variables, decision and utility variables.

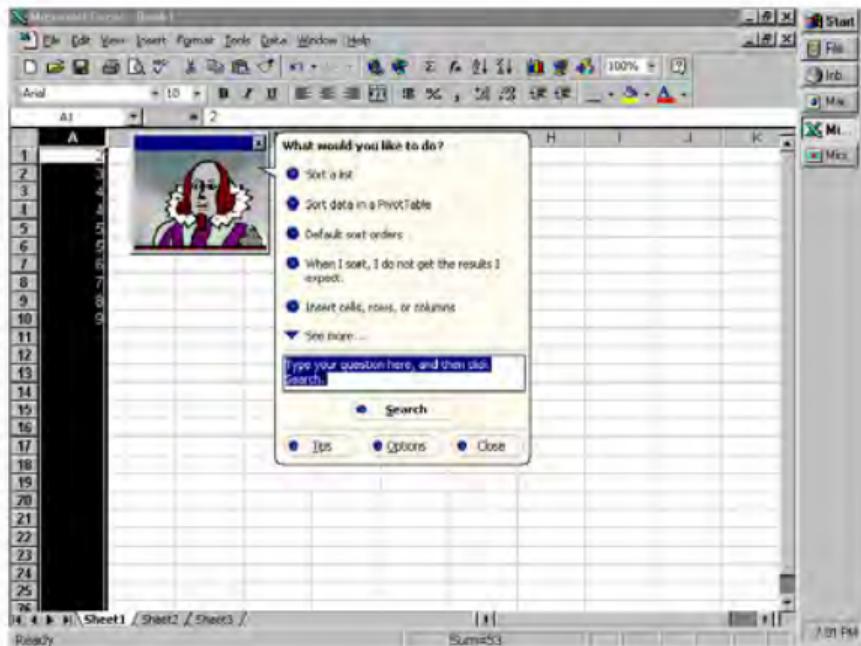
Common Probabilistic Graphical Models

Type	Directed / Undirected	Static / Dynamic	Probabilistic / Decisional
Bayesian Models	both	Static	Probabilistic
Markov Chains	Directed	Dynamic	Probabilistic
Hidden Markov Models	Directed	Dynamic	Probabilistic
Markov Random Fields	Undirected	Static	Probabilistic
Bayesian Networks	Directed	Static	Probabilistic
Dynamic Bayesian Networks	Directed	Dynamic	Probabilistic
Influence Diagrams	Directed	Static	Decisional
Markov Decision Processes	Directed	Dynamic	Decisional
Partially Observable MDPs	Directed	Dynamic	Decisional

Table of Contents

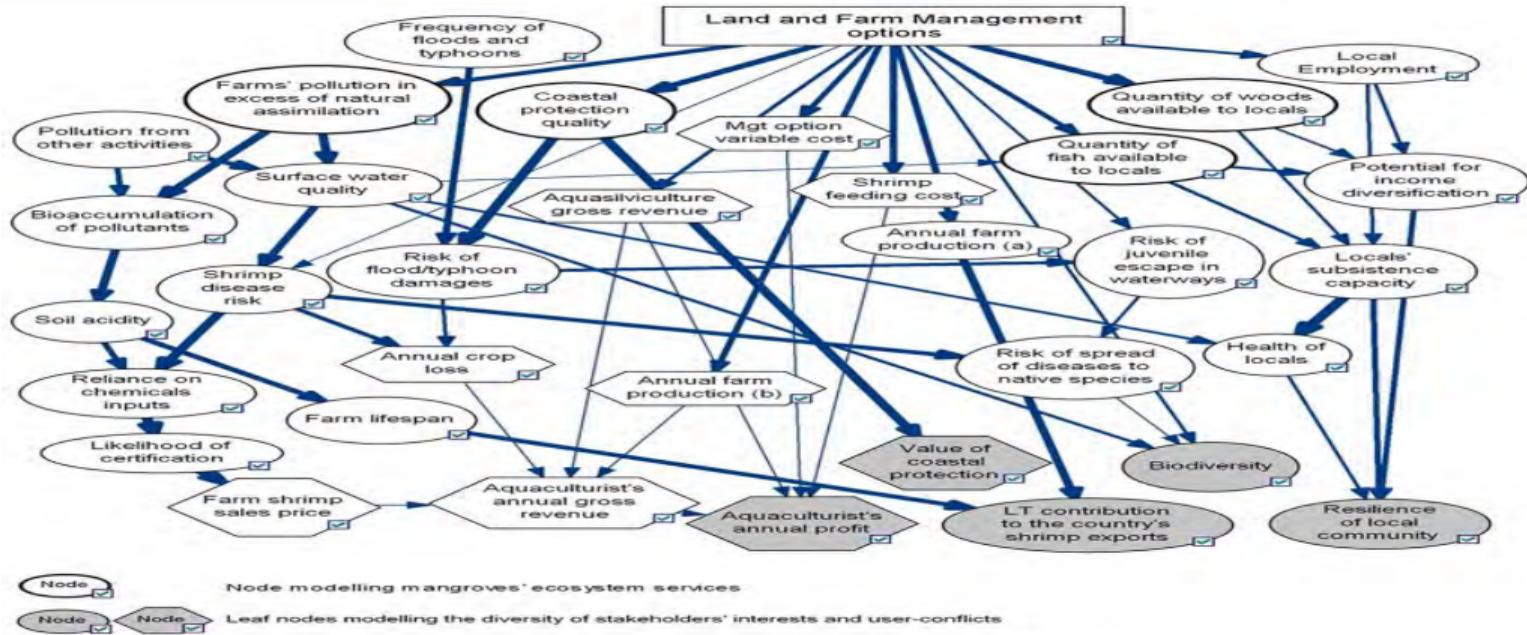
-
- 1 [Uncertainty](#)
 - 2 [Probabilistic Graphical Model](#)
 - 3 [Applications of Probabilistic Graphical Model](#)
 - 4 [Course Logistics](#)

Bayesian Models -Microsoft Lumiere Project



<http://erichorvitz.com/lumiere.htm>

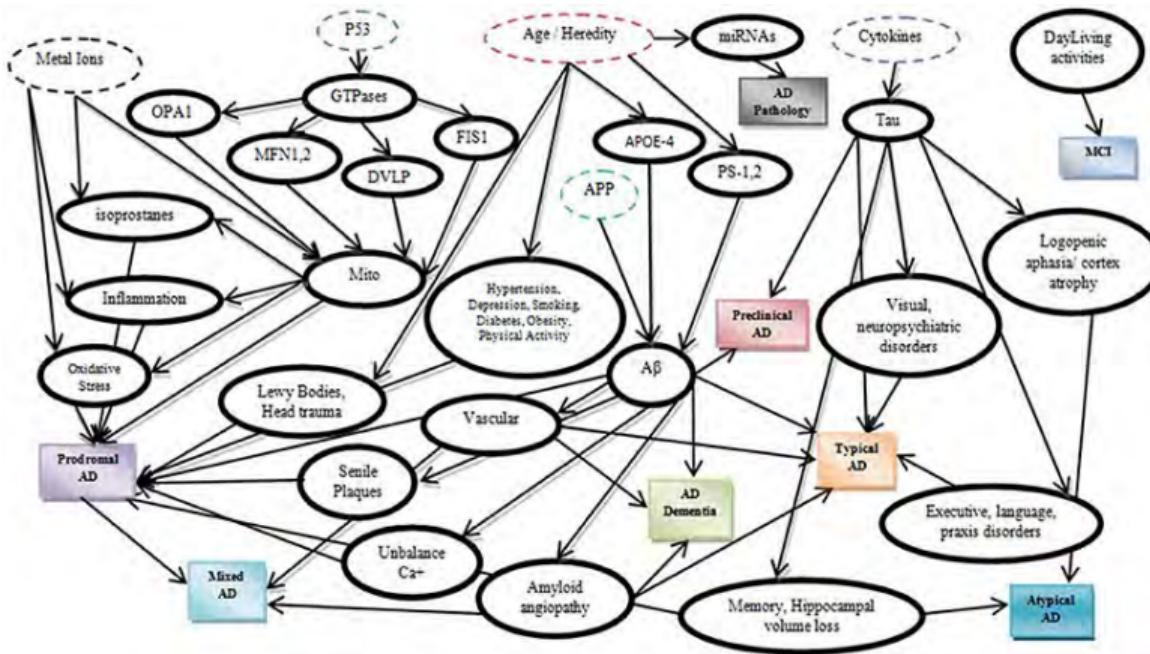
Bayesian Models - Life Sciences



Schmitt, Laetitia & Brugere, Cecile. (2013). Capturing Ecosystem Services, Stakeholders' Preferences and Trade-Offs in Coastal Aquaculture Decisions: A Bayesian Belief Network Application.

Bayesian Models – Medical Diagnosis

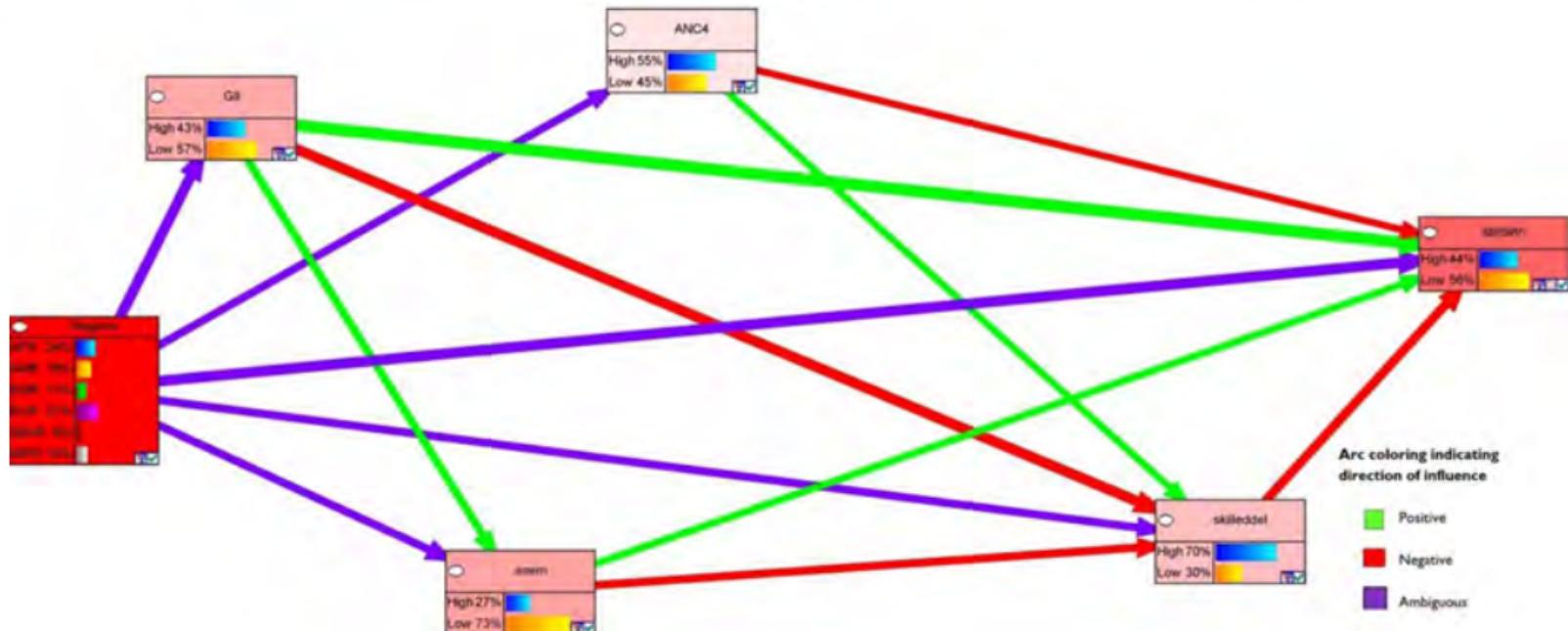
lead



<https://doi.org/10.3389/fnagi.2017.00077>



Bayesian Models – Spatial Relationship

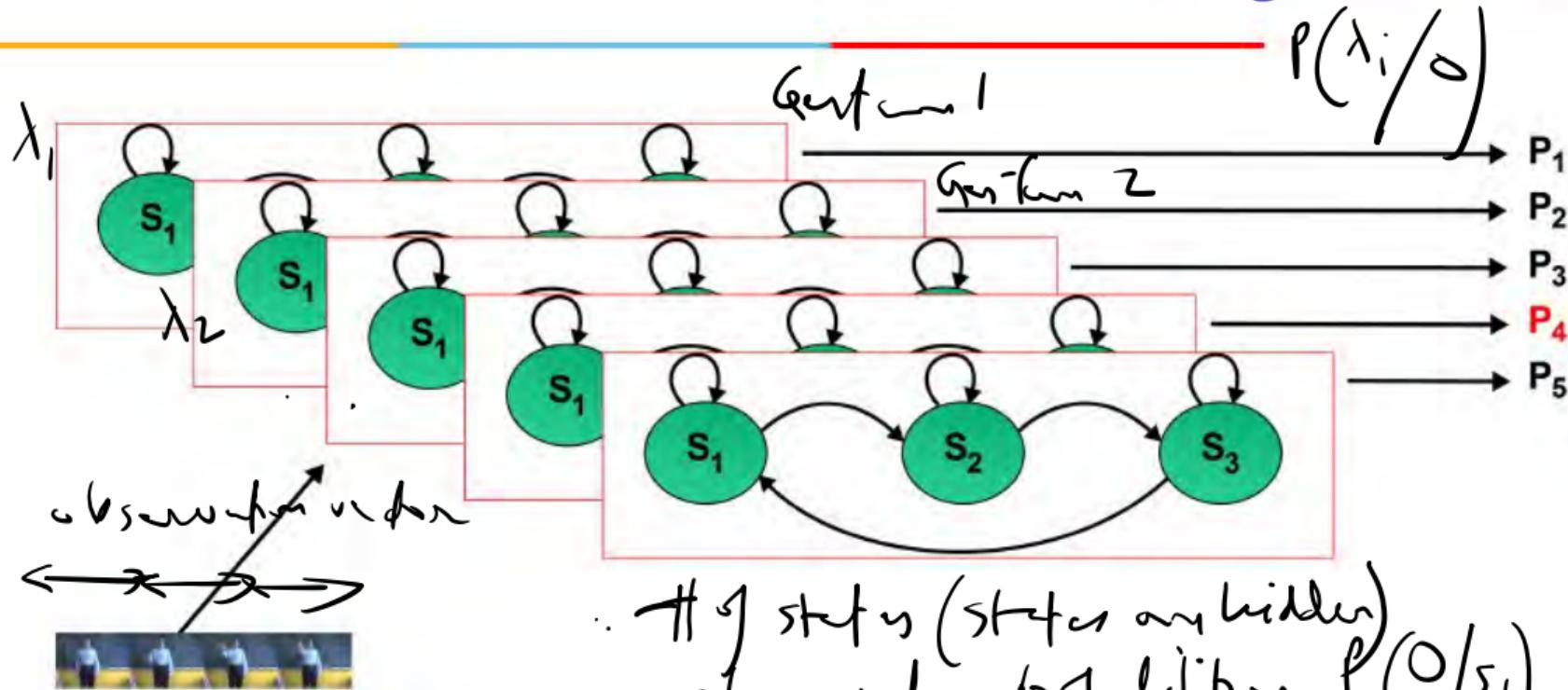


<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6930217/>

Sucar's book

lead

Hidden Markov Model - Gesture Recognition



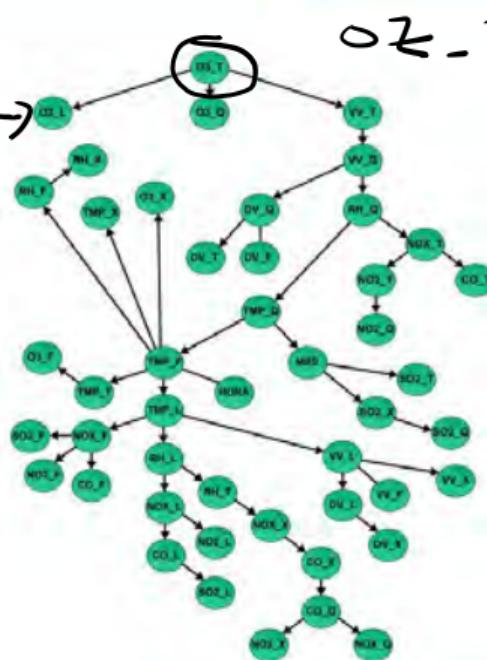
↑ 3 states (states are hidden)
observation probability $P(O|S_t)$
state transition prob $P(S_{t+1}|S_t)$

L E Sucar

Bayesian Networks – Ozone Prediction

Sucav's book

Tree

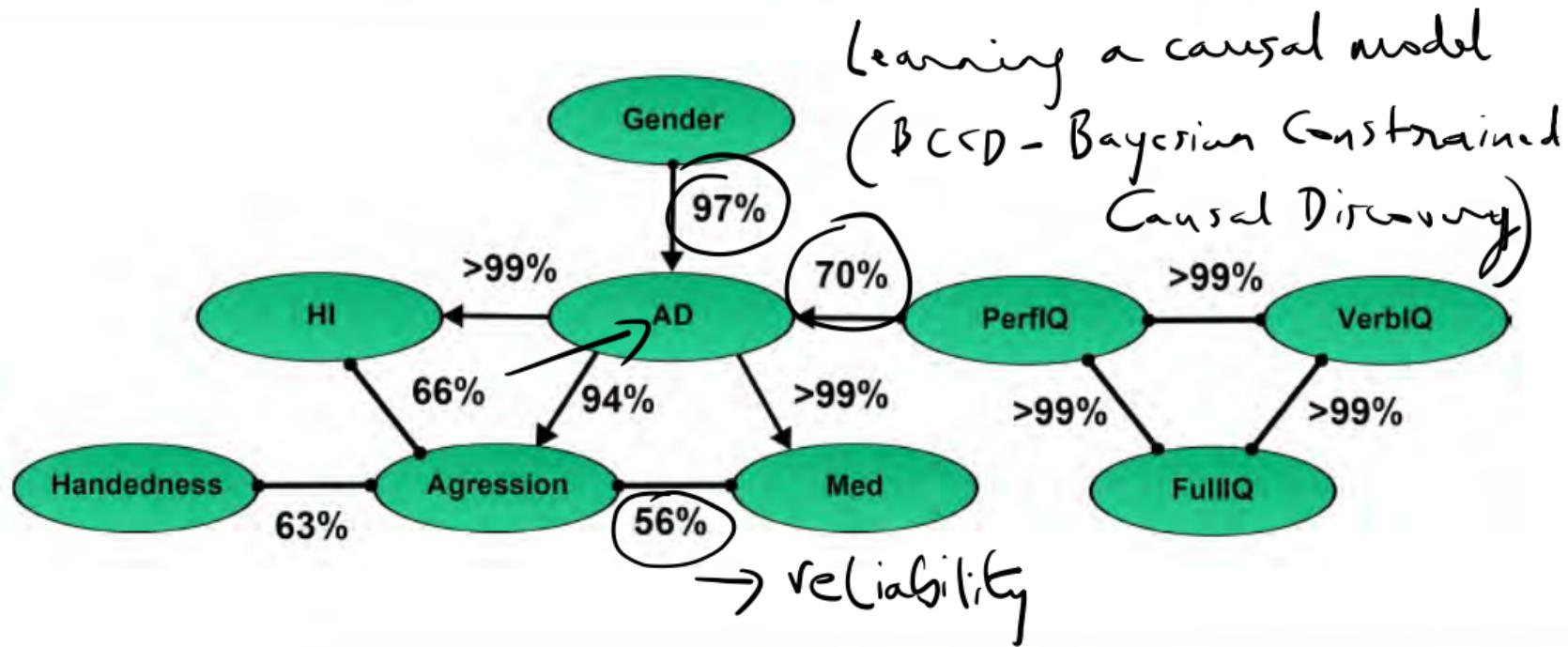


T
forecast pollution
levels several hours in
advance

Total of 47 variables
9 measurements each
for 5 stations + hour
+ month

L E Sucar

Bayesian Networks - Attention Deficit Model



L E Sucar

A Detailed Look

Consider a simple medical diagnosis problem - there are two diseases flu and hay fever. These diseases are not mutually exclusive

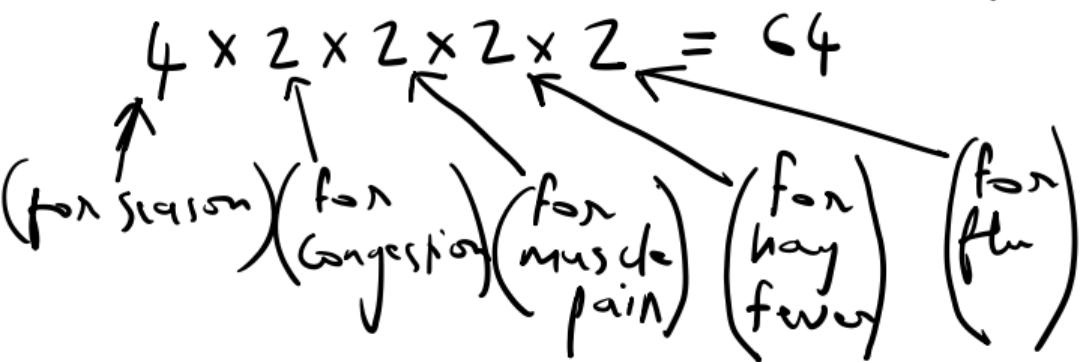
4-valued random variable \rightarrow Season (Winter, Spring, Summer, Autumn)

2-valued symptoms \rightarrow Congestion, Muscle pain

Total of 5 variables: Season, Congestion, Muscle pain, Flu, Hay Fever

A Detailed Look

What is the sample space size of this joint distribution?



Question: How likely is it that a patient has the flu given that it is fall, and the patient has sinus congestion but no muscle pain?

A Detailed Look

Daphne
Koller

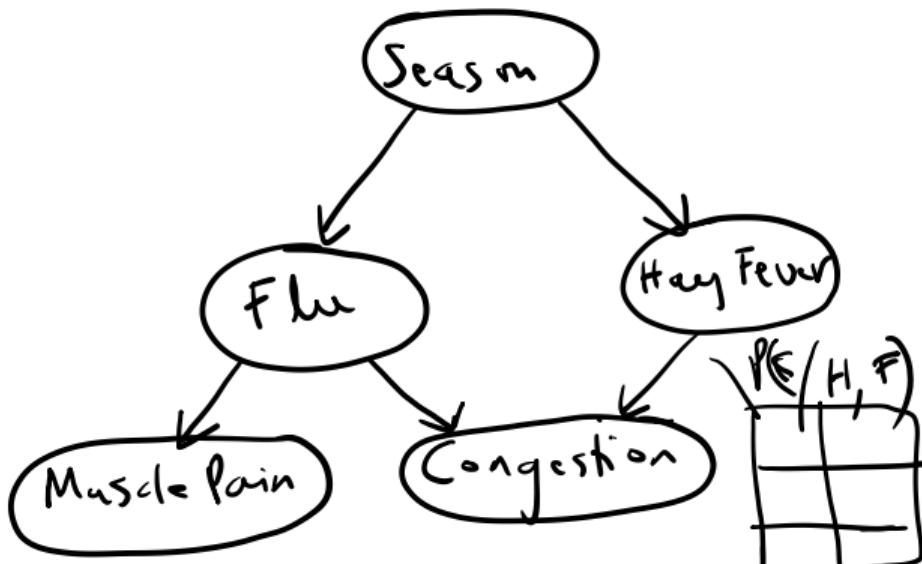
$$P(\text{Flu} = \underline{\text{True}} \mid \text{Season} = \underline{\text{Fall}}, \text{Congestion} = \underline{\text{True}}, \text{Muscle Pain} = \underline{\text{False}})$$

- Already looks complicated even when there are only 64 points in the sample space
- What happens when we have hundreds of attributes?
- Fortunately there is a way out \rightarrow can compactly encode a joint distribution over hundreds of variables using a graphical model.

$$P(F) \neq P(F/S)$$

A Detailed Look

$$P(F/S, H) = P(F/S)$$



$$P(\text{Season, flu, hay fever, ...}) = P(\text{Season}) P\left(\frac{\text{flu}}{\text{Season}}\right) P\left(\frac{\text{HF}}{\text{Season}}\right)$$

Independencies

$$(F \perp H | S)$$

$$(C \perp S | F, H)$$

$$(M \perp H, C | F)$$

$$(M \perp C | F)$$

; independent

A Detailed Look

The graph can be viewed in two perspectives

The graph is a compact representation of a set of independencies

for example

$$\begin{aligned} & P(\text{Congestion} | \text{Flu}, \text{HayFever}, \text{Season}) \\ & = P(\text{Congestion} | \text{Flu}, \text{HayFever}) \end{aligned}$$

Graph defines a skeleton for compactly representing a high-dimensional distribution

A Detailed Look

We can break up the distribution into smaller factors, each over a much smaller space of possibilities.

The overall distribution is a product of these smaller factors

factors

$$P(\text{Spring, no flu, hay fever, sinus congestion, muscle pain})$$

$$= P(\text{Season} = \text{Spring}) P(\text{Flu} = \text{false} | \text{Season} = \text{Spring}) \times$$

$$P(\text{Hay fever} = \text{true} | \text{Season} = \text{Spring}) P(\text{Congestion} = \text{true} | \text{Hay fever} = \text{true}, \text{Flu} = \text{false})$$

$$P(\text{Muscle pain} = \text{true} | \text{Flu} = \text{false})$$

A Detailed Look

This parameterisation is significantly more compact

→ requires only $\underline{3} + \underline{4} + 4 + 4 + 2$ parameters or
17 parameters → why?

$p(\text{season} = \text{spring})$, $p(\text{season} = \text{fall})$, $p(\text{season} = \text{winter})$
 $p(\text{season} = \text{summer})$ → only 3 non redundant parameters
since if we know 3 of them, the 4th can be obtained
from $1 - (\text{Sum of the given three})$

A Detailed Look

$P(\text{Flu} = \text{False} | \text{Season} = \text{Spring})$

$P(\text{Flu} = \text{False} | \text{Season} = \text{Summer})$

$P(\text{Flu} = \text{False} | \text{Season} = \text{Winter})$

$P(\text{Flu} = \text{False} | \text{Season} = \text{Fall})$

No redundant

$P(\text{Flu} = \text{True} | \text{Season} = \text{Spring})$

$P(\text{Flu} = \text{True} | \text{Season} = \text{Summer})$

$P(\text{Flu} = \text{True} | \text{Season} = \text{Winter})$

$P(\text{Flu} = \text{True} | \text{Season} = \text{Fall})$

Redundant

A Detailed Look

What about $P(\text{Congestion} | \text{Hay Fever}, \text{Flu})$?

$$P(\text{Congestion} = \text{True} | \text{Hay} = T, \text{Flu} = F)$$

$$P(\text{Congestion} = \text{True} | \text{Hay} = T, \text{Flu} = F)$$

$$P(\text{Congestion} = \text{True} | \text{Hay} = F, \text{Flu} = T)$$

$$P(\text{Congestion} = \text{True} | \text{Hay} = F, \text{Flu} = F)$$

Nonredundant

$$P(\text{Congestion} = \text{False} | \text{Hay} = T, \text{Flu} = F)$$

$$P(\text{Congestion} = \text{False} | \text{Hay} = T, \text{Flu} = T)$$

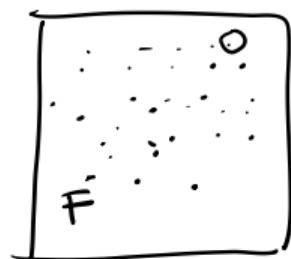
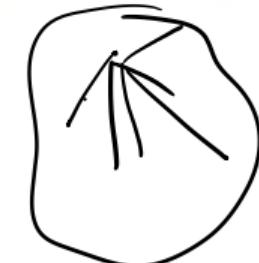
$$P(\text{Congestion} = \text{False} | \text{Hay} = F, \text{Flu} = T)$$

$$P(\text{Congestion} = \text{False} | \text{Hay} = F, \text{Flu} = F)$$

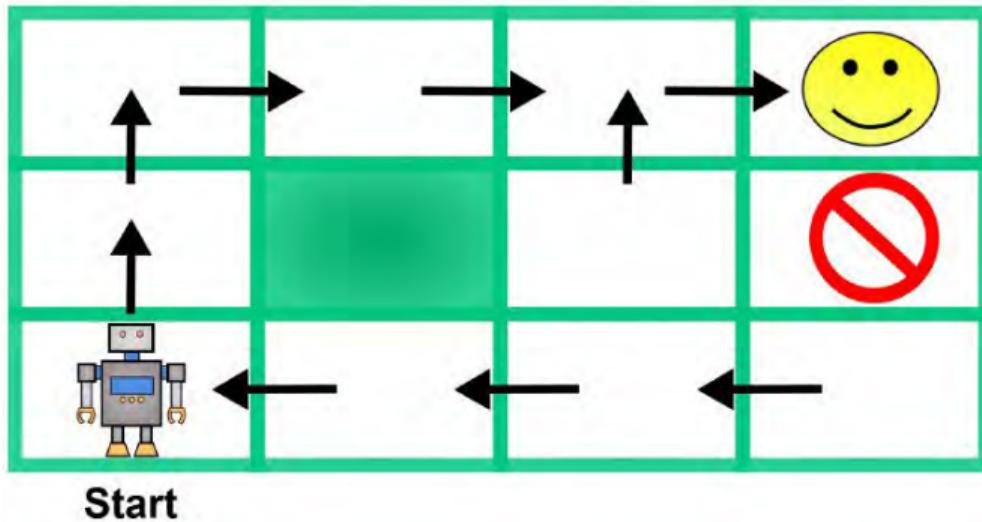
Redundant

$$P(F/N, E) = P(F/N)$$

Markov Random Fields - Image Segmentation



Markov Decision Process - Robot Motion Planning



Use Cases of PGM - Question Answer



D Sontag

Use Cases of PGM - Stereo Vision



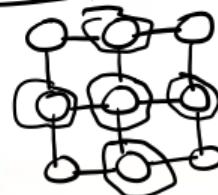
input: two images



output: disparity



Graphical Model



- node for each pixel
- infer depth for each pixel

D Sontag

MIT Ph.D

Google Translation Knowledge and Language Graph

The screenshot shows the Google Translate interface. At the top, there's a navigation bar with the Google logo, followed by "Translate", "From: English", "To: Spanish", and a "Translate" button. Below this, there are tabs for "Spanish", "Chinese", and "English". The main area displays a sentence in English on the left and its Spanish translation on the right. The English sentence is:

The top U.S. general, visiting Israel at a delicate and dangerous moment in the global standoff with Tehran, is expected to press for restraint amid fears that the Jewish state is nearing a decision to attack Iran's nuclear program.

The Spanish translation is:

El máximo general de EE.UU., de visita en Israel en un momento delicado y peligroso en el enfrentamiento global con Teherán, se espera que presione a la moderación en medio de temores de que el estado judío se acerca a una decisión de atacar el programa nuclear de Irán.

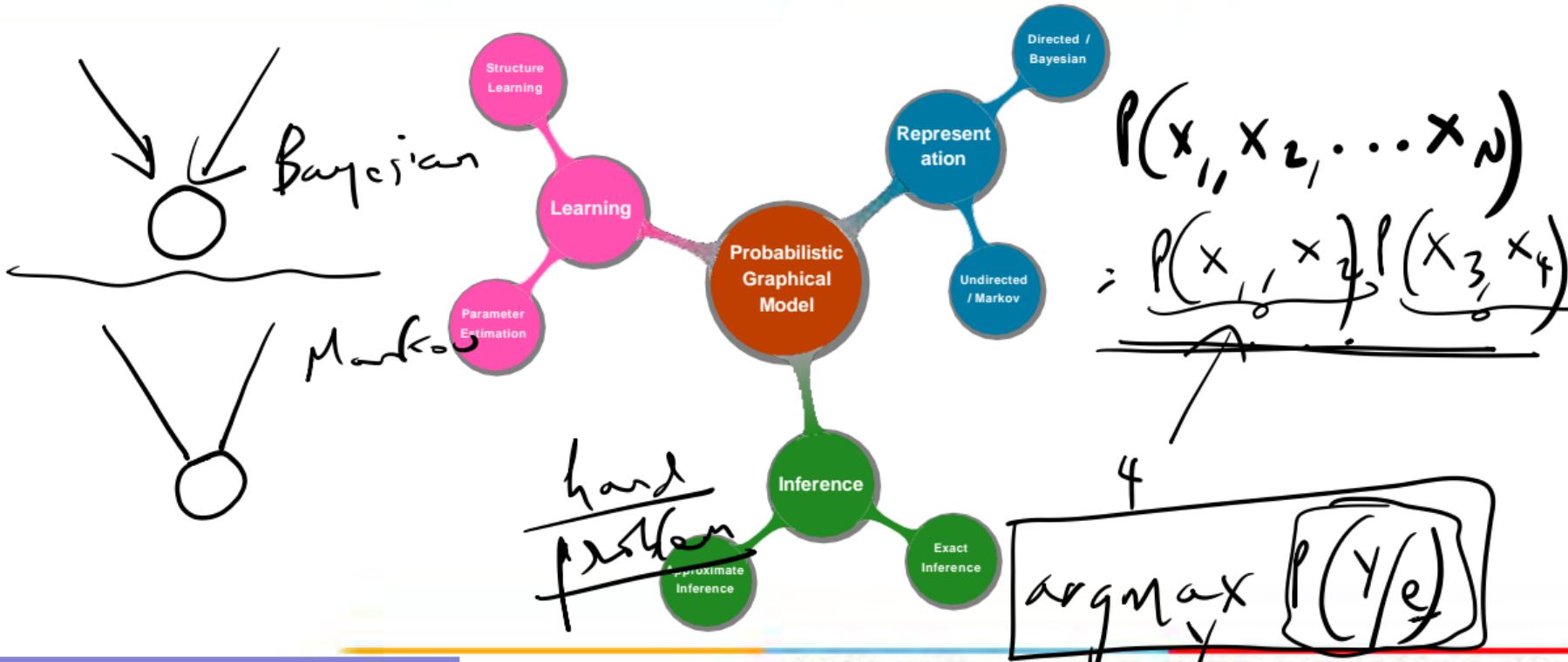
Below the translation boxes, there's a note: "New! Hold down the shift key, click, and drag the words above to reorder. [Dismiss](#)". At the bottom, there are links for "Turn off instant translation", "About Google Translate", "Mobile", "Privacy", "Help", and "Send feedback".

Table of Contents



- 1 Uncertainty
- 2 Probabilistic Graphical Model
- 3 Applications of Probabilistic Graphical Model
- 4 Course Logistics

Components of Probabilistic Graphical Model



Course Handout



- M1 Introduction
- M2 Mathematical Preliminaries
- M3 Directed Graphical Models
- M4 Undirected Graphical Models
- M5 Exact Inference
- M6 Approximate Inference
- M7 Parameter Learning
- M8 Structure Learning
- M9 Models

Lab Sessions



1 Python

2 pgmpy Library

L1 Bayesian model representation

L2 Markov Model representation

L3 MAP on Bayesian model

L4 MLE on Bayesian Model

L5 MLE on Markov Model

L6 Learning Structure in Bayesian Model

Details will be posted in Canvas.

Evaluation Components



Component	Weightage
-----------	-----------

Assignments	20 %
-------------	------

Quiz	10 %
------	------

Mid Sem	30 %
---------	------

Compre	40 %
--------	------

Further announcements will be posted in Canvas.

References

- 1 Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press. 2009
- 2 Mastering Probabilistic Graphical Models using Python by Ankur Ankan, Abhinash Panda. Packt Publishing 2015.
- 3 Learning in Graphical Models by Michael I. Jordan. MIT Press. 1999

Thank You !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 2 : MATHEMATICAL PRELIMINARIES

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.

Table of Contents



1 Uncertainty

2 Probability Theory

3 Joint Distribution

4 Graph Theory

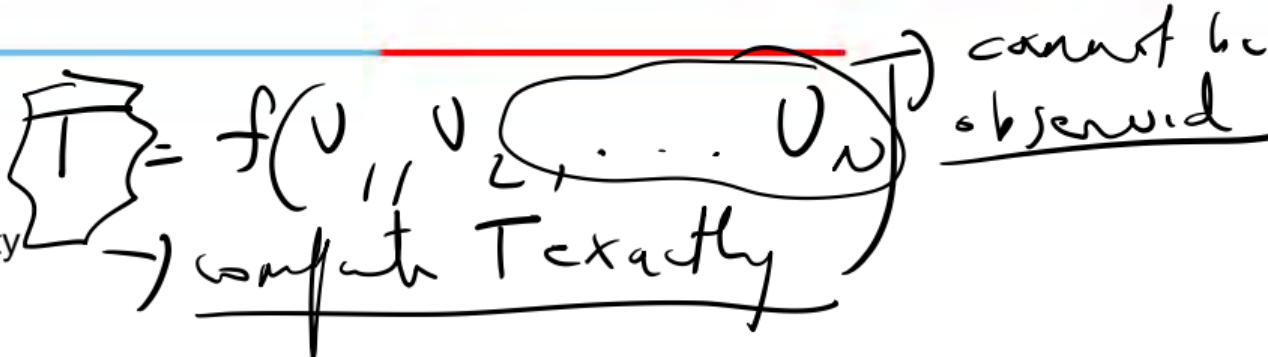
Uncertainty

- We select a course of actions among many possibilities.
- Decisions may be based on the information obtained from the environment, previous knowledge and the objectives.
- Eg: It looks cloudy. Should I carry an umbrella?
- The information and knowledge is incomplete or unreliable. So the decisions made are not certain. **We make decisions under uncertainty.**
- One of the goals of AI is to develop systems that can reason and make decisions under uncertainty.

Uncertainty

- Due to

- ▶ Partial observability
- ▶ Non-determinism



- Complexity increases

- ▶ Each piece of knowledge may not be independently used to arrive at decisions.
- ▶ Deduced facts are maintained along with new facts. This increases the knowledge base.

- Examples

- ▶ A medical doctor in an emergency.
- ▶ An autonomous vehicle that detects what might be an obstacle in its way.
- ▶ A financial agent needs to select the best investment.

$P(p_1(A), I(p_2/C) \dots I(p_n/A))$

Example

- Diagnosing a dental patients' toothache.
 - Toothache may have different causes.
- Equation using propositional logic:

$$P \Rightarrow Q, Q \Rightarrow T \\ \rightarrow P \Rightarrow T$$

$\boxed{\text{Toothache} \Rightarrow \text{Cavity} \vee \text{GumProblem} \vee \text{Abscess} \vee \dots}$

- Change to a causal rule.

~~GumProblem \Rightarrow Toothache~~
Cavity \Rightarrow Toothache

But not all cavity cause toothache.

- So make logically exhaustive.

Cavity 1 \Rightarrow Toothache
Cavity 2 \Rightarrow Toothache



Uncertain Reasoning

3 reasons for failure when using logic in Judgmental domains [medical diagnosis, law, business, design, automobile repair, gardening,]

- Laziness – complete set of antecedents and consequences
- Theoretical ignorance – no complete theory
- Practical ignorance – not all tests can be run

Belief and Degree of Belief



- Belief State is a representation of a set of all possible world states.
- Agent's knowledge can provide only a degree of belief.
- **Tool to deal with Degree of Belief is Probability Theory.**

Belief is derived from

- 1 statistical data.
- 2 some general knowledge.
- 3 combination of evidence sources.

Probabilistic Statements



- Probability statements instead of propositional logic.
- Probability statements are made with respect to knowledge state.

Example

- Probability that a patient has a cavity, given that she has toothache is 0.8.
- Probability that a patient has a cavity, given that she has toothache and a history of gum disease is 0.4.

$$P(C|T) = 0.8$$

why?

$$P(C|T, G) = 0.4$$

Table of Contents



1 Uncertainty

2 Probability Theory

3 Joint Distribution

4 Graph Theory

Sample Space

- A sample space Ω specifies set of all possible outcomes that we want to consider.

Coin toss $\Omega = \{H, T\}$

Die Roll

$\Omega = \{\square, \bullet, \cdot, \square\cdot, \square\square, \square\square\}$

isn't
sample space

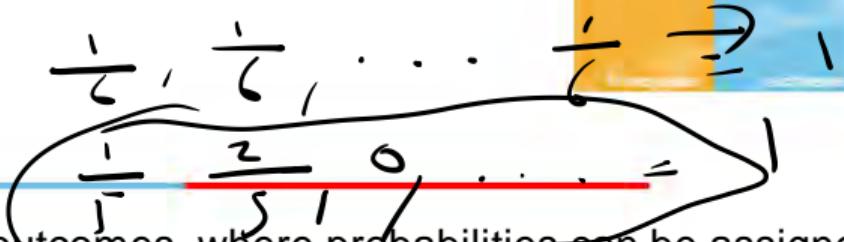
- Probability of an outcome $P(\omega)$ specifies the chance or probability with each possible outcome.

$$P(H) = 0.5$$

$$P(\square\square) = \frac{1}{6}$$

Measurable Event

lead



- An event S or Φ is a subset of outcomes, where probabilities can be assigned. We are interested in the set of outcomes.

Even die roll

$$E = \{ \text{die faces with 2 dots} \}$$

Prime die roll

$$M = \{ \text{die faces with 2, 3, 5 dots} \}$$

$$P(E) = \frac{3}{6} = \frac{1}{6} + \frac{1}{6} + \frac{1}{6}$$

Properties of Event Space

- Event space contains empty event \varnothing and the trivial event Ω .
- It is closed under union.

$$P(D) = \frac{2}{5}$$

If $\alpha, \beta \in S$, then $\overline{\alpha \cup \beta} \in S$

- It is closed under complementation.

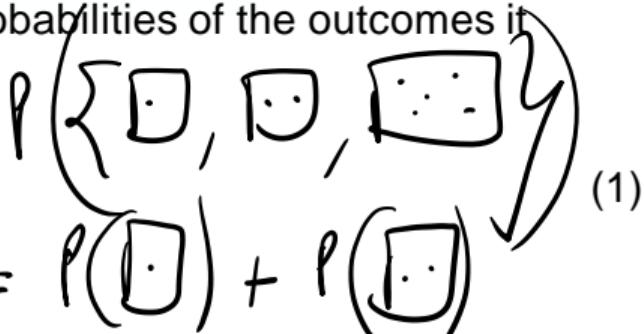
$$\alpha \rightarrow \bar{\alpha} \quad P(\bar{\alpha}) = 0$$

$$P(\bar{\alpha}) = \frac{1}{5}$$

Probability of Event

- Probability of an event is given by the sum of the probabilities of the outcomes it contains.

$$P(a) = \sum_{\omega \in a} P(\omega)$$



 $P(\{\square, \square\square, \square\square\square\})$

 $= P(\square) + P(\square\square) + P(\square\square\square)$
(1)

Even die roll $P(E) = \frac{3}{6} = 0.5$

Prime die roll $P(M) = \frac{3}{6} = 0.5$

$\neq P(\square\square\square)$

Prior Probability



- Prior or Unconditional probabilities refer to degree of belief in the absence of any other information.

$$P(\text{DieTotal} = 11) = P((5,6)) + P((6,5)) = 1/18$$

$$\frac{1}{36} + \frac{1}{36} = \frac{2}{36} = \frac{1}{18}$$

Evidence

- The information that has already been revealed is called **evidence**.
 - ▶ She is having toothache. $\text{Toothache} = \text{True}$ or $\text{toothache} = \text{false}$
 - ▶ We roll a dice and we get 5. $\text{Die}_1 = 5$



Posterior Probability

- Conditional or Posterior probability refer to the probability of some event occurring given a particular condition.

$$P(\text{cavity}|\text{toothache}) = 0.6$$

- Condition on all evidences that has been observed.

$$P(a|\beta) = \frac{P(a \wedge \beta)}{P(\beta)} \quad \text{where } P(\beta) > 0 \quad (2)$$

Probability Model



- Associate a numerical probability $P(\omega)$ with each event S .

- Axioms of probability theory

$$P(\omega) \geq 0 \quad (3)$$

$$P(\Omega) = \sum_{\omega \in \Omega} P(\omega) = 1 \quad (4)$$

$$P(A_1, A_2, \dots, A_N) = P(A_1) + P(A_2) + \dots + P(A_N) \text{ for disjoint events } A_1, A_2, \dots, A_N \quad (5)$$

$$P(a | \beta) = \frac{P(a \wedge \beta)}{P(\beta)} \quad \text{where } P(\beta) > 0 \quad (6)$$

$$P(a \vee \beta) = P(a) + P(\beta) - P(a \wedge \beta) \quad (7)$$

$$P(\alpha \vee \beta)$$

Bayes Rule



$$P(\alpha | \beta) = \frac{P(\alpha \wedge \beta)}{P(\beta)} = \frac{P(\alpha) P(\beta | \alpha)}{P(\beta)}$$

- Conditional probabilities can be derived from the prior given the evidence.

$$P(a|\beta) = \frac{P(\beta|a)P(a)}{P(\beta)} \quad \text{where } P(\beta) > 0 \quad (8)$$

$$\underline{P(\alpha \wedge \beta)} = \underline{P(\alpha) P(\beta | \alpha)} = \underline{P(\beta) \underbrace{P(\alpha | \beta)}}$$

Example 1

- Consider the student population, and let Smart denote smart students and GradeA denote students who got grade A. Based on estimates from past statistics assume that $P(\text{GradeA}|\text{Smart}) = 0.6$, the probability for students being smart is 0.3 and the prior probability of students receiving high grades is 0.2. Estimate the probability that the student is smart given GradeA.

$$\frac{P(\text{Smart})}{P(\text{GradeA})}$$

$$P(\text{GradeA}/\text{Smart}) \rightarrow P(\text{Smart}/\text{GradeA})$$

Solution

$$P(\text{Smart} | \text{Grade} = A) = \frac{P(\text{Grade} = A | \text{Smart}) \times P(\text{Smart})}{P(\text{Grade} = A)}$$

Given, $P(\text{Smart}) = 0.3$

$P(\text{Grade} = A) = 0.2$

$P(\text{Grade} = A | \text{Smart}) = 0.6$

According to Bayes' rule

$$P(\text{Smart} | \text{Grade} = A) = \frac{0.6 * 0.3}{0.2} = \underline{\underline{0.9}}$$

Example 2

- Suppose that a tuberculosis (TB) skin test is 95 percent accurate. Suppose that 1 in 1000 of the subjects who get tested is infected. What is the probability of getting a positive test result?

Solution

$$P(\text{Positive}) = P(\text{Positive} \cap TB) + P(\text{Positive} \cap \bar{TB})$$

\rightarrow not

$$P(A) = P(AB) + P(A\bar{B})$$

Given, $P(TB) = 0.001$

$P(\text{infected subjects get a positive result}) = 0.001 * 0.95$

$P(\text{uninfected subjects get a positive result}) = 0.999 * 0.05$

$$P(\text{Positive}) = \frac{0.001 * 0.95}{0.999 * 0.05} + \frac{0.999 * 0.05}{0.999 * 0.05} = 0.0509$$

According to Bayes' rule,

$$P(TB|\text{Positive}) = \frac{0.001 * 0.95}{0.0509} = \frac{0.0187}{\frac{P(\text{Positive} \cap TB)}{P(\text{Positive})}}$$

$< 2\%$



Table of Contents

1 Uncertainty

2 Probability Theory

3 Joint Distribution

4 Graph Theory

Random Variable



- Variables used in probability theory.
- Uppercase letter
- A random variable X is defined by a function that associates with each outcome in Ω a value or a state.

$$\mathbf{P}(X = x) = \mathbf{P}(\underline{\omega \in \Omega}: X(\omega) = x) \quad (9)$$

- Use $P(x)$ as a shorthand for $P(X = x)$.

$$\sum_{x \in \text{Val}(X)} P(X = x) = \sum_x P(x) = 1 \quad (10)$$

Domain of Random Variable

- Every random variable has a domain, the set of possible values it can take.
- Finite random variable or Infinite random variable.
- Domain can be discrete or continuous (integer or real).
Weather random variable – Domain = {*sunny, overcast, rainy, cloudy, snow*}
- Boolean Random variable
 - ▶ $\text{Domain} = \text{Val}(X) = \{\text{true}, \text{false}\}$
 - ▶ x^1 to denote *true* and x^0 to denote *false*.
 - ▶ The distribution of binary random variable is called a Bernoulli distribution.

$$P(X=0) = P \quad P(X=1) = 1 - P$$

Joint Distribution



- The distribution over several random variables are described using joint distribution.
- Set of random variables $X = \{X_1, X_2, \dots, X_n\}$
- Joint Distribution $\mathbf{P}(X) = P(X_1, X_2, \dots, X_n)$.
- Eg: Suppose random variable *Grade* reports the final grade of a student and the student's intelligence is given by *Intelligence*. Then the joint distribution

$\mathbf{P}(\text{Intelligence}, \text{Grade})$

$$\begin{aligned} & \mathbf{P}(\text{Grade} = A) \\ &= \mathbf{P}(G = A, I = l) + \mathbf{P}(G = A, I = h) \end{aligned}$$

$I(\text{Intelligence} = l \omega)$

$= P(I = l \omega, G = A)$

$+ P(I = l \omega, G = B)$

$+ P(I = l \omega, G = C)$

		Intelligence	
		low	high
Grade	A	0.07	0.18
	B	0.28	0.09
C	0.35	0.03	

Marginal Distribution

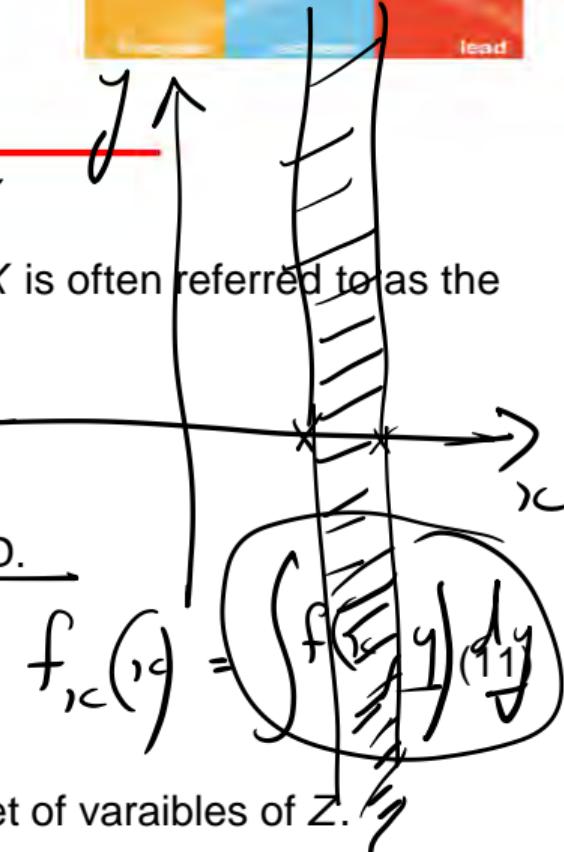
$$f(x, y) = P(X=x, Y=y)$$

- The distribution over events that can be described using X is often referred to as the marginal distribution over the random variable X .

- Summing out**
- Marginal distribution is denoted by $P(X)$.

- Row-wise or Column-wise summations in the JD gives MD.

$$P(Y) = \sum_{z \in Z} P(Y, z)$$



- Sum over all the possible combinations of values of the set of variables of Z .

$$P(X \leq 1 - \varepsilon) = 0$$

Marginal Distribution

$$P(X \leq 1 + \varepsilon) =$$



cdf ?

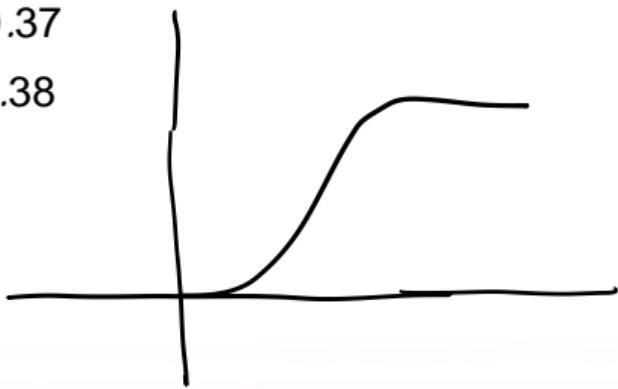
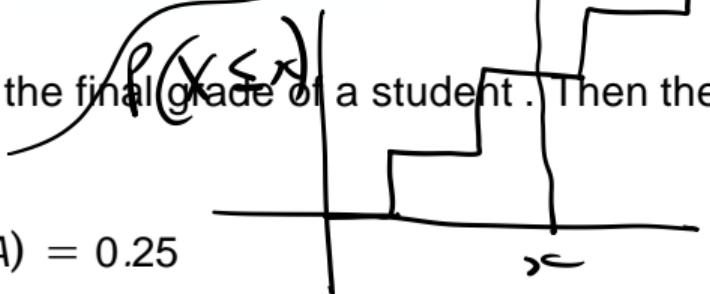
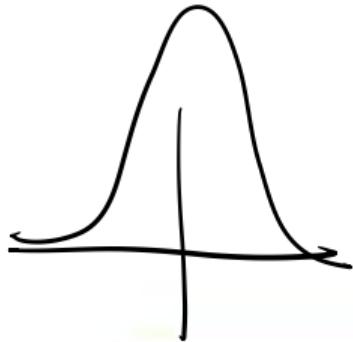
- Eg: Suppose random variable *Grade* reports the final grade of a student . Then the marginal distribution of *Grade*

$$P(\text{Grade} = A) = 0.25$$

$$P(\text{Grade} = B) = 0.37$$

$$P(\text{Grade} = C) = 0.38$$

$$P(\text{Grade}) = 1$$



Conditional Probability Distribution

$$P(Y) = P(Y \& z_1) + P(Y \& z_2) + \dots \quad \boxed{\text{marginalization}}$$

- The conditional distribution over a random variable given an observation of the value of another random variable is referred to as Conditional Probability Distribution.
- Compute conditional probability of some variable given evidence about others.

$$P(Y \& z_1) + P(Y \& z_2)$$

$$P(Y) = \sum_z P(Y \& z) P(z) \quad (12)$$

- Eg: What is the probability for the student to have high intelligence given that the grade scored is A.

$$= P(Y \& \text{Intelligence})$$

$$P(\text{Intelligence} = \text{high} | \text{Grade} = A) = \frac{0.18}{0.25} = 0.72$$

$\neq P(\text{Intelligence})$

\neq Marginal Distribution

Exercise

Given Joint Distribution $\mathbf{P}(\text{Cavity}, \text{Toothache}, \text{Catch})$ of 3 binary random variables.

		toothache		\neg toothache	
		catch	\neg catch	catch	\neg catch
cavity	toothache	0.108	0.012	0.072	0.008
	\neg cavity	0.016	0.064	0.144	0.576

Joint Distribution Entries
 $P(\text{toothache}, \text{cavity}, \neg \text{catch})$

- 1 Compute $P(\text{toothache})$
- 2 Compute $P(\text{cavity})$
- 3 Compute $P(\text{cavity} | \text{toothache})$ given the evidence of toothache .

Solution

$$\begin{aligned}
 P(\text{toothache}) &= P(\text{toothache}, \underline{\text{cavity}}, \underline{\text{catch}}) \\
 &+ P(\text{toothache}, \underline{\neg \text{cavity}}, \underline{\text{catch}}) + P(\text{toothache}, \underline{\text{cavity}}, \underline{\neg \text{catch}}) \\
 &+ P(\text{toothache}, \underline{\neg \text{cavity}}, \underline{\neg \text{catch}}) \\
 &= 0.108 + 0.016 + 0.012 + 0.064 \\
 &= 0.20
 \end{aligned}$$

$$P(\text{cavity}) = 0.108 + 0.012 + 0.072 + 0.008 = 0.2$$

$$P(\text{cavity}|\text{toothache}) = \frac{P(\text{cavity} \wedge \text{toothache})}{P(\text{toothache})} = \frac{0.108 + 0.012}{0.20} = \underline{\underline{\frac{12}{20}}}$$

Independence of Events



- An event α is independent of event β in P ,

$$\begin{array}{l} \square P(\alpha | \beta) = P(\alpha) \text{ or} \\ \square \end{array}$$

$$\alpha \perp \beta \quad \text{if} \quad P(\beta | \alpha) = P(\beta) \quad \text{or} \quad (13)$$

$$\square P(\alpha \wedge \beta) = P(\alpha)P(\beta)$$

- $\alpha \perp \beta$ implies $\beta \perp \alpha$
- Eg: Tossing two coins, Rolling a die.

Note that $\alpha \perp \beta$ if $P(\beta) = 0$.

Independence of Random Variables

lead

- Two random variables X and Y can be independent of each other.
- A random variable X is independent of another random variable Y ,

$$X \perp Y \quad \text{if} \quad \begin{array}{l} P(X | Y) = P(X) \quad \text{or} \\ P(Y | X) = P(Y) \quad \text{or} \\ P(X \wedge Y) = P(X)P(Y) \end{array} \quad (14)$$

- $X \perp Y$ implies $Y \perp X$

- Eg: Weather is independent of dental problems.

$$P(X \wedge Y) = P(X)I(Y/X) \rightarrow P(X)P(Y)$$

Conditional Independence of Events

- An event α is conditionally independent of event β given event γ in P

$$\boxed{\begin{array}{l} \square P(\alpha \cap \beta \mid \gamma) = P(\alpha \mid \gamma) \\ \square \quad \text{if } P(\beta \mid \gamma) = 0 \\ \square P(\alpha \wedge \beta \mid \gamma) = P(\alpha \mid \gamma)P(\beta \mid \gamma) \end{array}} \quad (15)$$

- Eg: Getting Admission in MIT is independent of getting admission in Stanford, given the student has scored Grade A.

$$P(\text{MIT/Stanford, Grade A}) = P(\text{MIT/Grade A}) \xrightarrow[\text{Stanford}]{} \xrightarrow[\text{Grade A}]{} \text{MIT Admission}$$

Conditional Independence of Random Variables



- A random variable X is conditionally independent of random variable Y given random variable Z

$$\begin{array}{c} \square \\ \square P(X | Y, Z) = P(X | Z) \\ X \perp Y | Z \quad \text{if} \quad P(Y, Z) \quad = \bigcirc \\ \square \\ \square P(X, Y | Z) = P(X | Z)P(Y | Z) \end{array} \quad (16)$$

- Eg: The random variables Toothache and Catch are independent given the presence or absence of Cavity.

Student Example



- Model the difficulty of a course, intelligence of students, Grade the students score in a particular course.
- Let D represent the difficulty of a course.

Domain of D = {easy, hard}

$$\mathbf{P(D)} = \{d^0, d^1\}$$

$$= \{0.6, 0.4\}$$

- Let I represent the intelligence of a student.

Domain of I = {low, high}

$$\mathbf{P(I)} = \{i^0, i^1\}$$

$$= \{0.7, 0.3\}$$

Student Example

- Let G represent the grade a student gets for a course.

Domain of \mathbf{G} = {A, B, C}

$$\mathbf{P}(\mathbf{G}) = \{g^1, g^2, g^3\}$$

- $\mathbf{P}(\mathbf{D}, \mathbf{I}, \mathbf{G})$ denotes the probabilities of all combinations of the values of the 3 random variables.
L
- These $2 * 3 * 3 = 12$ values can be represented using a Joint Distribution Table.

Joint Probability Distribution

$$P(\text{high, high, easy})$$

$$P(\text{low, low, difficult})$$

- Joint Probability Distribution completely represents the joint distribution for all random variables.
- In the students example, the $P(D, I, G)$, the 12 parameters cannot be determined by the value of the other parameters. Hence called Independent parameters.
- Independent parameters are parameters whose values are not completely determined by the values of the other parameters.

P	d^0	g^0	→
i^0	d^0	g^1	→
i^0	d^0	g^2	→

Student Example - Joint Distribution

<i>I</i>	<i>D</i>	<i>G</i>	<i>P</i>
i^0	d^0	g^1	0.126
		g^2	0.168
		g^3	0.126
i^0	d^1	g^1	0.009
		g^2	0.045
		g^3	0.126
i^1	d^0	g^1	0.052
		g^2	0.0224
		g^3	0.0056
i^1	d^1	g^1	0.069
		g^2	0.036
		g^3	0.024

$P(g_1 | i^0, d^0)$
 $P(g_2 | i^0, d^0)$
 $P(g_3 | i^0, d^0)$
 $P(D, I, G) = 1$
 $P(g_1 | i^1, d^1)$

$$P(X=x, Y=y | Z=z) = P(X=x | Z=z) P(Y=y | Z=z)$$

Conditional Independence

Let X, Y, Z be sets of random variables. We say that X is conditionally independent of Y given Z in a distribution P if P satisfies $(X = x \perp Y = y | Z = z)$ for all values $x \in \text{Val}(X)$, $y \in \text{Val}(Y)$, and $z \in \text{Val}(Z)$. The variables in the set Z are often said to be observed. If the set observed variable Z is empty, then instead of writing $(X \perp Y | \emptyset)$, we write $(X \perp Y)$ and say that X and Y are marginally independent.

(Definition 2.4 from Daphne Koller's book)

Conditional Independence

Proposition 2.3 The distribution P satisfies $(X \perp Y | Z)$ if and only if $P(X, Y | Z) = P(X | Z)P(Y | Z)$. Suppose we learn about a conditional independence.

Can we conclude other independence properties that must hold in the distribution?

Symmetry: $(X \perp Y | Z) \Rightarrow (Y \perp X | Z)$.

Properties that hold

(2.7) Decomposition: $(X \perp Y, W | Z) \Rightarrow (X \perp Y | Z)$.

→ (2.8) Weak union: $(X \perp Y, W | Z) \Rightarrow (X \perp Y | Z, W)$.

→ (2.9) Contraction: $(X \perp W | Z, Y) \& (X \perp Y | Z) \Rightarrow (X \perp Y, W | Z)$.

$$P(X) = \sum_Y P(X, Y)$$

Why is decomposition true?

$$P(X, Y | Z) = \underbrace{\sum_w P(X, Y, w | Z)}_{\text{Summing over all } \omega} \quad (\text{Summing over all } \omega)$$

$$= \underbrace{\sum_{\omega} P(X | Z)}_{\text{because } (X \perp Y, W | Z)} P(Y, w | Z)$$

$$= P(X | Z) \left(\sum_{\omega} P(Y, w | Z) \right). \quad \text{But } \sum_{\omega} P(Y, \omega | Z) = \underbrace{P(Y | Z)}_{\text{marginal}}$$

$$= P(X | Z) P(Y | Z).$$

Why is weak union property true?

Given $X \perp Y, \omega/z$. To prove $X \perp Y/\omega, z$

First we note that $P(X, Y/z, \omega) = P(X/y, \omega, z)P(Y/\omega, z)$

Now since $X \perp Y, \omega/z$ we see that $P(X/y, \omega, z) = P(X/z)$

Thus we have $P(X, Y/z, \omega) = P(X/z)P(Y/\omega, z) - (1)$

If we replace $P(X/z)$ by $P(X/\omega, z)$ in (1) we are done.
How is this move justified?

Why is the weak union property true?

From Decomposition we have $X \perp Y, W/Z \Rightarrow X \perp W/Z$

Thus we have $P(X/W, Z) = P(X/Z)$ and so ①

becomes

$$P(X, Y/W, Z) = P(X/W, Z)P(Y/W, Z)$$

What about contraction?

We are given $X \perp \omega | Z, Y$ & $X \perp Y | Z$. To prove

$$X \perp Y, \omega | Z.$$

Proof: $P(X, Y, \omega | Z) = P(X | Y, \omega, Z) P(Y, \omega | Z)$ from Bayes rule

Now since $X \perp \omega | Z, Y$ we can see that

$P(X | Y, \omega, Z) = P(X | Y, Z)$. Further since $X \perp Y | Z$ this means $P(X | Y, Z) = P(X | Z)$. Thus $P(X, Y, \omega | Z) = P(X | Z) P(Y, \omega | Z)$ or $X \perp Y, \omega | Z$ as needed.

Querying a Distribution

Compute $P(Y|E=e)$

E = evidence random variables instantiated to e .

$Y \rightarrow$ query variables, a subset of random variables in the network

We want the posterior probability distribution over the values y given that $E=e$

MAP Query

Find a high probability joint assignment to some subset of variables

$$\text{MAP}(\omega | e) = \arg \max_{\omega} P(\omega, e)$$

How is a MAP query different from a probability query?

MAP Query

To find the most likely assignment to a single variable A, we could simply compute $P(A | e)$ and then pick the most likely value.

The assignment where each variable individually picks its most likely value can be quite different from the most likely joint assignment to all variables simultaneously.

[Joint Distribution Matters]

Example

Example 2.4 from the textbook [Daphne Koller]

Let random variables A and B be both binary valued

a^o	b^o
a^i	b^i
0.4	0.6

		A	
		b^o	b^i
a^o	b^o	0.1	0.9
	b^i	0.5	0.5

} These are conditional probabilities
 $P(B = b^i | A = a^j)$

Example

Now $\text{MAP}(A) = a'$ since $P(a') > P(\hat{a})$

What is $\text{MAP}(A, B)$?

It is (a', b') since $P(a', b') = P(a') P(b' | a')$

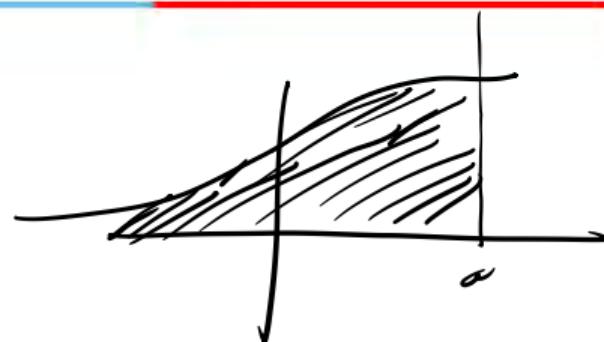
$= 0.4 \times 0.9 = 0.36$ is greater than for any other combination.

$\boxed{\arg\max_{a,b} P(a,b) \neq (\arg\max_a P(a), \arg\max_b P(b))}$

Continuous Spaces

$$\int p(x) dx = 1$$

Val(X)



$$P(X \leq a) = \int_{-\infty}^a p(x) dx$$

$$\sum \rightarrow \int$$

$$P(a \leq X \leq b) = \int_a^b p(x) dx \rightarrow \sum_{a \leq x_i \leq b} p(x_i)$$

Continuous Space

$$P(a_1 \leq X_1 \leq b_1, a_2 \leq X_2 \leq b_2, \dots, a_n \leq X_n \leq b_n)$$

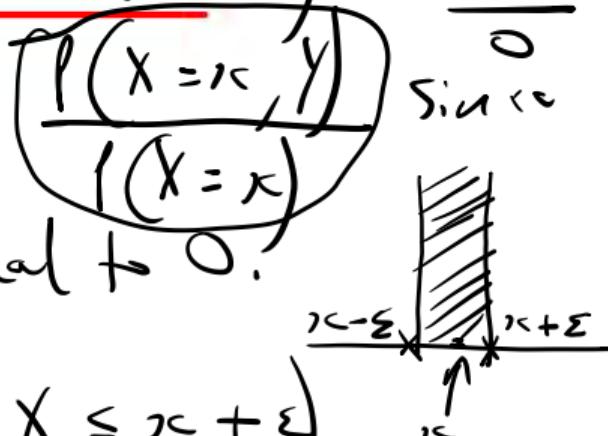
$$= \int_{a_1}^{b_1} \int_{a_2}^{b_2} \dots \int_{a_n}^{b_n} P(x_1, x_2, \dots, x_n) dx_n dx_{n-1} \dots dx_1$$

P is integrable, joint density function,
 $P(x_1, x_2, \dots, x_n) \geq 0$ for all values
 x_1, x_2, \dots, x_n

Conditional Density Functions

$$P(X = x_0) = \frac{0}{0}$$

We cannot write $P(Y/X = x)$ as $\frac{P(X=x, Y)}{P(X=x)}$ since both these probabilities are equal to 0.



Define $P(Y/x)$ as $\lim_{\varepsilon \rightarrow 0} P(Y/x - \varepsilon \leq X \leq x + \varepsilon)$

$$P(a \leq Y \leq b | x - \varepsilon \leq X \leq x + \varepsilon) = \frac{P(a \leq Y \leq b, x - \varepsilon \leq X \leq x + \varepsilon)}{P(x - \varepsilon \leq X \leq x + \varepsilon)}$$

Conditional Density Functions

$$\frac{\int \int_{\substack{y \\ a < x' - \varepsilon}}^{\substack{y \\ x + \varepsilon}} p(x', y) dx' dy}{\int_{\substack{x' - \varepsilon \\ x - \varepsilon}}^{x + \varepsilon} p(x') dx'} = \frac{\int_a^b 2\varepsilon p(x, y) dy}{2\varepsilon p(x)}$$

Conditional Density Functions

$$P(a \leq Y \leq b / c - \varepsilon \leq X \leq c + \varepsilon)$$

$$\approx \int_a^b 2\varepsilon p(x, y) dy$$

$$\frac{2\varepsilon p(x)}{\int_a^{c+\varepsilon} p(x) dx}$$

$$= \int_a^b \frac{p(x, y)}{p(x)} dy = \text{a constant}$$

$$\therefore P(Y/x = c) = \frac{p(x, y)}{p(x)}$$

Table of Contents



1 Uncertainty

2 Probability Theory

3 Joint Distribution

4 Graph Theory

Graph

2^N possible values \rightarrow regular JPDF

- Data structure used to represent the probability distribution of data.
- A graph is a data structure K consisting of a set of nodes and a set of edges.

Graph $K = (\underline{X}, \underline{E})$

$$X \longrightarrow Y \quad (17)$$

- The set of nodes denote each random variable.

 causality

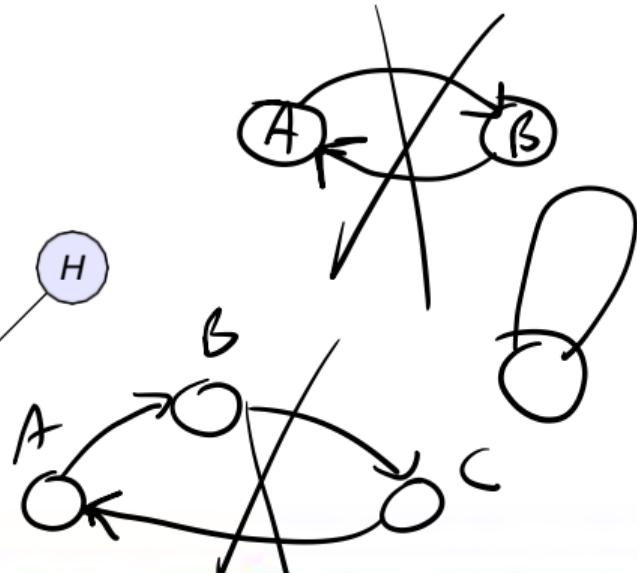
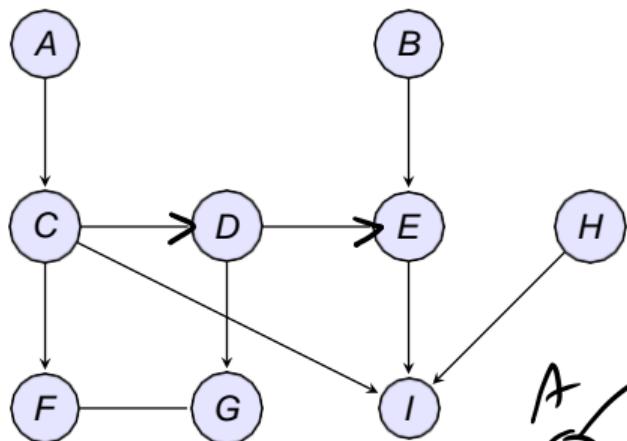
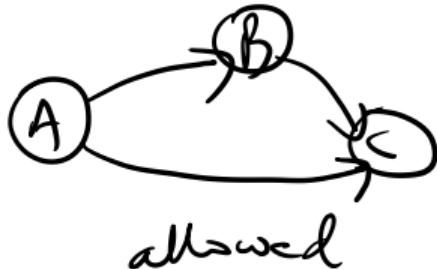
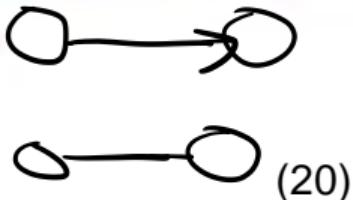
Set of Nodes $X = \{X_1 \dots X_n\}$ (18)

- A pair of nodes X_i, X_j can be connected by a directed edge $X_i \rightarrow X_j$ or an undirected edge $X_i - X_j$.

Set of Edges $E = X_i \rightarrow X_j \text{ or } X_i - X_j$ (19)

Directed Graph

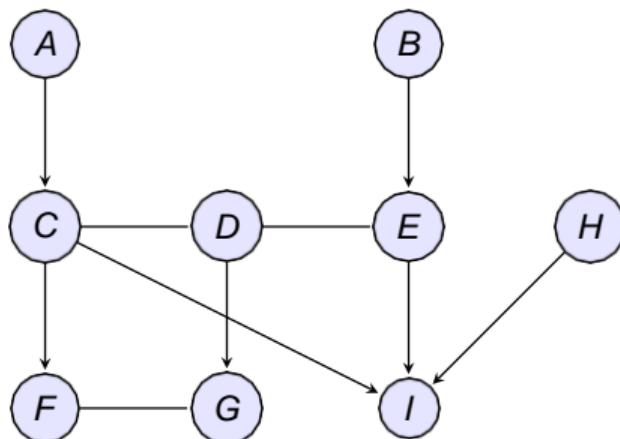
- A graph is directed if all edges are directed. $X_i \rightarrow X_j$.
- $G = (X, E)$ where $E = \{X_i \rightarrow X_j\}$



Undirected Graph

- A graph is undirected if all edges are undirected. $X_i - X_j$.

$$H = (X, E) \text{ where } E = \{X_i - X_j\} \quad (21)$$



Parent and Child



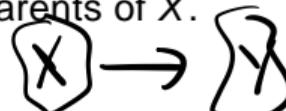
Graph $K = (X, E)$

where $E = \{X \rightarrow Y\}$

■ Parent

- ▶ X is called the parent of Y .
- ▶ Pa_X denote parents of X .

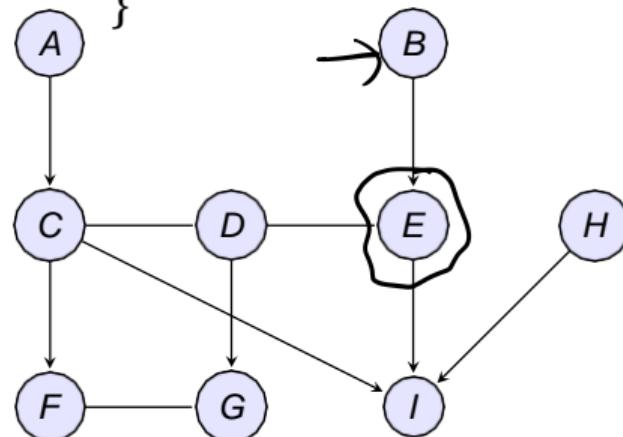
■ Child



- ▶ Y is called the child of X .
- ▶ Ch_X denote children of X .

■ Example: Identify the parents and children of Node E .

Ans: $\text{Pa}_E = B$ $\text{Ch}_E = I$



Neighbor and Boundary



■ Neighbor

- Whenever $X \rightarrow Y \in E$, X and Y are adjacent.
- Nb_x denote neighbors of X .

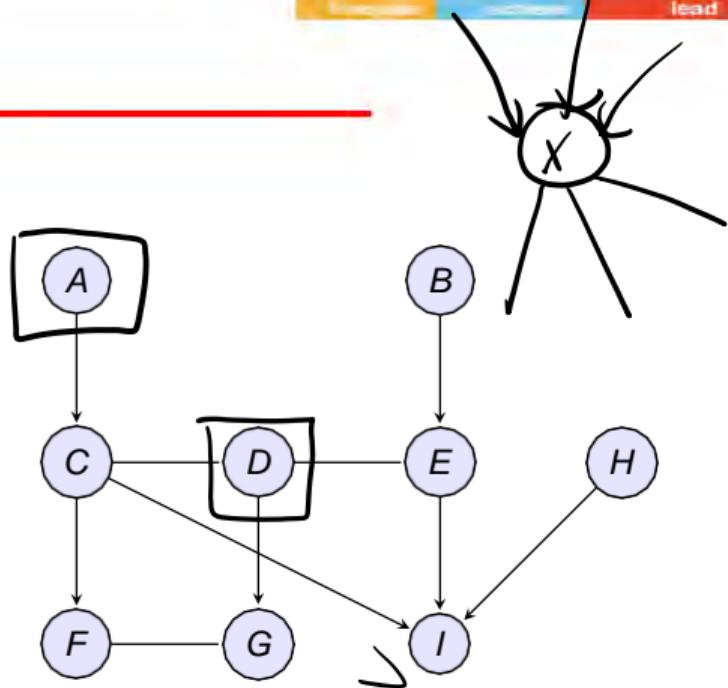
■ Boundary

- In Directed graph, $\text{Boundary}_x = Pax$
- In Undirected graph, $\text{Boundary}_x = Nb_x$.

$$\text{Boundary}_x = Pax \cup Nb_x$$

- Example: Identify the neighbors and boundary of Node C .

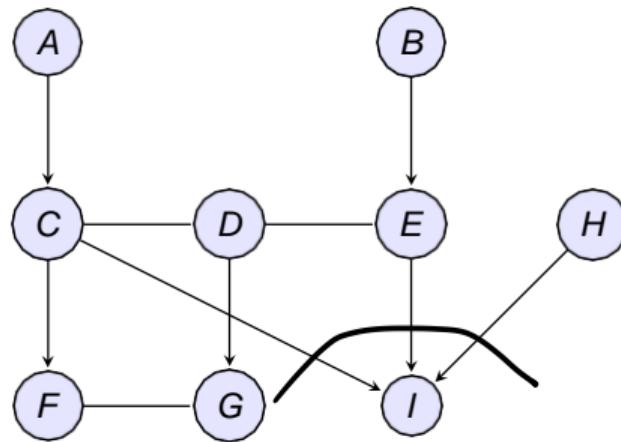
Ans: $Nb_C = \cancel{A}, \cancel{D}, \cancel{E}$ $B_C = A, D, \cancel{E}$



Degree of a Graph



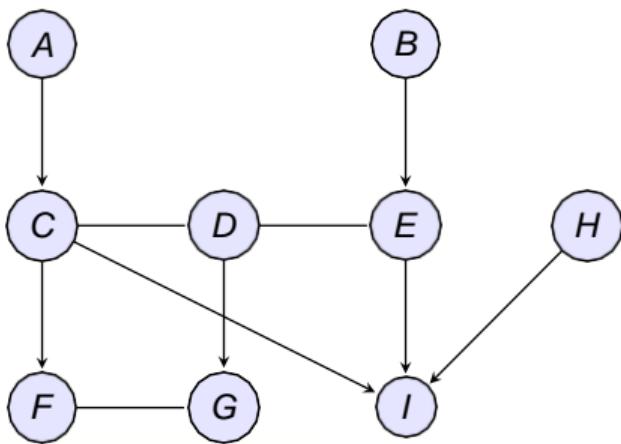
- The **degree** of a node X is the number of edges in which it participates.
- The **in-degree** of a node is the number of directed edges $Y \rightarrow X$.
- The **degree of a graph** is the maximal degree of a node in the graph.
- Example: Identify the degree Node
I. Ans=3



Path

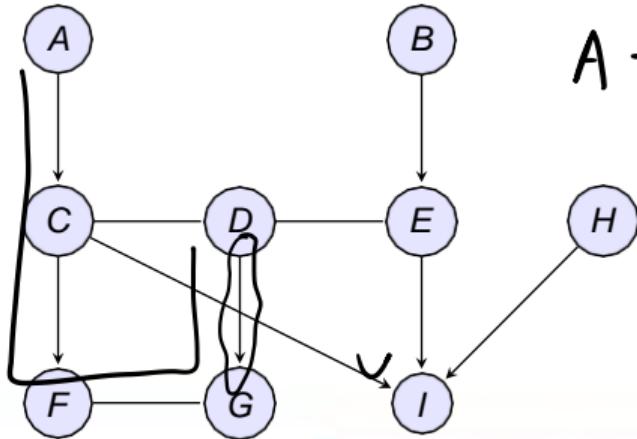
- X_1, \dots, X_k form a path in the graph $K = (X, E)$, if for every $i = 1, \dots, k - 1$ we have either $X_i \rightarrow X_{i+1}$ or $X_i = X_{i+1}$.
- Example: Identify a path. Ans: $A \rightarrow C \rightarrow I$

$X_i \leftarrow X_{i+1}$
 $X_i \rightarrow X_{i+1}$



Trail

- X_1, \dots, X_k form a **trail** in the graph $K = (X, E)$, if for every $i = 1, \dots, k - 1$ we have either $X_i \leftrightarrow X_{i+1}$ or $X_i \geq X_{i+1}$ (any sort of edge)
- A graph is **connected**, if there is a trail between X_i and X_j .
- Example: Identify a trail. Ans: $A \rightarrow C - D - E \rightarrow I$ (both path & trail)



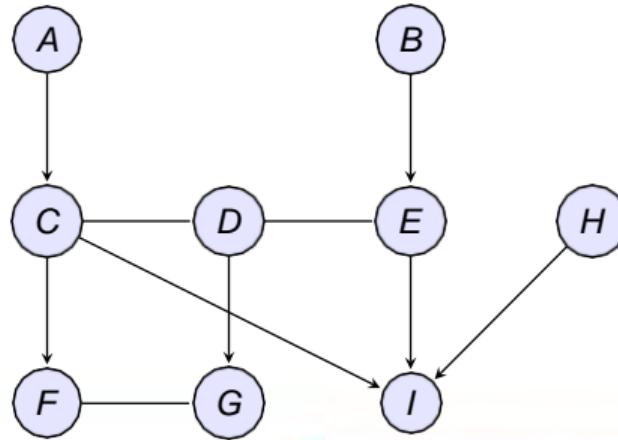
$A - C - F - G - D$

(trail, but not a path)

Ancestor and Descendant



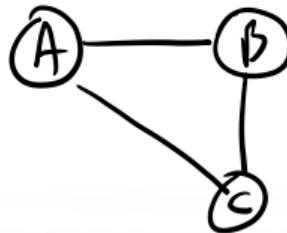
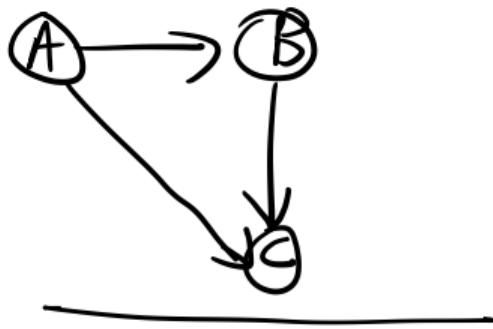
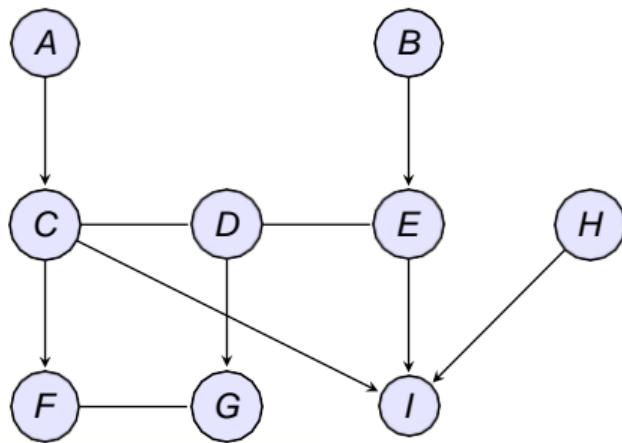
- X is an **ancestor** of Y in a graph K if there is a directed path X_1, \dots, X_k with $X_1 = X$ and $X_k = Y$.
- Y is the **descendant** of X .
- $Ancestor_X$ and $Descendant_X$



Cycle

Directed Acyclic
DAG Graph

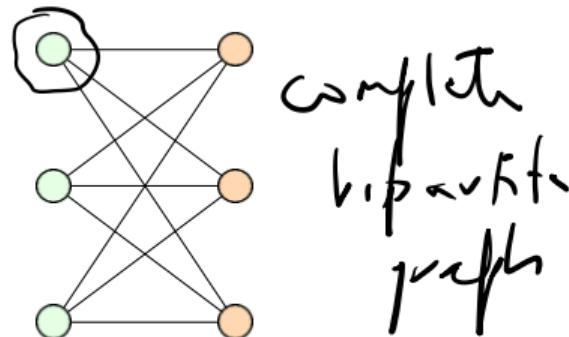
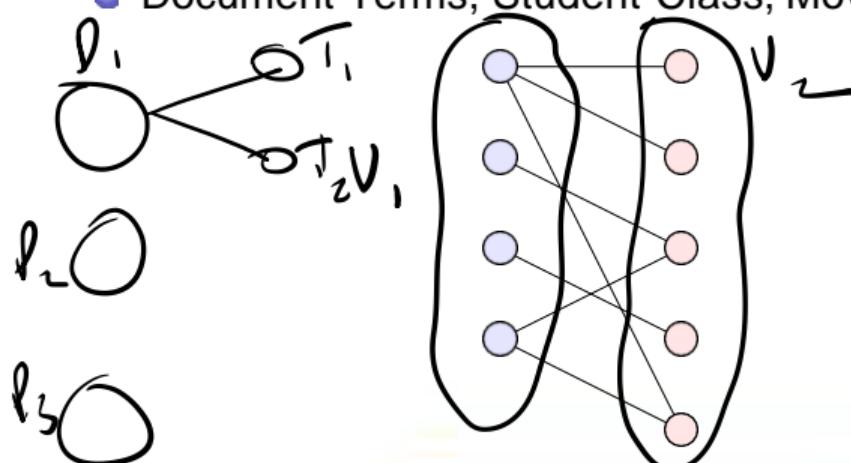
- A **cycle** in graph K is a directed path X_1, \dots, X_k with $X_1 = X_k$.
 - Example: Identify a cycle.
- Ans: No cycle



Bipartite Graphs

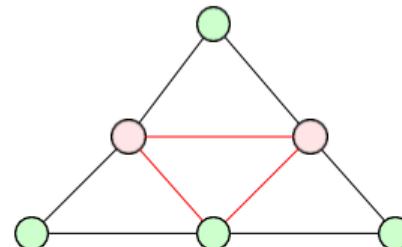
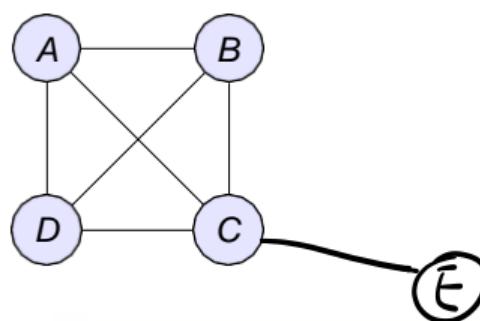


- If the vertex-set of a graph G can be split into two disjoint sets, V_1 and V_2 , in such a way that each edge in the graph joins a vertex in V_1 to a vertex in V_2 , and there are no edges in G that connect two vertices in V_1 or two vertices in V_2 , then the graph G is called a bipartite graph.
- Document-Terms, Student-Class, Movie preference of viewers



Clique

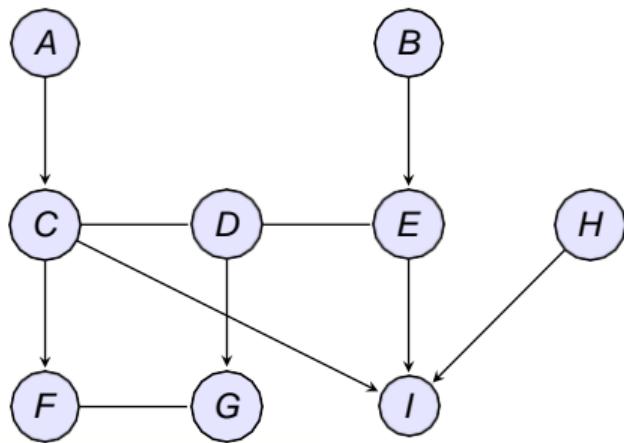
- Clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent.
- A maximum clique of a graph, is a clique, such that there is no clique with more vertices.



Directed Acyclic Graph (DAG)



- A graph is acyclic if it contains no cycles.
- A directed acyclic graph is a graph that has directed edges but no cycles.
- DAG is the basic graphical representation of Bayesian Networks.





References

- 1 Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press. 2009
- 2 Artificial Intelligence: A Modern Approach (3rd Edition) by Stuart Russell, Peter Norvig
- 3 Mastering Probabilistic Graphical Models using Python by Ankur Ankan, Abhinash Panda. Packt Publishing 2015.
- 4 Learning in Graphical Models by Michael I. Jordan. MIT Press. 1999

Thank You !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 3 : BAYESIAN MODEL

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.

TABLE OF CONTENTS

1 PROBABILISTIC GRAPHICAL MODEL

2 JOINT DISTRIBUTION

3 FACTOR

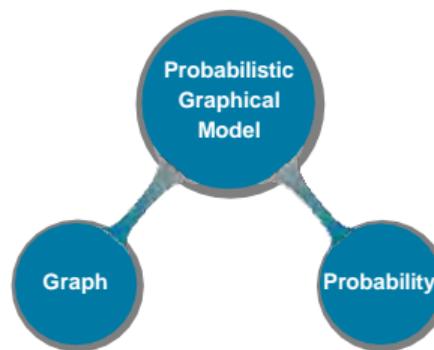
4 INDEPENDENCE

5 BAYESIAN NETWORK

6 HOME WORK

PROBABILISTIC GRAPHICAL MODELS

- Probabilistic Graphical Model is a model that is standalone, where probability distributions and its semantics represent uncertainty about state of world.



COMPONENTS OF PROBABILISTIC GRAPHICAL MODEL

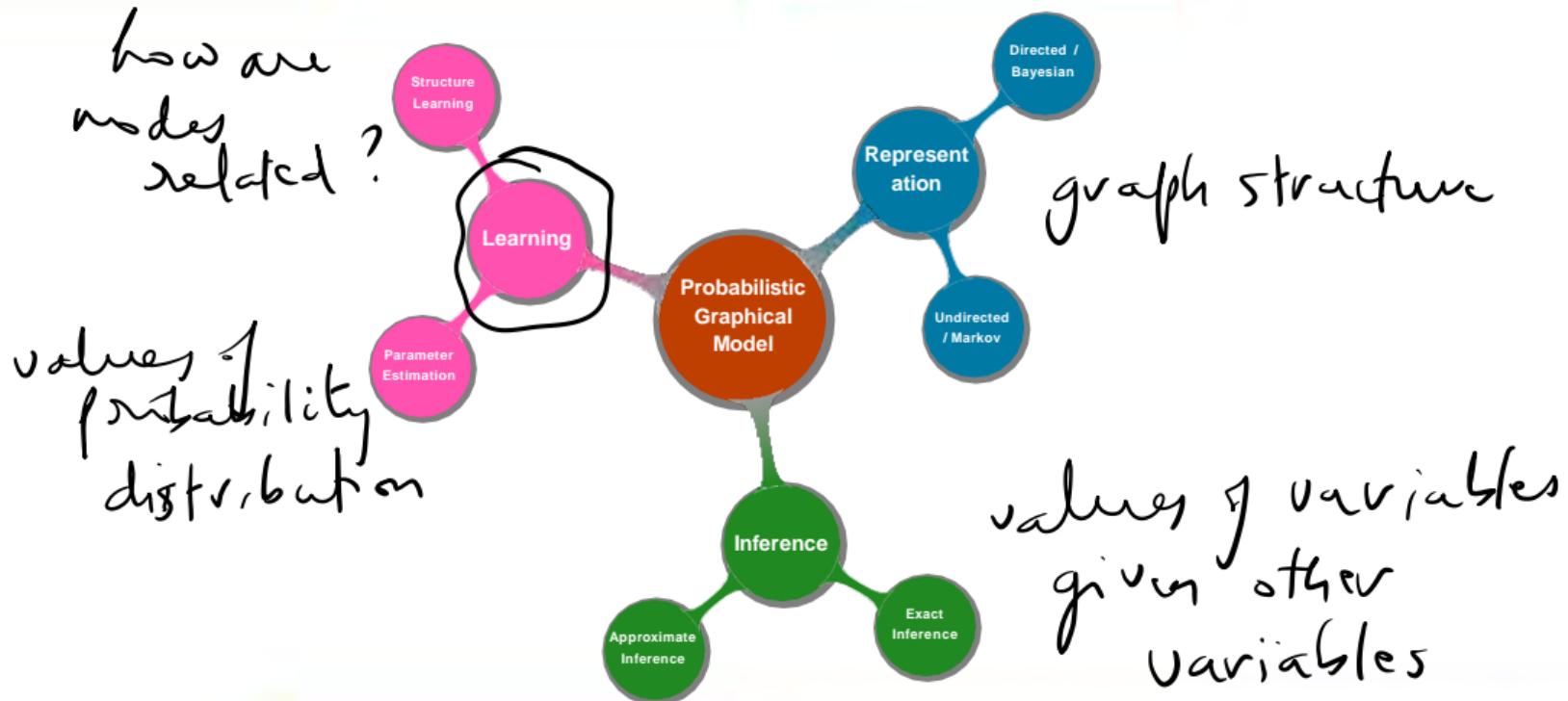


TABLE OF CONTENTS

1 PROBABILISTIC GRAPHICAL MODEL

2 JOINT DISTRIBUTION

3 FACTOR

4 INDEPENDENCE

5 BAYESIAN NETWORK

6 HOME WORK

STUDENT EXAMPLE

- Model the difficulty of a course, intelligence of students, Grade the students score in a particular course.
- Let D represent the difficulty of a course.

$$\begin{aligned} \text{Domain of } D &= \{\underline{\text{easy}}, \underline{\text{hard}}\} = \{\underline{d^0}, \underline{d^1}\} \\ P(D) &= \{\underline{0.6}, \underline{0.4}\} \end{aligned}$$

- Let I represent the intelligence of a student.

$$\begin{aligned} \text{Domain of } I &= \{\underline{\text{low}}, \underline{\text{high}}\} = \{\underline{i^0}, \underline{i^1}\} \\ P(I) &= \{\underline{0.7}, \underline{0.3}\} \end{aligned}$$

STUDENT EXAMPLE

- Let G represent the grade a student gets for a course.

$$\text{Domain of } G = \{A, B, C\} = \{g^1, g^1, g^2\}$$

- How do we represent Joint distribution of the 3 random variables? How many parameters are required?
- $P(I, D, G)$ denotes the probabilities of all combinations of the values of the 3 random variables.
- These 2 *2 *3 = 12 parameters can be represented using a Joint Distribution.

STUDENT EXAMPLE - JOINT DISTRIBUTION

I	D	G	$P(I, D, G)$
i^0	d^0	$\underline{g^1}$	0.126
		$\underline{g^2}$	0.168
		$\underline{g^3}$	0.126
i^0	d^1	$\underline{g^1}$	0.009
		$\underline{g^2}$	0.045
		$\underline{g^3}$	0.126
i^1	d^0	$\underline{g^1}$	0.252
		$\underline{g^2}$	0.0224
		$\underline{g^3}$	0.0056
i^1	d^1	$\underline{g^1}$	0.060
		$\underline{g^2}$	0.036
		$\underline{g^3}$	0.024

(i^0, d^0, g^1)
 (i^0, d^0, g^2)
 (i^0, d^0, g^3)
 (i^0, d^1, g^1)
 (i^0, d^1, g^2)
 (i^0, d^1, g^3)
 (i^1, d^0, g^1)
 (i^1, d^0, g^2)
 (i^1, d^0, g^3)
 (i^1, d^1, g^1)
 (i^1, d^1, g^2)
 (i^1, d^1, g^3)

What is the sum of the joint distribution?

$$\sum P(I, D, G) = 1 \quad (1)$$

OPERATIONS ON JOINT DISTRIBUTION

- 1 Conditioning
- 2 Renormalization
- 3 Marginalization

1. CONDITIONING ON JOINT DISTRIBUTION

- Suppose a student score 'A' grade.
- Observation: $G = g^1$.
- This conditioning gives a reduced Joint distribution.
- Conditioning reduces Joint distribution.

I	D	G	$P(I, D, g^1)$
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	0.060

What is sum of the distribution now?

$$\sum P(I, D, g^1) \cancel{=} 1$$

(2)

2. RENORMALIZATION OF CONDITIONED JD

I	D	G	$P(I, D, g^1)$
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	0.060
			0.447

normalize

I	D	G	$P(I, D g^1)$
i^0	d^0	g^1	$0.126/\underline{0.447} = 0.282$
i^0	d^1	g^1	$0.009/\underline{0.447} = 0.020$
i^1	d^0	g^1	$0.252/0.447 = 0.564$
i^1	d^1	g^1	$0.060/0.447 = 0.134$

$$P(i^0, d^1 | g^1) = 0.282 \quad (3)$$

$$P(i^0, d^1 | g^1) = 0.020$$

$$P(I, D, g^1) \xrightarrow{\text{normalize}} P(I, D | g^1)$$

3. MARGINALIZATION ON JD

Marginalization on JD = Summing Out

I	D	$P(I, D)$
i^0	d^0	0.282
i^0	d^1	0.020
i^1	d^0	0.564
i^1	d^1	0.134

$$P(D = d^0) = P(I=i^0, D=d^0) + P(I=i^1, D=d^0)$$

D	$P(D)$
d^0	0.846
d^1	0.154

$$P(D = D_s) = \sum_I P(I, D = D_s)$$

$$\sum_I P(I, D) = P(D) \quad (4)$$

TABLE OF CONTENTS

1 PROBABILISTIC GRAPHICAL MODEL

2 JOINT DISTRIBUTION

3 FACTOR

4 INDEPENDENCE

5 BAYESIAN NETWORK

6 HOME WORK

FACTOR $P(A, B, C, D) = \frac{1}{Z} \phi_1(A, B) \phi_2(B, C) \phi_3(C, D) \phi_4(D, A)$

~~normalizing~~

- A **factor** Φ is a function or a table that maps a set of random variables to a real value.

$$\Phi : \underline{\text{Val}(X_1, \dots, X_n)} \rightarrow \mathbb{R} \quad (5)$$

- The argument of the factor is called **scope** of the factor.

$$\text{Scope} : \{ \underline{X_1, \dots, X_n} \} \quad (6)$$

- Factors are building blocks used for defining high dimensional spaces and distributions.
- Factors are used to define an exponentially large probability distribution of N random variables.
- Factors are manipulated in the same way as probability distributions.

$$\hat{P} \rightarrow P(X_1, X_2, \dots, X_n) = \phi_1(X_1, X_2) \phi_2(X_2, X_3) \phi_3(X_3, X_4)$$

4x3

JOINT DISTRIBUTION IS A FACTOR

I	D	G	$P(I, D, G)$
i^0	d^0	g^1	0.126
		g^2	0.168
		g^3	0.126
i^0	d^1	g^1	0.009
		g^2	0.045
		g^3	0.126
i^1	d^0	g^1	0.252
		g^2	0.0224
		g^3	0.0056
i^1	d^1	g^1	0.060
		g^2	0.036
		g^3	0.024

Scope : { I, D, G }

UNNORMALIZED CONDITIONED JD IS A FACTOR

I	D	G	$P(I, D, g^1)$
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	0.060
			0.447

$$\phi(i^0, d^0, g^1) = 0.126$$

$$\phi(i^0, d^1, g^1) = 0.009$$

Scope : $\{I, D\}$

not $\{I, D, G\}$

CONDITIONAL PROBABILITY DISTRIBUTION

- CPD is a factor, which gives the conditional probability of a random variable, when other random variables are observed or known.
- For every combination of I and D , the value of G is observed.

		$P(G I, D)$		
		g^1	g^2	g^3
i^0, d^0	i^0, d^0	0.3	0.4	0.3
	i^0, d^1	0.05	0.25	0.7
i^1, d^0	i^1, d^0	0.9	0.08	0.02
	i^1, d^1	0.5	0.3	0.2

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

$P(g^2 | i^0, d^0)$

$P(g^3 | i^0, d^0)$

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

$P(g^2 | i^0, d^1)$

$P(g^3 | i^0, d^1)$

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

$P(g^2 | i^1, d^0)$

$P(g^3 | i^1, d^0)$

$P(g^1 | i^1, d^1)$

$P(g^2 | i^1, d^1)$

$P(g^3 | i^1, d^1)$

legitimate joint distribution

- Each row sums to 1.

$$\sum_{Pi^1, d^1} = 1$$

OPERATIONS ON FACTORS

- 1 Factor Product
- 2 Factor Marginalization
- 3 Factor Reduction

1. FACTOR PRODUCT

- Factor product is the cross product of two factors.

$$\Phi_1(A, B) \times \Phi_2(B, C)$$

		$\{A, B\} \cup \{B, C\} = \{A, B, C\}$		$A \quad B \quad C$	$\Phi_3(A, B, C) = \Phi_1 * \Phi_2$
A	B	$\Phi_1(A, B)$		$a^1 \quad b^1 \quad c^1$	$0.5 * 0.5 = 0.25$
a^1	b^1	0.5	$B \quad C$	$a^1 \quad b^1 \quad c^2$	$0.5 * 0.7 = 0.35$
a^1	b^2	0.8	$b^1 \quad c^1$	$a^1 \quad b^2 \quad c^1$	$0.8 * 0.1 = 0.08$
a^2	b^1	0.2	$b^1 \quad c^2$	$a^1 \quad b^2 \quad c^2$	$0.8 * 0.2 = 0.16$
a^2	b^2	0	$b^2 \quad c^1$	$a^2 \quad b^1 \quad c^1$	$0.2 * 0.5 = 0.25$
			$b^2 \quad c^2$	$a^2 \quad b^1 \quad c^2$	$0.2 * 0.7 = 0.35$
				$a^2 \quad b^2 \quad c^1$	$0 * 0.1 = 0$
				$a^2 \quad b^2 \quad c^2$	$0 * 0.2 = 0$

2. FACTOR MARGINALIZATION

- Remove one random variable.

A	B	C	$\phi_1(A, B, C)$
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	0.25
a^2	b^1	c^2	0.35
a^2	b^2	c^1	0
a^2	b^2	c^2	0

$$\phi_2(A, C) = \sum_B \phi_1(A, B, C)$$

A	C	$\Phi_2(A, C)$ marginalized on B
a^1	c^1	$0.25 + 0.08 = 0.33$
a^1	c^2	$0.35 + 0.16 = 0.51$
a^2	c^1	$0.25 + 0 = 0.25$
a^2	c^2	$0.35 + 0 = 0.35$

3. FACTOR REDUCTION

- Extract only one random variable.
- Observe $C = c^1$.

A	B	C	$\Phi_1(A, B, C)$
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	0.25
a^2	b^1	c^2	0.35
a^2	b^2	c^1	0
a^2	b^2	c^2	0

Diagram illustrating factor reduction:

```

graph LR
    A1B1C1["A1 B1 C1"] --> F1["Φ1(A, B, C)"]
    A1B1C2["A1 B1 C2"] --> F1
    A1B2C1["A1 B2 C1"] --> F1
    A1B2C2["A1 B2 C2"] --> F1
    A2B1C1["A2 B1 C1"] --> F2["Φ1(A, B, C1)"]
    A2B1C2["A2 B1 C2"] --> F2
    A2B2C1["A2 B2 C1"] --> F2
    A2B2C2["A2 B2 C2"] --> F2
    
```

The diagram shows the factor $\Phi_1(A, B, C)$ being reduced to $\Phi_1(A, B, C^1)$. The first four rows of the table are grouped by a bracket under $\Phi_1(A, B, C)$, and the last four rows are grouped by a bracket under $\Phi_1(A, B, C^1)$. Arrows point from each group to its corresponding row in the reduced table.

Factors and JPF

Let us say we have 4 random variables A, B, C, D and factors defined over them as follows:

$\phi_1[A, B]$	$\phi_2[B, C]$	$\phi_3[C, D]$	$\phi_4[D, A]$
$a^0 b^0 \frac{30}{100}$	$b^0 c^0 \frac{100}{100}$	$c^0 d^0 \frac{1}{100}$	$d^0 a^0 \frac{100}{100}$
$a^0 b' 5$	$b' c^0 1$	$c^0 d' 100$	$d^0 a' 1$
$a' b^0 1$	$b' c^0 1$	$c' d^0 100$	$d' a^0 1$
$a' b' 10$	$b' c' 100$	$c' d' 1$	$d' a' 100$

Example JPF using factors

$$P(A, B, C, D) = \frac{1}{Z} \phi_1(A, B) \phi_2(B, C) \phi_3(C, D) \phi_4(A, D)$$

Z = partition function used to normalize the probabilities

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

Example JPF using factors

What is the probability associated with the assignment $(\hat{a}^{\circ} b^{\circ} c^{\circ} d^{\circ})$?

$$\begin{aligned}&= \phi_1(a^{\circ} b^{\circ}) \phi_2(b^{\circ} c^{\circ}) \phi_3(c^{\circ} d^{\circ}) \phi_4(d^{\circ} a^{\circ}) \\&= 30 \times 100 \times 1 \times 100 = 300000\end{aligned}$$

After Normalization $\underline{0.04}$

$$f(x, y) = f_x(x) f_y(y) e^{x^2 + y^2 + xy}$$

Example JPDF using factors

$$e^{x^2 + xy} \quad e^{y^2}$$

There is a tight connection between independence properties

$$X \perp Y | Z \Leftrightarrow p(x, y, z) = \underbrace{\phi_1(x, z)}_{\text{"X is independent of Y given Z"}} \underbrace{\phi_2(y, z)}_{\text{When Z takes a fixed value } \phi_1 \text{ and } \phi_2 \text{ are separable functions.}}$$

"X is independent of Y given Z"

When Z takes a fixed value ϕ_1 and ϕ_2 are separable functions.

TABLE OF CONTENTS

1 PROBABILISTIC GRAPHICAL MODEL

2 JOINT DISTRIBUTION

3 FACTOR

4 INDEPENDENCE

5 BAYESIAN NETWORK

6 HOME WORK

INDEPENDENCE

- Independent parameters are parameters whose values are not completely determined by the values of the other parameters.
- Random variables $X = \{X_1, X_2, \dots, X_n\}$ can be considered independent if

$$\frac{P(\{X_1, X_2, \dots, X_n\})}{P(\{X_1, X_2, \dots, X_n\})} = \frac{\underset{n}{\underbrace{P(X_1)P(X_2)\dots P(X_n)}}}{\underset{Y_n}{\underbrace{\prod_{i=1}^n P(X_i)}}} \quad \text{← } n^2 \text{ values (7)}$$

(8)

- A set of random variables are independent of each other, if their joint probability distribution is equal to the product of probabilities of each individual random variable.

Another Perspective

$$P(A|B) = \frac{P(A, B)}{P(B)} \quad (\text{by definition})$$

$$P(A, B) = P(A) P(B) \quad \text{whenever } A \& B \text{ are independent}$$

$$\therefore P(A, B) = P(A|B) P(B) = P(A) P(B)$$

$$\Rightarrow P(A) = P(A|B)$$

[Knowing B does not change the probability of A]

STUDENT EXAMPLE

- A company is trying to hire a recent intelligent college graduate. The company has access to the student's SAT scores.
- The probability space is induced by Intelligence I and SAT score S .

$$I = \{ \text{high}, \text{low} \} = \{ i^1, i^0 \}$$

$$S = \{ \text{high}, \text{low} \} = \{ s^1, s^0 \}$$

STUDENT EXAMPLE - JOINT DISTRIBUTION

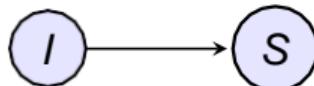
The joint distribution of $P(I, S)$ is given as

I	S	$P(I, S)$
i^0	s^0	0.665
i^0	s^1	0.035
i^1	s^0	0.06
i^1	s^1	0.24

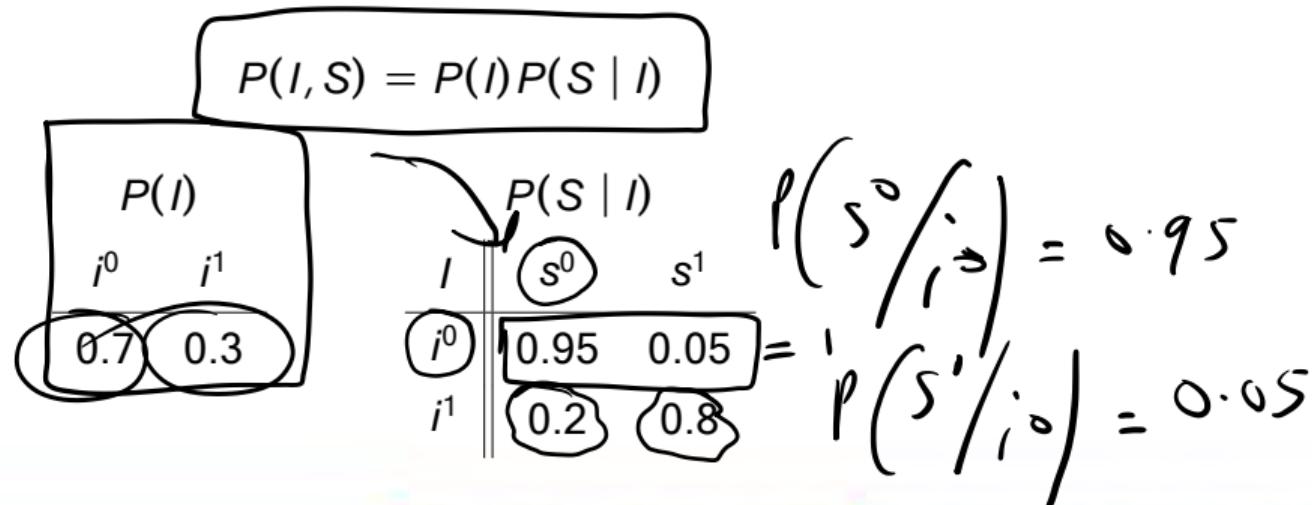
I

STUDENT EXAMPLE - CONDITIONAL DISTRIBUTION

- The student's SAT score is determined by his intelligence. This represents **causality**.

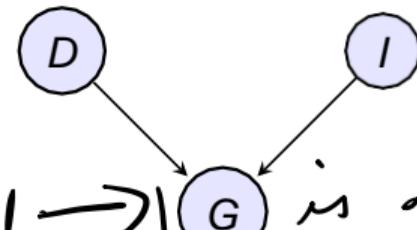


- Joint distribution $P(I, S)$ can be computed by using chain rule.



STUDENT EXAMPLE - CONDITIONAL DISTRIBUTION

- The grade student score depends on her intelligence and the difficulty of the course.
(by intuition)



$P_A(\text{Grade})$

= {D, difficulty, Intelligence}

Intelligence is a random variable

- Joint distribution $P(I, D, G)$ can be computed by using chain rule.

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

$$P(D|I) = P(D)$$

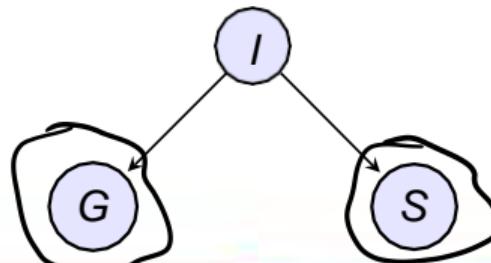
What can we say about D and I?

STUDENT EXAMPLE - CONDITIONAL INDEPENDENCE

- With 3 random variables, Intelligence I , Grade G and SAT score S , the JD has 12 entries.
- Both the SAT score and the grade are highly correlated on student's intelligence.
- If I is known, knowing Grade = A no longer gives information that $S = \text{high}$.
- If I is known, knowing $S = \text{high}$ no longer gives information that $\text{Grade} = A$.

$$S \perp G \mid I$$

- The student's intelligence is the only reason why his grade and SAT score might be correlated.



$$S = A + I$$

$$G = B + I$$

Student Example

Another way to look at this situation:

$$P(S/G, I) = P(S/I) \rightarrow S \perp G/I$$

If we know the student's intelligence then knowing his Grade will give us further information about his SAT score.

STUDENT EXAMPLE - CONDITIONAL INDEPENDENCE

- Joint distribution $P(I, S, G)$ can be computed by using chain rule.

$$\begin{aligned}
 & P(I, S, G) = P(I)P(S, G | I) \quad \text{[Bayes rule]} \\
 & P(S, G | I) = P(S | I)P(G | I) \quad \text{[why is this true?]} \\
 & P(I, S, G) = P(I)P(S | I)P(G | I) \\
 & P(S | G, I) = P(S | I) \Rightarrow P(I)P(G | I)P(S | I)
 \end{aligned}$$

■ 3 CPDs fully specify the JD.

$P(I)$	
i^0	i^1
<u>0.7</u>	<u>0.3</u>

$P(S I)$		
I	S^0	S^1
i^0	0.95	0.05
i^1	0.2	0.8

$P(G I)$			
I	G^1	G^2	G^3
i^0	0.2	0.34	0.46
i^1	0.74	0.17	0.09

TABLE OF CONTENTS

1 PROBABILISTIC GRAPHICAL MODEL

2 JOINT DISTRIBUTION

3 FACTOR

4 INDEPENDENCE

5 BAYESIAN NETWORK

6 HOME WORK

STUDENT EXAMPLE

Difficulty of course D	$Val(D) = \{ \text{hard}, \text{easy} \}$	$\{ d^1, d^0 \}$
Intelligence I	$Val(I) = \{ \text{high}, \text{low} \}$	$\{ i^1, i^0 \}$
Grade G	$Val(G) = \{ A, B, C \}$	$\{ g^1, g^2, g^3 \}$
SAT score S	$Val(S) = \{ \text{high}, \text{low} \}$	$\{ s^1, s^0 \}$
Recommendation Letter L	$Val(L) = \{ \text{strong}, \text{weak} \}$	$\{ l^1, l^0 \}$

- Joint distribution is given by

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

- $JD = \underline{2} * \underline{3} * \underline{2} * \underline{2} * \underline{2} = \underline{48 \text{ entries.}}$

STUDENT EXAMPLE

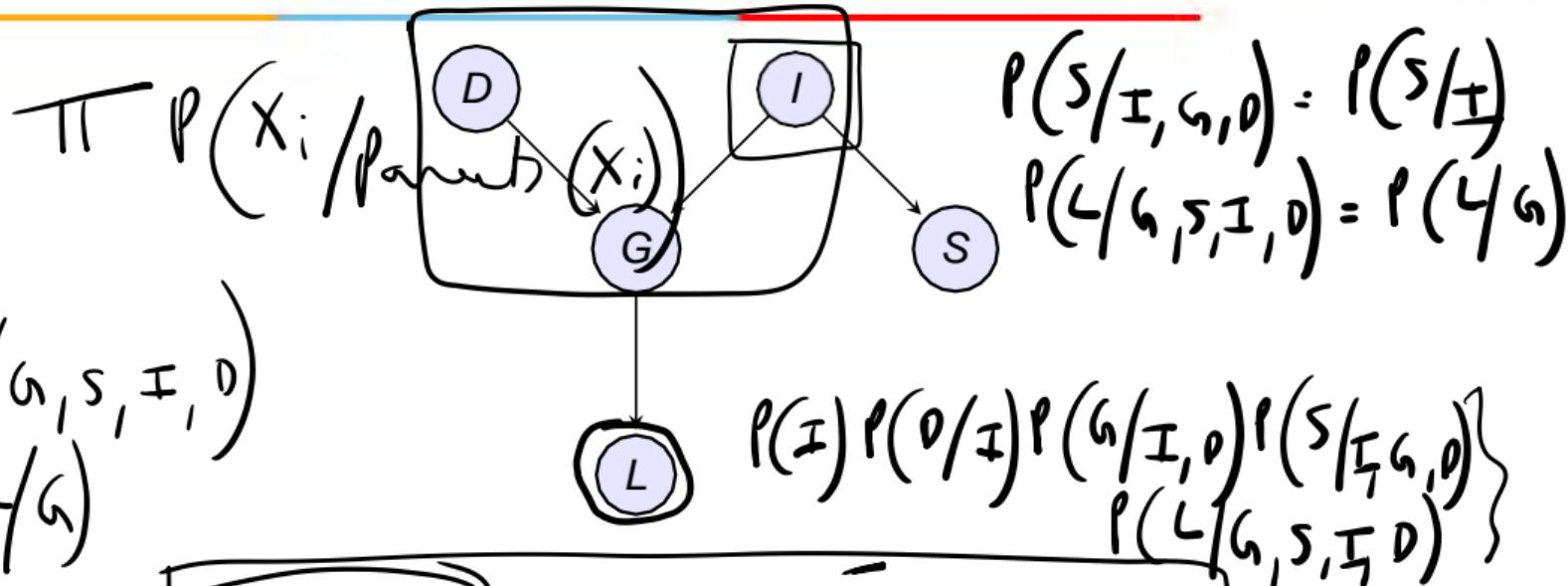
- Assume that the grade depends on *Difficulty* of the course and *Intelligence* of the student.
- The *SAT* score depends on *Intelligence* of the student
- Assume that the quality of the Recommendation *Letter* depends on *Grade*.

STUDENT EXAMPLE

$$\sum_{G \in \text{entries}} P(G | I, D) = 1 \quad 48 \text{ entries}$$

$$2 \times 2 \times 2 \times 3 \times 2 = 48$$

lead



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

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

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

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

How many parameters?

- Parameters = $1 + 1 + 8 + 2 + 3 = 15$ entries.

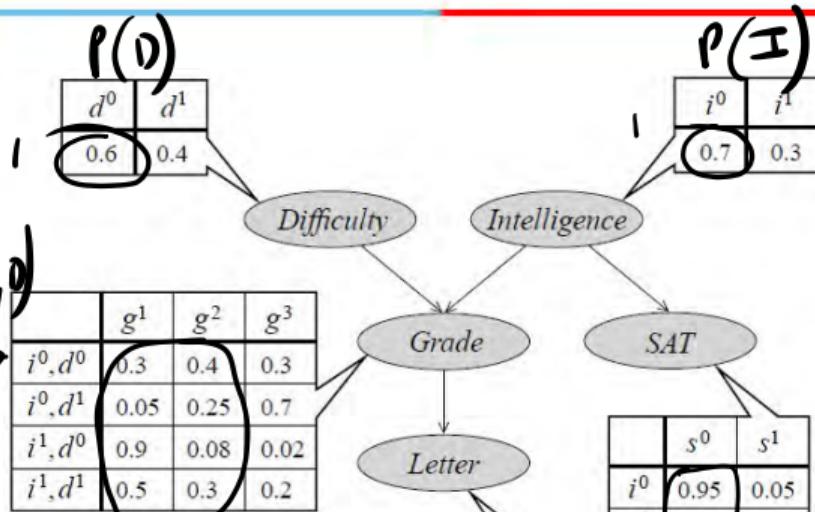
how did we get this?
Non-redundant

STUDENT EXAMPLE - B Student



$$P(X/P_a(X)) \quad P(G/I, D)$$

(node/parent) \rightarrow



circled values
= non-redundant
parameters

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

$$P(L|G)$$

$$+ 1 + 8 + 2 + 3 = 15$$

BAYESIAN NETWORK

- A Bayesian Network is a data structure to represent dependencies among random variables.
- Compact and natural representation.
- Represented using Directed acyclic graph (DAG) G
 -) Each node is a random variable.
 -) A set of directed edge connects pairs of nodes. Edges correspond to direct influence of one node on another.
- A data structure that provides the skeleton for representing a joint distribution compactly in a factorized way.
- A compact representation for a set of conditional independence assumptions about a distribution.

BAYESIAN NETWORK - TOPOLOGY

- Topology specifies the conditional independencies.

Cause = Parent(Effects)

- A Bayesian network represents the joint distribution of all random variables.
- Network structure together with its CPDs is called a **Bayesian network or local probability model**.

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (9)$$

BAYESIAN NETWORK - CONSTRUCTION



1 Nodes

-) Determine the set of random variables that are required to model the domain.
-) Order them such that the causes precedes the effects.

$$\{X_1, \dots, X_n\}$$

$$X_i \rightarrow X_j$$

if $i < j$

2 Links: For each node X_i ,

-) Choose a set of parents $Pa(X_i)$.
-) For each parent, insert a link from the Parent to the node X_i .
-) Write down the conditional probability table $P(X_i | Pa(X_i))$.

TABLE OF CONTENTS

1 PROBABILISTIC GRAPHICAL MODEL

2 JOINT DISTRIBUTION

3 FACTOR

4 INDEPENDENCE

5 BAYESIAN NETWORK

6 HOME WORK

RESTAURANT EXAMPLE

- Let Q represent the random variable for the quality of food.

Q	Good	Average	Bad
$P(Q)$	0.3	0.5	0.2

- Let L represent the random variable for the location of restaurant.

L	Good	Bad
$P(L)$	0.6	0.4

- Random variables Q and L are independent of each other.

RESTAURANT EXAMPLE

- Let C represent the cost of food.

$$C = \{ \text{high}, \text{low} \}$$

- Cost C is dependent on the quality Q of food and the location L of the restaurant.

- Let N represent the number of people visiting the restaurant.

$$N = \{ \text{high}, \text{low} \}$$

- N is affected by C which in turn is affected by Q .



RESTAURANT EXAMPLE

- What is the size of joint distribution $P(Q, L, C, N)$?
- List all the independencies and conditionally dependencies.
- Draw the Bayesian Network.
- How many parameters are required to represent $P(Q, L, C, N)$?
- Write the expression for $P(Q, L, C, N)$.

RESTAURANT EXAMPLE

- What is the size of joint distribution $P(Q, L, C, N)$?

$$3 * 2 * 2 * 2 = 24$$

- How many parameters are required to represent $P(Q, L, C, N)$?

$$\begin{array}{r} \boxed{(3 - 1) + (2 - 1) + (6 - 2) + (4 - 1) = 10} \\ \cancel{2} + \cancel{1} + \cancel{6} + \cancel{4} = 13 \end{array} X$$

- Write the expression for $P(Q, L, C, N)$.

According to Bayesian Network ,

$$P(Q, L, C, N) = P(Q)P(L)P(C|L, Q)P(N|C, L)$$

RESTAURANT EXAMPLE

- List all the independencies and conditionally dependencies.

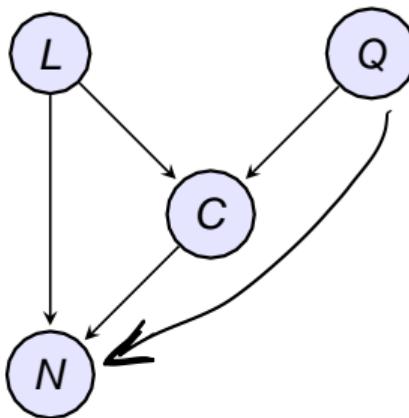
$$Q \perp L$$

$$\cancel{C|Q, L}$$

$$\cancel{N|C, L}$$

$$Q \perp N|C$$

- Draw the Bayesian Network.



REFERENCES

- 1 Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press. 2009
- 2 Artificial Intelligence: A Modern Approach (3rd Edition) by Stuart Russell, Peter Norvig
- 3 Mastering Probabilistic Graphical Models using Python by Ankur Ankan, Abhinash Panda. Packt Publishing 2015.
- 4 Learning in Graphical Models by Michael I. Jordan. MIT Press. 1999

Thank You !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 5 : BAYESIAN MODEL

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.



Table of Contents

1 Probabilistic Influence

2 Directed Separation

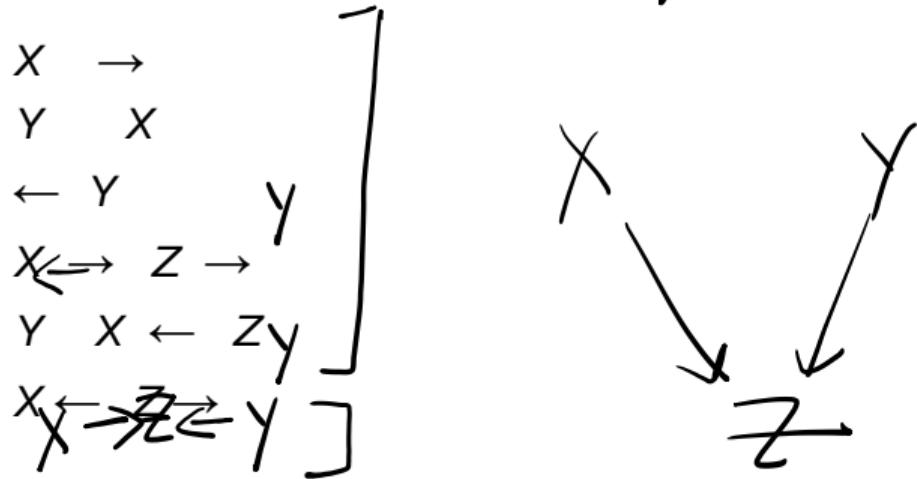
3 CPD Representation

4 Bayesian Network Summary

Influence

- Influence means identifying the conditions when one random variable changes the beliefs about another random variable.
- When can X influence Y ?

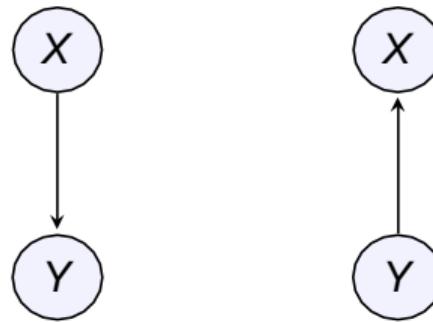
$$P(Y/X) \neq P(Y)$$



Direct Influence

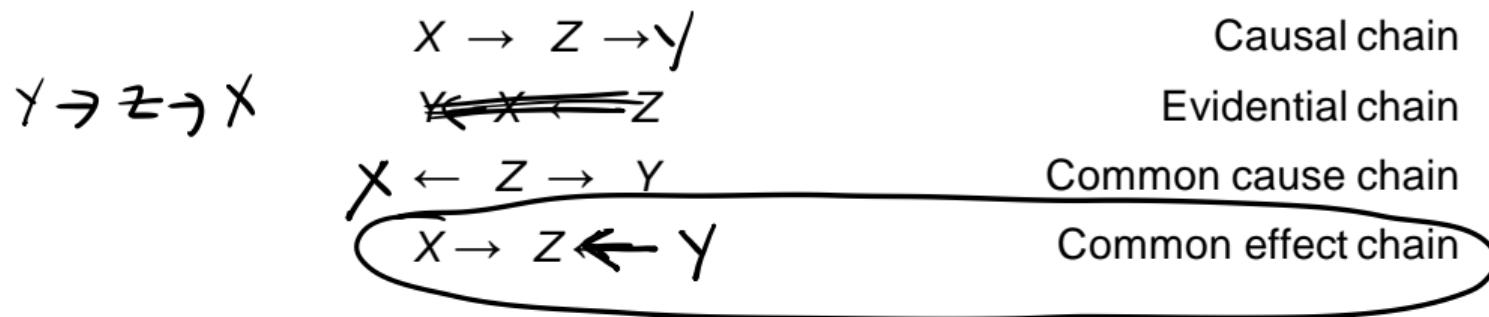
- X and Y are directly connected.
- Direct parent child relation.
- They influence each other.

$X \rightarrow$
 $Y \ X$
 $\leftarrow \ Y$



Indirect Influence

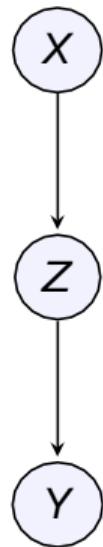
- X and Y are not directly connected, but there is a **trail** between them in the graph.
- X and Y connected by a trail through Z.



X_1, \dots, X_k form a **trail** in the graph, if for every $i = 1, \dots, k - 1$ we have either $X_i = X_{i+1}$.

Indirect Influence

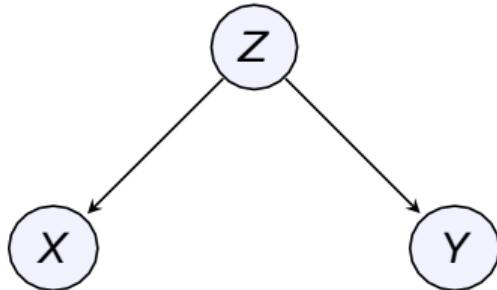
Causal



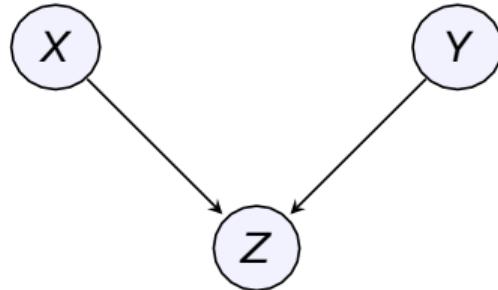
Evidential



Common Cause

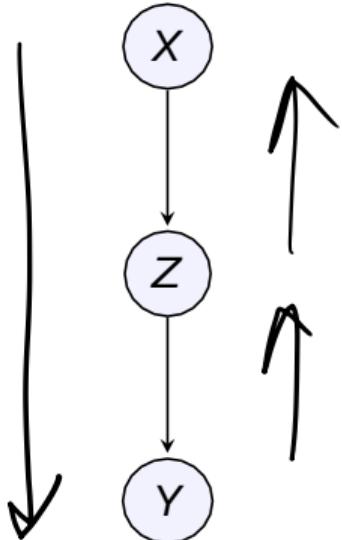


Common Effect



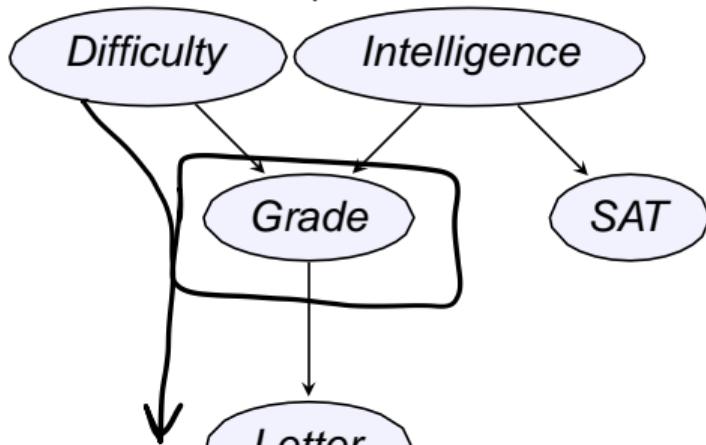
Indirect Causal Effect

L and D are not independent, $P(L|D) \neq P(L)$

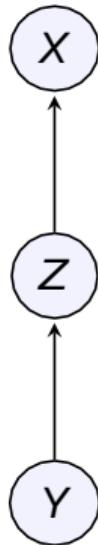


- $X \rightarrow Z \rightarrow Y$
- X can influence Y via Z , if Z is not observed. Eg:
 $D \rightarrow G \rightarrow L$
- X **cannot** influence Y via Z if Z is observed.
Eg: $(L \perp I|G)$

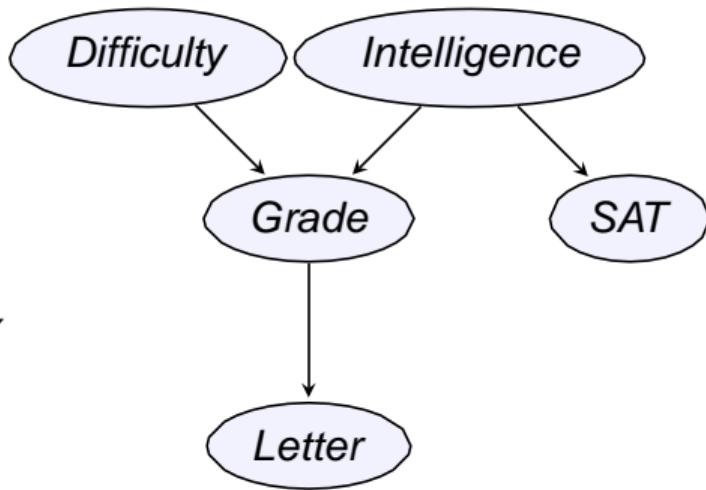
$$P(L|D, G) = P(L|G)$$



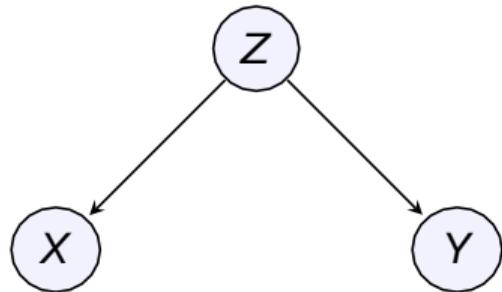
Indirect Evidential Effect



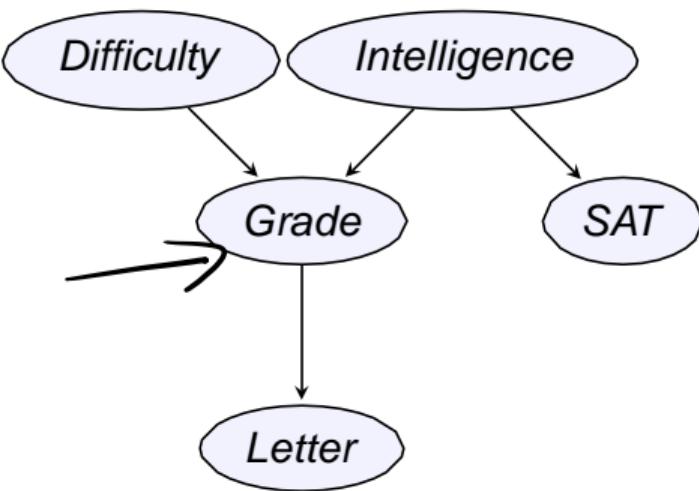
- $X \leftarrow Z \leftarrow Y$
- Y can influence X via Z , if Z is not observed. Eg:
 $D \rightarrow G \rightarrow L$
- Y **cannot** influence X via Z if Z is observed.
Eg: $(L \perp I/G)$



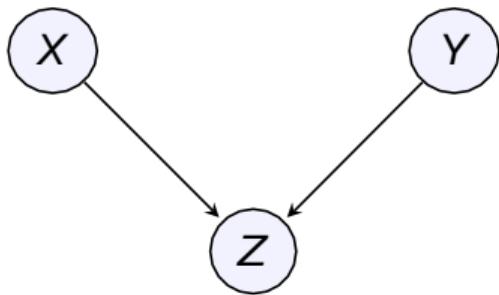
Common Cause Chain



- $X \leftarrow Z \rightarrow Y$
- X can influence Y via Z , if Z is not observed. Eg:
 $G \leftarrow I \rightarrow S$
- X **cannot** influence Y via Z if Z is observed.
Eg: $(S \perp G/I)$

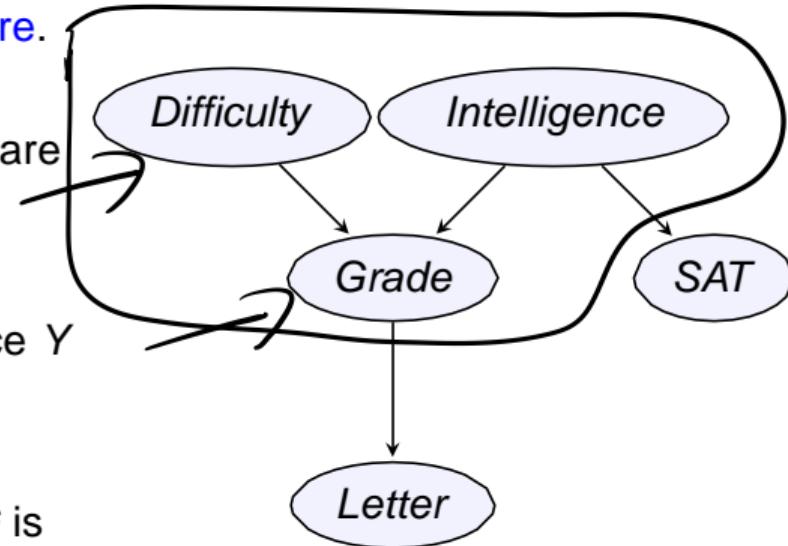


Common Effect Chain



- $X \rightarrow Z \leftarrow Y$. This is called **v-structure**.
- When G is not observed I and D are independent. Eg:

$$D \rightarrow G \leftarrow I$$
- **X cannot** influence Y via Z , if Z is not observed.
- When evidence G is observed, I and D are correlated.



Indirect Influence Flow

- X and Y are not directly connected, but connected by a **trail** through Z .
- If Z is **not** observed,

Causal chain	$: X \rightarrow Z \rightarrow Y$: active; Yes
Evidential chain	$: X \leftarrow Z \leftarrow Y$: active; Yes
Common Cause chain	$: X \leftarrow Z \rightarrow Y$: active; Yes
Common Effect chain	$: X \rightarrow Z \leftarrow Y$: inactive; NO

- V-structure is active if and only if either Z or one of Z 's descendants are observed.

$$X_1 \leftarrow X_2 \leftarrow X_3$$

Definition

In a Bayesian Network G with a trail $X_1 = \dots = X_n$, let a subset Z of variables be observed. The trail is **active** given Z if

- whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in Z .
- no other node along the trail is in Z . (not in v-structure)

One node can influence another if there is any trail along which influence can flow.

Influence Flow

Definition

In a Bayesian Network G with a trail $X_1 = \dots = X_n$, let a subset Z of variables be observed. The trail is **active** given Z if

- whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in Z .
- no other node along the trail is in Z . (not in v-structure)

Influence Flow

Definition

In a Bayesian Network G with a trail $X_1 = \dots = X_n$, let a subset Z of variables be observed. The trail is **active** given Z if

- whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in Z .
- no other node along the trail is in Z . (not in v-structure)

One node can influence another if there is any trail along which influence can flow.

For almost all parameterizations P of the graph G , the d-separation test precisely characterizes the independencies that hold for P .



Table of Contents

1 Probabilistic Influence

2 Directed Separation

3 CPD Representation

4 Bayesian Network Summary

Directed Separation (d-separation)

The notion of d-separation provides us with a notion of separation between nodes in a directed graph.

Definition

In a Bayesian Network G , let X, Y, Z , be three sets of nodes. X and Y are d-separated, if there is no active trail between X and Y given Z .

$$I(G) = \{(X \perp Y | Z) : d - sep_G(X; Y | Z)\} \quad (1)$$

This set is also called the set of global Markov independencies.

Factorization & Independence in BN

Theorem

If P factorizes over G and $d = \text{sep}_G(X; Y/Z)$, then P satisfies, $(X \perp Y/Z)$.

(restatement of Theorem 3.3 from Daphne Koller's book)

$$\begin{aligned} d - \text{sep}_G(X; Y/Z) \Rightarrow X \perp Y/Z \in I(G) \text{ and } I(G) \subseteq I(P) \\ \Rightarrow X \perp Y/Z \in I(P) \end{aligned}$$

COMP538: Introduction to Bayesian Networks - Lecture 3: Probabilistic Independence and Graph Separation (hkust.edu.hk)

Factorization & Independence in BN

Theorem

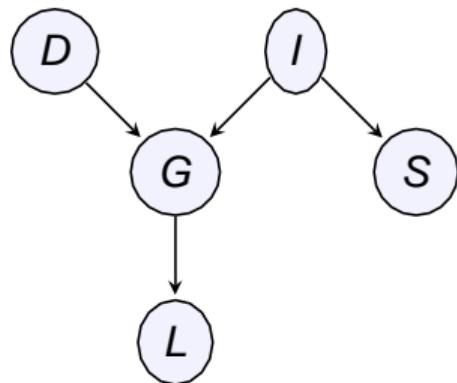
If P factorizes over G and $d = \text{sep}_G(X; Y/Z)$, then P satisfies, $(X \perp Y/Z)$.

Proof for $D \perp S$

Active Trail $S \leftarrow I \rightarrow G \leftarrow D$

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

$$\begin{aligned} P(D, S) &= \sum_{G, L, I} P(I)P(D)P(G/I, D)P(L/G)P(S/I) \\ &= \sum_I P(I)P(D)P(S/I) \sum_G P(G/I, D) \sum_L P(L/G) \\ &= P(D) \sum_I P(I)P(S/I) = P(D)P(S) \Rightarrow D \perp S \end{aligned}$$



Factorization & Independence in BN

Theorem

If P factorizes over G ; then any node is d -seperated from its non-descendants given its parents.

Factorization & Independence in BN

Theorem

If P factorizes over G ; then any node is d -separated from its non-descendants given its parents.

For L descendants = J

For L nondescendants = D, G, I, S

1. Trail $S \leftarrow I \rightarrow G \rightarrow L$

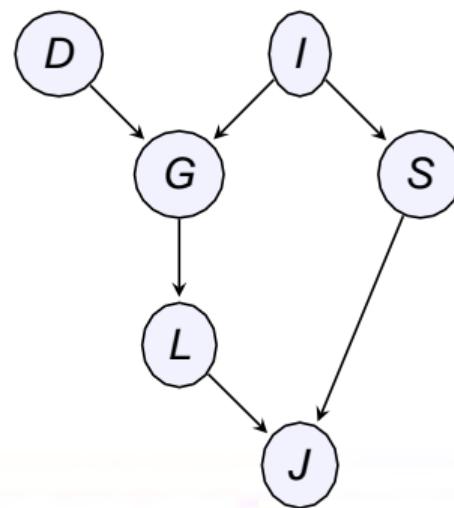
Trail not active as G is observed.

G parent of L , blocks the trail.

2. Trail $S \leftarrow J \rightarrow L$

Trail not active as only G is observed.

J is descendant and is not observed.



I-map & d-separation

Definition

In a Bayesian Network G, P satisfies the corresponding independence statements.

$$I(G) = \{(X \perp Y/Z) : d - sep_G(X; Y/Z)\} \quad (2)$$

If P satisfies $I(G)$, then G is an I-map of P .



Table of Contents

1 Probabilistic Influence

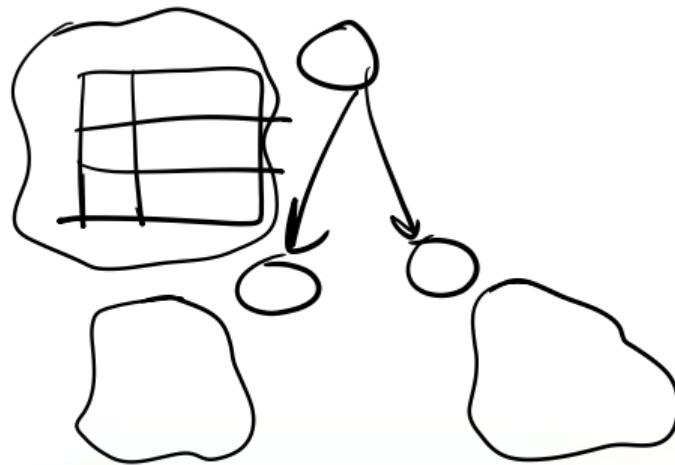
2 Directed Separation

3 CPD Representation

4 Bayesian Network Summary

Tabular CPD

- Take all the possible combinations of different states of a variable and represent them in a tabular form.
- Tabular CPD is not the best choice to represent CPDs always.



Tabular CPD

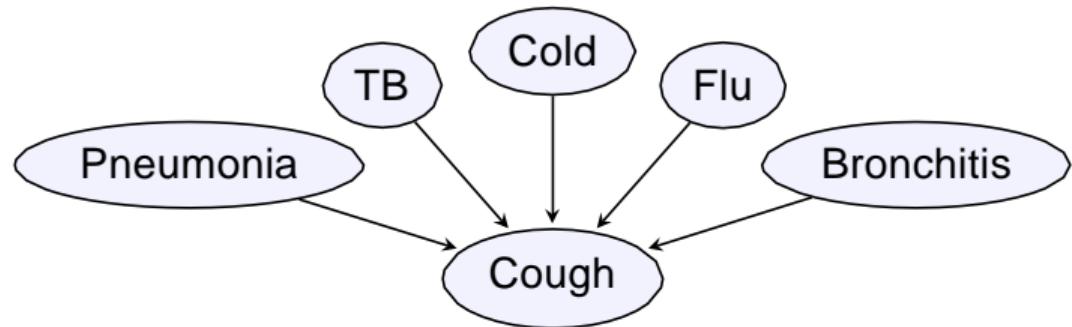
- Take all the possible combinations of different states of a variable and represent them in a tabular form.
- Tabular CPD is not the best choice to represent CPDs always.

X	Y	$P(X, Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

Tabular CPD

- Take all the possible combinations of different states of a variable and represent them in a tabular form.
- Tabular CPD is not the best choice to represent CPDs always.

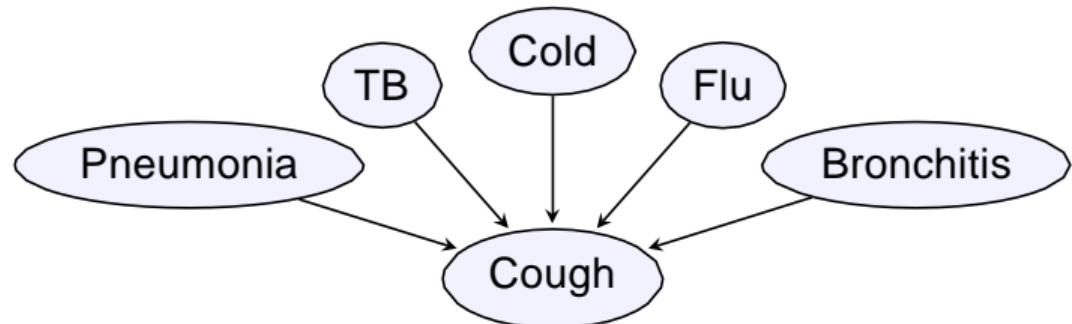
X	Y	$P(X, Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48



Tabular CPD

- Take all the possible combinations of different states of a variable and represent them in a tabular form.
- Tabular CPD is not the best choice to represent CPDs always.

X	Y	$P(X, Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48



- For binary valued k parents, the size of the tabular CPD will be of $O(2^k)$.

Deterministic CPD

- Deterministic random variable are those, whose value depends only on the values of its parents in the model.

$$P(X/Pa_X) = \begin{cases} 1 & \text{if } x = Val(Pa_X) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

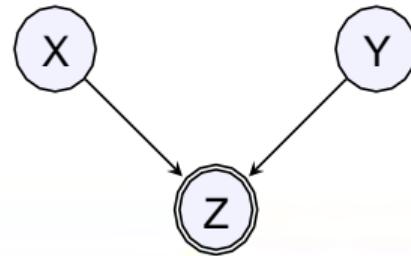
- Denote a deterministic variable by double circles.

Deterministic CPD

- Deterministic random variable are those, whose value depends only on the values of its parents in the model.

$$P(X/Pa_X) = \begin{cases} 1 & \text{if } x = Val(Pa_X) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

- Denote a deterministic variable by double circles.
- Eg: A Bayesian network for a logic gate. X and Y are the inputs, A and B are the outputs and Z is a deterministic variable representing the operation of the logicgate.



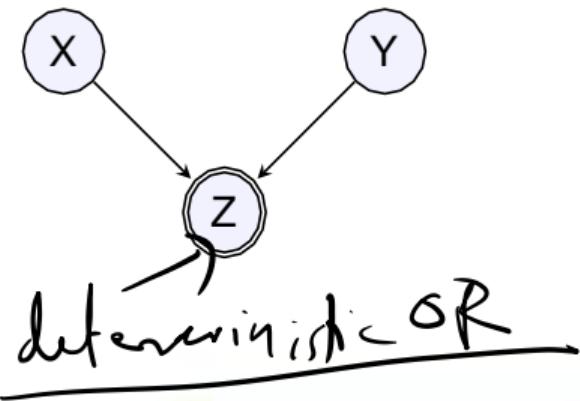
Context Specific CPD

- Context specific independence is a type of independence for random variables X, Y, Z and an assignment c .
- The independence statement only holds for a particular value of conditioning variable c .

$$\left(X \perp Y \middle| \text{ } \right) \quad P(X, Y | Z) = P(X | Z) P(Y | Z) \quad P \models (X \perp_c Y | Z, c) \quad (4)$$

Context Specific CPD

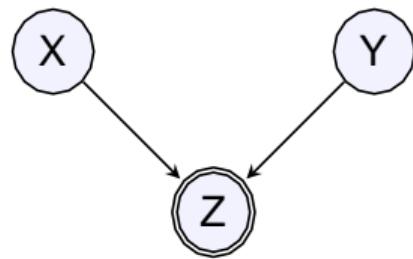
Which of the following Context specific independences hold when Z is a deterministic OR of X and Y?



- 1 $(Z \perp X | y^0)$
- 2 $(Z \perp X | y^1)$
- 3 $(X \perp Y | z^0)$
- 4 $(X \perp Y | \bar{z})$

Context Specific CPD

Which of the following Context specific independences hold when Z is a deterministic OR of X and Y?



- ① $(Z \perp X | y^0)$ **False**

When $Y = 0$, $Z = X$. So not independent.

- ② $(Z \perp X | y^1)$ **True**

When $Y = 1$, $Z = 1$. So context specific independent.

- ③ $(X \perp Y | z^0)$ **True**

When $Z = 0$, $X \perp Y$. So context specific independent.

- ④ $(X \perp Y | z^1)$ **False**

When $Z = 1$, $X \not\perp Y$. So not independent.

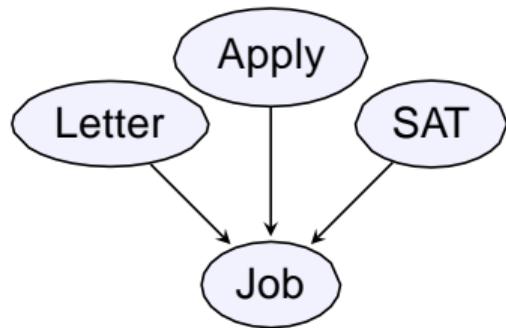
Tree Structured CPD

- Tree Structured CPD encode dependence of a child on a parent.

Tree Structured CPD

- Tree Structured CPD encode dependence of a child on a parent.

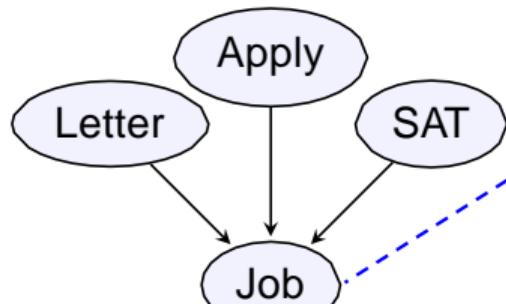
Bayesian Network



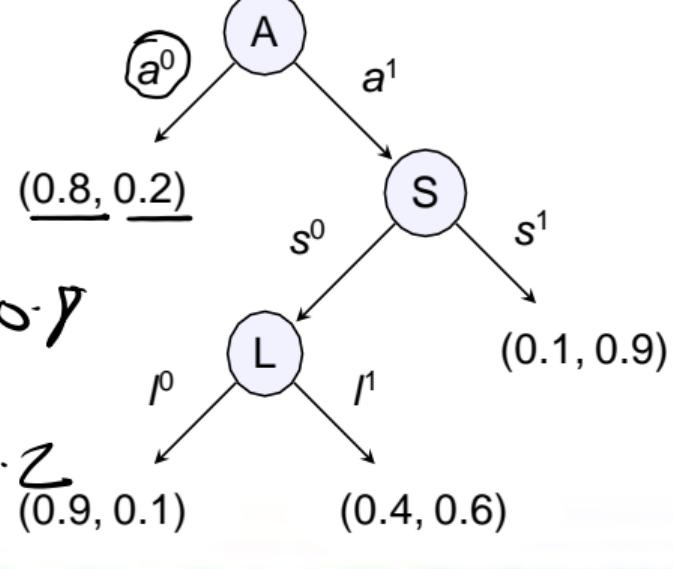
Tree Structured CPD

- Tree Structured CPD encode dependence of a child on a parent.

Bayesian Network



CPD of Job(no, yes)



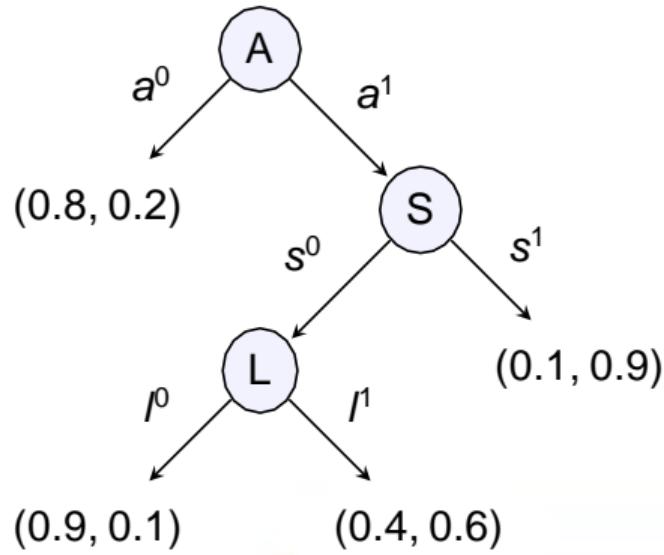
$$P(J = \text{no} | a^0) = 0.8$$

$$P(J = \text{yes} | a^0) = 0.2$$

Tree Structured CPD

Which of the following Context specific independences hold?

CPD of Job(no, yes)

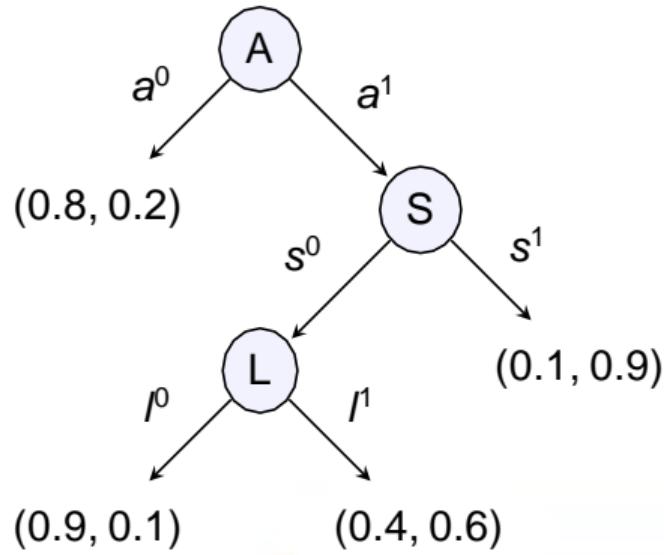


- 1 $(J \perp_c L / a^1, s^1)$
- 2 $(J \perp_c L / a^1)$
- 3 $(J \perp L / s^1, A)$
- 4 $(J \perp_c L, S / a^0)$

Tree Structured CPD

Which of the following Context specific independences hold?

CPD of Job(no, yes)



- ① $(J \perp_c L / a^1, s^1)$ True
Context specific independent.
- ② $(J \perp_c L / a^1)$ False
Not independent.
- ③ $(J \perp_c L / s, A)$ True
Context specific independent.
- ④ $(J \perp_c L, S / a)$ True
Context specific independent.



Table of Contents

1 Probabilistic Influence

2 Directed Separation

3 CPD Representation

4 Bayesian Network Summary

Independence

Definition

Independence: $P(\{X_1, X_2, \dots, X_n\}) = \prod_{i=1}^n P(X_i)$

Local Independence: $I_L(G) : (X_i \perp \text{NonDescendants}_{X_i} / \text{Pa}_{X_i}) \quad \forall X_i$

Independency Map: G is an I-map for P if $I_L(G) \subset I(P)$

Theorem

P satisfies $I_L(G)$ if P is representable as a set of CPDs associated with G .

Bayesian Network

Definition

A Bayesian Network $B = (G, P)$ where P factorizes over G and where P is specified as a set of CPDs associated with G .

$$\begin{aligned} P \text{ factorizes over } G &\text{ if } P(X_1, X_2, \dots, X_n) = \prod_i P(X_i | \text{Pa}_i^G(X)) \\ G \text{ encodes } I_1(G) &\text{ if } \forall X_i: (X_i \perp \text{NonDescendants}_{X_i} | \text{Pa}_{X_i}) \end{aligned}$$

Theorem

If G is an I-map for P , then P factorizes G .

If P factorizes according to G , then G is an I-map for P .

Reasoning Patterns

Causal reasoning Queries that predict the effects of various factors or features are called causal reasoning.

Evidential reasoning Queries that reason from effects to causes are called evidential reasoning.

Intercausal reasoning Explaining away is an instance of intercausal reasoning, where different causes of the same effect can interact.

- Inference by Enumeration

$$P_B(Y = y | E = e)$$

Influence Flow

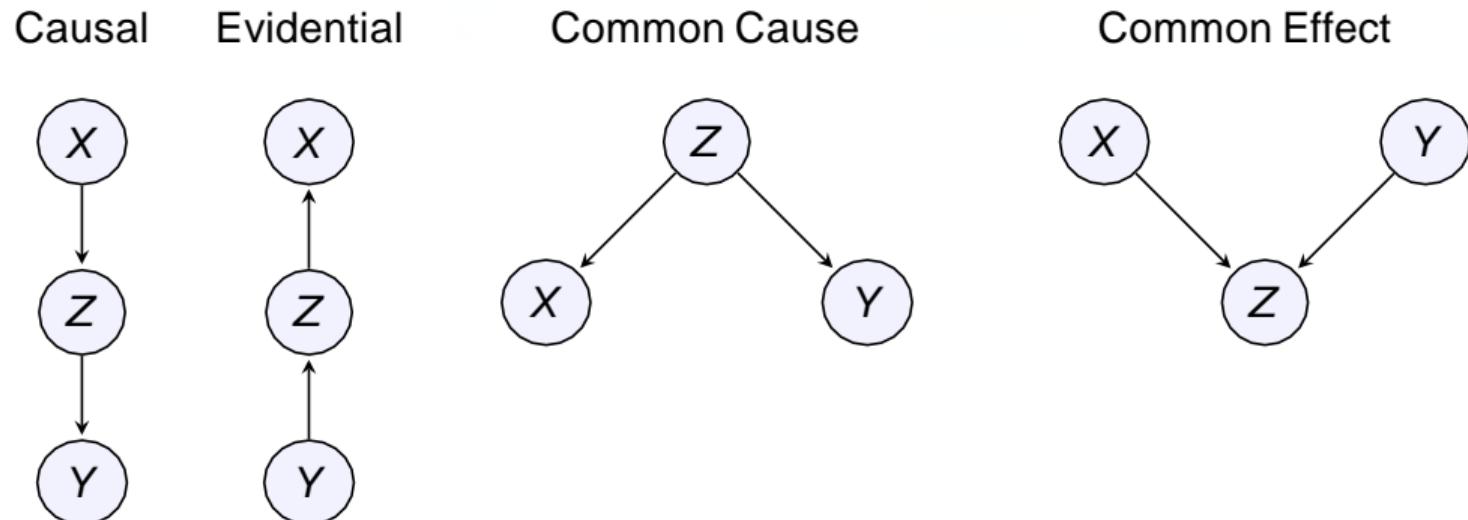
Definition

In a Bayesian Network G with a trail $X_1 \not\rightarrow \dots \not\rightarrow X_n$, let a subset Z of variables be observed. The trail is **active** given Z if

- whenever we have a v-structure $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, then X_i or one of its descendants are in Z .
- no other node along the trail is in Z . (not in v-structure)

One node can influence another if there is any trail along which influence can flow.

Indirect Influence



If Z is **not** observed

Directed Separation (d-separation)

Definition

In a Bayesian Network G , let X, Y, Z , be three sets of nodes. X and Y are d-separated, if there is no active trail between any node given Z .

$$I(G) = \{(X \perp\!\!\!\perp Y/Z) : d - sep_G(X; Y/Z)\} \quad (5)$$

Questions

- 1 Given a Bayesian Network, find the appropriate factorization of joint distribution.
- 2 Given a Bayesian Network, identify the active trails.
- 3 Given a Bayesian Network, identify the I-maps.
- 4 Given a Bayesian Network, identify the d-seperations.
- 5 In a Bayesian Network, infer by enumeration, the probability of an event when some evidences are observed.
- 6 Given a toy application, generate the CPD and Bayesian Network.
- 7 Given CPDs, generate a Bayesian Network.
- 8 Given a joint distribution in the factorized form, generate a Bayesian Network.
- 9 Given a Bayesian Network, identify the conditional independencies.
- 10 Given a CPD, identify the context specific independencies.

References

- 1 Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press. 2009
- 2 Artificial Intelligence: A Modern Approach (3rd Edition) by Stuart Russell, Peter Norvig
- 3 Mastering Probabilistic Graphical Models using Python by Ankur Ankan, Abhinash Panda. Packt Publishing 2015.
- 4 Learning in Graphical Models by Michael I. Jordan. MIT Press. 1999

Thank You !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 4 : BAYESIAN MODEL

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.



TABLE OF CONTENTS

1 BAYESIAN NETWORK

2 REASONING PATTERNS

3 INDEPENDENCY MAP

BAYESIAN NETWORK

DEFINITION (GLOBAL SEMANTICS)

A Bayesian Network is a directed acyclic graph G whose nodes represent the random variables $\{X_1, X_2, \dots, X_n\}$ and represents a joint distribution via the chain rule for the Bayesian Networks.

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | Pa(X_i)) \quad (1)$$

- Each node is associated with a CPD.

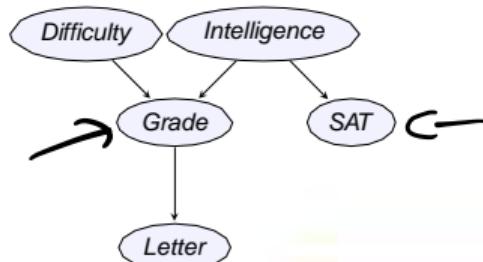


$$CPD(X_i) = P(X_i | Pa(X_i)) \quad (X_1, X_2, \dots, X_n) = P(X_1) P(X_2 | X_1) P(X_3 | X_1, X_2) \dots$$

BAYESIAN NETWORK IS LEGAL

A BN is a legal distribution; if

- $P \geq 0$
 -) P is a product of CPDs.
 -) CPDs are non negative.
- $\sum P = 1$



D - I - G - S \rightarrow L₀ for $(P, I, G, S) \{ \} \rightarrow (P, I, G, S)$
 lead

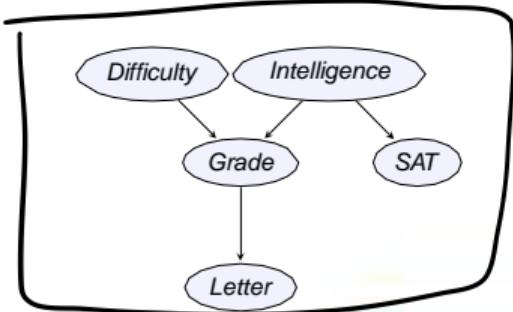
BAYESIAN NETWORK IS LEGAL

A BN is a legal distribution; if

- $P \geq 0$

- P is a product of CPDs.
- CPDs are non negative.

- $\sum P = 1$



$$\begin{aligned}
 \sum P &= \boxed{P(I, D, G, S, L)} \\
 &= \sum_{D, I, G, S, L} P(I)P(D)P(G|I, D)P(S|I)P(L|G) \\
 &= \sum_{D, I, G, S} P(I)P(D)P(G|I, D)P(S|I) \cancel{\sum P(L|G)} \\
 &= \sum_{D, I, G} P(I)P(D)P(G|I, D) \sum_S P(S|I) = 1 \\
 &= \sum_{D, I} P(I)P(D) \sum_G P(G|I, D) \sum_L P(L|G) = 1 \\
 &= \left(\sum_I P(I) \right) \left(\sum_D P(D) \right) = 1
 \end{aligned}$$



TABLE OF CONTENTS

1 BAYESIAN NETWORK

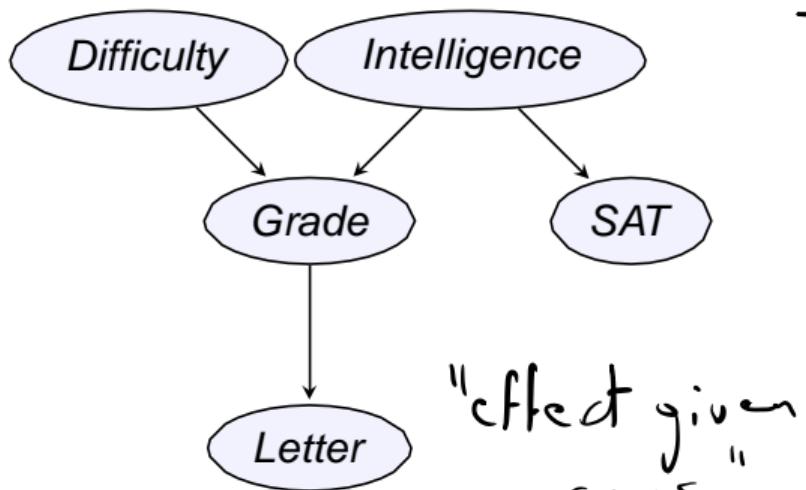
2 REASONING PATTERNS

3 INDEPENDENCY MAP

REASONING PATTERNS

- 1 Causal reasoning
- 2 Evidential reasoning
- 3 Intercausal reasoning

CAUSAL REASONING



- How likely will a student get a strong recommendation?

$$\cancel{P(I^1)} = ?$$

- Given that the student is not so intelligent, what is chance that he gets a strong letter?

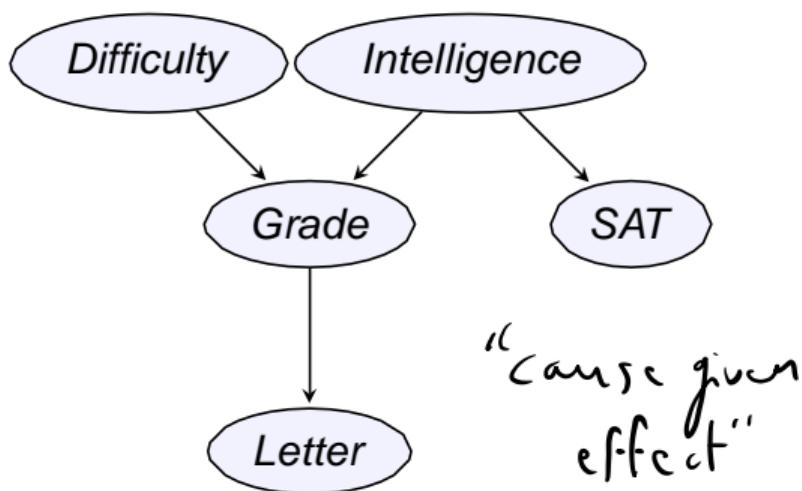
$$\cancel{P(I^1 | \underline{i}^0)} = ?$$

- What if the course is easy?

$$\cancel{P(I^1 | \underline{L}^0, d^0)} = ?$$

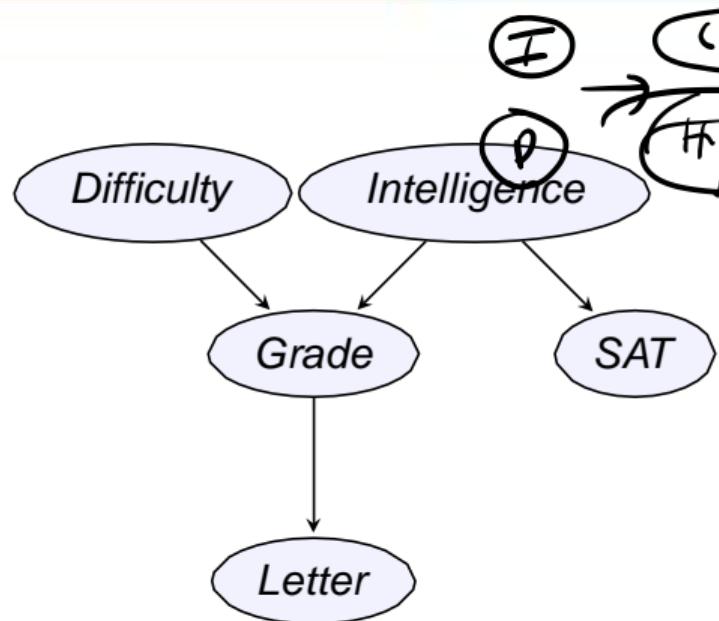
- Queries that predict the effects of various factors or features are called causal reasoning.

EVIDENTIAL REASONING



- Given that a student gets C grade for a course, comment on his intelligence.
 $P(i^1|g^3) = ?$
- Given that the student got a weak letter, comment on his intelligence.
 $P(i^1|l^0) = ?$
- $P(\underline{i^1}|l^0, \underline{g^3}) = ?$
- **Queries that reason from effects to causes are called evidential reasoning.**

INTERCAUSAL REASONING



given evidence, prob of multiple causes?

Causal reasoning Example

In the absence of any other information

$$P_{\text{Student}}(l) = 0.5^{0.2}$$

How do we calculate this? → From CPDs in
the Bayesian network.

Causal Reasoning Example

From the CPDs we have

$$P(e) = \underbrace{P(e/g)}_{P(g)} P(g) + \underbrace{P(e/g^2)}_{P(g^2)} P(g^2) + \underbrace{P(e/g^3)}_{P(g^3)} P(g^3)$$

$$\begin{aligned} P(g) &= \underbrace{P(g/i^{\circ}, d^{\circ})}_{P(i^{\circ}, d^{\circ})} P(i^{\circ}, d^{\circ}) + \underbrace{P(g/i^{\circ}, d^{\circ})}_{P(i^{\circ}, d^{\circ})} P(i^{\circ}, d^{\circ}) \\ &\quad + \underbrace{P(g/i^{\circ}, d^{\circ})}_{P(i^{\circ}, d^{\circ})} P(i^{\circ}, d^{\circ}) + \underbrace{P(g/i^{\circ}, d^{\circ})}_{P(i^{\circ}, d^{\circ})} P(i^{\circ}, d^{\circ}) \end{aligned}$$

Causal Reasoning Example

$$P(i^o, d^o) = P(i^o) P(d^o)$$

We have $P(g) = (0.3)(0.42) + (0.05)(0.28) + (0.9)(0.18)$
 $+ (0.5)(0.12) = \underline{0.362}$

$$P(g^2) = (0.4)(0.42) + (0.25)(0.28) + (0.08)(0.18)
+ (0.3)(0.12) = \underline{0.2884}$$

$$P(g^3) = (0.3)(0.42) + (0.7)(0.28) + (0.02)(0.18) + (0.7)(0.12)
= \underline{0.3496}$$

Causal reasoning Example

$$P(I) = \underline{0.9}(\underline{0.3}c_2) + \underline{0.4}(\underline{0.288}4) + \underline{0.01}(\underline{0.349})$$

$\approx \underline{0.582}$

If we know that George is not so intelligent, what is the probability that he gets a strong letter of recommendation?

$$P(I'/I_0) \rightarrow ?$$

Causal Reasoning Example

$$P(l^i; i^o) = P(l^i; i^o, d^o) + P(l^i; i^o, \bar{d}) \quad [\text{Marginalization}]$$

$$P(l^i; i^o, d^o) = \frac{P(l^i | g^1) P(g^1 | i^o, d^o) P(i^o, d^o)}{P(l^i | g^1) P(g^1 | i^o, d^o) P(i^o, d^o) + P(l^i | g^2) P(g^2 | i^o, d^o) P(i^o, d^o) + P(l^i | g^3) P(g^3 | i^o, d^o) P(i^o, d^o)}$$

Substituting for the various probabilities we have

$$P(l^i; i^o, d^o) = \underline{0.21546}$$

Causal Reasoning Example

$$\text{Similarly } P(l, i^o, d') = 0.056$$

$$P(l, i^o) = P(l, i^o, d) + P(l, i^o, d') = 0.27146$$

$$P(l | i^o) = \frac{P(l, i^o)}{P(i^o)} = \frac{0.27146}{0.7} \approx 0.389$$

The probability that the student gets a good letter
of recommendation goes down given low intelligence

Evidential Reasoning Example

Suppose a student received a grade $C \rightarrow g^3$
 what is the probability of high intelligence now?

$$P(i'|g^3) = 0.079$$

cause
effect

$$P(d') = 0.40 \text{ but } P(d|g^3) = 0.62$$

$$P(i'|g^3, s) = 0.578$$

"cause given
effect"

grade is poor, but SAT score
is high, so high intelligence
is favoured

Intercausal Reasoning

$$P(i'|g^3) = 0.079$$

If we discover that the subject is hard $\rightarrow d'$
 $P(i'|g^3, d') = 0.11 \rightarrow$ this is a partial explanation
for the student's low grade (the student can be
intelligent and have still got a poor grade because
the subject was hard)

Intercausal Reasoning

If the student gets a B grade (g^2) :

$$P(i'/g^2) = 0.175$$

$$P(i'/g^2, d') = 0.34$$

We have explained away the poor grades
using the difficulty of the class.



TABLE OF CONTENTS

1 BAYESIAN NETWORK

2 REASONING PATTERNS

3 INDEPENDENCY MAP

DEPENDENCY IN BN

- A node depends directly only on its parents.
- If the student's grade is known, the quality of his recommendation letter is not influenced by information about any other variable. L is conditionally independent of all other nodes in the network given its parent G

$$(L \perp \{I, D, S\} \mid G)$$

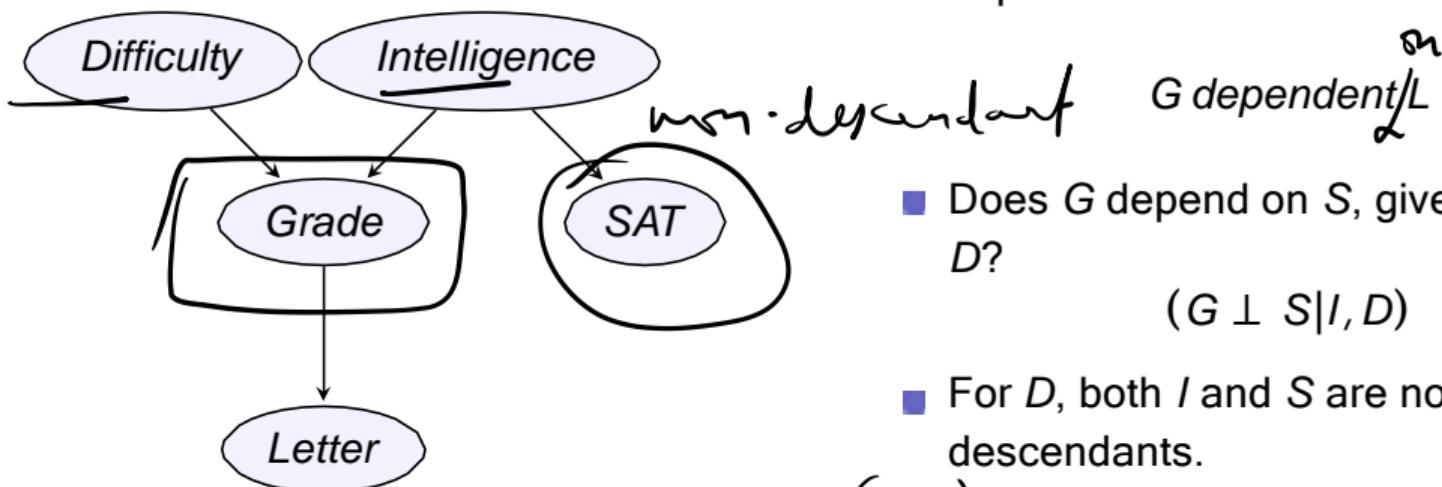
~~($L \perp I, D \mid G$)~~



$$(S \perp D, G, L \mid I)$$

DEPENDENCY IN BN

- Given the parents, a node can depend on its descendants.



- Does G depend on S , given I and D ?

$$(G \perp S | I, D)$$

- For D , both I and S are non descendants.

$$(D \perp I, S)$$

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

Dependency in BN

Is G independent of C given its parents I and D ?

No. We have:

$$P(g'|i', d', e) > P(g|i', d')$$

If we know that the student got a strong letter of recommendation, it increases the probability that the student had a good grade.

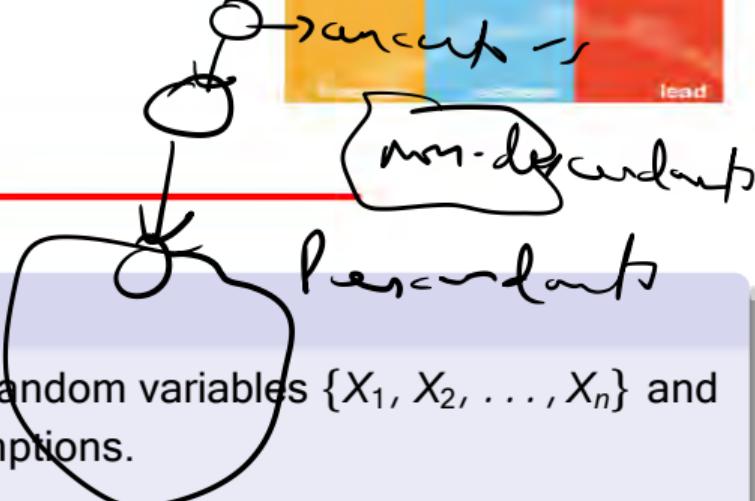
BAYESIAN NETWORK STRUCTURE

DEFINITION (LOCAL SEMANTICS)

A directed acyclic graph G whose nodes represent random variables $\{X_1, X_2, \dots, X_n\}$ and G encodes a set of conditional independence assumptions.

$$\text{For each variable } X_i : (X_i \perp \text{NonDescendants}_{X_i} | \text{Pa}_{X_i}) \quad (2)$$

- Pa_{X_i} represent parents of X_i in G .
- $\text{NonDescendants}_{X_i}$ represent the random variables that are not descendants of X_i .
- $I_L(G)$ represents the set of conditional independence assumptions called local independencies.



BAYESIAN NETWORK SEMANTICS

LOCAL SEMANTICS BN encodes a set of conditional independence assumptions.

For each variable X_i : $(X_i \perp \text{NonDescendants}_{X_i} | \text{Pa}(X_i))$

GLOBAL SEMANTICS BN represents a joint distribution via the chain rule.

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{Y_n} P(X_i | \text{Pa}(X_i))$$

MARKOV BLANKET A node is conditionally independent of all other nodes in the Bayesian Network, given its parents, children and children's parents.

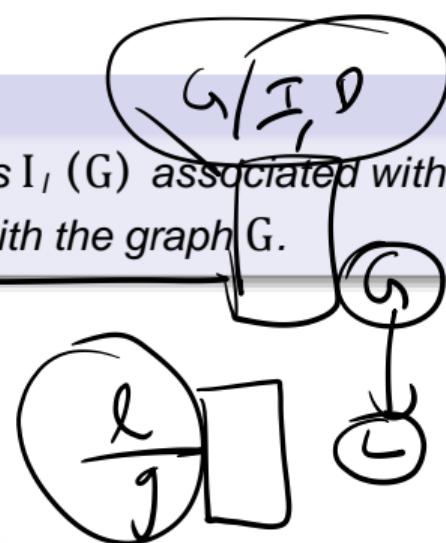
For each variable X_i : $(X_i \perp \text{other nodes} | \text{Pa}(X_i), \text{Ch}(X_i), \text{Pa}(\text{Ch}(X_i)))$

INDEPENDENCY MAP



THEOREM

A distribution P satisfies local independencies I_G associated with G if and only if P is representable as a set of CPDs associated with the graph G .



INDEPENDENCY MAP OR I-MAP

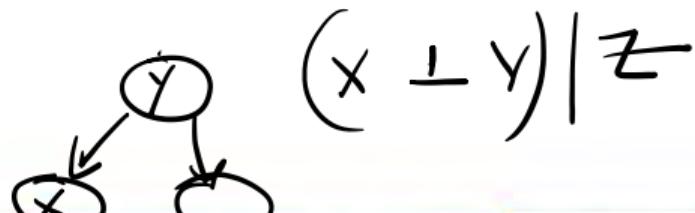
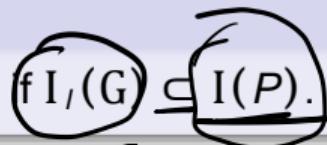
$$\boxed{X, X \perp\!\!\!\perp X} \quad \boxed{0.01 \rightarrow P(X, Y | Z) = P(X|Z)P(Y|Z)} \quad \boxed{I(P)}$$

- P be a distribution over X .
- $I(P)$ be the set of independence assertions ($X \perp\!\!\!\perp Y | Z$) that hold in P .
- Any independence that G asserts must also hold in P .



DEFINITION

G is called an **I-map** for P if $I_G \subseteq I(P)$.



$$P(X_0, Y_0) = P(X_0)P(Y_0) = \frac{0.4 \times 0.2}{0.08} P(X_0) = 0.4$$

I-MAP EXAMPLE 1

$$P(Y_0) = 0.2$$



X	Y	$P(X, Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

are X and Y
independent in P?

$$\underline{P(X, Y)}$$

$$P(X_0)P(Y_0) = P(X_0, Y_0)$$

$$P(X_0)P(Y_1) = P(X_0, Y_1)$$

$$P(X_1)P(Y_0) = P(X_1, Y_0)$$

$$X \perp Y$$

- Is G_ϕ : $X \perp Y$ an I-map of P?

if we simply the graph



(no edge between X & Y)

I-MAP EXAMPLE 1

X	Y	$P(X, Y)$
x^0	y^0	0.08
x^0	y^1	0.32
x^1	y^0	0.12
x^1	y^1	0.48

- Is $G_\phi : X \perp Y$ an I-map of P ?

$\underbrace{G_\phi}_{\text{encodes the assumption}} \text{ that } X \perp Y.$

- $P(x^1) = 0.48 + 0.12 = 0.60$
- $P(y^1) = 0.32 + 0.48 = 0.80$
- $P(x^1, y^1) = 0.48 = P(x^1)P(y^1)$
- Hence X and Y are independent i.e $(X \perp Y)$
- $(X \perp Y) \in I(P)$.
- G_ϕ is an I-map of P .

Similarly $P(x^0, y^0) = P(x^0)P(y^0)$
 $P(x^0, y^1) = P(x^0)P(y^1)$ & $P(x^1, y^0) = P(x^1)P(y^0)$

I-MAP EXAMPLE 2

X	Y	$P(X, Y)$
x^0	y^0	0.4
x^0	y^1	0.3
x^1	y^0	0.2
x^1	y^1	0.1

$$P(X, Y) \stackrel{?}{=} P(X) P(Y)$$

- Is $G_\phi : X \perp Y$ an I-map of P ?

I-MAP EXAMPLE 2

X	Y	$P(X, Y)$
x^0	y^0	0.4
x^0	y^1	0.3
x^1	y^0	0.2
x^1	y^1	0.1

- Is $G_\phi : X \perp\!\!\!\perp Y$ an I-map of P ?

- $P(x^1) = 0.2 + 0.1 = \cancel{0.3} \quad \text{d} \cdot 3$
- $P(y^1) = \underline{0.3 + 0.1} = \underline{0.4}$
- $P(x^1, y^1) \neq P(x^1)P(y^1)$
- Hence X and Y are not independent.
- $(X \perp\!\!\!\perp Y) \notin I(P)$.
- G_ϕ is not an I-map of P .

STUDENT EXAMPLE - B^{Student}

We know independence assumptions in G

$$(D \perp I) \implies$$

$$\underline{P(D|I)} = \underline{P(D)}$$

$$(L \perp I, D|G) \implies$$

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

$$(S \perp D, G, L|I) \implies$$

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

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

by chain rule

$$= P(I)P(D)P(G|I, D)\cancel{P(L|I, D, G)}P(S|I, D, G, L)$$

$$= P(I)P(D)P(G|I, D)\cancel{P(L|G)}P(S|I, D, G, L)$$

$$= P(I)P(D)P(G|I, D)\cancel{P(S|I)}P(L|G)$$

$P(x_i | Pa(x_i))$

A distribution P factorizes G if P can be represented as a chain rule of CPDs

P FACTORIZES OVER G

DEFINITION

Bayesian Network graph G over the variables X_1, X_2, \dots, X_n with a distribution P over the same space factorizes according to G if P can be expressed as a product of its CPDs.

$$P \text{ factorizes over } G \text{ if } P(X_1, X_2, \dots, X_n) = \prod_i^Y P(X_i | Pa^G(X_i)) \quad (3)$$

THEOREM

For a Bayesian Network graph G over the variables X_1, X_2, \dots, X_n with a distribution P over the same space and G is an I-map for P , then P factorizes G .

$\rightarrow G$ is an IMA for $P \Rightarrow P$ factorizes G

Proof of the theorem

Assume that $X_1, X_2 \dots X_n$ form a topological ordering (parents of X_i have subscripts smaller than i ; children of X_i have larger subscripts)

Using the chain rule

$$P(X_1, X_2, \dots, X_n) = P(X_1) P(X_2/X_1) P(X_3/X_1, X_2) \dots P(X_i/X_1, X_2, \dots, X_{i-1}) \dots$$

proof of the theorem

Consider one of the factors, $l(x_i | x_1, x_2, \dots, x_{i-1})$

Since g is an I-map for P we have

$$(x_i \perp \text{NonDescendants } x_i) \mid P^g$$

Now all of x_i 's parents are in the set

$\{x_1, x_2, \dots, x_{i-1}\}$ and none of x_i 's descendants can be in this set

Proof of th. theorem

$$\{x_1, x_2, \dots, x_{i-1}\} = Pa(x_i) \cup z \text{ where}$$

$z \subseteq \text{Non Descendants } x_i$

We know that $(x \perp z | Pa(x_i))$ which means that

$$P(x_i | Pa(x_i) \cup z) = P(x_i / Pa(x_i)) \prod P(x_i / Pa(x_i))$$

Now apply this to all the factors in the chain rule
to see that P factorizes G .



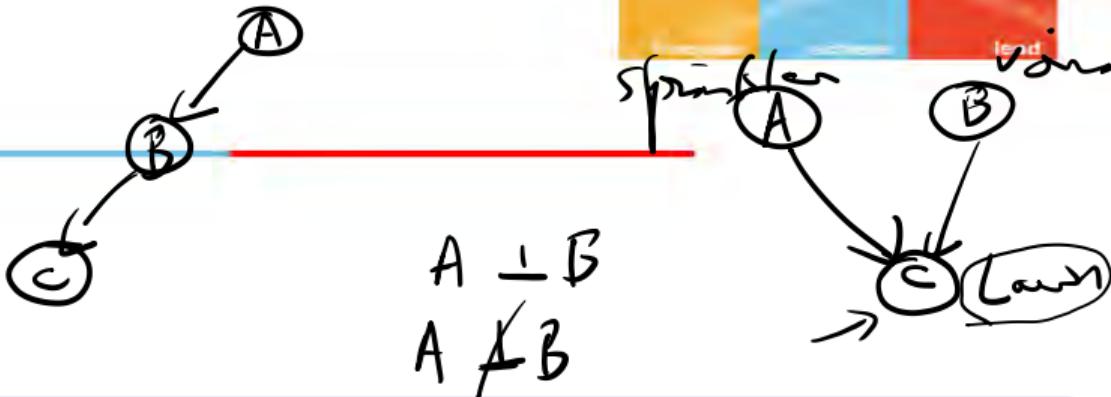
BAYESIAN NETWORK - ANOTHER DEFINITION

DEFINITION

A Bayesian network is a pair $B = (G, P)$ where P factorizes over G and where P is specified as a set of CPDs associated with G .

G IS AN I-MAP OF P

P(rain/sprinkler, law)



THEOREM

For a Bayesian Network graph G over the variables X_1, X_2, \dots, X_n with a distribution P over the same space and if P factorizes according to G, then G is an I-map for P.

G is an I-map for $P \Rightarrow P$ factorizes according to G

P factorizes according to $G \Rightarrow G$ is an IMAP for P

$$\sum P(S/I) = 1 \quad P(S) = \sum_I P(S/I)P(I)$$

STUDENT EXAMPLE - B_{Student}

Given: P factorizes according to G

By chain rule $\frac{P(I, D, G, S, L)}{P(I, D, G, S, L)} = \frac{P(I)P(D)P(G|I, D)P(S|I)P(L|G)}{P(I, D, G, S, L)}$

By definition $P(S|I, D, G, L) = \frac{P(I, D, G, S, L)}{P(I, D, G, L)}$

Marginalize over S $\boxed{P(I, D, G, L)} = \sum_S \underbrace{P(I, D, G, S, L)}_{S}$

Show that $(S \perp D, G, L | I)$ in P

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

$$P(A|P(B|A)) = P(A, B) = \frac{P(I)P(D)P(G|I, D)P(L|G)}{\sum S P(S|I)}$$

$\sum P(S/I)$?

STUDENT EXAMPLE - B^{Student}

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

$$\begin{aligned} P(S|I, D, G, L) &= \frac{P(I, D, G, S, L)}{P(I, D, G, L)} \\ &= \frac{P(I)P(D)P(G|I, D)P(S|I)P(L|G)}{P(I)P(D)P(G|I, D)P(L|G)} \\ &= P(S|I) \\ (S \perp D, G, L | I) \end{aligned}$$

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

$$(S \perp D, G, L | I)$$

Independence assumption holds. G is an I-map for P.

More generally:

We need to show that if P factorizes according to G
then G is an I-map for P

P Factorizes according to $G \Rightarrow X_i \perp \text{Non Descendants}(X_i)$
given Parents(X_i)

This means $P(X_i | \text{Non Descendants}(X_i)) = P(X_i | \text{Parents}(X_i))$
Note that $\text{Non Descendants}(X_i) \supseteq \text{Parents}(X_i)$

More Generally

$$P(X_i / ND(X_i)) = \frac{P(X_i, ND(X_i))}{P(ND(X_i))} ; ND(X_i) = \text{Non Descendants}(X_i)$$

Given the factorization of P we see that

$$P(X_i, ND(X_i)) = \prod P(Y_i / \text{Parents}(Y_i)) P(X_i / \text{Parents}(X_i))$$

where $Y_i \in ND(X_i)$

$$P(ND(X_i)) = \prod P(Y_i / \text{Parents}(Y_i)) \text{ where } Y_i \in ND(X_i)$$

More Generally

If $y_i \in ND(x_i)$, then $Parents(y_i) \subseteq ND(x_i)$

Then

$$\begin{aligned} P(x_i | ND(x_i)) &= \frac{\prod_{y_i \in ND(x_i)} P(y_i | Parents(y_i)) P(x_i | Pa(x_i))}{\prod_{y_i \in ND(x_i)} P(y_i | Parents(y_i))} \\ &= P(x_i | Pa(x_i)) \text{ as needed} \end{aligned}$$



REFERENCES

- 1 Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press. 2009
- 2 Artificial Intelligence: A Modern Approach (3rd Edition) by Stuart Russell, Peter Norvig
- 3 Mastering Probabilistic Graphical Models using Python by Ankur Ankan, Abhinash Panda. Packt Publishing 2015.
- 4 Learning in Graphical Models by Michael I. Jordan. MIT Press. 1999

Thank You !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 6 : UNDIRECTED GRAPHICAL MODEL

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.



Table of Contents

1 Undirected Graphical Models

Scenario 1

- Four people; Alice, Bob, Charlie, Diana; go out for dinner in different groups of two.
- Alice goes out with Bob, Bob goes out with Charlie, Charlie with Diana, and Diana with Alice.
- Bob doesn't go with Diana, and Alice doesn't go with Charlie.
- Let's think about the probability of them ordering food of the same cuisine.
- From our social experience, we know that people interacting with each other may influence each others choice of food.
- Alice can influence Bob's choice of cuisine. Bob can influence Charlie's choice of cuisine. But Alice and Charlie won't agree.
- How can we represent this in Bayesian Network?

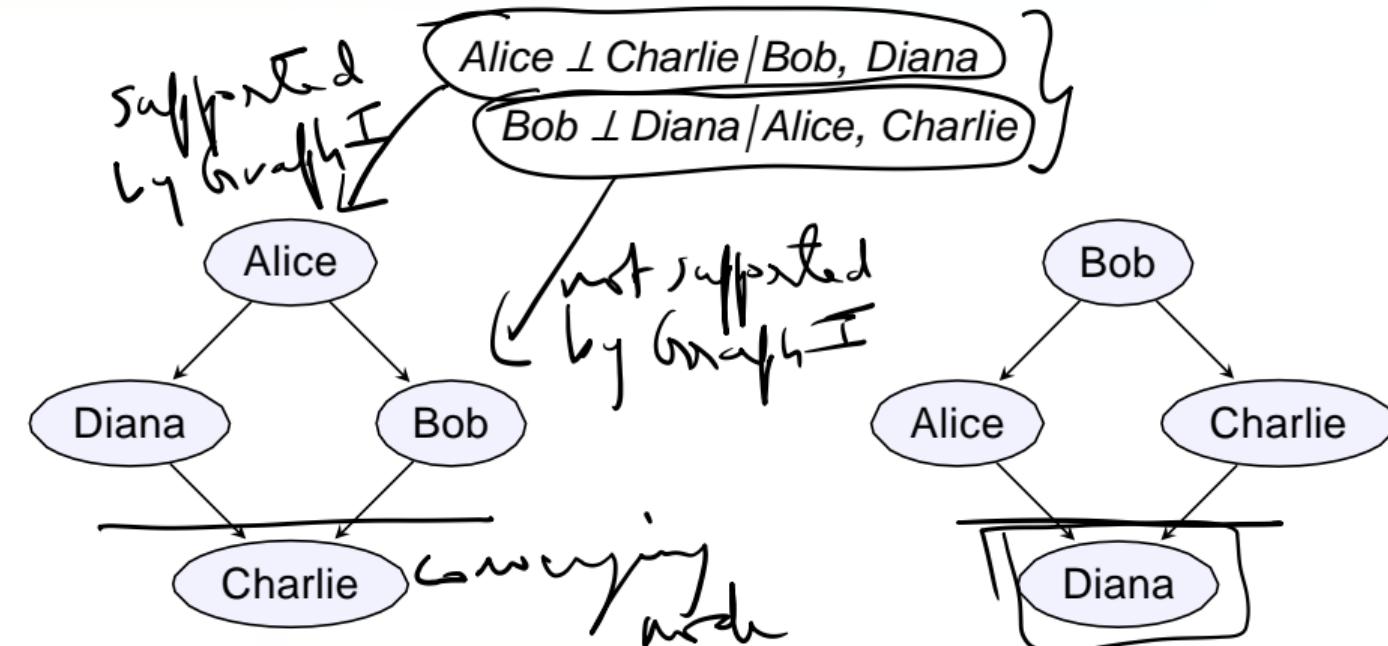
$Alice \perp Charlie | Bob, Diana$

$Bob \perp Diana | Alice, Charlie$

} cannot be
represented by
any Bayesian network

Scenario 1

split
rain
new
~~is saturated~~



What is the problem?

Scenario 1:

Bob and Diana are both serial users, so specifying them makes the paths from Alice to Charlie inactive.

Thus $A \perp C | B, D$

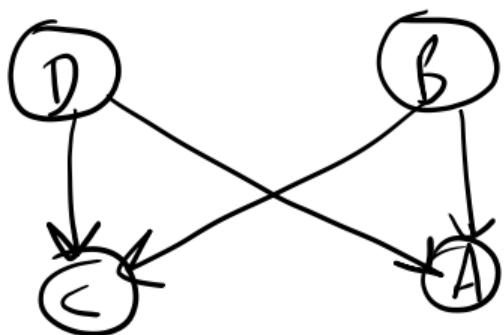
What is the problem?

Consider $B \perp D | A, C$

Here C is a converging node, so
specifying it creates a path of influence
between B and D

$\therefore B \not\perp D | A, C$

What's wrong here?



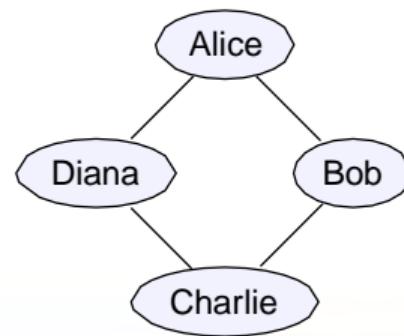
$$\begin{aligned} A \perp C \mid B, D \\ B \not\perp D \mid A, C \end{aligned}$$

B, D marginally
independent (i.e. $B \perp D \mid \emptyset$)

Scenario 1

- Directed models have a limitation that they cannot represent symmetric interactions.
- Undirected graphical model to encode influence flows in both directions.
- Example:

$Alice \perp Charlie | Bob, Diana$
 $Bob \perp Diana | Alice, Charlie$



Markov Network

Definition

Markov network is an undirected graph, where

- the nodes represent the random variables and
 - the dependencies or direct probabilistic interaction between these random variables are represented with undirected edges.
-
- No parent-child relationship.
 - So we do not use CPD.
 - Use factor to represent how likely it is for some states of a variable to agree with the states of other variables.

Parameterizing Markov Network

$$\overline{\pi p(x_i | P_a(x_i))}$$

or

$$\overline{\pi \phi_i(x)}$$

$$\phi(A, B, C)$$

- Markov Networks are parameterized using factors.
- Factors help in symmetric parameterization of random variables.
- Factors capture the affinities between related variables.
- Factors do not represent the probability.
- Factors are not constrained to sum up to 1 or to be in the range [0,1].
- The parameterization of the Markov network defines the local interactions between directly related variables.
- The scope of a factor to be the set of random variables over which it is defined.

Factor

- A **factor** Φ is a function or a table that maps a set of random variables to a real value.

$$\Phi : \text{Val}(X_1, \dots, X_n) \rightarrow \mathbb{R} \quad (3)$$

- The argument of the factor is called **scope** of the factor.

$$\text{Scope} : D = \underline{\{X_1, \dots, X_n\}} \quad (4)$$

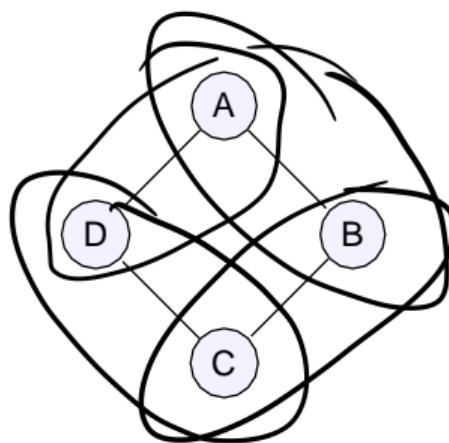
- Operations on a factor (Refer Session 3 for details)

- Marginalize a factor φ whose scope is W with respect to a set of random variables X , sum out all the entries of X , to reduce its scope to $\{W - X\}$.
- Reduction of a factor φ whose scope is W to the context $X = x'$ means removing all the entries from the factor where $X \neq x^i$. This reduces the scope to $\{W - X\}$.
- Factor product refers to the product of factors φ_1 with a scope X and φ_2 with scope Y to produce a factor φ_3 with a scope $X \cup Y$.

Factor

4

D	A	$\varphi(D, A)$
d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100



A	B	$\varphi(A, B)$
a^0	b^0	90
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

C	D	$\varphi(C, D)$
c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

B	C	$\varphi(B, C)$
b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

Queries using Factors

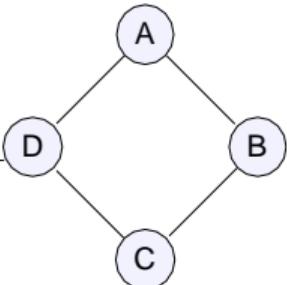
- Compute the probability corresponding to a^1, b^1, c^0, d^1 .

$$\begin{aligned} P(a^1, b^1, c^0, d^1) &= \varphi_1(a^1, b^1) \times \varphi_2(b^1, c^0) \times \varphi_3(c^0, d^1) \times \varphi_4(d^1, a^1) \\ &= \underline{10} \times \underline{1} \times \underline{100} \times \underline{100} = \underline{\underline{700,000}} \end{aligned}$$

↙

Factor Product

D	A	$\phi_4(D, A)$
d^0	a^0	80
d^0	a^1	60
d^1	a^0	20
d^1	a^1	10



C	D	$\phi_3(C, D)$
c^0	d^0	10
c^0	d^1	1
c^1	d^0	100
c^1	d^1	90

A	B	$\phi_1(A, B)$
a^0	b^0	90
a^0	b^1	100
a^1	b^0	1
a^1	b^1	10

B	C	$\phi_1(B, C)$
b^0	c^0	10
b^0	c^1	80
b^1	c^0	70
b^1	c^1	30

un-normalized

A	B	C	D	$P^*(A, B, C, D) = \Phi(A, B, C, D)$
a^0	b^0	c^0	d^0	$90 \cdot 10 \cdot 10 \cdot 80$
a^0	b^0	c^0	d^1	$90 \cdot 10 \cdot 1 \cdot 20$
a^0	b^0	c^1	d^0	$90 \cdot 80 \cdot 100 \cdot 80$
a^0	b^0	c^1	d^1	$90 \cdot 80 \cdot 90 \cdot 20$
a^0	b^1	c^0	d^0	$100 \cdot 70 \cdot 10 \cdot 80$
a^0	b^1	c^0	d^1	$100 \cdot 70 \cdot 1 \cdot 20$
a^0	b^1	c^1	d^0	$100 \cdot 30 \cdot 100 \cdot 80$
a^0	b^1	c^1	d^1	$100 \cdot 30 \cdot 90 \cdot 20$
a^1	b^0	c^0	d^0	$1 \cdot 10 \cdot 10 \cdot 60$
a^1	b^0	c^0	d^1	$1 \cdot 10 \cdot 1 \cdot 10$
a^1	b^0	c^1	d^0	$1 \cdot 80 \cdot 100 \cdot 60$
a^1	b^0	c^1	d^1	$1 \cdot 80 \cdot 90 \cdot 10$
a^1	b^1	c^0	d^0	$10 \cdot 70 \cdot 10 \cdot 60$
a^1	b^1	c^0	d^1	$10 \cdot 70 \cdot 1 \cdot 10$
a^1	b^1	c^1	d^0	$10 \cdot 30 \cdot 100 \cdot 60$
a^1	b^1	c^1	d^1	$10 \cdot 30 \cdot 90 \cdot 10$

Factor Product

- Factor Product

$$\tilde{P}(A, B, C, D) = \varphi_1(A, B) \times \varphi_2(B, C) \times \varphi_3(C, D) \times \varphi_4(D, A) \quad (5)$$

is un-normalized. It is not a probability distribution.

- Normalize $\tilde{P}(A, B, C, D)$ using partition function Z . Z is called the partition function and is a function of the parameters.

$$Z = \sum_{A, B, C, D} \tilde{P}(A, B, C, D) \quad (6)$$

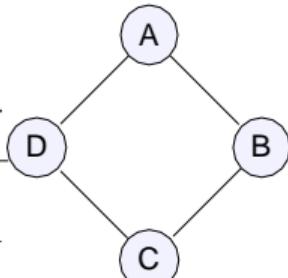
- Normalized factor product sums to all variants

$$P(A, B, C, D) = \frac{1}{Z} \tilde{P}(A, B, C, D) \quad (7)$$

~~negative
not dist.~~

Normalized Factor Product

D	A	$\phi_4(D, A)$
d^0	a^0	80
d^0	a^1	60
d^1	a^0	20
d^1	a^1	10



C	D	$\phi_3(C, D)$
c^0	d^0	10
c^0	d^1	1
c^1	d^0	100
c^1	d^1	90

A	B	$\phi_1(A, B)$
a^0	b^0	90
a^0	b^1	100
a^1	b^0	1
a^1	b^1	10

B	C	$\phi_2(B, C)$
b^0	c^0	10
b^0	c^1	80
b^1	c^0	70
b^1	c^1	30

A	B	C	D	$P^*(A, B, C, D)$	$P(A, B, C, D)$
a^0	b^0	c^0	d^0	720,000	0.0055
a^0	b^0	c^0	d^1	18,000	0.0001
a^0	b^0	c^1	d^0	57600,000	0.4365
a^0	b^0	c^1	d^1	12960,000	0.0982
a^0	b^1	c^0	d^0	5600,000	0.0424
a^0	b^1	c^0	d^1	140,000	0.0011
a^0	b^1	c^1	d^0	24000,000	0.1819
a^0	b^1	c^1	d^1	5400,000	0.0409
a^1	b^0	c^0	d^0	6,000	0.0000
a^1	b^0	c^0	d^1	100	0.0000
a^1	b^0	c^1	d^0	480,000	0.0036
a^1	b^0	c^1	d^1	72,000	0.0005
a^1	b^1	c^0	d^0	420,000	0.0318
a^1	b^1	c^0	d^1	70,000	0.0005
a^1	b^1	c^1	d^0	1800,000	0.1364
a^1	b^1	c^1	d^1	270,000	0.0205
				109493,100	1.0

Queries using Factor Product

- Compute the probability of B.

Marginalize wrt A,C,D

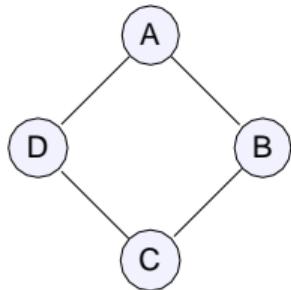
$$P(b^1) = 0.4555$$

$$P(b^0) = 0.5445$$

- Compute the probability of B agreeing with C given c^0 .

$$P(b^1 | c^0) = 0.0759$$

Factors vs Probability Distribution

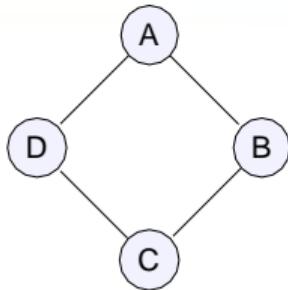


A	B	$\phi(A, B)$	factor
a^0	b^0	90	
a^0	b^1	100	
a^1	b^0	1	
a^1	b^1	10	

prob dist

Marginal Probability of A and B				$P_\phi(A, B)$
A	B	C	D	$P(A, B, C, D)$
a^0	b^0	c^0	d^0	0.0055
a^0	b^0	c^0	d^1	0.0001
a^0	b^0	c^1	d^0	0.4365
a^0	b^0	c^1	d^1	0.0982
a^0	b^1	c^0	d^0	0.0424
a^0	b^1	c^0	d^1	0.0011
a^0	b^1	c^1	d^0	0.1819
a^0	b^1	c^1	d^1	0.0409
a^1	b^0	c^0	d^0	0.0000
a^1	b^0	c^0	d^1	0.0000
a^1	b^0	c^1	d^0	0.0036
a^1	b^0	c^1	d^1	0.0005
a^1	b^1	c^0	d^0	0.0318
a^1	b^1	c^0	d^1	0.0005
a^1	b^1	c^1	d^0	0.1364
a^1	b^1	c^1	d^1	0.0205
				0.1892

Factors vs Probability Distribution



		$\varphi_1(A, B)$	
A	B	$\varphi_1(A, B)$	
a^0	b^0	90	
a^0	b^1	100	
a^1	b^0	1	
a^1	b^1	10	

		$P_\varphi(A, B)$	
A	B	$P_\varphi(A, B)$	
a^0	b^0	0.5403	
a^0	b^1	0.2663	
a^1	b^0	0.0042	
a^1	b^1	0.1892	

Marginal Probability of A and B

A	B	C	D	$P(A, B, C, D)$	$P_\varphi(A, B)$
a^0	b^0	c^0	d^0	0.0055	
a^0	b^0	c^0	d^1	0.0001	
a^0	b^0	c^1	d^0	0.4365	
a^0	b^0	c^1	d^1	0.0982	0.5403
a^0	b^1	c^0	d^0	0.0424	
a^0	b^1	c^0	d^1	0.0011	
a^0	b^1	c^1	d^0	0.1819	
a^0	b^1	c^1	d^1	0.0409	0.2663
a^1	b^0	c^0	d^0	0.0000	
a^1	b^0	c^0	d^1	0.0000	
a^1	b^0	c^1	d^0	0.0036	
a^1	b^0	c^1	d^1	0.0005	0.0042
a^1	b^1	c^0	d^0	0.0318	
a^1	b^1	c^0	d^1	0.0005	
a^1	b^1	c^1	d^0	0.1364	
a^1	b^1	c^1	d^1	0.0205	0.1892

There is no natural mapping between factors and probability distribution.

Factorization and Independencies

$$\varphi_5(B, D, A) \quad \varphi_6(B, D, C)$$

$$\varphi_7(A, C, D)$$

- $P \models (B \perp D | A, C)$ should have a decomposition

$$P = \frac{1}{Z} \underbrace{[\varphi_1(A, B) \times \varphi_2(B, C)]}_{F(A, B, C)} \times \underbrace{\varphi_3(C, D) \times \varphi_4(D, A)}_{G(A, C, D)}$$

B and D are separated given A and C.

- $P \models (A \perp C | B, D)$ should have a decomposition

$$P = F(B) G(D)$$

$$P = \frac{1}{Z} \underbrace{[\varphi_4(D, A) \times \varphi_1(A, B)]}_{F(A)} \times \underbrace{\varphi_2(B, C) \times \varphi_3(C, D)}_{G(C)}$$

A and C are separated given B and D.

Factorization and Independencies

$P \models X \perp Y/Z$ if and only if contains

$$P = \varphi_1(X, Z)\varphi_2(Y, Z) \quad (8)$$

- Independence properties of the distribution P correspond directly to separation properties in the graph over which P factorizes.

Factors can be misleading

a^o	b^o	c^o	d^o	0.04
a^o	b^o	c^o	d^o	0.04
a^o	b^o	c^i	d^o	0.04
a^o	b^o	c^i	d^i	4.1×10^{-6}
a^o	b^i	c^o	d^o	1.9×10^{-5}
a^o	b^i	c^i	d^i	6.9×10^{-5}
a^i	b^o	c^i	d^o	0.69
a^o	b^i	c^i	d^i	6.9×10^{-5}
a^i	b^o	c^i	d^o	1.4×10^{-5}
a^i	b^o	c^o	d^i	0.14
a^i	b^o	c^i	d^o	1.4×10^{-5}
a^i	b^o	c^i	d^i	1.4×10^{-5}
a^i	b^i	c^o	d^o	1.4×10^{-5}
a^i	b^i	c^i	d^i	0.014
a^i	b^i	c^i	d^o	0.014
a^i	b^i	c^i	d^i	0.014

Joint Distribution
 for the
 Misconception
Example

Factors for Misclassification Example

$$\phi_1(A, B)$$

a^o	b^o	30
a^o	b'	5
a'	b^o	1
a'	b'	10

$$\phi_2(B, C)$$

b^o	c^o	100
b^o	c'	1
b'	c^o	1
b'	c'	100

$$\phi_3(C, D)$$

c^o	d^o	1
c^o	d'	100
c'	d^o	100
c'	d'	1

$$\phi_4(D, A)$$

d^o	a^o	100
d^o	a'	1
d'	a^o	1
d'	a'	100

The factor $\phi_1(A, B)$ suggests that A and B are mostly in agreement.

Marginal Distribution for A, B in Misconception Example

A	B	
a	b	0.13
a	b'	0.69
a'	b	0.14
a'	b'	0.04

Here we see that A and B are mostly in disagreement unlike in $\phi_1(A, B)$

This is because of the influence of the other factors on the distribution

Influence of other factors

$\phi_3(C, D)$ asserts that Charles and Debbie disagree

$\phi_2(B, C)$ asserts that Bob and Charles agree.

$\phi_4(D, A)$ asserts that Debbie and Alice agree

The implication of the above is that Alice and Bob disagree.

$$B - C \cancel{\rightarrow} D - A$$

Gibbs Distribution

Definition

A distribution P_{Φ} is called a Gibbs distribution parameterized by a set of factors $\Phi = \{\varphi_1(D_1), \dots, \varphi_k(D_k)\}$ if it can be expressed as product of the factors.

$$P_{\Phi}(X_1, \dots, X_n) = \frac{1}{Z_{\Phi}} [\varphi_1(D_1) \times \dots \times \varphi_k(D_k)]$$

$$\tilde{P}(X_1, \dots, X_n) = \prod_{i=1}^k \varphi_i(D_i) \quad (9)$$

$$Z_{\Phi} = \sum_{X_1, \dots, X_n} \tilde{P}(X_1, \dots, X_n) \quad (10)$$

$$P_{\Phi}(X_1, \dots, X_n) = \frac{1}{Z_{\Phi}} \tilde{P}(X_1, \dots, X_n) \quad (11)$$

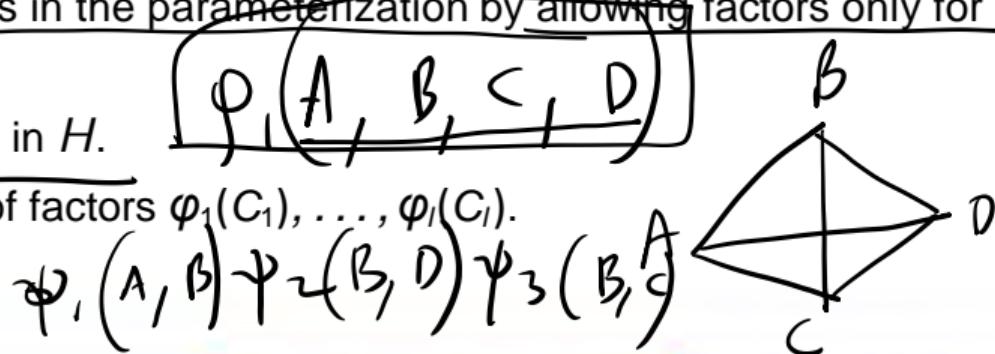
Gibbs Distribution

$$\prod p(x_i/p_a(x_i))$$

Definition

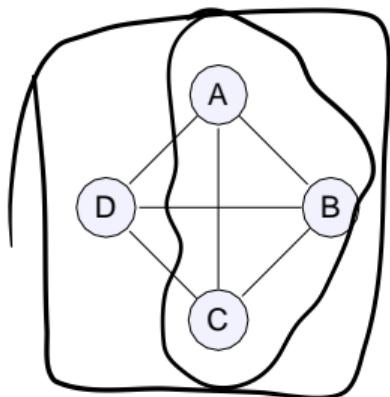
A distribution P_Φ with $\Phi = \{\varphi_1(D_1), \dots, \varphi_k(D_k)\}$ factorizes over a Markov Network H if each D_k is a complete subgraph of H .

- The factors that parameterize a Markov network are often called **clique potentials**.
- Reduce the number of factors in the parameterization by allowing factors only for maximal cliques.
- Let C_1, \dots, C_k be the cliques in H .
- Parameterize P using a set of factors $\varphi_1(C_1), \dots, \varphi_l(C_l)$.



Gibbs Distribution Example

$$P_i \neq \psi_1(\underline{A, B}) \psi_2(\underline{B, C}) \psi_3(\underline{C, D}) \dots$$



- Cliques (Option 1):
 $\{A, B\}, \{B, C\}, \{C, D\},$
 $\{D, A\}, \{D, B\}, \{A, C\}$
- Cliques (Option 2):
 $\{A, B, D\}, \{B, C, D\}$
- Cliques (Option 3):
 $\{A, B, C\}, \{A, C, D\}$

Pairwise Markov Network

$\varphi(A, B)$ ← too many parameters
not needed

Definition

Pairwise Markov Network is an undirected graph whose nodes X_1, \dots, X_n and edges $X_i - X_j$ are associated with a factor $\varphi_{ij}(X_i, X_j)$.

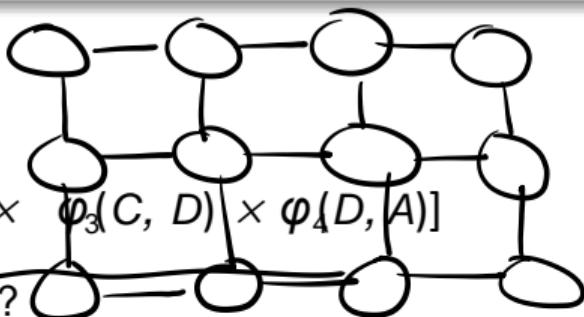
- A subclass of Markov networks.

- Eg:

↗

$$P(A, B, C, D) = \frac{1}{Z} [\varphi_1(A, B) \times \varphi_2(B, C) \times \varphi_3(C, D) \times \varphi_4(D, A)]$$

- How many parameters for n RV with d values each?



Number of parameters in Pairwise Markov Network $= O(n^2 d^2)$ (13)

$$P(X_1, X_2, \dots, X_n) = \varphi_1(X_1, X_2) \times \varphi_2(X_2, X_3) \times \varphi_3(X_3, X_4) \dots \varphi_{n-1}(X_{n-1}, X_n)$$

Induced Markov Network

Definition

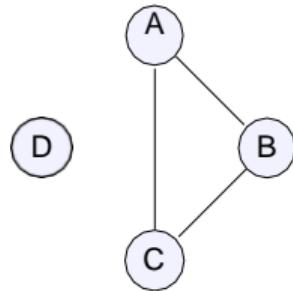
For a set of factors φ_i , with a scope D_i , the Induced Markov Network H_Φ , has an edge between a pair of variables X_i and X_j whenever there exists a factor $\varphi_m \in \Phi$ such that $X_i, X_j \in D_m$.

- X and Y will have an undirected edge
 -) if they appear together in some factor φ
 -) if there exists a factor $\varphi(X, Y)$.

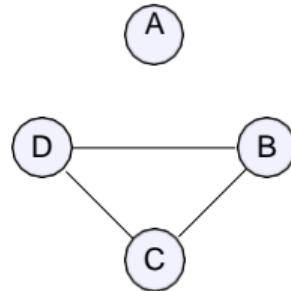
Induced Markov Network

Consider 4 RVs A,B,C, and D. The factor and its induced Markov Network is given below.

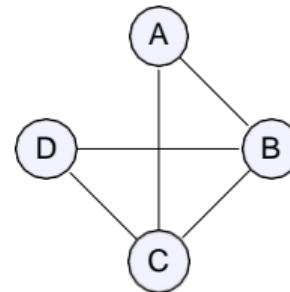
$$\varphi_1(A, B, C)$$



$$\varphi_2(B, C, D)$$



$$\Phi = \varphi_1(A, B, C) \times \varphi_2(B, C, D)$$



P factorizes H

Definition

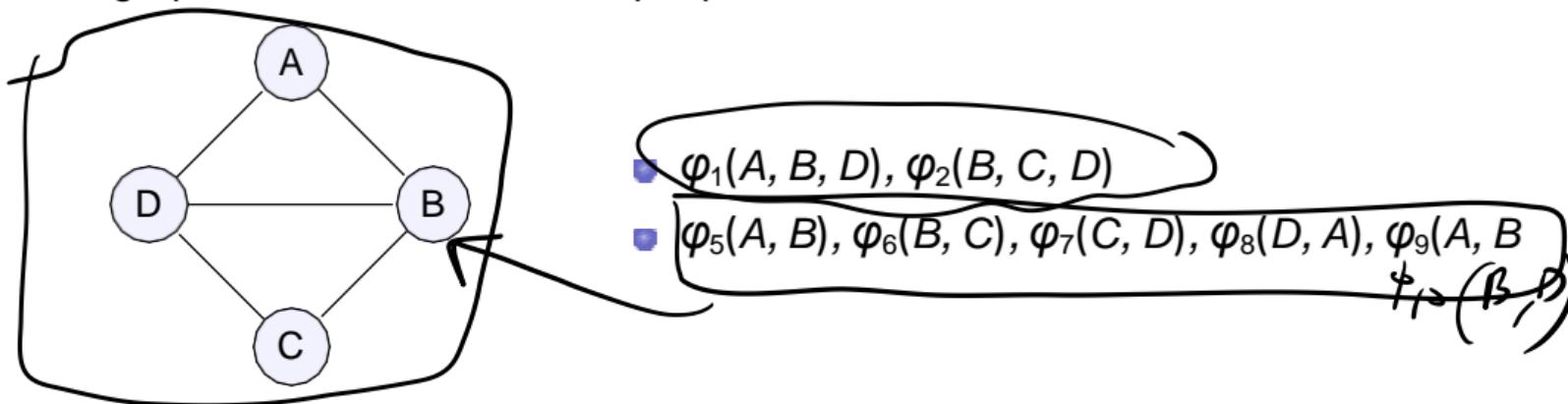
Gibbs distribution P factorizes a Markov Network H if there exists $\Phi = \{\varphi_1(D_1), \dots, \varphi_k(D_k)\}$ such that

- $P = P_\Phi$, normalized product of factors φ_i
- H is the induced graph for Φ .

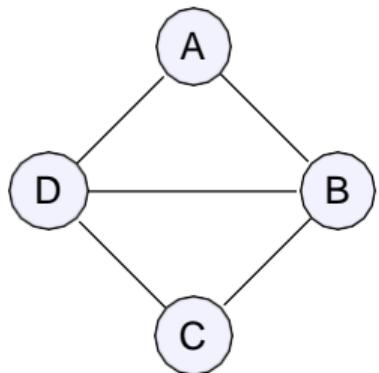
$$\frac{1}{Z} P$$

P factorizes H

- From an induced Markov network H , we cannot read the factorization P_Φ from the graph, as there can be multiple possible factorizations.



Flow of Influence

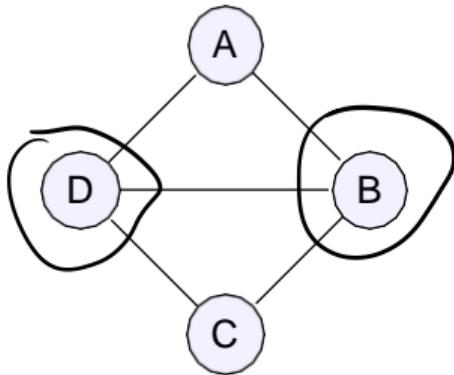


- $\varphi_1(A, B, D), \varphi_2(B, C, D)$
- $\varphi_5(A, B), \varphi_6(B, C), \varphi_7(C, D), \varphi_8(D, A), \varphi_9(B, D)$
- When can B influence D ?
- When can A influence C ?

D -> A

H -> A

Flow of Influence



- $\varphi_1(A, B, D), \varphi_2(B, C, D)$
- $\varphi_5(A, B), \varphi_6(B, C), \varphi_7(C, D), \varphi_8(D, A), \varphi_9(B, D)$
- When can B influence D?
 -) Direct influence
 -) $\varphi_1(A, B, D)$
 -) $\varphi_9(B, D)$

- When can A influence C?

-) Indirect influence
-) Through B or D
-) $\varphi_1(B, C, D)$

$\varphi_1(A, B, D)\varphi_2(B, C, D)$

$\varphi_5(A, B), \varphi_6(B, C)$

$\varphi_7(C, D), \varphi_8(D, A)$

A, \cancel{D}, C , $A, \cancel{B}, \cancel{C}$
 $A, \cancel{B}, \cancel{C}$, A, \cancel{B}, C

Flow of Influence

- Parameterization of the distributions are different.
- The trails in the graph through which influence can flow are the same.
- Active trails depend only on the graph structure.

References

- 1 Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press. 2009
- 2 Artificial Intelligence: A Modern Approach (3rd Edition) by Stuart Russell, Peter Norvig
- 3 Mastering Probabilistic Graphical Models using Python by Ankur Ankan, Abhinash Panda. Packt Publishing 2015.
- 4 Learning in Graphical Models by Michael I. Jordan. MIT Press. 1999

Thank You !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

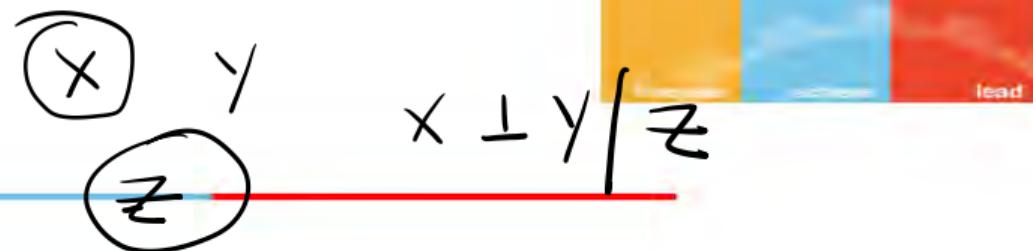
PROBABILISTIC GRAPHICAL MODEL SESSION # 7 : UNDIRECTED GRAPHICAL MODEL

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



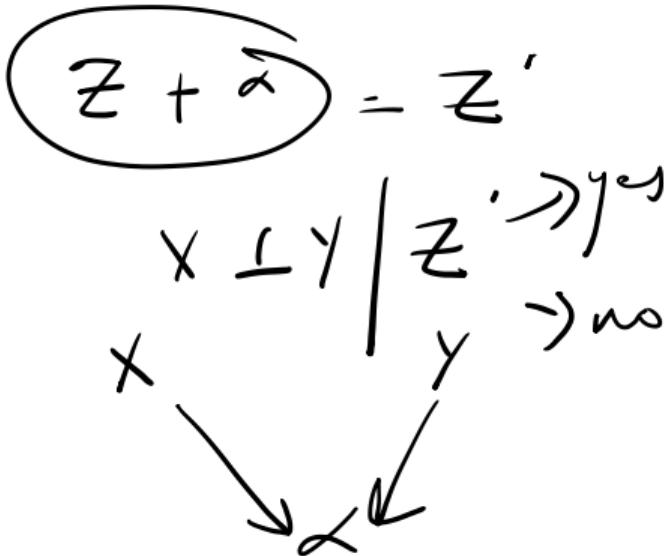
The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.

Table of Contents



1 Markov Network Independencies

2 Bayesian Network vs Markov Network



Active Trail in Markov Network

Definition

Let H be a Markov network structure and let $X_1 - , \dots, - X_n$ be a path in H .

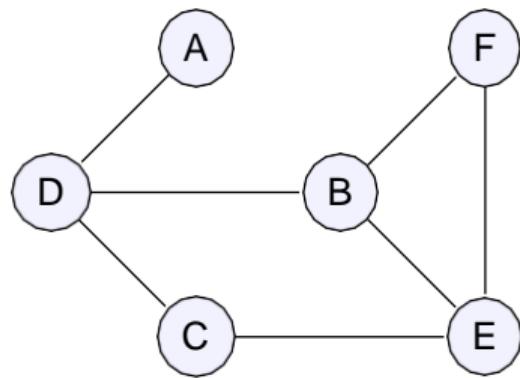
Let $Z \subseteq X$ be a set of observed variables.

The path $X_1 - , \dots, - X_n$ is **active** given Z if none of the X_i is in Z .

- Influence has to flow through unobserved variables along the trail.
- Once a variable is observed along the trail, the influence is blocked.

Markov Network - Example

- Find the active trails given B is observed.



$A - D - C - E - F$

Separation in Markov Network

Definition

A set of nodes Z separates X and Y in H , a Markov network structure, if there is no active path between any node in X and Y given Z .

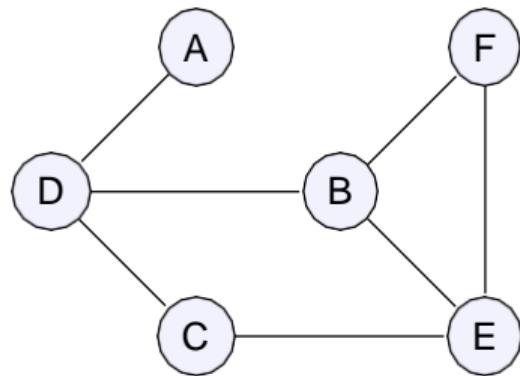
- Denote Separation as $\text{sep}_H(X; Y|Z)$
- Global Independencies

$$I_g(H) = \{(X \perp Y|Z) : \text{sep}_H(X; Y|Z)\} \quad (1)$$

- The Independencies in $I(H)$ are precisely those that are guaranteed to hold for every distribution P over H .

Markov Network - Example

- Find the global Independencies.



$$\begin{aligned}I_g(H) &= \{(A \perp B|D) : sep_H(A; B|D)\} \\&= \{(A \perp C|D) : sep_H(A; C|D)\} \\&= \{(A \perp E|D) : sep_H(A; E|D)\} \\&= \{(A \perp F|D) : sep_H(A; F|D)\}\end{aligned}$$

Factorization implies Independence

Theorem

Let P be a distribution over X and H a Markov Network structure over X .
If P is a Gibbs distribution that factorizes over H ~~and~~ H is an I-map for P .

then

Proof

Let X, Y and Z be any 3 disjoint sets in \mathcal{X} such that Z separates X and Y in H

Consider two cases:

① $X \cup Y \cup Z = \mathcal{X}$ (set of all vertices in the graph)

We need to show that $I \models (X \perp Y | Z)$
(X is independent of Y given Z)

Proof

Since Z separates X and Y , there are no direct edges between X and Y

Therefore any clique in H is completely contained in $X \cup Z$ or $Y \cup Z$

Let I_X be the indexes of the cliques completely contained in $X \cup Z$. Let I_Y be the remaining cliques.

Proof

Since P is a Gibbs distribution that factorizes over H we can write

$$P(x_1, x_2, \dots, x_n) = \frac{1}{Z} \prod_{i \in I_x} \phi_i(D_i) \cdot \prod_{i \in I_y} \phi_i(D_i)$$

None of the factors in the first product involve any variable in Y and none in the second product involve any variable in X .

Proof

We can rewrite $P(X_1, X_2, \dots, X_n)$ such that

$$P(X_1, X_2, \dots, X_n) = \frac{1}{Z} f(x, z) g(y, z)$$

\rightarrow this Z is the partition function

Thus $(X \perp Y | Z)$

Case b: $X \cup Y \cup Z \subset X$ (proper subset of X)

Let $U = X - X \cup Y \cup Z$

Proof

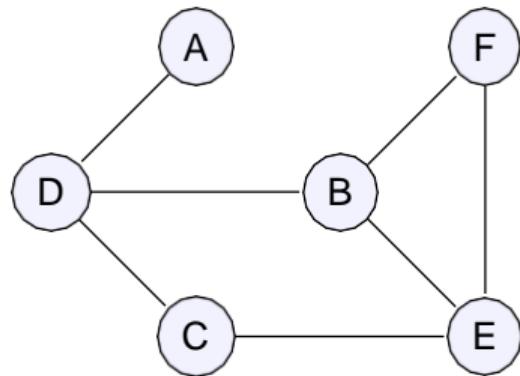
Partition \cup into disjoint sets \cup_1 and \cup_2 such that Z separates $X \cup \cup_1$ from $Y \cup \cup_2$ in H .

Using the previous argument we can conclude $X, \cup_1 \perp Y, \cup_2 | Z$

Using the decomposition property we can conclude that $X \perp Y | Z$

Markov Network - Example

- Identify a possible factorization for the Markov Network.



$$\begin{aligned}P_G = & \varphi_1(A, D)\varphi_2(B, E, F) \\& \varphi_3(D, B)\varphi_4(D, C)\varphi_5(C, E)\end{aligned}$$

Independence implies Factorization

Theorem

Hammersley-Clifford theorem:

Let P be a positive distribution over X and H a Markov Network structure over X .
If H is an I-map for P , then P is a Gibbs distribution that factorizes over H .

Need for positive distribution

Look at Example 4.4 from Daphna Koller's book

Distribution P over X_1, X_2, X_3, X_4 which has a value $\frac{1}{8}$ for 8 combinations which are

(0000) (1000) (1100) (1110)

(0001) (0011) (0111) (1111) and 0 for all other combinations.

Need for positive distribution

The distribution is not a true distribution as some entries are 0s

Let H be the graph

$$x_1 - x_2 - x_3 - x_4 - x_1$$

P satisfies the global independencies with respect to H

Need for positive distribution

The graph asserts that $x_1 \perp x_3 | x_2, x_4$

Does P satisfy this relationship?

$$\text{We have } P(x_1=1 | x_2=1, x_4=0) = 1$$

So for this assignment of x_2, x_4 we can see
that x_1 is independent of x_3

Similarly for other assignments & other independencies

Need for positive distribution

Does this distribution factorize over H ?

Can we express the distribution P as

$$\phi_1(x_1, x_2) \phi_2(x_2, x_3) \phi_3(x_3, x_4) \phi_4(x_4, x_1) ?$$

$\phi_1(x_1, x_2)$	$\phi_2(x_2, x_3)$	$\phi_3(x_3, x_4)$	$\phi_4(x_4, x_1)$
0 0 α_1	0 0 β_1	0 0 γ_1	0 0 ρ_1
0 1 α_2	0 1 β_2	0 1 γ_2	0 1 ρ_2
1 0 α_3	1 0 β_3	1 0 γ_3	1 0 ρ_3
1 1 α_4	1 1 β_4	1 1 γ_4	1 1 ρ_4

Need for positive distribution

We need to find values for the parameters

$$(\alpha_1, \alpha_2, \gamma_3, \gamma_4) \dots (\beta_1, \beta_2, \beta_3, \beta_4)$$

$$P(0000) = \frac{1}{8} \quad \alpha_1 \beta_1 \gamma_1 \beta_1 \propto \frac{1}{8}$$

$$P(0010) = 0 \quad \alpha_1 \underline{\beta_2 \gamma_3} \beta_1 \propto 0 \Rightarrow \text{one } \beta$$

β_2, γ_3 must be 0

$$P(0011) = \frac{1}{8} \quad \alpha_1 \beta_2 \gamma_4 \beta_3 \propto \frac{1}{8} \Rightarrow \beta_2 \text{ cannot be 0}$$

Need for positive distribution

So γ_3 must be a 0.

But $P(1110) = \frac{1}{8} \Rightarrow \alpha_4 \beta_4 \gamma_3 \rho_2 \neq \frac{1}{8}$

So γ_3 cannot be a 0

[contradiction]

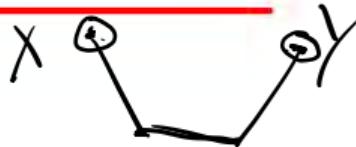
Proof of Hammersley-Clifford Theorem

Take a look at

https://vision.in.tum.de/_media/teaching/ss2017/pgmcv/in2329-02_gm.pdf

The statement of the theorem here is different from Koller's book (but equivalent) and this is a full proof → focus on Forward Direction in the above document

Pairwise Independencies



Definition

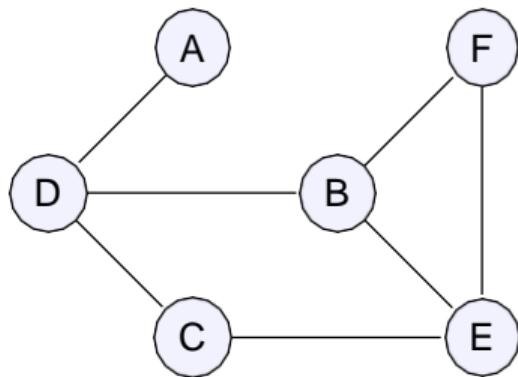
Let H be a Markov network structure. Pairwise Independencies associated with H is defined as

$$I_p(H) = \{(X \perp Y | X - \{X, Y\}) : \text{edge}(X - Y) \notin H\} \quad (2)$$

X and Y are independent given all the remaining nodes in the network.

Markov Network - Example

- Find the pairwise independencies.



For node A

$$\begin{aligned}I_p(H) &= (A \perp C | D, B, E, F : \text{edge}(A - C) \notin H) \\&= (A \perp B | D, C, E, F : \text{edge}(A - B) \notin H) \\&= (A \perp E | D, B, C, F : \text{edge}(A - E) \notin H) \\&= (A \perp F | D, B, E, C : \text{edge}(A - F) \notin H)\end{aligned}$$

Markov Blanket

Definition

For a given graph H , the Markov blanket of X in H is defined as neighbours of X in H .

$$MB_X = \{Pa(X), Ch(X), Pa(Ch(X))\}$$

this defn does not apply here (3)

- Markov blanket is the set of nodes containing parents, children, and children's parents.

Local Independencies

Definition

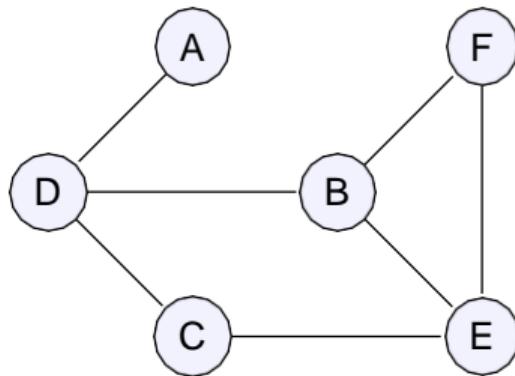
Let H be a Markov network structure. Local Independencies associated with H is defined as

$$I_L(H) = \{(X \perp X - \{X + MB_H(X)\} / MB_H(X)) : X \in X\} \quad (4)$$

X is independent of the rest of the nodes in the graph
given its immediate neighbours

Markov Network - Example

- Find the local independencies and Markov blanket.



For node ~~A~~ ~~D~~

$$MB_{\cancel{A}} = \{A, D, B, C\}$$

$$I_1(H) = (\cancel{A} \perp E, F / MB_{\cancel{A}})$$

Independencies

Definition

For any Markov network H and any distribution P ,

$$\text{if } P \models I_1(H) \text{ then } P \models I_p(H) \quad (5)$$

$$\text{if } P \models I_g(H) \text{ then } P \models I_1(H) \quad (6)$$

- $I_p(H)$ is strictly weaker than $I_1(H)$ which is strictly weaker than $I_g(H)$
- For a positive distribution P ,

$$P \models I_p(H)$$

$$P \models I_1(H)$$

$$P \not\models I_g(H)$$

Pairwise and Local Dependencies

Theorem: For any Markov network H and any distribution P , we have that if $P \models I_e(H)$ then $P \models I_p(H)$

let us look at its proof

Proof

$$\begin{array}{c} z \\ \bullet \\ x \end{array} \equiv \begin{array}{c} y \\ = w \end{array}$$

First we need to prove the Weak Union property of all other nodes, other than $X \perp Y \mid Z$

$$(X \perp Y, W \mid Z) \Rightarrow (X \perp Y \mid W, Z) \text{ pairwise indep}$$

local indep
If $P(W = w, Z = z) = 0$ then the implication

follows trivially

Pr-of

Assume that $P(z) \neq 0$ and $P(z, \omega) \neq 0$. Then

$$P(x, y | z, \omega) = \frac{P(x, y, \omega | z)}{P(\omega | z)}$$

Since $(x \perp y, \omega | z)$ we can write $P(x, y, \omega | z)$

$$= P(x | z) P(y, \omega | z)$$

P_{v-f}

$$\text{Then we can write } P(X, Y | Z, \omega) = \frac{P(X|Z)P(Y, \omega | Z)}{P(\omega | Z)}$$

$$= \frac{P(X|Z)P(Y|\omega, Z)P(\omega | Z)}{P(\omega | Z)} = P(X|Z)P(Y|Z, \omega)$$

According to the Decomposition Rule

$$(X \perp Y, \omega | Z) \Rightarrow (X \perp \omega | Z)$$

Pr-of

Using the Decomposition Rule we have

$$(X \perp Y, \omega / Z) \Rightarrow (X \perp \omega / Z) \text{ which means}$$

$$P(X/\omega, Z) = P(X/Z)$$

$$\text{We had } P(X, Y / Z, \omega) = P(X/Z) P(Y/Z, \omega) \text{ from which}$$

$$\begin{aligned} \text{we have } P(X, Y / Z, \omega) &= P(X/\omega, Z) P(Y/Z, \omega) \\ &\Rightarrow (X \perp Y / Z, \omega) \end{aligned}$$

Part

Armed with the weak union property, we can prove the result we want:

Let $Z = N_H\{X\} \rightarrow$ neighbor set of X

Let $W = \pi - \{X, Y\} - Z$

Markov Assumption ($I_2(H)$) tells us that

$$(X \perp\!\!\!\perp Y \mid W \mid Z)$$

Pr-F

Applying the just proved Weak Union property
we have

$$(X \perp \{Y\} \text{ given } Z) \Rightarrow (X \perp Y \mid \omega, Z)$$

i.e. Markov Property \Rightarrow Pairwise Independence

Local and Global Independencies

Proposition 4.4 For any Markov network H and any distribution P we have that if

$I \models I(H)$ then $P \models I_L(H)$



$$I(H) = \{ (X \perp Y | Z) : \text{sep}_H(X; Y | Z) \}$$

$$I_L(H) = \{ X \perp \chi - \{X\} - MB_H(X) \mid MB_H(X) \}$$

When are $I(H)$, $I_{\ell}(H)$, $I_P(H)$ all equivalent?

For positive distributions

Theorem 4.4 Let P be a positive distribution.
~~If P satisfies $I_P(H)$ then P satisfies $I(H)$~~

Earlier we showed that

$$P \models I(H) \Rightarrow P \models I_{\ell}(H) \Rightarrow P \models I_P(H)$$

for any distribution P

Proof of Theorem 4.4

We want to show that $P \models I(H)$ given that $P \models I_P(H)$ for all disjoint sets X, Y and Z .
 $P \models I(H)$ means that whenever it is true that $\text{sep}_H(X; Y|Z)$, then $P \models (X \perp Y|Z)$.

Proof of Theorem 4.4

Proof is by induction on the size of Z

When $|Z| = n - 2$, the statement follows trivially since $I_p(H)$ and $I(H)$ then mean the same thing (X and Y are individual nodes in this case)

Assume that $\text{Sep}_H(X; Y | Z) \Rightarrow P \models (X \perp Y | Z)$ holds for every Z whose $|Z| = k$.

Proof of Theorem 4.4

Let Z be any set such that $|Z| = k - 1$

There are two cases:

Case (a): $X \cup Y \cup Z = X$ (set of all vertices)

As $|Z| < n - 2$ we must have either $|X| > 2$

or $|Y| > 2$. Assume that $|Y| > 2$

There exists $A \in Y$. Let $Y' = Y - \{A\}$

Proof of Theorem 4.4

Since $\text{Sep}_H(X; Y|Z)$ we must also have

$\text{Sep}_H(X; Y'|Z)$ and $\text{Sep}_H(X; A|Z)$

Separation is monotonic so we have

$\text{Sep}_H(X; Y'(Z \cup \{A\}))$ and $\text{Sep}_H(X; A(Z \cup Y'))$

Each of the sets $Z \cup \{A\}$ and $Z \cup Y'$ has size at least k

Proof of Theorem 4.4

Therefore the induction hypothesis applies
and we must have P satisfies:

$$\textcircled{1} \quad X \perp Y' / Z \cup \{A\} \text{ and } \textcircled{2} \quad X \perp A / Z \cup Y'$$

Since P is a positive distribution the
intersection property applies and we have

$$P \models X \perp Y' \cup \{A\} / Z, \text{ ie } X \perp Y / Z \text{ (done)}$$

Proof of Theorem 4.4

Case (b): $X \cup Y \cup Z \subseteq \chi$

In this case there is a node A that does not belong to $X \cup Y \cup Z$.

We have $\text{sep}_H(X; Y | Z)$. From monotonicity of separation we have $\text{sep}_H(X; Y | Z \cup \{A\})$

Proof of Theorem 4.4

Since X and Y are separated given Z , there cannot exist a path between X and A and between Y and A . At most one of these paths can exist \rightarrow assume that there is no path between X and A given Z .

By monotonicity, we must have $\text{Sep}_{\text{IT}}(X; A | Z \cup Y)$

Proof of Theorem 4.4

As before $Z \cup \{A\}$ and $Z \cup Y$ have size at least K

Therefore the induction hypothesis applies and we must have

$$X \perp Y | Z \cup \{A\} \text{ and } X \perp A | Z \cup Y$$

Use intersection to get $X \perp Y, A | Z$ and Decomposition to get $X \perp Y | Z$

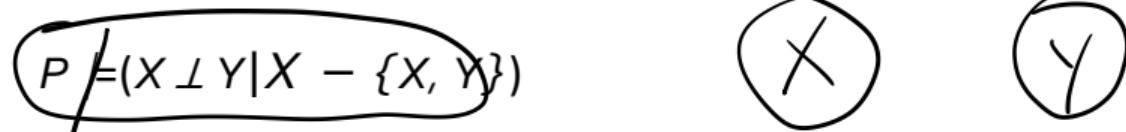
Constructing Graphs from Distributions

- A fully connected graph has no independence conditions and, hence, it can be an I-Map of any probability distribution.
- To encode the Independencies in a given distribution P using a graph structure, use minimal I-map.
- Two approaches for constructing a minimal I-map
 - 1 using the pairwise Markov Independencies.
 - 2 using the local Independencies.

Constructing Graphs from Distributions

Pairwise Markov independencies

- Let P be a positive distribution.
- Let H be defined by introducing an edge $\{X, Y\}$.
- If the edge $\{X, Y\}$ is not in H , then X and Y must be independent given all other nodes in the graph.
- To guarantee that H is an I-map, add direct edges between all pairs of nodes X and Y such that

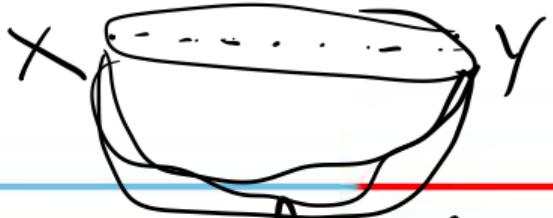
$$P \not\models (X \perp Y | X - \{X, Y\})$$


- Then Markov network H is the unique minimal I-map for H .
- To guarantee that H is an I-map, add direct edges between all pairs of nodes X and Y , such that they are dependent even on observing all the other variables in the network.

Minimal T-map

Theorem: Let P be a positive distribution and let H be obtained by introducing an edge $x - y$ whenever $P \not\models x \perp y | X - \{x, y\}$. Then the Markov network is the unique minimal T-map by construction.

Proof



Why is H an I-map for P ? By construction P satisfies $I_P(H)$. Since P is a positive distribution $I_P(H)$ is equivalent $\xrightarrow{\text{pairwise independency}}$ $I(H)$

Why is H a minimal I-map? If we eliminate some edge $x-y$ from H it would mean that $x \perp y | X - \{x, y\}$ which we know to be false for P .

Proof

Why is it a unique minimal 1-map

Let us say that there is another 1-map H' which is also minimal.

H' must contain all the edges of H ; otherwise it would imply some independencies that don't exist in P . If H' contains any additional edges then it is no longer minimal

~~fairwise local~~ ~~local~~ global

I-map \rightarrow global independencies

Constructing Graphs from Distributions

Local Independencies

- Let P be a positive distribution.
- For each variable X , define the neighbors of X to be a minimal set of nodes Y that render X independent of the rest of the nodes. i.e. Markov Blanket of X .
- A set U is a Markov blanket of X in a distribution P if $X \not\in U$ and if U is a minimal set of nodes such that

$$(X \perp X - \{X+U\}/U) \in I(P)$$

- Then define a graph H by introducing an edge $\{X, Y\}$ for all X and all $Y \in MB_P(X)$
- Then Markov network H is the unique minimal I-map for P .
- For each variable X , find the minimal set of nodes. Observing these makes the variable independent of all the variables. Then, add an edge between the variable and all the nodes in the set.





Table of Contents

1 Markov Network Independencies

2 Bayesian Network vs Markov Network

Bayesian Network and Markov Network

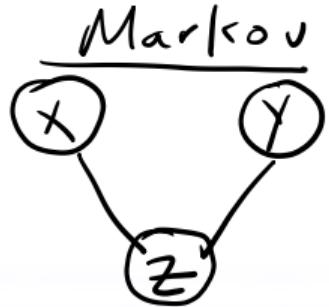
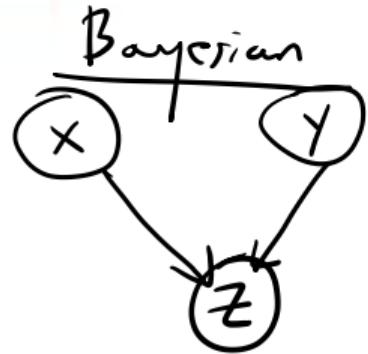
$$I(x_i | \text{pa}(x_i))$$

$$f(a, b, c)$$

Both

is it not a factor?

- Parametrize a probability distribution using a graphical model.
- Encode the Independencies among the random variable.



Convert Bayesian Network to Markov Network

$$B \quad C \quad P(A|B, C) = ?$$

Two perspectives

- 1 Parameterization perspective – represent the probability distribution of the Bayesian model using a fully parameterized Markov model.
- 2 Independencies perspective – represent the independence constraints encoded by the Bayesian model using the Markov model.

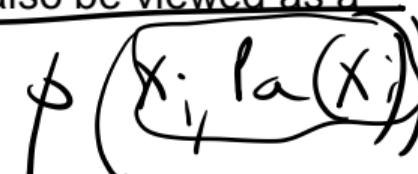


$$P(X_i | Pa(X_i)) \rightarrow$$



Convert Bayesian Network to Markov Network

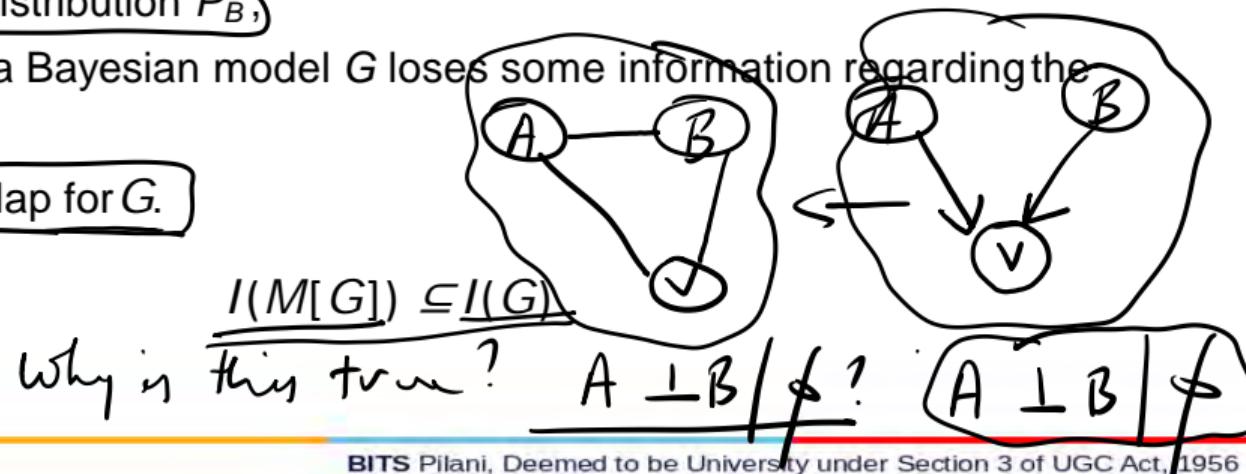
Parameterization perspective

- Probability distribution P_B , B is a parameterized Bayesian network over a graph G .
- The parameterization of the Bayesian network B , can also be viewed as a parameterization of a Gibbs distribution.
- Each CPD $P(X_i | Pa_{X_i})$ is a factor with scope $\{X_i, Pa_{X_i}\}$. 
- This set of factors defines a Gibbs distribution with the partition function equal to 1.

Convert Bayesian Network to Markov Network

Independencies perspective

- 1 Replace all the directed edges between the nodes with undirected edges.
 - 2 Add additional undirected edges between nodes that are parents of the node.
- This new structure is called moral graph of Bayesian network.
- $M[G]$ is an I-Map for distribution P_B ,
- Moral graph $M[G]$ of a Bayesian model G loses some information regarding the Independencies.
- $M[G]$ is a minimal I-Map for G .



$$I(M(G)) \subseteq I(G)$$

Consider a Z that does not separate X and Y in the Bayesian network G . Could it happen that Z will separate X and Y in $M(G)$ [the Markov network] and thus prevent $I(M(G)) \subseteq I(G)$?

$$X \rightarrow \dots \xrightarrow{a} \overset{v}{\leftarrow} \underset{b}{\leftarrow} \dots Z \text{ in } G$$

$$X - \underset{a}{\overbrace{\quad}} \overset{v}{\leftarrow} \underset{b}{\overbrace{\quad}} - Z \text{ in } M(G)$$

$$\mathcal{I}(M(G)) \subseteq \mathcal{I}(G)$$

The only danger lies in paths bearing converging nodes. Specifying a converging node v in G activates the path between X and Y in G but inactivates that path in $M(G)$. Fortunately, there is another path in $M(G)$ that comes to the rescue since v 's parents a and b are joined by an edge.

Why is $M(G)$ a minimal I-map for G ?

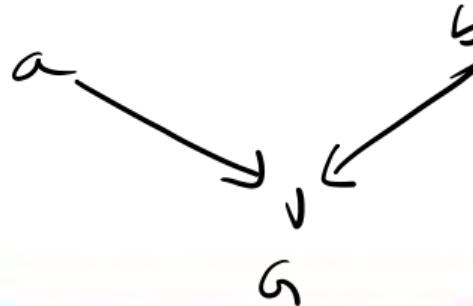
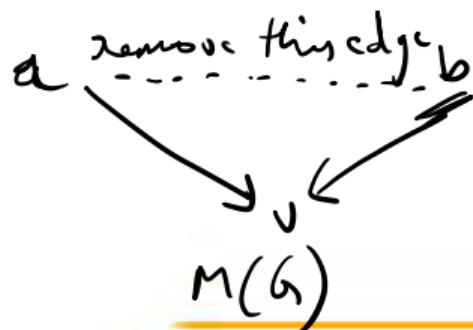
$M(G) = G + \text{edges between parents of a given node}$.
 Can we get rid of some edges in $M(G)$ such that it will
 continue to remain an I-map for G ?

If we get rid of an edge in $M(G)$ that has a directed
 equivalent in G , we will have an dependency
 introduced in $M(G)$ that is not there in G , i.e.
 $X \perp Y \mid \text{all nodes in } M(G)$

$$X \rightarrow Y$$

Why is $M(G)$ a minimal I-map for G ?

If we remove an edge that was added between the parents of a node in G , we introduce an independency $a \perp b | v$ in $M(G)$ that is not there in G , $\therefore I(M(G)) \subsetneq I(G)$



\therefore we cannot remove any edges in $M(G)$

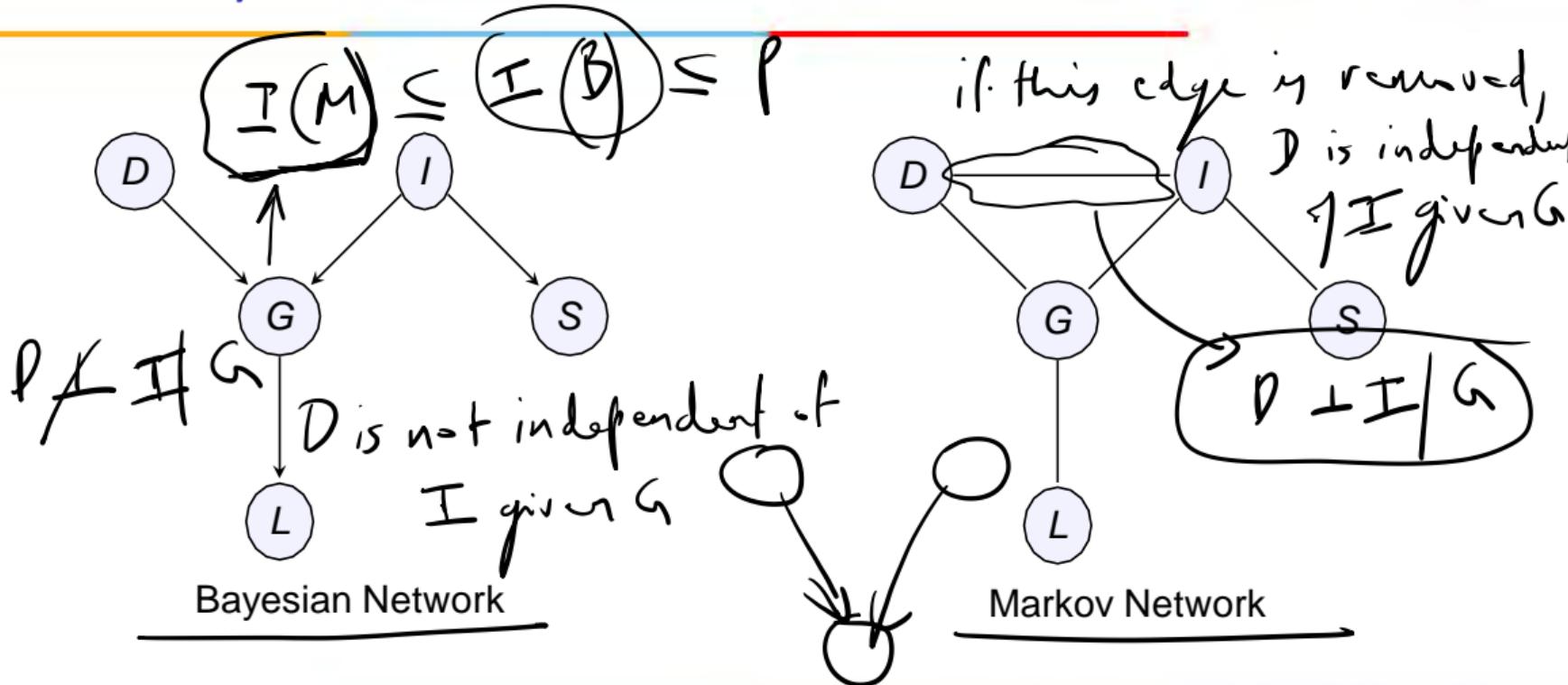
Moral Graph

Definition

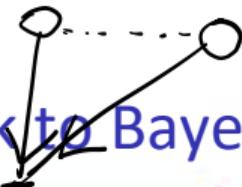
The moral graph $M[G]$ of a Bayesian network structure G over X is the undirected graph over X that contains an undirected edge between X and Y if:

- (a) there is a directed edge between them (in either direction) or
- (b) X and Y are both parents of the same node.

Convert Bayesian Network to Markov Network

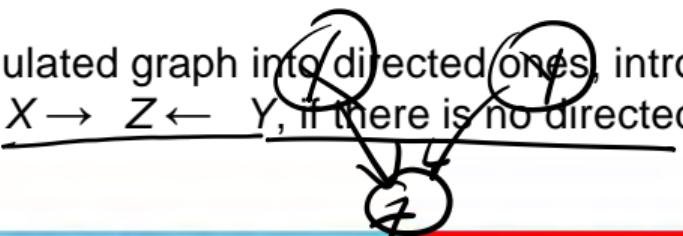


Convert Markov Network to Bayesian Network



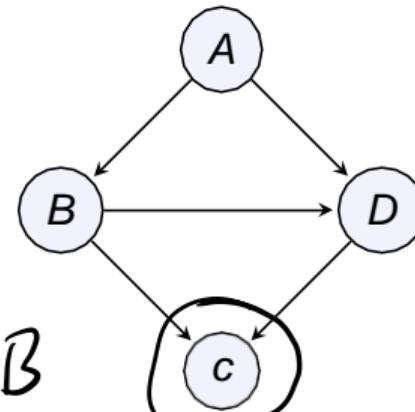
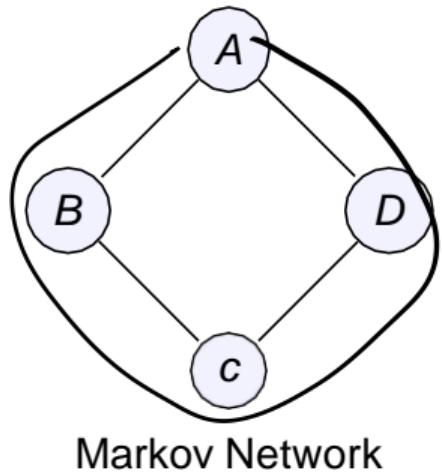
Independencies perspective

- 1 Replace all the undirected edges between the nodes with directed edges.
 - 2 Partition all loops into triangles. Add edges to the network to make it chordal.
- Any Bayesian network I-map for the given Markov network must add triangulating edges into the graph, so that the resulting graph is chordal. This process is called triangulation.
- A triangulated or chordal graph is a graph in which each of its cycles of four or more vertices has a chord.
 - By simply converting edges of a non-triangulated graph into directed ones, introduces immoralities. An immorality is a v-structure $X \rightarrow Z \leftarrow Y$, if there is no directed edge between X and Y.

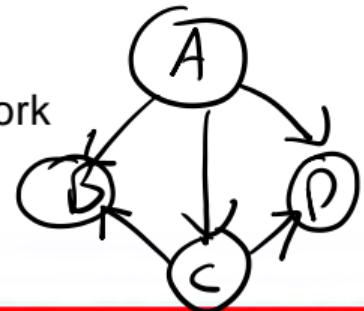


Convert Markov Network to Bayesian Network

A, B, D, C

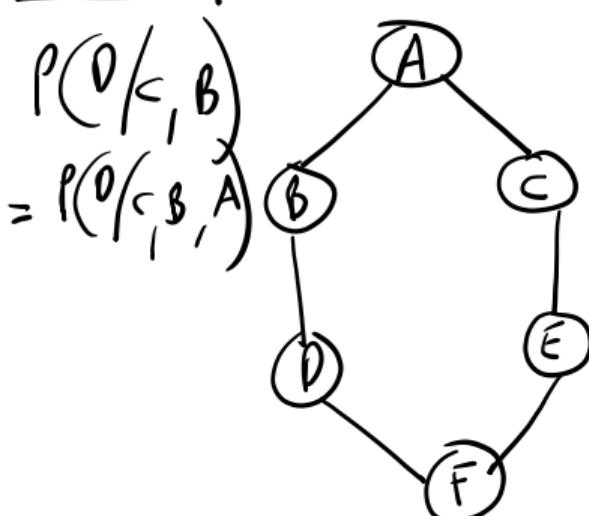


A, C, D, B



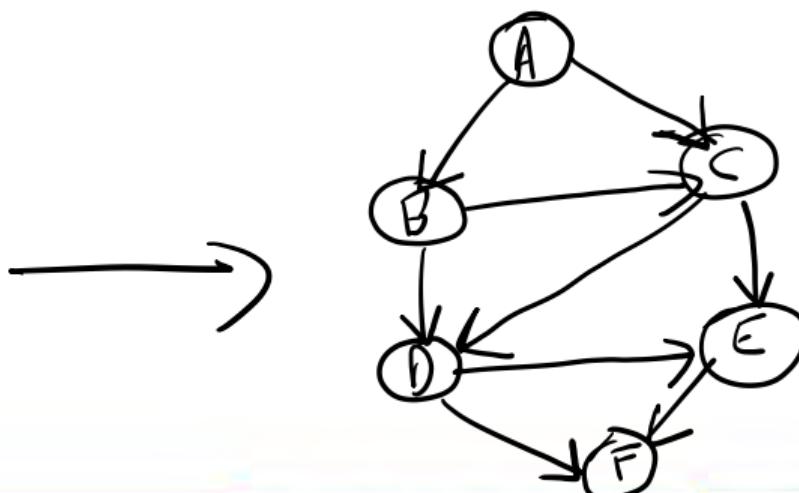
Markov Network to Bayesian network

Example 4.17



Enumerate nodes in the order

A, B, C, D, E, F



Reasoning

A is the first node in the ordering \rightarrow if has no parents

B can only have A as its parent

Consider C :

We can consider only A as C's parent

Is C independent of B given A? \rightarrow No! There is
a path C, E, F, D, B

Reasoning

So we add an edge $B \rightarrow C$

Consider node D:

- We must have B as a parent of D
- Is D independent of C given B? No
- So we add C as a parent to D.
- Now D is independent of A given B and C

Finally E's parents must be C and D.

$$P(x_1, x_2, x_3) = P(x_3|x_1, x_2)P(x_2|x_1)P(x_1)$$

lead

Why does this procedure give a minimal 1-map?

We are given distribution P . Pick a particular ordering of variables

- We construct a graph G such that each node x_i has as parents a subset U of $\{x_1, x_2, \dots, x_{i-1}\}$ such that $(x_i \perp \!\!\! \perp x_1, x_2, \dots, x_{i-1} | U)$
- This ensures that $P(x_i | x_1, x_2, \dots, x_{i-1}) = P(x_i | U)$
- Thus P factorizes over G

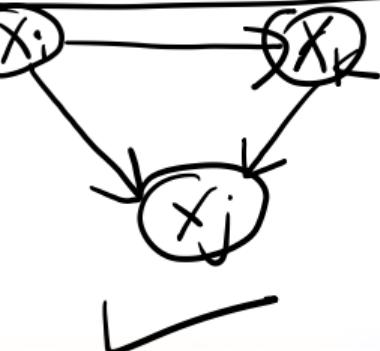
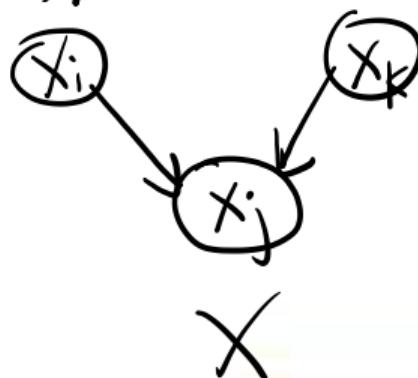
Why does this procedure work?

Theorem 3.2 then says that if P factorizes over G then G is an I-Maf for P .

G is minimal by construction, removing a single edge will cause it to have some dependency that does not belong to P .

Theorem

Let H be a Markov network structure and G be a Bayesian network minimal I-map for H . Then G can have no immoralities



Proof Sketch

By contradiction

Assume that an immorality exists and X_i and X_k can both be parents of X_j without an edge between X_i and X_k

Proof Sketch

Since there is an edge $X_i \rightarrow X_j$ in G we conclude that X_i cannot be d-separated from X_j by all of X_j 's other parents [Note we assume $i < k \vee j$]

H contains a path between X_i and X_j that is not cut-off by any other parents of X_j . Similarly there exists a path between X_j and X_k

Proof Sketch

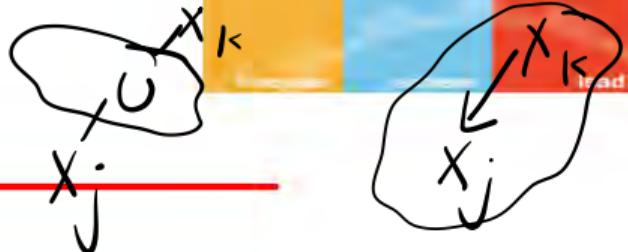
Let \cup be the parent set chosen for X_k .

Why was X_i ($i < k$) not chosen as a parent of X_k leading to the immorality?

Since there are one or more paths from X_i to X_k via X_j , all these paths are cut by \cup

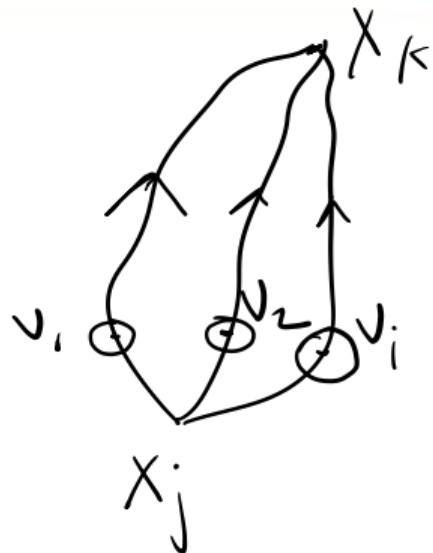
\cup can separate X_i from X_j or X_j from X_k

Proof Sketch



Let \cup separate X_j from X_k . Consider the choice \cup parent set for X_j . This is a minimal subset of X_1, X_2, \dots, X_{j-1} , which separates X_j from other nodes. Since \cup separates X_k from X_j , X_k cannot be the first node encountered on some uncut path from $X_j \Rightarrow X_k$ cannot be adjacent to $X_j \Rightarrow$ contradiction

Proof Sketch continued



circled points represent the first
non-candidate-parents of X_j that
 w_C encounters on all paths between
 X_j and X_k . Candidate parents of X_j
are $\{X_1, X_2, \dots, X_{j-1}\}$

Such points exist since we know
that $U \subseteq \{X_1, X_2, \dots, X_{j-1}\}$ separates X_j from X_k

Pr--f sketch continued

Now v_1, v_2, \dots, v_i are all part of the minimal subset of $\{x_1, x_2, \dots, x_{j-1}\}$ that constitutes the set of parents of $x_j \rightarrow \underline{\text{why}}$. This is because specifying all the candidate parents will not block a path from x_j to each v_i consisting only of non-candidate parents.

Pr-F sketch continued

Now X_k , $k < j$ cannot be a parent of X_j
since all paths between X_j and X_k are blocked
by the V_i 's which are definitely parents of

X_j .

Questions

- 1 Given a Markov Network, find the appropriate factorization of joint distribution.
- 2 Given a Markov Network, identify the active trails.
- 3 Given a Markov Network, identify the I-maps.
- 4 Given a Markov Network, identify the d-separations.
- 5 Given a toy application, generate Markov Network and the factors associated with it.
- 6 Given factors, generate a Markov Network.
- 7 Given a joint distribution in the factorized form, generate a Markov Network.
- 8 Given a Markov Network, identify the conditional Independencies.

References

- 1 Probabilistic Graphical Models: Principles and Techniques by Daphne Koller and Nir Friedman. MIT Press. 2009
- 2 Artificial Intelligence: A Modern Approach (3rd Edition) by Stuart Russell, Peter Norvig
- 3 Mastering Probabilistic Graphical Models using Python by Ankur Ankan, Abhinash Panda. Packt Publishing 2015.
- 4 Learning in Graphical Models by Michael I. Jordan. MIT Press. 1999

Thank You !!!



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 10 : BELIEF PROPAGATION

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in

TABLE OF CONTENTS

1 VARIABLE ELIMINATION ALGORITHM

2 CLUSTER GRAPH

3 CLIQUE TREE

4 MESSAGE PASSING

5 BELIEF UPDATE

VARIABLE ELIMINATION ALGORITHM



Variable elimination algorithm is the manipulation of the factors.

1 Create a factor ψ_i by multiplying existing factors.

2 Eliminate a variable in ψ_i to generate a new factor T_i . \rightarrow Summation

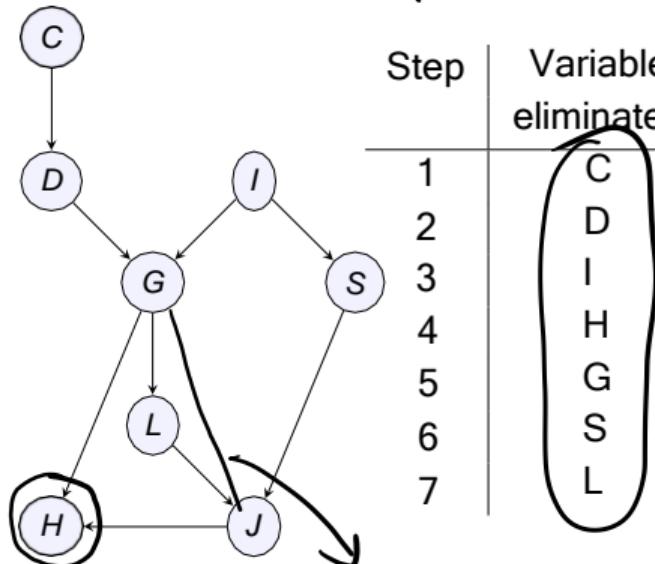
3 T_i is then used to create another factor.

All factors used in the creation of T_i are thrown out. Factors containing A contain B

VARIABLE ELIMINATION IN BN

$$P(P/C) \rightarrow \phi_D(C, D)$$

$$\phi_A(C, I, D) = P(C/I, D) T_1(D) = \sum_C \phi_C(C) \phi_D(C, D)$$



Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C)$	(C, D)	$T_1(D)$
2	D	$T_D(D) \cdot \phi_G(G, I, D)$	(G, I, D)	$T_2(G, I)$
3	I	$\phi_I(I) \cdot \phi_S(S, I) \cdot T_2(G, I)$	(G, S, I)	$T_3(G, S)$
4	H	$\phi_H(H, G, J)$	(H, G, J)	$T_4(G, J)$
5	G	$T_3(G, S) \cdot T_4(G, J) \cdot \phi_L(L, G)$	(G, L, S, J)	$T_5(J, L, S)$
6	S	$\phi_J(J, L, S) \cdot T_5(L, S)$	(J, L, S)	$T_6(J, L)$
7	L	$T_6(J, L)$	(J, L)	$T_7(J)$

there because of marginalization

TABLE OF CONTENTS

1 VARIABLE ELIMINATION ALGORITHM

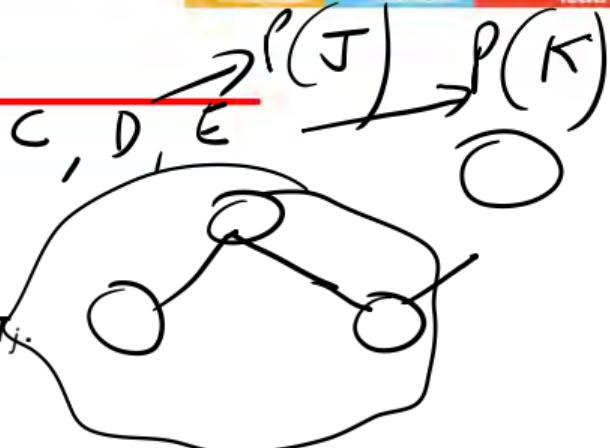
2 CLUSTER GRAPH

3 CLIQUE TREE

4 MESSAGE PASSING

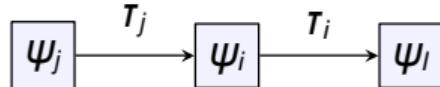
5 BELIEF UPDATE

VARIABLE ELIMINATION ALGORITHM



Different Perspective ~~super factor~~

- 1 Factor ψ_i is a data structure, which takes messages T_j .
- 2 T_j is generated by the other factor ψ_j .
- 3 Factor ψ_i generates message T_i .
- 4 Message T_i is used by another factor ψ_l .



CLUSTER GRAPH

DEFINITION

A cluster graph U for a set of factors over X

- an undirected graph
- nodes i are clusters $C_i \subseteq X$
- edges between C_i and C_j associated with sepset $S_{i,j} \subseteq C_i \cap C_j$.

Family Preserving Property

- Each factor $\varphi \in \Phi$, the distribution, must be associated with a cluster C_i such that

$$\text{Scope}[\varphi] \subseteq C_i$$

GENERATE CLUSTER GRAPH

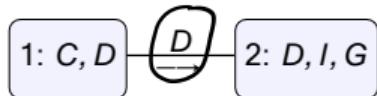
$$\psi(x_1, x_2, x_3) \rightarrow C_1, C_2, C_3 \in \text{Scope}(\psi)$$

- An execution of variable elimination defines a cluster graph.
- Mark a cluster for each factor ψ_i used in the computation, which is associated with the set of variables $C_i = \text{Scope}[\psi_i]$.
- Draw an edge between two clusters C_i and C_j if the message T_i is produced by eliminating a variable in ψ_i is used in the computation of T_j .

VARIABLE ELIMINATION AND CLUSTER GRAPH

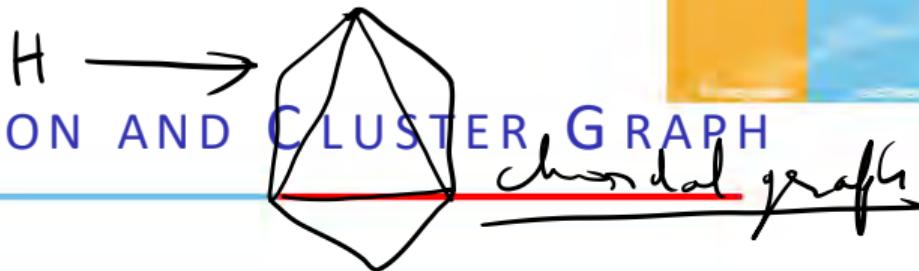
Step	VE	New factor	Message
1	C	$\psi_1(D)$	$\tau_1(D)$
2	D	$\psi_2(G, I)$	$\tau_2(G, I)$
3	I	$\psi_3(G, S)$	$\tau_3(G, S)$
4	H	$\psi_4(G, J)$	$\tau_4(G, J)$
5	G	$\psi_5(J, L, S)$	$\tau_5(J, L, S)$
6	S	$\psi_6(J, L)$	$\tau_6(J, L)$
7	L	$\psi_7(J)$	$\tau_7(J)$

- $C_1 = \psi_1 = \varphi_C(C) \cdot \varphi_D(C, D)$
- $C_2 = \psi_2 = \psi_1 \cdot \varphi_G(G, I, D)$
- C_1 generates τ_1 .
- τ_1 is used for computing ψ_2 .
- Hence an edge between C_1 and C_2 .

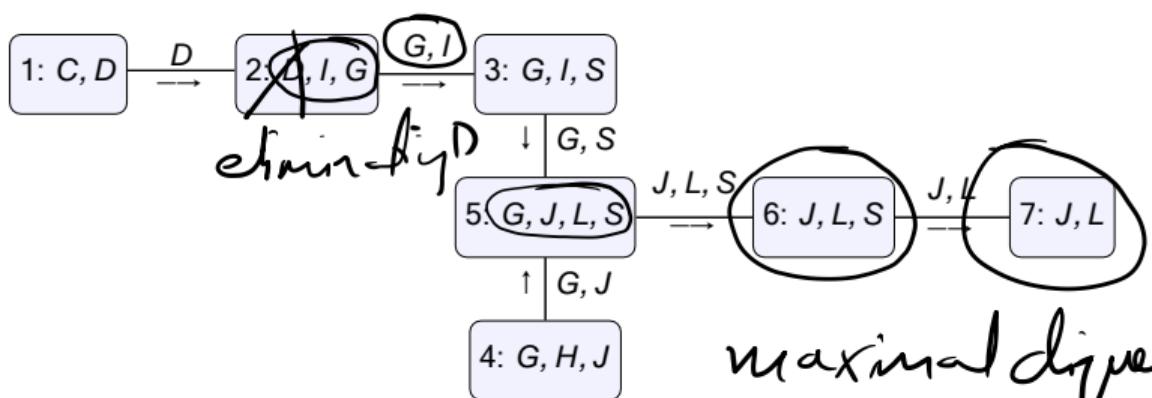


$\tau_1(\textcircled{D})$

VARIABLE ELIMINATION AND CLUSTER GRAPH



Step	VE	New factor	Message
1	C	$\psi_1(D)$	$\tau_1(D)$
2	D	$\psi_2(G, I)$	$\tau_2(G, I)$
3	I	$\psi_3(G, S)$	$\tau_3(G, S)$
4	H	$\psi_4(G, J)$	$\tau_4(G, J)$
5	G	$\psi_5(J, L, S)$	$\tau_5(J, L, S)$
6	S	$\psi_6(J, L)$	$\tau_6(J, L)$
7	L	$\psi_7(J)$	$\tau_7(J)$



Each of the factors in the initial set of factors Φ is also associated with a cluster C_i .

$$\ell(G, J, I, S)$$

TABLE OF CONTENTS

1 VARIABLE ELIMINATION ALGORITHM

2 CLUSTER GRAPH

3 CLIQUE TREE

4 MESSAGE PASSING

5 BELIEF UPDATE

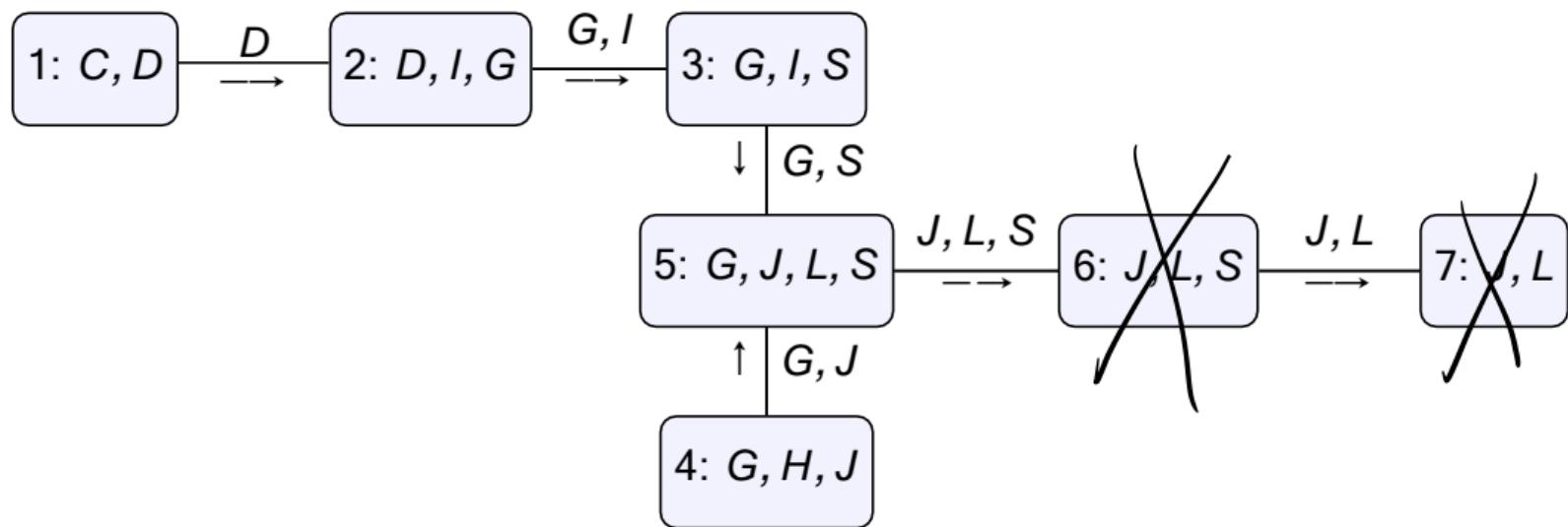
CLIQUE TREE

DEFINITION

A clique tree T for a set of factors over X

- an undirected tree
 - nodes i are clusters $C_i \subseteq X$
 - edges between C_i and C_j associated with sepset $S_{i,j} \subseteq C_i \cap C_j$.
-
- also called Junction Tree or Join Tree.
 - The clusters are called **cliques**.
 - Used to draw inferences.

CLIQUE TREE EXAMPLE



Cliques C_6 and C_7 are nonmaximal or redundant. Hence can be removed.

PROPERTIES OF CLIQUE TREE



1. Family Preserving Property

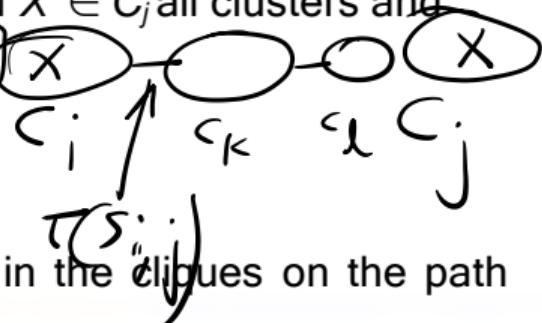
- Each factor $\varphi \in \Phi$ must be associated with a cluster C_i such that

$$\text{Scope}[\varphi] \subseteq C_i \quad S_{i,j} = C_j - \{ \text{climated variables} \}$$

2. Running Intersection Property

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

$$S_{i,j} = C_i \cap C_j$$



Example: G is present in C_2 and in C_4 , so it is also present in the cliques on the path between them: C_3 and C_5 .

CLIQUE TREE THEOREM 1

THEOREM

Let T be a cluster tree induced by a variable elimination algorithm over a set of factors Φ . Then T satisfies the running intersection property.

Proof

Let c and c' be two clusters that contain X
Let c_x be the cluster at which X is finally
eliminated

We shall show

(1) all clusters along the path from c to c_x

contain X

(2) all clusters along the path from c' to c_x contain X

Proof

Since we are dealing with a cluster tree, the path between C and C_X contains C' or C_X falls on the path from C to C' 's. We either have

$$(1) C - C' - C_X \quad \text{or} \quad (2) C - C_X - C'$$

In (1), since we show $G - C_X$ has X in all nodes in between, there must be X in all nodes between C and C' .

Proof

In (2) $c - c_x$ has X in all intermediate nodes
and so does $c' - c_x$. Therefore $c - c'$ has X
in all intermediate nodes.

We must prove that $c - c_x$ has X in
all intermediate nodes

Proof

- The computation at C_x happens after C
- When X is eliminated at C_x , all factors involving X are multiplied into a big 'factor' and X is summed out
 - N - duster node layout C_x will contain X
 - X is not eliminated in C , so the message computed at C must contain X and goes upstream

Proof

Since X is not eliminated on the path from C to C_X it is present on all nodes between C and C_X

CLIQUE TREE THEOREM 2

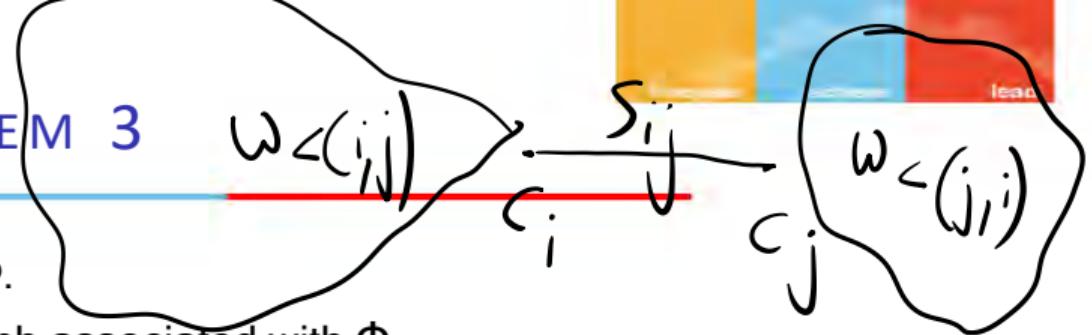
THEOREM

Let T be a cluster tree induced by a variable elimination algorithm over a set of factors Φ . Let C_i and C_j be two neighbouring clusters, such that C_i passes the message T_i to C_j . Then the scope of the message T_i is precisely $C_i \cap C_j$.

$$\text{Scope}[T_i] = S_{i,j} = C_i \cap C_j$$

CLIQUE TREE THEOREM 3

- Let T be a cluster tree over Φ .
- Let H_Φ be the undirected graph associated with Φ .
- $W_{<(i,j)}$ = all variables that appear only on C_i side of T .
- $W_{>(j,i)}$ = all variables that appear only on C_j side of T .
- $S_{(i,j)}$ = all variables that appear on both sides.



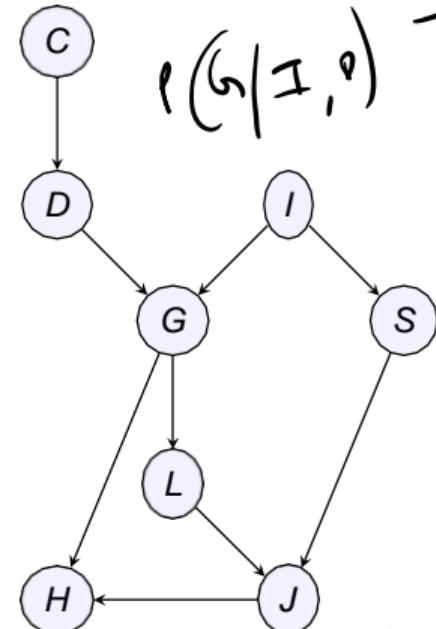
THEOREM

T satisfies the running intersection property, if and only if,
for every sepset $S_{i,j}$, the nodes $W_{<(i,j)}$ and $W_{>(j,i)}$ are separated in H_Φ given $S_{i,j}$.

$$P_\Phi \models (W_{<(i,j)} \perp W_{>(j,i)} | S_{(i,j)})$$

CLIQUE TREE INDEPENDENCE

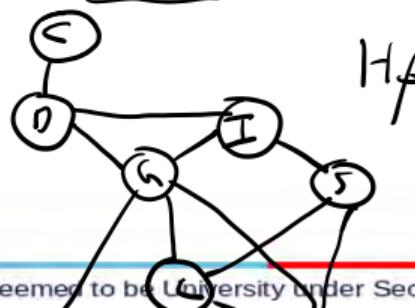
$$\ell(\mathcal{G} | \mathcal{I}, \emptyset) \rightarrow \phi(\mathcal{G}, \mathcal{I}, \emptyset)$$



Bayesian network



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

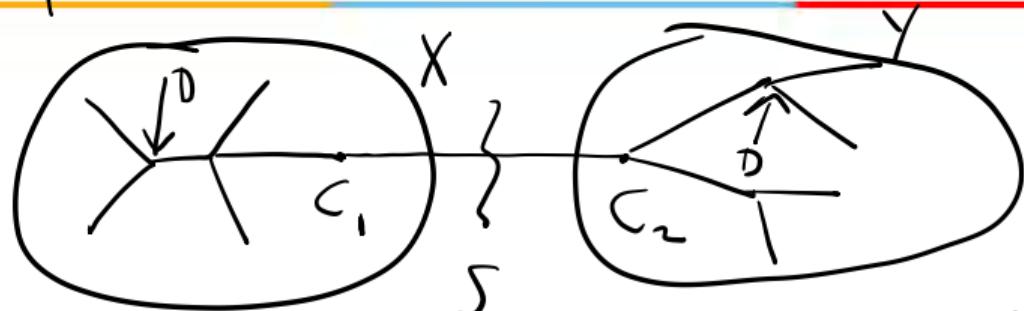


Proof

(1) Using intersection property in clique tree
⇒ original Markov networks can be separated
into conditionally independent pieces

Let S be a clique separator in clique tree
such that variables X are on one side of the
separator and variables Y are on the other side
Show that $\text{sep}_I(X; Y | S)$

Prof



Lemma: All nodes that are both in X and Y
are in S

Proof of lemma: In digraph T_1 , let S separate
digraphs C_1 and C_2 . Consider a node D that is in

Proof

both X and Y.

D is in some clique C_α in the left hand side of the separator and in some clique C_β on the right hand side of the separator

There exists a path from C_α to C_β in T through C_1 and C_2

Proof

Since the running intersection property holds true
D is in both C_1 and C_2 and also in $S = C_1 \cap C_2$

Now, show that $\text{sep}_I(X; Y | S)$

We will prove this result by contradiction

Suppose S does not separate X and Y

Proof

There exists a node A in X a node B in Y such that a path π in H_f between A and B does not pass through S.

This implies that each node in π is either in X or in Y but not in both X and Y since by the lemma above such a node would be in S.

Proof

The path π must pass through some node D in X + some node E in Y which means DE is an edge in H_f . So DE forms a clique in H_f which must be part of some maximal clique in T . But we know that all cliques in T are subsets of $X \cup Y$, so there cannot exist a clique containing D, E .

Pr-of

reverse Direction

separability in $H_\phi \Rightarrow$ running intersection property in T

Also by contradiction

assume separability in H_ϕ and that running intersection property in T does not hold and arrive at a contradiction

Proof sepset

Assume S is a separating set in H^f . T is a cluster tree corresponding to H^f where the nonempty intersection property does not hold.



Let $A \in C_1$ and $\overset{S}{\in} A \in C_k$ but $A \notin C_2$

Pr-f

in H_f

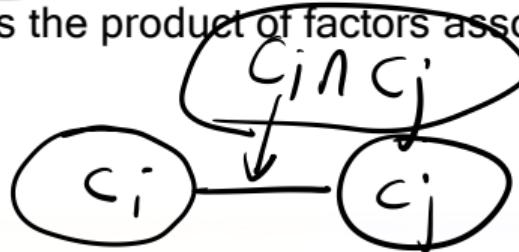
There exists a path between a vertex in C_1 and a vertex in C_k through A where this path does not go through S

$\therefore \underline{S \text{ is not a separator which is}}$
a contradiction

CONSTRUCT A CLIQUE TREE

→ without a variable elimination ordering
 - Section 10.4.2 in Daphne Koller's book

- 1 Triangulate the graph G over factor Φ to create a chordal graph H^* .
- 2 Find the maximal cliques in H^* and assign them as nodes to an undirected graph.
- 3 Assign weights to the edges between two nodes of the undirected graph as the numbers of elements in the sepset of the two nodes.
- 4 Construct the clique tree using the maximum spanning tree algorithm.
- 5 Compute the cluster potential for each cluster as the product of factors associated with the nodes present in the cluster.



CONSTRUCT A CLIQUE TREE – EXAMPLE

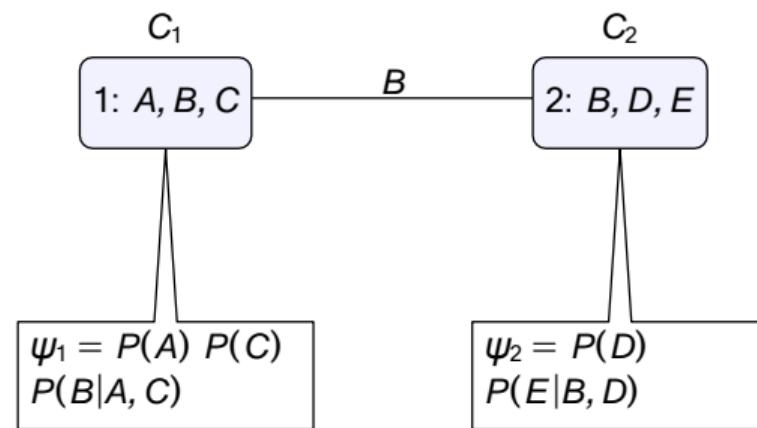
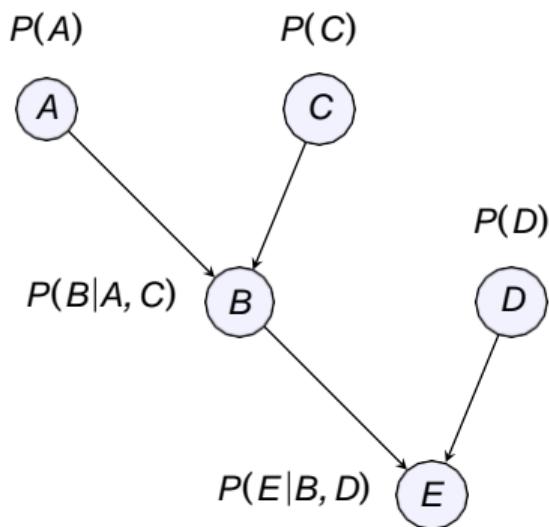


TABLE OF CONTENTS

1 VARIABLE ELIMINATION ALGORITHM

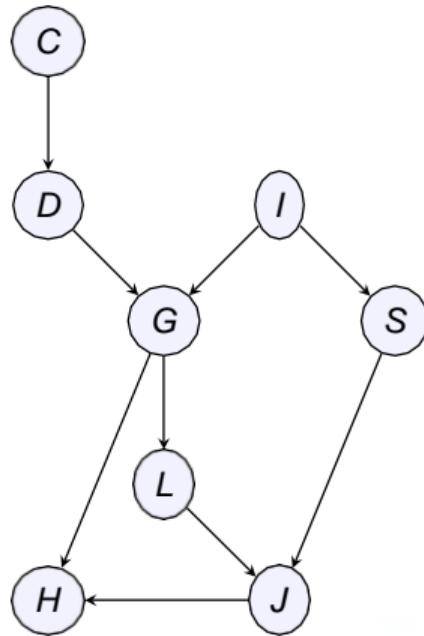
2 CLUSTER GRAPH

3 CLIQUE TREE

4 MESSAGE PASSING

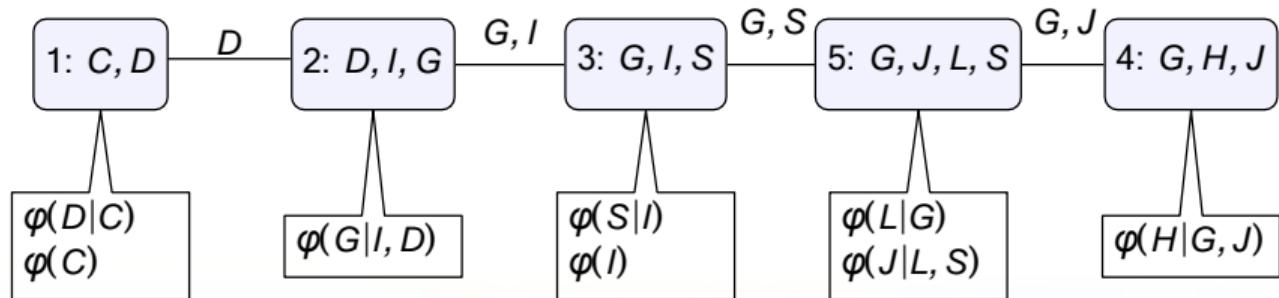
5 BELIEF UPDATE

CLIQUE TREE FOR A BAYESIAN NETWORK



$$\Phi = \varphi(C) \cdot \varphi(D|C) \cdot \varphi(I) \cdot \varphi(G|D, I) \cdot \\ \varphi(S|I) \cdot \varphi(L|G) \cdot \varphi(J|L, S) \cdot \varphi(H|G, J)$$

Construct the clique tree T over Φ .



CLIQUE TREE MESSAGE PASSING

- Compute the **initial potentials** associated with each clique.

$$\psi_1(C, D) = \varphi_C(C) \cdot \varphi_D(C, D)$$

$$\psi_2(G, I, D) = \varphi_G(G, I, D)$$

$$\psi_3(S, I) = \varphi_I(I) \cdot \varphi_S(S, I)$$

$$\psi_4(H, G, J) = \varphi_H(H, G, J)$$

$$\psi_5(J, L, G, S) = \varphi_L(L, G) \cdot \varphi_J(J, L, S)$$

CLIQUE TREE MESSAGE PASSING

Compute the probability $P(J)$.

- Select root clique as a clique that contains J . $\text{root} = C_5$
- In C_1 :
 -) Eliminate C by performing $\sum_C \psi_1(C, D)$.
 -) The resulting factor $\delta_{1 \rightarrow 2}(D)$ is send as a message to C_2 .
- In C_2 :
 -) Define $\beta_2(G, I, D) = \delta_{1 \rightarrow 2}(D) \cdot \psi_2(G, I, D)$.
 -) Eliminate D by performing $\sum_D \beta_2(G, I, D)$.
 -) The resulting factor $\delta_{2 \rightarrow 3}(G, I)$ is send as a message to C_3 .
- In C_3 :
 -) Define $\beta_3(G, I, S) = \delta_{2 \rightarrow 3}(G, I) \cdot \psi_3(G, I, S)$.
 -) Eliminate I by performing $\sum_I \beta_3(G, I, S)$.
 -) The resulting factor $\delta_{3 \rightarrow 5}(G, S)$ is send as a message to C_5 .

CLIQUE TREE MESSAGE PASSING

- In C_4 :

-) Eliminate H by performing $\sum_H \psi_4(H, G, J)$.
 -) The resulting factor $\delta_{4 \rightarrow 5}(G, J)$ is send as a message to C_5 .

- In C_5 :

-) Define $\beta_5(G, J, S, L) = \delta_{3 \rightarrow 5}(G, S) \cdot \delta_{4 \rightarrow 5}(G, J) \cdot \psi_5(G, J, S, L)$.
 -) Compute $P(J)$ by summing out G, L, S .

$$\sum \sum P(G_i, I, S) = P(G)$$

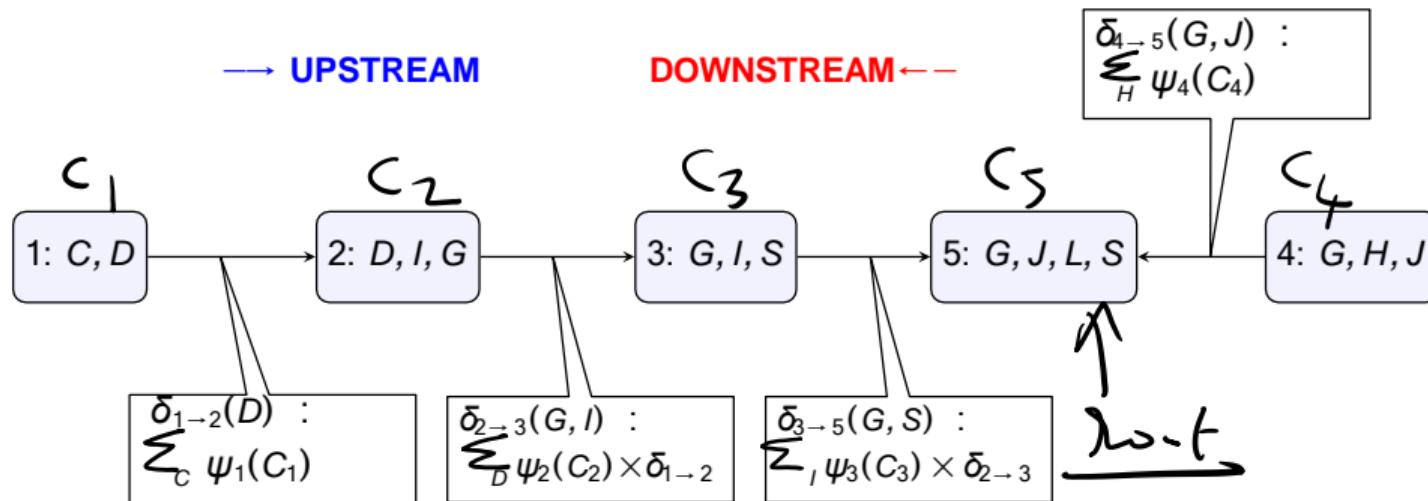
CLIQUE TREE MESSAGE PASSING

$$P(G, J, C, S)$$

Compute $P(J)$.

$\text{root} = C_5$

$$S_5 \rightarrow 3$$

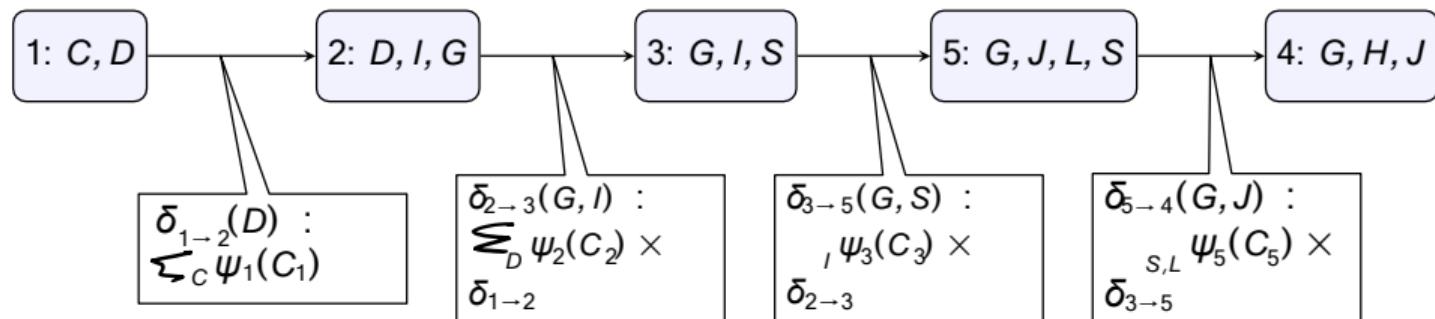


$$\beta_5(G, J, S, L) = \delta_{3 \rightarrow 5}(G, S) \cdot \delta_{4 \rightarrow 5}(G, J) \cdot \psi_5(G, J, S, L)$$

CLIQUE TREE MESSAGE PASSING

Compute $P(J)$.

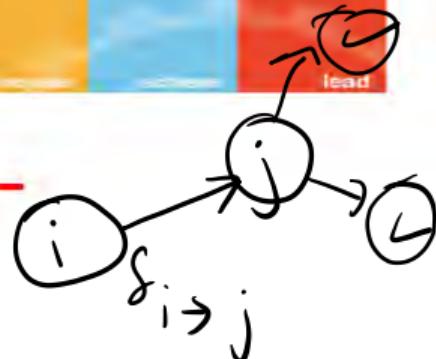
root = C_4



$$\beta_4(H, G, J) = \delta_{5 \rightarrow 4}(G, J) \cdot \psi_4(H, G, J)$$

CLIQUE TREE MESSAGE PASSING

- 1 Assign each factor φ to some clique $a(\varphi)$.
- 2 Compute the **initial potentials** associated with each clique.



$$\underline{\psi_j(C_j)} = \prod_{\varphi: a(\varphi)=j} \varphi$$

- 3 Except for the root, the message from C_i to another clique C_j is computed using the **sum-product message passing** computation.

$$\delta_{i \rightarrow j} = \sum_{C_i - S_{i,j}} \psi_i \cdot \prod_{k \in (Nb_i - \{j\})} \delta_{k \rightarrow i}$$

- 4 At the root, after receiving all messages, compute the **belief factor**.

$$\beta_r(C_r) = \psi_r \cdot \prod_{k \in Nb_{Cr}} \delta_{k \rightarrow r}$$

CLIQUE TREE AND VARIABLE ELIMINATION

Goal: Compute the marginal probability of any set of query nodes Y which is fully contained in some clique.

- 1 Select one such clique C_r to be the root.
- 2 Perform the Clique Tree message passing toward that root.
- 3 Extract $\tilde{P}_\phi(Y)$ from the final potential at C_r by summing out the other variables $C_r - Y$.



$$\tilde{P}_\phi(Y) = \sum_{C_r - Y} \prod_{\phi} \varphi$$

$$\sum_{C_r - Y} \beta_r(\zeta_r)$$

(Z)

CLIQUE TREE MESSAGE PASSING ALGORITHM

Algorithm 10.1 Upward pass of variable elimination in clique tree

Procedure CTree-SP-Upward (

Φ , // Set of factors

T , // Clique tree over Φ

α , // Initial assignment of factors to cliques

C_r // Some selected root clique

)

1 Initialize-Cliques

2 while C_r is not ready

3 Let C_i be a ready clique

4 $\delta_{i \rightarrow p_r(i)}(S_{i, p_r(i)}) \leftarrow \text{SP-Message}(i, p_r(i))$

5 $\beta_r \leftarrow \psi_r \cdot \prod_{k \in \text{Nb}_{C_r}} \delta_{k \rightarrow r}$

6 return β_r

Procedure Initialize-Cliques (

)

for each clique C_i

$\psi_i(C_i) \leftarrow \prod_{\phi_j : \alpha(\phi_j)=i} \phi_j$

Procedure SP-Message (

i , // sending clique

j , // receiving clique

)

1 $\psi(C_i) \leftarrow \psi_i \prod_{k \in (\text{Nb}_i - \{j\})} \delta_{k \rightarrow i}$

2 $\tau(S_{i,j}) \leftarrow \sum_{C_i = S_{i,j}} \psi(C_i)$

3 return $\tau(S_{i,j})$

Correctness of Algorithm 10.1

Proposition: Assume that X is eliminated when a message is passed from C_i to C_j . Then X does not appear anywhere on the C_j -side of the edge $(i-j)$.

Proof: Assume the contrary

visit by
raising intersection

Correctness of Algorithm 10.1

Theorem: Let $s_{i \rightarrow j}$ be the message passed from c_i to c_j . Then

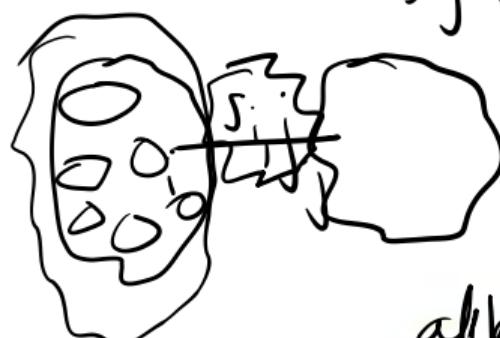
$$s_{i \rightarrow j}(s_{i,j}) = \sum_{V \subset (i \rightarrow j)} \prod_{\phi \in F \subset (i \rightarrow j)} \phi$$

set of factors
 on the i-side

set of variables

appearing on the i-side of $i \rightarrow j$ but not in the

subset -





Correctness of Algorithm 10.1

Proof is by induction. Given formula is trivially true at a leaf

Inductive Step: Let i_1, i_2, \dots, i_m be neighbours of i other than j

$$\sum_{V<(i \rightarrow j)} \prod_{\phi \in F_{<(i \rightarrow j)}} = \sum_{Y_i} \sum_{V<(i_1 \rightarrow i)} \sum_{V<(i_2 \rightarrow i)} \dots \sum_{V<(i_m \rightarrow i)}$$

$$\prod_{\phi \in F_{<(i_1 \rightarrow i)}} \phi \quad \prod_{\phi \in F_{<(i_2 \rightarrow i)}} \phi \dots \prod_{\phi \in F_{<(i_m \rightarrow i)}} \phi$$

Correctness of Algorithm 10.1

This sum-product can be rearranged as

$$\sum_{y_i} \left(\prod_{\phi \in F_i} \phi \right) \sum_{V < (i \rightarrow)} \left(\prod_{\phi \in F_{<(i \rightarrow i)}} \phi \right) \dots \sum_{V < (m \rightarrow i)} \left(\prod_{\phi \in F_{<(m \rightarrow i)}} \phi \right)$$

$$= \sum_{y_i} \psi \cdot s_{i \rightarrow i} s_{i \rightarrow i} \dots s_{i \rightarrow i}$$

which is what we understand by $s_{i \rightarrow j}$

Correctness of Algorithm 10.1

Let us use the formula to compute β_r at the root

$$\beta_r \leftarrow \rho_r \cdot \prod_{k \in N_G(r)} \delta_{k \rightarrow r} \quad [\text{from Algo 10.1}]$$

Let i_1, i_2, \dots, i_m be the neighbours to root r

$$\delta_{i \rightarrow r} = \sum_{v \in (i \rightarrow r)} \prod_{\phi \in F(v \rightarrow r)} \phi$$

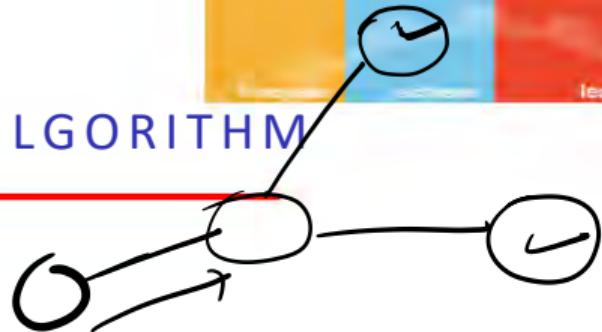
Correctness of Algorithm 10.1

$$\beta_r = \psi_r \cdot s_{i_1 \rightarrow r} s_{i_2 \rightarrow r} \dots s_{i_m \rightarrow r}$$

$$\beta_r = \psi_r \cdot \left(\sum_{\forall i_1 \rightarrow r} \prod_{\phi \in F_{\leq i_1 \rightarrow r}} \phi \right) \left(\sum_{\forall i_2 \rightarrow r} \prod_{\phi \in F < (i_2 \rightarrow r)} \phi \right)$$

$$\beta_r = \underbrace{\sum_{X - C_r} \psi_r \prod_{\phi} \phi}_{= C_r} = \underline{\Pr(C_r)}$$

CLIQUE TREE MESSAGE PASSING ALGORITHM



- A message sent between two cliques in the same direction is necessarily the same.
- For any given clique tree, each edge has two messages associated with it: one for each direction of the edge.
- For any given clique tree with c cliques, there are $c - 1$ edges and $2(c - 1)$ messages are computed.
tree - clique tree on c vertex
- The messages are computed by following a simple asynchronous algorithm.
- Uses dynamic programming.

$k \times (\text{cost})$

k variable

SUM-PRODUCT BELIEF PROPAGATION ALGORITHM

Algorithm 10.2 Calibration using sum-product message passing in a clique tree

```

Procedure CTree-SP-Calibrate (
     $\Phi$ , // Set of factors
     $T$  // Clique tree over  $\Phi$ 
)
1 Initialize-Cliques
2 while exist  $i, j$  such that  $i$  is ready to transmit to  $j$ 
3    $\delta_{i \rightarrow j}(S_{i,j}) \leftarrow \text{SP-Message}(i, j)$ 
4   for each clique  $i$ 
5      $\beta_i \leftarrow \psi_i \cdot \prod_{k \in \text{Nb}_i} \delta_{k \rightarrow i}$ 
6   return  $\{\beta_i\}$ 

```

$$\delta_{i \rightarrow j} = \sum_{C \ni i} \psi_i \prod_{k \in C \setminus \{i\}} \delta_{k \rightarrow i}$$

when i has received messages from all its neighbours except j , i is ready to transmit to j

SUM-PRODUCT BELIEF PROPAGATION ALGORITHM

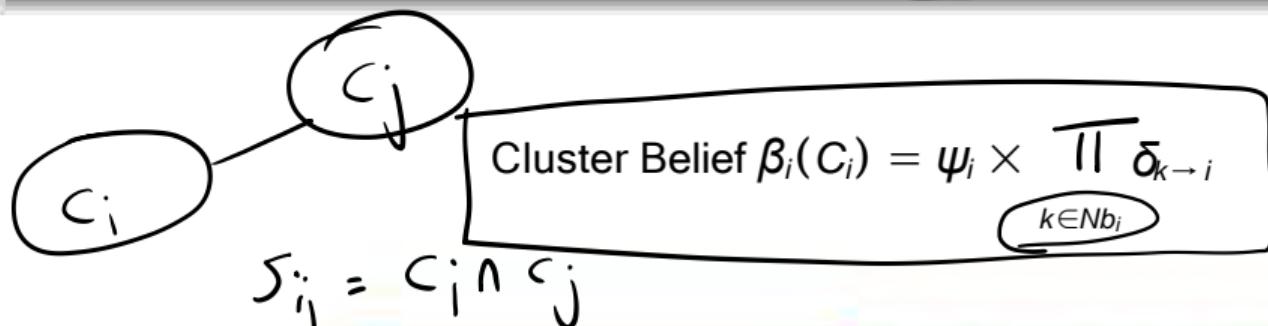
- Uses dynamic programming.
- The algorithm is defined asynchronously, with each clique sending a message as soon as it is ready.
- In the upward pass, first pick a root and send all messages toward the root.
- In the downward pass, the root sends all the downward pass messages to all of its children.
- The algorithm continues until the leaves of the tree are reached.

CLIQUE CALIBRATION

DEFINITION

Two adjacent cliques C_i and C_j are calibrated if every pair of adjacent clusters C_i and C_j agree on their sepset $S_{i,j}$.

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



CLIQUE CALIBRATION

DEFINITION

A clique tree T is calibrated if all pairs of adjacent cliques are calibrated. For a calibrated clique tree, clique beliefs are $\beta_i(C_i)$ and sepset beliefs are $\mu_{i,j}$

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

$$\beta_i(S_{i,j}) = \beta_j(S_{i,j})$$

TABLE OF CONTENTS

1 VARIABLE ELIMINATION ALGORITHM

2 CLUSTER GRAPH

3 CLIQUE TREE

4 MESSAGE PASSING

5 BELIEF UPDATE

MESSAGE PASSING WITH DIVISION

- For any edge between two clusters C_i and C_j , two messages are computed; $\delta_{j \rightarrow i}$ and $\delta_{i \rightarrow j}$.
- Let the first message be passed from C_i to C_j .
- Then the return message from C_j to C_i would be passed when C_j has received all messages from its neighbours.
- Once C_j has received all messages from its neighbours, compute its belief β_j

$$\beta_j(C_j) = \psi_j \times \prod_{k \in Nb_j} \delta_{k \rightarrow j}$$

- Alternatively, message from C_j to C_i can be computed as

$$\delta_{j \rightarrow i} = \sum_{C_i - S_{i,j}} \psi \cdot \prod_{k \in (Nb_j - i)} \delta_{k \rightarrow j}$$

MESSAGE PASSING WITH DIVISION

■ Rewrite

$$\mu_{ij}(s_{ij})$$

$$\begin{aligned} \beta_j(c_j) &= \sum_{s_i - s_{ij}} \prod_{k \in Nb_j} \delta_{k \rightarrow j} \\ &= \overline{\delta_{i \rightarrow j}} \sum_{c_i - s_{ij}} \psi_i \times \prod_{k \in Nb_j} \delta_{k \rightarrow j} \\ &= \overline{\delta_{i \rightarrow j}} \overline{\delta_{j \rightarrow i}} \end{aligned}$$

$\delta_{i \rightarrow j}$ is s_{ij}

■ Hence,

$$\delta_{j \rightarrow i} = \frac{\beta_i(c_i)}{\delta_{i \rightarrow j}}$$

L AURITZEN-SPIEGELHALTER ALGORITHM

Message Passing with Division

- 1 For each cluster C_i , initialize the cluster belief β_i as its cluster potential ψ_i and sepset potential $S_{i,j}$ between adjacent clusters C_i and C_j as 1.
- 2 In each iteration, the cluster belief β_i is updated by multiplying it with the message from its neighbours and the sepset potential $\{i - j\}$ is used to store the previous message passed along the edge $(i - j)$, irrespective of the direction of the message.
- 3 Whenever a new message is passed along an edge, it is divided by the old message to ensure that we don't count this message twice.
- 4 Marginalize the belief to get the message passed.

Diagram illustrating message passing with division:

$$\delta_{i \rightarrow j} = \frac{\sum_{C_i - S_{i,j}} \beta_i}{\mu_{i,j}}$$

The diagram shows the calculation of $\delta_{i \rightarrow j}$. It starts with a cluster belief β_i (circled) and a message $\sum \beta_j$ (circled) over a sepset $S_{i,j}$ (circled). An arrow points from this product to the message $\delta_{i \rightarrow j}$ (circled). This message is then divided by the old message $\mu_{i,j}$ (circled) to produce the "actual message passed". A handwritten note next to the diagram states: "actual message passed = belief of i Marginalized over J's set".

L AURITZEN-SPIEGELHALTER ALGORITHM



Message Passing with Division

- 5 Update the belief by multiplying it with the message from its neighbours.

$$\beta_j \leftarrow \beta_j \cdot \delta_{i \rightarrow j}$$

~~β_i *correct noisy*
incorrect message~~

- 6 Update the sepset belief

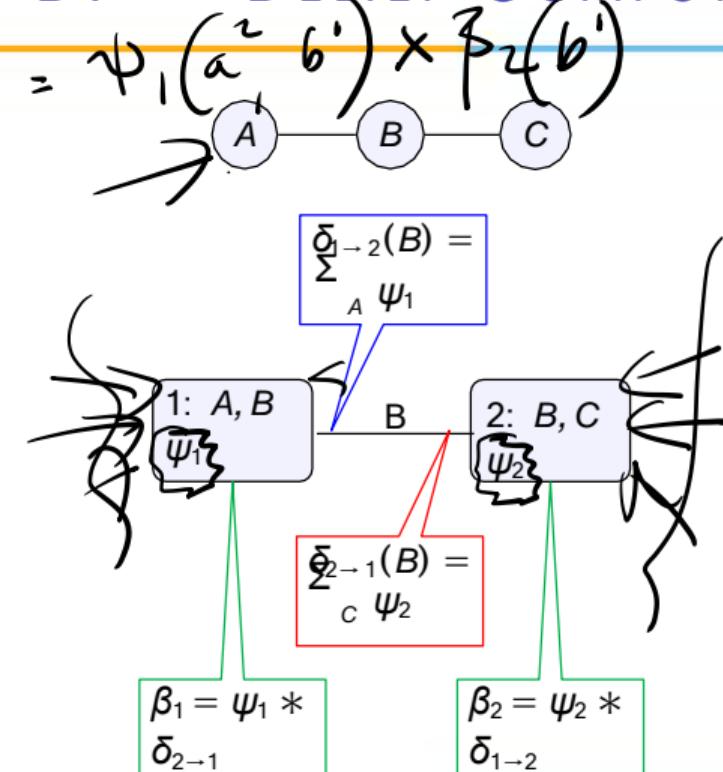
$$\mu_{i,j} = \sum_{C_i - S_{i,j}} \beta_i$$

- 7 Repeat Steps 2 and 3 until the tree is calibrated for each adjacent edge $(i - j)$

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

$\beta_1(a^1, b^1)$ $\psi_1(a, b)$ $\psi_1(a = a^1, b = b^1)$

BP - BELIEF COMPUTATION - EXAMPLE



ψ_1	a^1, b^1	3	
	a^2, b^1	-1	
	a^1, b^2	0	
	a^2, b^2	1	

ψ_2	b^1, c^1	4	
	b^1, c^2	-1	
	b^2, c^1	1	
	b^2, c^2	2	

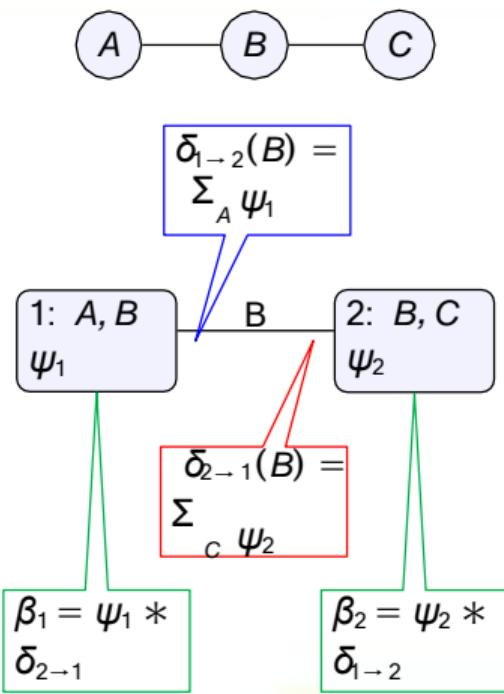
$\delta_{1 \rightarrow 2}$	b^1	2	
	b^2	1	

$\delta_{2 \rightarrow 1}$	b^1	3	
	b^2	3	

β_1	a^1, b^1	$\frac{3*3=9}{-1*3=-3}$	
	a^2, b^1		
	a^1, b^2	0*3=0	
	a^2, b^2	1*3=3	

β_2	b^1, c^1	$4*2=8$	
	b^1, c^2	$-1*2=-2$	
	b^2, c^1	$1*1=1$	
	b^2, c^2	$2*1=2$	

B P – CALIBRATION – EXAMPLE



ψ_1	$a^1 b^1$	3
	$a^2 b^1$	-1
	$a^1 b^2$	0
	$a^2 b^2$	1

ψ_2	$b^1 c^1$	4
	$b^1 c^2$	-1
	$b^2 c^1$	1
	$b^2 c^2$	2

$\delta_{1 \rightarrow 2}$	b^1	2
	b^2	1

$\delta_{2 \rightarrow 1}$	b^1	3
	b^2	3

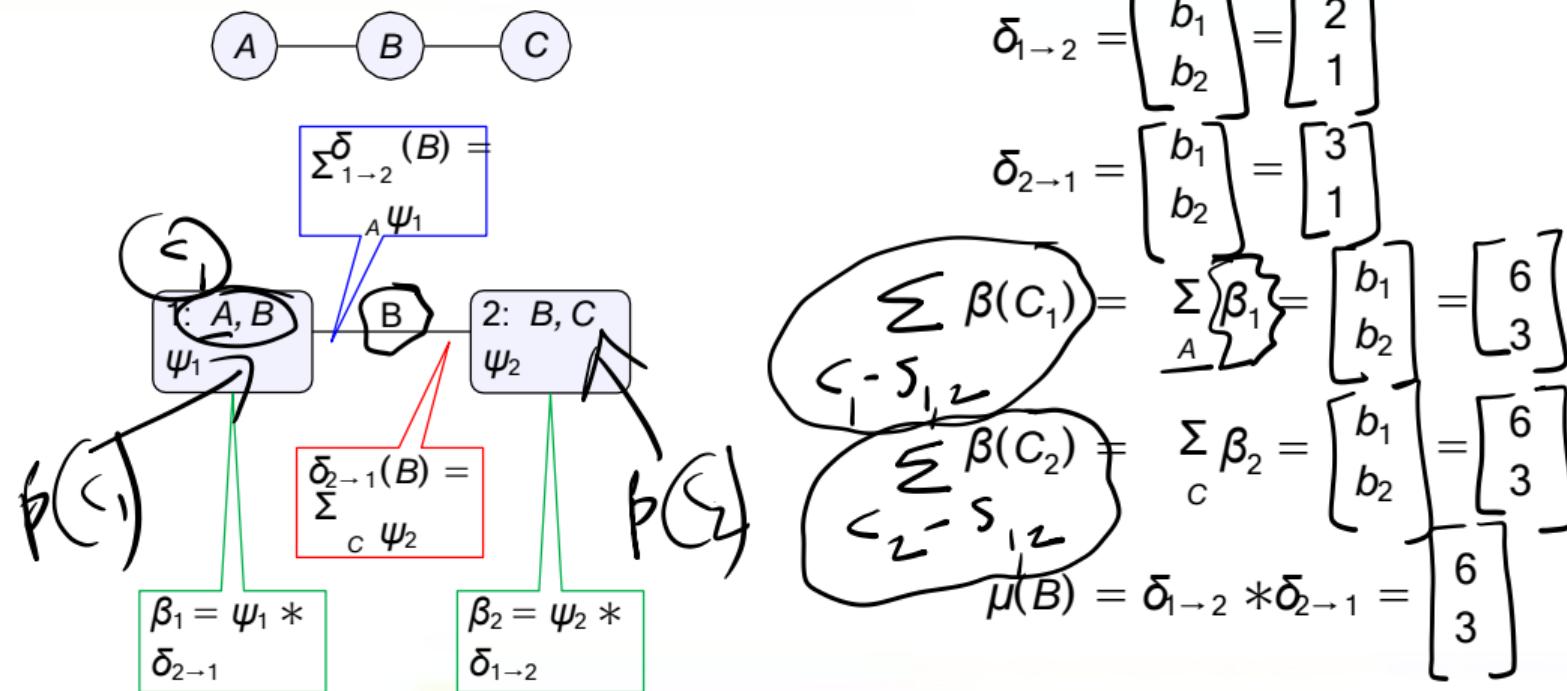
β_1	$a^1 b^1$	$3 * 3 = 9$
	$a^2 b^1$	$-1 * 3 = -3$
	$a^1 b^2$	$0 * 3 = 0$
	$a^2 b^2$	$1 * 3 = 3$

$\sum_A \beta_1$	b^1	⑥
	b^2	3

 $=$

$\sum_c \beta_2$	b^1	6
	b^2	3

B P – CALIBRATION – EXAMPLE



CALIBRATED CLIQUE TREE AS DISTRIBUTION

- At Convergence of clique tree calibration algorithm,

$$\beta_i = \psi_i \cdot \prod_{k \in Nb_i} \delta_{k \rightarrow i} \quad (1)$$

$$\mu_{i,j}(S_{i,j}) = \delta_{i \rightarrow j} \delta_{j \rightarrow i} \quad (2)$$

$$\tilde{P}_\phi(X) = \frac{\prod_{i \in V_T} \beta_i(C_i)}{\prod_{(i-j) \in E_T} \mu_{i,j}(S_{i,j})} \quad (3)$$

- Clique and sepsets beliefs provide a reparameterization of the unnormalized measure. This property is called the **clique tree invariant**.
- The distribution represented by a clique tree

$$Q_T = \frac{\prod_{i \in V} \beta_i(C_i)}{\prod_{(i-j) \in E_T} \mu_{i,j}(S_{i,j})} \quad (4)$$

Equivalence between sum-product and belief update

Theorem 10.5: Consider a set of sum-product initial potentials $\{\psi_i, i \in V_T\}$ and messages $\{\delta_{i \rightarrow j}, \delta_{j \rightarrow i} : i - j \in E_T\}$, and a set of belief-update beliefs $\{\beta_i : i \in V_T\}$ and messages $\{\mu_{i \rightarrow j} : i - j \in E_T\}$, for which equation (10.8) and equation (10.9) hold. For any pair of neighboring cliques C_i, C_j , let $\{\delta'_{i \rightarrow j}, \delta'_{j \rightarrow i} : i - j \in E_T\}$ be the set of sum-product messages following an application of SP-Message(i, j), and $\{\beta'_i : C_i \in T\}, \{\mu'_{i \rightarrow j} : (i - j) \in E_T\}$, be the set of belief-update beliefs following an application of BU-Message(i, j). Then equation (10.8) and equation (10.9) also hold for the new beliefs $\delta'_{i \rightarrow j}, \beta'_i, \mu'_{i \rightarrow j}$.

Equivalence between sum-product and belief update

For reference Equation 10.8 and Equation 10.9 are:

$$\beta_i = \psi_i \cdot \prod_{k \in N_b} \delta_{k \rightarrow i} \quad (10.8)$$

$$\mu_{i,j}(s_{i,j}) = \delta_{i \rightarrow j} \cdot \delta_{j \rightarrow i} \quad (10.9)$$

Equivalence Proof

Consider SPMessages_{i,j}

$$\beta_j = \psi_j \prod_{k \in N_{b_j} \setminus \{i\}} \delta_{k \rightarrow j} \mu_{ij}(\sigma_{ij}) = \delta_{j \rightarrow i} \delta_{i \rightarrow j}$$

$$\beta'_j = \psi_j \prod_{k \in N_{b_j} \setminus \{i\}} \delta'_{k \rightarrow j} \delta'_{i \rightarrow j}$$

$$\boxed{\beta'_j = \beta_j \frac{\delta'_{i \rightarrow j}}{\delta_{i \rightarrow j}}}$$

Equivalence Proof

Now consider $B \cup \text{Message}(i, j)$

$$\beta_j = \beta_j \frac{\sigma_{i \rightarrow j}}{\mu_{i,j}}$$

$$\begin{aligned}
 \sigma_{i \rightarrow j} &= \sum_{c_i - \{s_{i,j}\}} \beta_i = \sum_{c_i - \{s_{i,j}\}} \psi_i \prod_{k \in N_{b_i}} \delta_{k \rightarrow i} \\
 &= \sum_{c_i - \{s_{i,j}\}} \left(\psi_i \prod_{k \in N_{b_i} - \{j\}} \delta_{k \rightarrow i} \right) \delta_{j \rightarrow i} = \delta_{i \rightarrow j} \delta_{j \rightarrow i}
 \end{aligned}$$

Equivalence Proof

Thus in $B\cup \text{Message}(i, j)$ we have -

$$\beta_j' = \frac{\beta_j s_{i \rightarrow j} \cancel{s_{j \rightarrow i}}}{\cancel{s_{i \rightarrow j}} \cancel{s_{j \rightarrow i}}} = \frac{\beta_j s_{i \rightarrow j}'}{\cancel{s_{i \rightarrow j}}}$$

Same as β_j' in $S\cup \text{Message}(i, j)$

Equivalence proof

In $BUMay(i, j)$

$$u'_{i,j}(s_{ij}) = \delta'_{i \rightarrow j}$$

We already saw that $\delta'_{i \rightarrow j} = \delta'_{i \rightarrow j} \delta_{j \rightarrow i}$

$$\therefore u'_{i,j}(s_{ij}) = \delta'_{i \rightarrow j} \delta_{j \rightarrow i}$$

But this is the same in $SPMay(i, j)$ too!

QUESTIONS

- 1 Construct a cluster graph for a BN or MN.
- 2 Construct a clique tree for a BN or MN and verify the properties.
- 3 Complete the missing statements in the given code snippet which implements belief propagation algorithm.
- 4 Complete the missing statements in the given code snippet which implements Lauritzen-Spiegelhalter algorithm.



BITS Pilani

Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 8 : PROBLEMS



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.

Problem 1 Statement

We have 3 biased coins A, B, and C such that the probabilities of getting heads on a single toss of each of them are respectively 0.2, 0.8 and 0.8. One of the coins is selected uniformly at random and then flipped 3 times to get outcomes

Problem | Statement

- 1) Draw a Bayesian network corresponding to this setup.
- 2) Which coin is most likely to have been drawn out of the bag if the observed values of X_1, X_2 and X_3 are Heads, Heads and Tails?

Problem | Solution

The Bayesian network is as below:

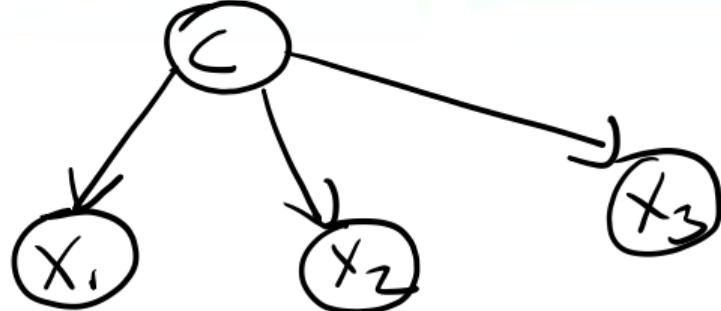
$P(x_2) = P(x_2/x_1, a)$

```
graph TD; C1((C1)) --> X1((X1)); C1 --> X2((X2)); C1 --> X3((X3)); X1 --> X2;
```

$C_1 \rightarrow$ random variable representing which coin is selected

$x_1, x_2, x_3 \rightarrow$ flips on the selected coin

Pr-Ver1 Solution



$X_1/a \rightarrow$
 X_1/b
 X_1/c

	C	H	T
a	0.2	0.8	
b	0.6	0.4	
c	0.8	0.2	

$$P(X_1 = T/a) \rightarrow P(H/a)$$

	C	H	T
a	0.2	0.8	
b	0.6	0.4	
c	0.8	0.2	

	C	H	T
a	0.2	0.8	
b	0.6	0.4	
c	0.8	0.2	

$$P(T/a)$$

Problem | Solution

To solve the 2nd part of the question we calculate $P(HHT/a)$, $P(HHT/b)$ and $P(HHT/c)$ and take the one that gives us the largest probability.

$$\begin{aligned}
 P(HHT/a) &= P(X_1 = H, X_2 = H, X_3 = T/a) \\
 &= P(X_1 = H/a)P(X_2 = H/a)P(X_3 = T/a) \\
 &= (0.2)^2(0.08) = 0.0032
 \end{aligned}$$

Problem 1 Solution

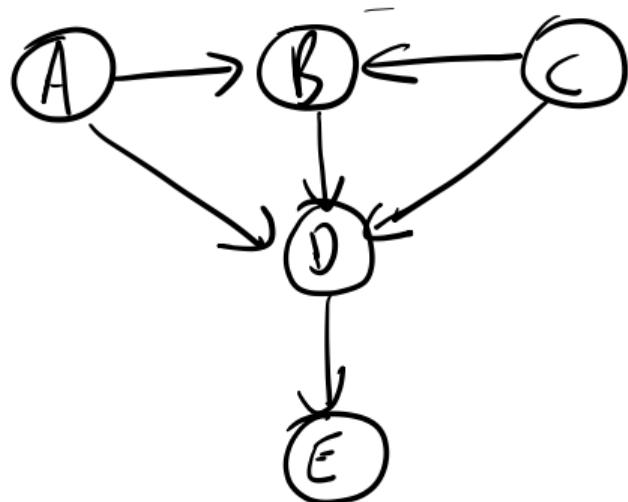
$$P(HHT/b) = (0.9)^2(0.4) = 0.144$$

$$P(HHT/c) = (0.8)^2(0.2) = 0.128$$

$\therefore b$ is the coin that was most likely to have produced HHT

Pr - lem 2 Statement

Consider the Bayesian network below:



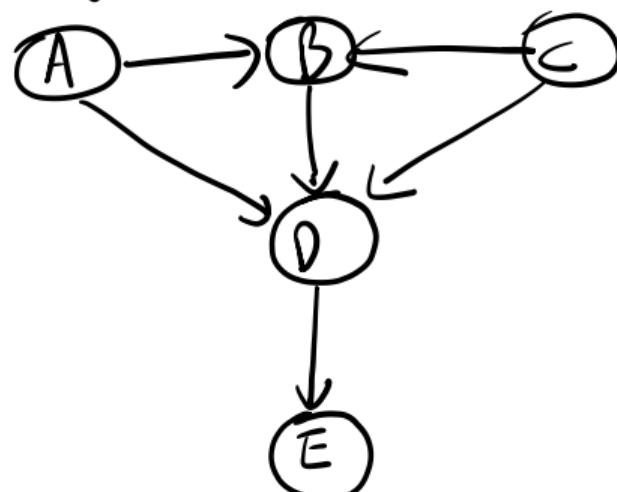
- 1) $P(A, B, C) \stackrel{?}{=} P(A) P(B) P(C)$
- 2) $P(E|D) \stackrel{?}{=} P(E|D, B)$
- 3) $P(C|A, B, D) \stackrel{?}{=} P(C|A, B, D, E)$

Problem 2 Solution

- (i) is false since there is an active path between A and B, and between B and C
- (ii) The path between B and E becomes inactive when D is specified, so (ii) is true
- (iii) C is conditionally independent of E given A, B, D
since all paths between C and E become inactive
 \therefore (iii) is true

Problem 3 Statement

For the graph below, calculate $P(A=0, B=0, C=1, D=0, E=0)$



$$\prod_{\text{overall}} p(x_i / p_a(x_i))$$

with CPDs given as in the next slide.

Problem 3 Statement

$$P(A=0) = 0.1, \quad P(C=0) = 0.1$$

A	B	C	$P(D)$
0	0	0	0.9
0	0	1	0.8
0	1	0	0.0
0	1	1	0.0
1	0	0	0.2
1	0	1	0.1
1	1	0	0.0
1	1	1	0.0

$$P(D=0 | A, B, C)$$

$P(B=0 A, C)$		
A	C	$P(B)$
0	0	0.9
0	1	0.5
1	1	0.5
1	0	0.1

D	$P(E)$
0	0.9
1	0.0

$$P(E=0 | D)$$

Problem 3 Statement

- 1) Calculate $P(A=0, B=0, C=1, D=0, E=0)$
- 2) Calculate $P(E=0 | A=0, B=0, C=0)$

Problem 3 Solution

① The joint probability can be expressed as a factorization

$$P(A=0)P(B=0/A=0, C=1)P(C=1)P(D=0/A=0, B=0)$$

$$P(E=0/D=0) \leftarrow \text{where did we get this from?}$$

$$= 0.9 \times 0.5 \times 0.9 \times 0.8 \times 0.9 \\ = 0.2916$$

Problem 3 Solution

$$P(E=0 | A=0, B=0, C=0) = \\ = \left[\frac{P(E=0, A=0, B=0, C=0, D=0)}{P(E=0, A=0, B=0, C=0, D=1)} \right] = P(E=0 | A=0, B=0, C=0)$$

$$P(A=0, B=0, C=0)$$

$$P(E | A, B, C)$$

$$= \frac{P(E, A, B, C)}{P(A, B, C)}$$

Problem 3 Solution

$$P(E=0, A=0, B=0, C=0, D=0)$$

$$= P(E=0/D=0) P(A=0) P(B=0/A=0, C=0) P(C=0) P(D=0/A=0, B=0, C=0)$$

$$P(E=0, A=0, B=0, C=0, D=1)$$

$$= P(E=0/D=1) P(A=0) P(B=0/A=0, C=0) P(C=0) P(D=1/A=0, B=0, C=0)$$

PrBLEM 3 Solution

$$\begin{aligned}
 & P(A=0)P\left(\frac{B=0}{A=0, C=0}\right)P(C=0) \left[P\left(\frac{E=0}{D=1}\right)P\left(\frac{D=1}{A=0, B=0, C=0}\right) \right. \\
 & \quad \left. + P\left(\frac{E=1}{D=0}\right)P\left(\frac{D=0}{A=0, B=0, C=0}\right) \right]
 \end{aligned}$$

$$P(E=0 | A=0, B=0, C=0)$$

$$P(A=0, B=0, C=0)$$

Problem 3 Solution

$$P(A=0, B=0, C=0) = P(A=0) P\left(\frac{B=0}{A=0, C=0}\right) P\left(\frac{C=0}{D, E}\right) \sum_{D, E} P\left(\frac{D}{A=0, B=0, C=0}\right) P(E/D)$$

After substitution we get

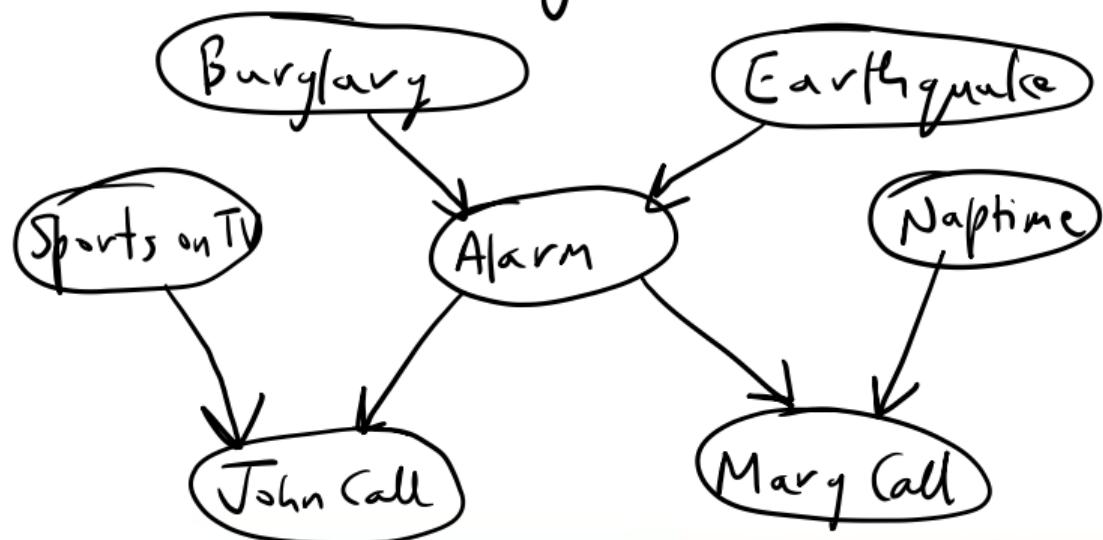
$$\frac{\sum_D P(D/A=0, B=0, C=0) P(E/D)}{\sum_D P(D/A=0, B=0, C=0) \sum_E P(E/D)}$$

Problem 3 Solution

$$= \frac{0.9x^0.9 + 0.1x^0}{0.9x^0.1 + 0.1x^0 + 0.9x^0.9 + 0.1x^1} \approx 0.81$$

Problem 4 Statement

Consider the following alarm network shown below:



Problem 4 Statement

Construct a Bayesian network structure over the nodes Burglary, Earthquake, Sports on TV, Naptime, John Call, Mary Call which is a minimal I-map for the marginal distribution over those variables defined by the network in the previous slide. Be sure to get all dependencies that remain from the original network.

Problem 4 Solution

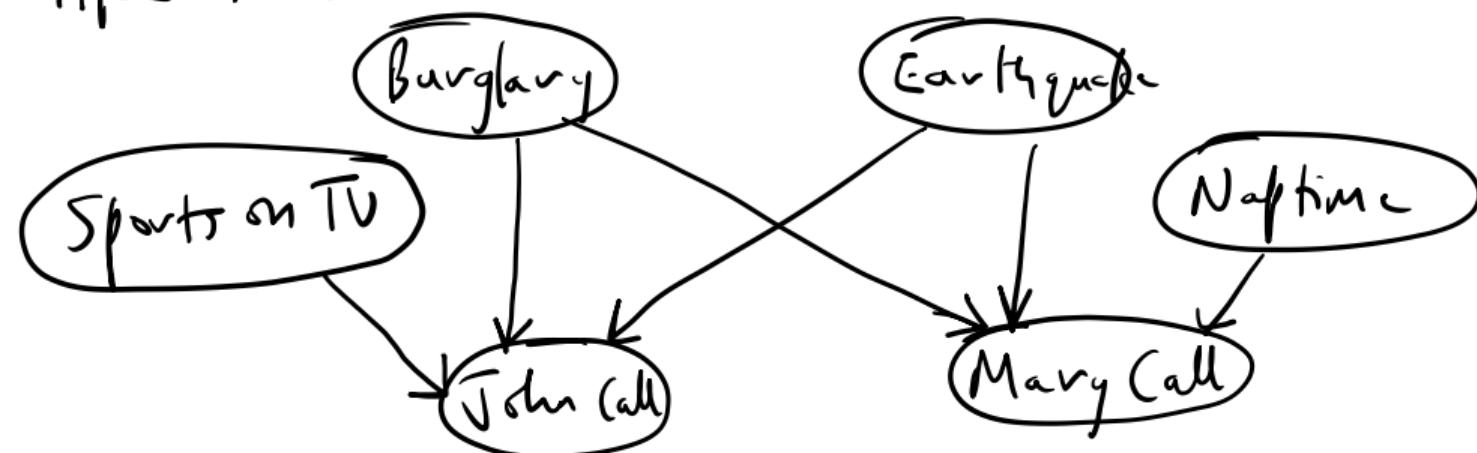
The key idea is to regard Alarm as unobserved.
Since we marginalize over Alarm.

We must preserve all the old independencies and add new edges to account for the fact that Alarm is removed.

If Alarm is unobserved, there exist active paths between Alarm's parents and Alarm's children

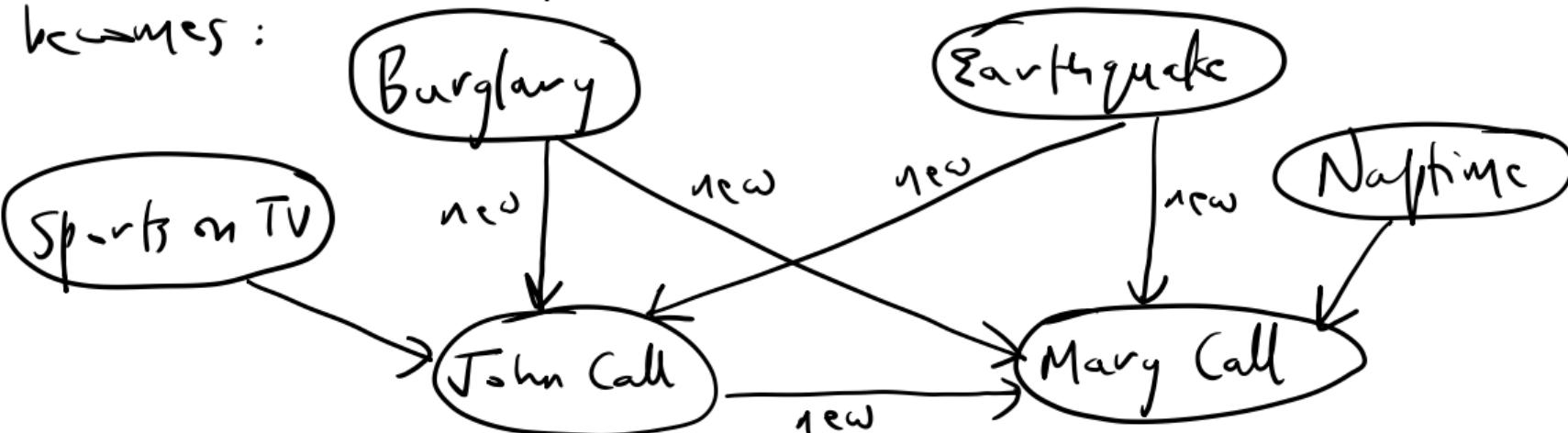
Problem 4 Solution

Thus as a first step - our graph needs to look like this:



Problem & Solution

Now we observe that any two children of Alarm also have an active path between them, so the graph becomes :



Problem 4 Solution

Now it appears that there is an active path between Sports on TV and Mary call through John call such that

Sports on TV \perp Mary Call | John Call

However in the original graph we had
Sports on TV \rightarrow John Call \leftarrow Alarm \rightarrow Mary Call

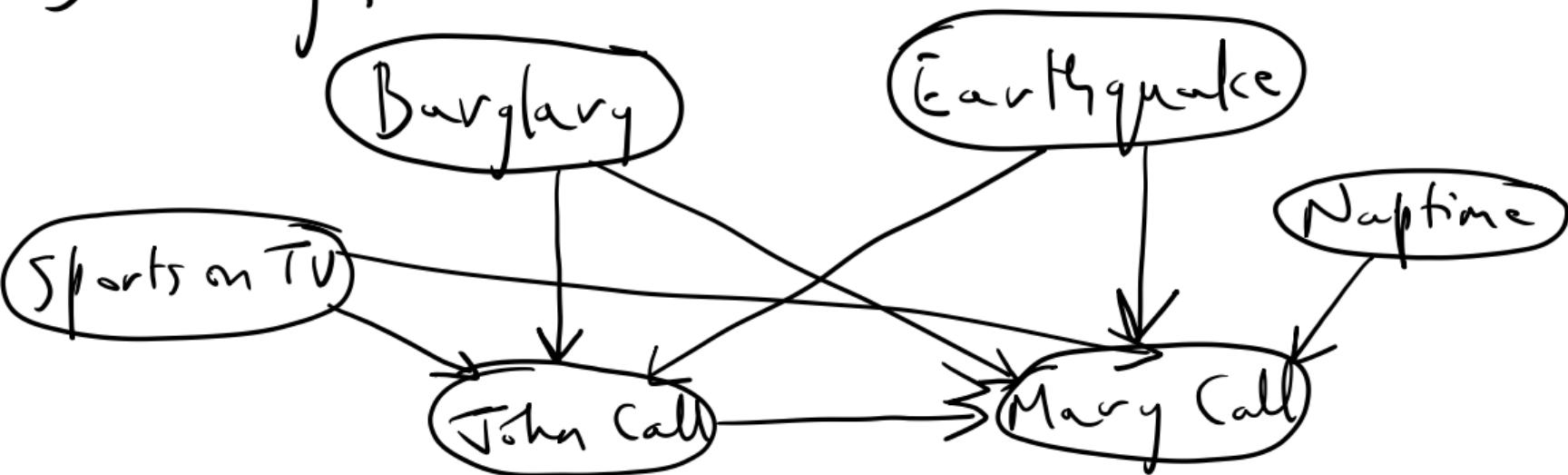
Problem & Solution

John Call is a converging node, specifying it leads to an active path from Sports on TV to Mary Call, which is the opposite of what we have in our new graph

We compensate for this by adding a direct edge between Sports on TV and Mary Call

Problem 4 Solution

So we get



Problem & Solution

Any more edges? The original graph looked symmetric, should we not add an edge between Naptime and John Call to preserve symmetry?

In the old graph:

Naptime → Mary Call ← Alarm → John Call -
(Mary Call converging)

Problem 4 Solution

Now, we have

Naptime \rightarrow MaryCall \leftarrow JohnCall

Naptime and JohnCall related in the same way as before (since Alarm is unobserved)
So no edge needed between Naptime and John Call

Problem 5

This problem will construct a practical example of a non-positive distribution where local independencies do not imply global ones.

Let P be any distribution over $X = \{X_1, X_2, \dots, X_n\}$ and let $X' = \{X'_1, X'_2, \dots, X'_n\}$

We construct a distribution $P'(X, X')$ whose marginal over X_1, X_2, \dots, X_n is the same as P and where each X_i is deterministically equal to X'_i .

Problem 5

Let H be a Markov network containing no edges other than $X_i - X_i'$. Question: Is H an I-map $P \rightarrow P'$? by which we mean, does every global dependency asserted by H also exist in P ?

Problem 5 Solution

$$\begin{array}{c} x_1 \longrightarrow x'_1 \\ x_2 \longrightarrow x'_2 \end{array} \quad \left. \begin{array}{c} \\ \\ \vdots \end{array} \right\} H$$

In graph H we see local independencies of the form
 $(x_i \perp \text{all variables other than } x_i \& x'_i | x'_i)$

These local independencies are also satisfied in P' since
 once x'_i is known, x_i has to be equal to x'_i and has no
dependence on any other variable.

Problem 5 Solution

What about global independencies asserted by H ? We see that X_i is D-separated from every other node X_j given the empty set, so X_i should be independent of X_j .

But do these independencies hold in P' ?

No because P was any distribution on X_1, X_2, \dots, X_n and may not support these independencies.

Problem 6

We shall show now that for a non-positive distribution, the pairwise independencies do not imply local independencies.

Let P be any distribution over $X = \{X_1, X_2, \dots, X_n\}$ and consider two auxiliary sets of variables X' and X'' and define $X^* = X \cup X' \cup X''$.

We now construct a distribution P' over X^* such that its marginal over X_1, X_2, \dots, X_n is the same as that of P , and such that $X_i = X'_i = X''_i$ deterministically.

Problem 6

Let H be the empty Markov network over X^* . Does H satisfy pairwise independence assertions in P' ?

Are all local independencies asserted by H also found in P' ?

Pr-blem 6 Solution

We note that $X_i \perp X_j \mid X^* - \{X_i, X_j\}$ since

$X^* - \{X_i, X_j\}$ contains X_i'' and $X_i = X_i'' = X_i'$ so

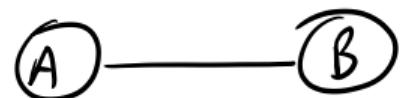
$$P(X_i/X_i'', X_i') = P(X_i/X_i'')$$

Similarly X_i and X_j are independent given all other nodes

Thus H satisfies all pairwise independencies but not local or global independencies

Problem 7

Let a, b, c, d be binary variables taking the values 0 and 1. Let $P(a, b, c, d)$ be the joint distribution such that $P(a^0, b^0, c^0, d^0) = 0.5$ and $P(a^1, b^1, c^1, d^1) = 0.5$. Can the Markov network constructed according to local independencies be the following graph?



Problem 7 Solution

We see that if we know the value of B (either 0 or 1), the value of A is known \rightarrow we do not care about the value of C or D .

$$\therefore P(A|B, C, D) = P(A|B). \text{ Thus } (A \perp C, D | B)$$

In the given graph A is H-separated from C and D once B is specified \rightarrow does this mean that the given graph is a I-map?

Prblm 7 Solution

We see similarly $B \perp C, D | A$, $C \perp A, B | D$ and so on. All these independencies are supported by the distribution and the graph.

However the graph is not an I-map for P because it asserts fake independencies like $A \perp C$ given the empty set.



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL

SESSION # 11: APPROXIMATE INFERENCE and MAP INFERENCE

SEETHA PARAMESWARAN

seetha.p@pilani.bits-pilani.ac.in



Table of Contents

- 1 Inference as Optimization
- 2 Exact Inference as an Optimization Problem
- 3 Propagation-Based Approximation

Constrained Optimization Problem



- Define a target class Q of **easy** distributions Q .
- Then search for an instance within that class that is the **best** approximation to P_Φ .
- Queries can then be answered using inference on Q rather than on P_Φ .
- This approach reformulates the inference task as one of optimizing an objective function over the class Q .

- This problem falls into the category of **constrained optimization**.



Table of Contents

- 1 Inference as Optimization
- 2 Exact Inference as an Optimization Problem
- 3 Propagation-Based Approximation

Exact Inference as an Optimization Problem

- Factorized distribution

$$P_\Phi(X) = \frac{1}{Z} \prod_{\varphi \in \Phi} \varphi(U_\varphi)$$

- $U_\varphi = \text{Scope}[\varphi] \subseteq X$ are scope of each factor φ in the distribution P_Φ .
- Queries about the distribution P_Φ include queries about marginal probabilities of variables and queries about the partition function Z .
- Exact inference finds a set of calibrated beliefs that represent $P_\Phi(X)$.
- We can view exact inference as searching over the set of distributions over the set of distributions Q that are representable by the cluster tree to find a distribution Q^* that matches P_Φ .

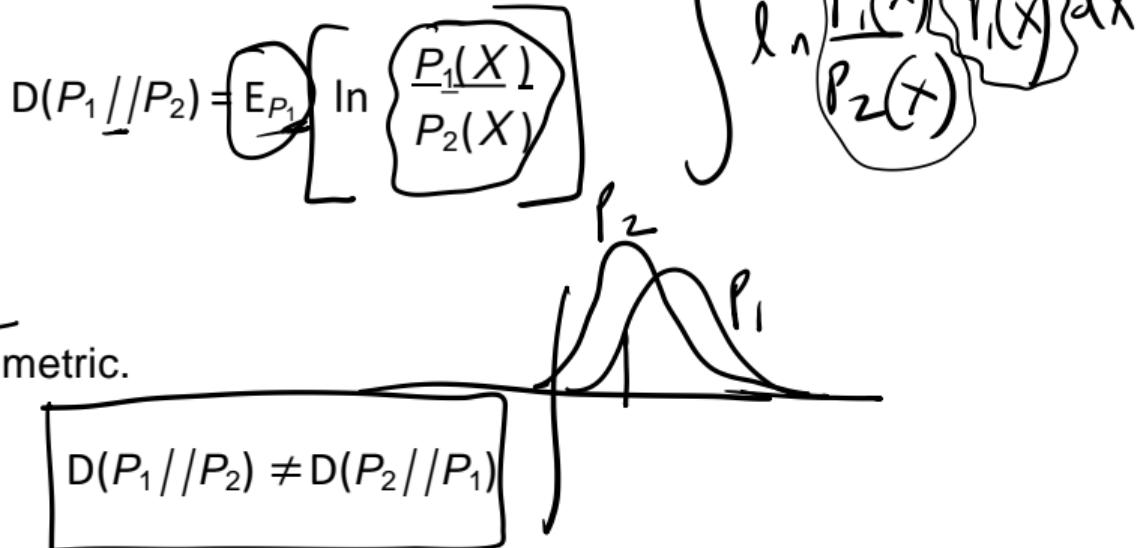
Exact Inference as an Optimization Problem

- Searching for a calibrated distribution that is as close as possible to P_Φ .
- Aim is to avoid performing inference with the distribution P_Φ .
- **Relative Entropy or KL Divergence** is used as a distance measure to find an approximation Q to P_Φ , such that the relative entropy is minimized.

Relative Entropy

$$E(X) = \int x p(x) dx$$

- Relative entropy between two distributions P_1 and P_2



Exact Inference as an Optimization Problem

- Goal is to search for a distribution Q that minimizes $D(Q // P_\Phi)$.
- Suppose the clique tree structure T for P_Φ satisfies running intersection property and family preservation property.

these are functions

$$Q = \underbrace{\{\beta_i : i \in V_T\}}_{\text{these are functions}} \cup \underbrace{\{\mu_{i,j} : (i - j) \in E_T\}}$$

$$Q(X) = \frac{\prod_{i \in V_T} \beta_i(C_i)}{\prod_{i \in E_T} \mu_{i,j}(S_{i,j})}$$

$$P_\Phi(X) = \frac{\prod_i p_i}{\prod_j m_j}$$

- Due to calibration requirement we have

$$\beta_i[c_i] = Q(c_i)$$

$$\mu_{i,j}[S_{i,j}] = Q(S_{i,j})$$

- Search for a Q that is representable by a set of beliefs over the cliques and sepsets in a particular clique tree structure T .

Exact Inference as an Optimization Problem

- ## CTree-Optimize-KL:

$$\text{Find } Q = \{\beta_i : i \in V_T\} \cup \{\mu_{i,j} : (i - j) \in E_T\}$$

Maximizing $-D(Q//P_\Phi)$ \equiv

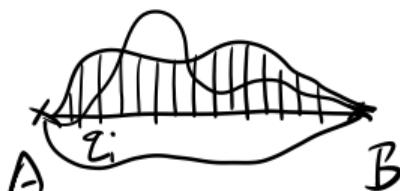
$$\text{subject to } \mu_{i,j}[s_{i,j}] = \sum_{c_i - s_{i,j}} \beta_i[c] \quad \forall (i-j) \in E_T, \forall s_{i,j} \in Val(S_{i,j})$$

$$\sum_{c_j} \beta_i[c_j] = 1 \quad \forall i \in V_T$$

- Optimization problem CTree-Optimize-KL has a unique solution.

Exact Inference as an Optimization Problem

- Applying Relative entropy equation in P_Φ



$$D(Q//P_\Phi) = \ln Z - F[\tilde{P}_\Phi, Q]$$

$$F[\tilde{P}_\Phi, Q] = \sum_{\varphi \in \Phi} E_d[\ln \varphi] + H_Q(X)$$

Calculus of Variations $D(Q//P_\Phi) \geq 0 \rightarrow P_\Phi \leftarrow \text{entropy}$

$$\ln Z \geq F[\tilde{P}_\Phi, Q]$$

a function of a function

$F[\tilde{P}_\Phi, Q]$ is called the **energy functional**.

The first term is called the **energy term**.

The second term is called the **entropy term**.

Minimizing the relative entropy is equivalent to maximizing the energy functional.

Exact Inference as an Optimization Problem

$$\begin{aligned} & \min f(x, y, z) \xrightarrow{\text{Lagrangian multiplier}} \nabla f(x, y, z) \\ & \text{s.t. } g(x, y, z) = c \\ & = \lambda \nabla g(x, y, z) \end{aligned}$$

CTree-Optimize:

Find $Q = \{\beta_i : i \in V_T\} \cup \{\mu_{i,j} : (i - j) \in E_T\}$

Maximizing $F[\tilde{P}_\Phi, Q]$

subject to $\mu_{i,j}[s_{i,j}] = \sum_{c_i \in S_{i,j}} \beta_i[c] \quad \forall (i - j) \in E_T, \forall s_{i,j} \in \text{Val}(S_{i,j})$

$$\sum_{c_i \in C_i} \beta_i[c] = 1 \quad \forall i \in V_T$$

$$\beta_i(c_i) \geq 0 \quad \forall i \in V_T; c_i \in \text{Val}(C_i)$$

Fixed-point characterisation

We need first the concept of Lagrange multipliers since we will convert the given constrained optimization problem to an unconstrained one.

To understand the derivation in the book, we need Lagrange multipliers + functionals \Rightarrow out of scope

Fixed-point characterisation

Setting up Lagrange multiplier equations for the constrained optimization problem and finding stationary points leads to the same sort of equations as the message passing algorithm

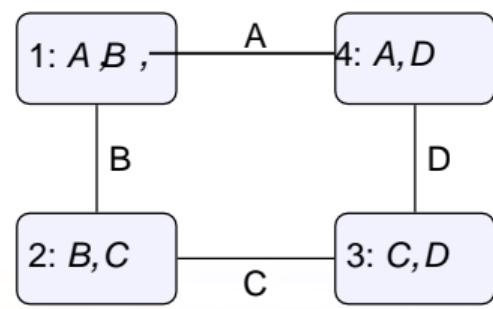
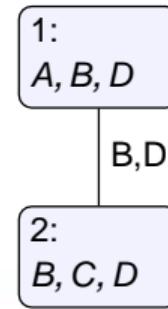
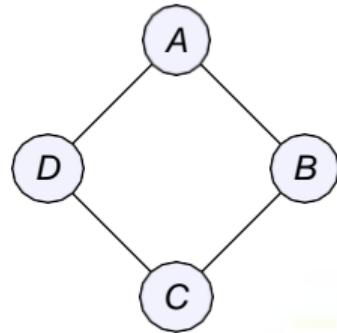


Table of Contents

- 1 Inference as Optimization
- 2 Exact Inference as an Optimization Problem
- 3 Propagation-Based Approximation

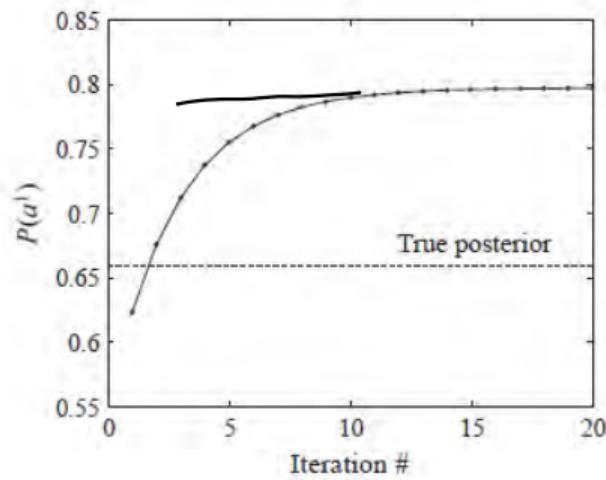
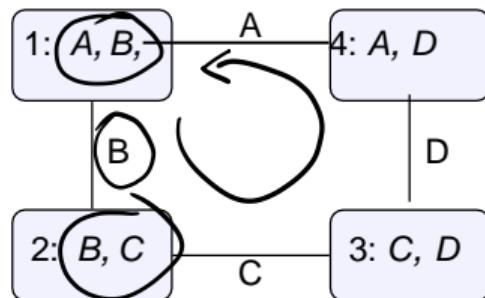
Propagation-Based Approximation

- Use the same message propagation as in exact inference.
- Use Cluster graph instead of clique tree.
- The cluster graph contains loops (undirected cycles), such graphs are often called **loopy**.
- The BP algorithm is called **Loopy belief propagation**, since it uses propagation steps used by algorithms for Markov trees, but applied to networks with loops.



Propagation-Based Approximation

- Message propagation process may not converge in two passes, since information from one pass will circulate and affect the next round.
- In some cases, the propagation of beliefs may not converge at all.
- An example run of loopy belief propagation is given below.



What happens if we use CTrees-B0-Update?

Let us propagate messages in the order $m_{12}, m_{23},$
 m_{34}, m_{41}

In the first message m_{12} , the cluster AB
passes information to cluster BC using a
marginal distribution on B

What happens if we use CTree-BU-Update?

Suppose all clusters favour consensus joint assignments

$\beta_1(a^0, b^0)$ and $\beta_1(a^1, b^1)$ much larger than
 $\beta_1(a^0, b^1)$ and $\beta_1(a^1, b^0)$

If μ_{12} strengthens the belief that $B = b'$,
then μ_{23} strengthens the belief that $C = c'$

what happens if we use CTree-BU-Update?

Going around the loop, cluster AB will get a message that strengthens the belief that $A = a$

This message will be treated as being independent of the initial propagation when it is not so. \Rightarrow This procedure overestimates the Marginal probability of A

Cluster-graph Belief Propagation

We say that U satisfies the running intersection property if, whenever there is a variable X such that $X \in C_i$ and $X \in C_j$, then there is a single path between C_i and C_j for which $X \in S_e$ for all edges e in the path

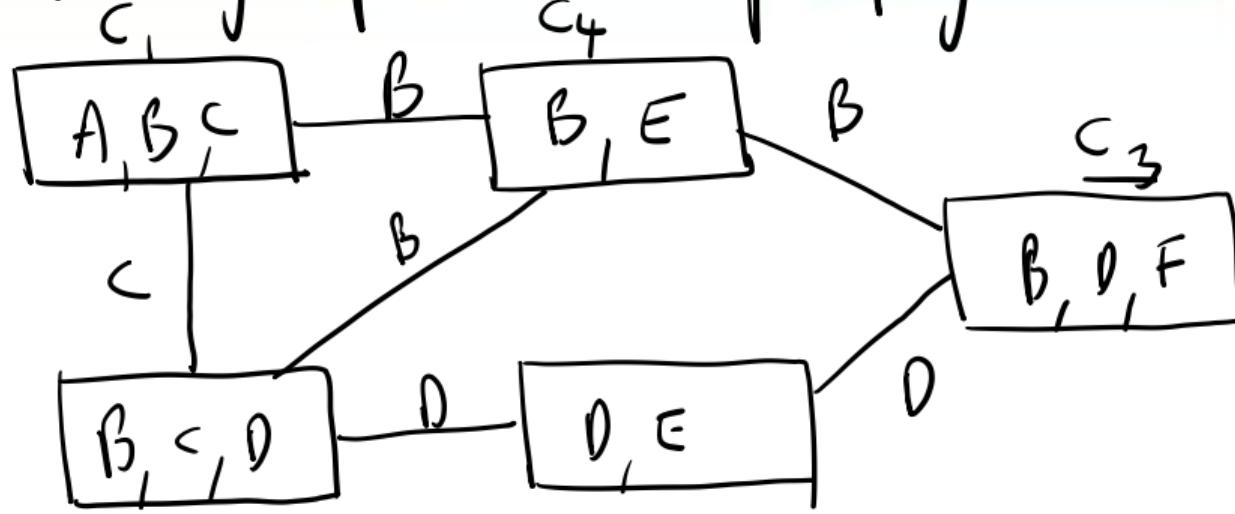


Cluster-graph belief propagation

There is only a single path in which information about X can flow in the graph.

- All clusters must agree on a marginal distribution of X .
- At most one path means information about X cannot endlessly cycle in a loop.

Cluster-graph belief propagation



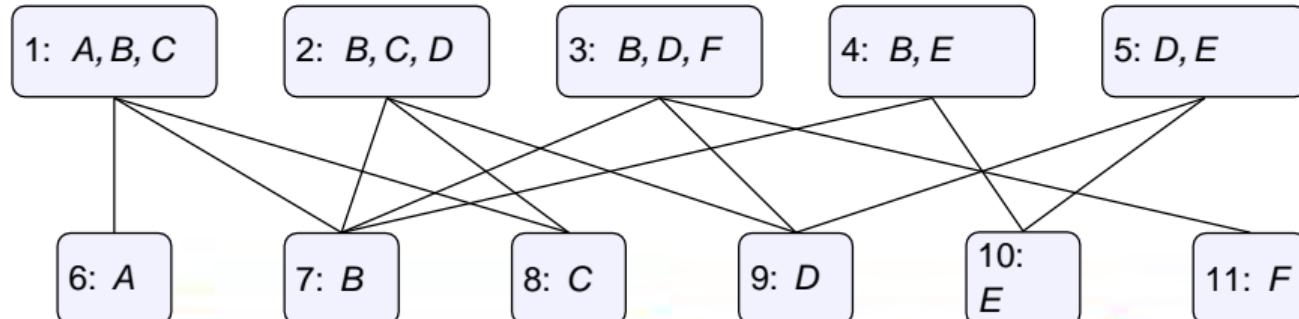
Two paths from C_3 to C_2

Cluster-graph belief propagation

- The first path from c_3 to c_2 goes through c_4 and propagates information about B
- The second path from c_2 to c_3 goes through c_5 and propagates information about D
 - We can still get circular reasoning
 - For edge $i-j$ having $c_i, c_j \in \beta_i = \sum_{c_i - s_{i,j}} \beta_j$

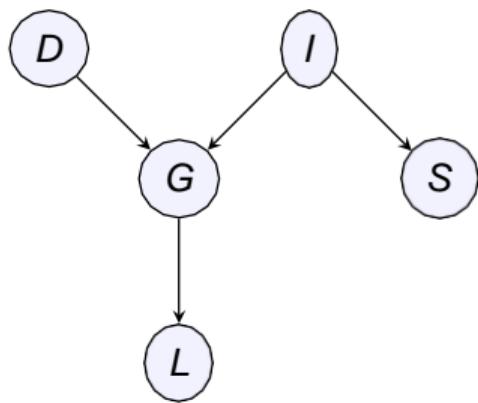
Bethe Cluster Graph

- Bethe cluster graph, uses a bipartite graph.
- The first layer graph consists of “large” clusters, with one cluster for each factor φ in Φ , whose scope is $\text{Scope}[\varphi]$.
- These clusters satisfy the family-preservation property.
- The second layer consists of “small” univariate clusters, one for each random variable.
- Place an edge between each univariate cluster X on the second layer and each cluster in the first layer that includes X ; the scope of this edge is X itself.



MAP and Variable Elimination

$\gamma_1 \frac{D}{}$



$$\max_{S, I, D, L, G} P(D, I, G, S, L)$$

$$= \max_{L, G} [\varphi_L(L, G)] \cdot \text{Max}_D [\varphi_D(D) \cdot T(G, D)]$$

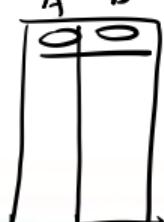
$$= \max_{L, G} [\varphi_L(L, G)] \cdot T_3(G)$$

$$= \max_G [T_3(G)] \cdot \max_L [\varphi_L(L, G)]$$

$$= \max_G [T_3(G)] \cdot T_4(G)$$

$$= T_5(\theta)$$

$$\max(\phi_1, \phi_2) = \phi_1 + \max_x \phi_2$$



A	B	C
0	0	
0	1	
1	0	
1	1	

MAP and Variable Elimination

Step	Variable eliminated	Factors used	Intermediate factor	New factor
1	<u>S</u>	<u>$\varphi_S(S, I)$</u>	<u>$\psi_1(I, S)$</u>	$T_1(I)$
2	<u>I</u>	$\varphi(I) \cdot \varphi_G(G, I, D) \cdot T_1(I)$	<u>$\psi_2(G, I, D)$</u>	$T_2(G, D)$
3	<u>D</u>	<u>$\varphi_D(D) \cdot T_2(G, D)$</u>	<u>$\psi_3(G, D)$</u>	$T_3(G)$
4	<u>L</u>	$\varphi_L(L, G)$	<u>$\psi_4(L, G)$</u>	$T_4(G)$
5	<u>G</u>	$T_3(G) \cdot T_4(G)$	<u>$\psi_5(G)$</u>	<u>$T_5(\theta)$</u>

$$\text{Max Marg}(G) = \psi_5(G)$$

Now choose the maximizing value x_i^* for X_i .

$$\max_{a,b} P(a, b) = \max_a P(a) \max_b P(b|a) \cancel{P(a)}$$

MAP, Variable Elimination and Traceback

lead

$$\max \cancel{P(b|a)} \quad \begin{matrix} a, \\ b \end{matrix} \quad \begin{matrix} \cancel{P(a)} \rightarrow P(b|a) \\ \text{chain t.} \end{matrix}$$

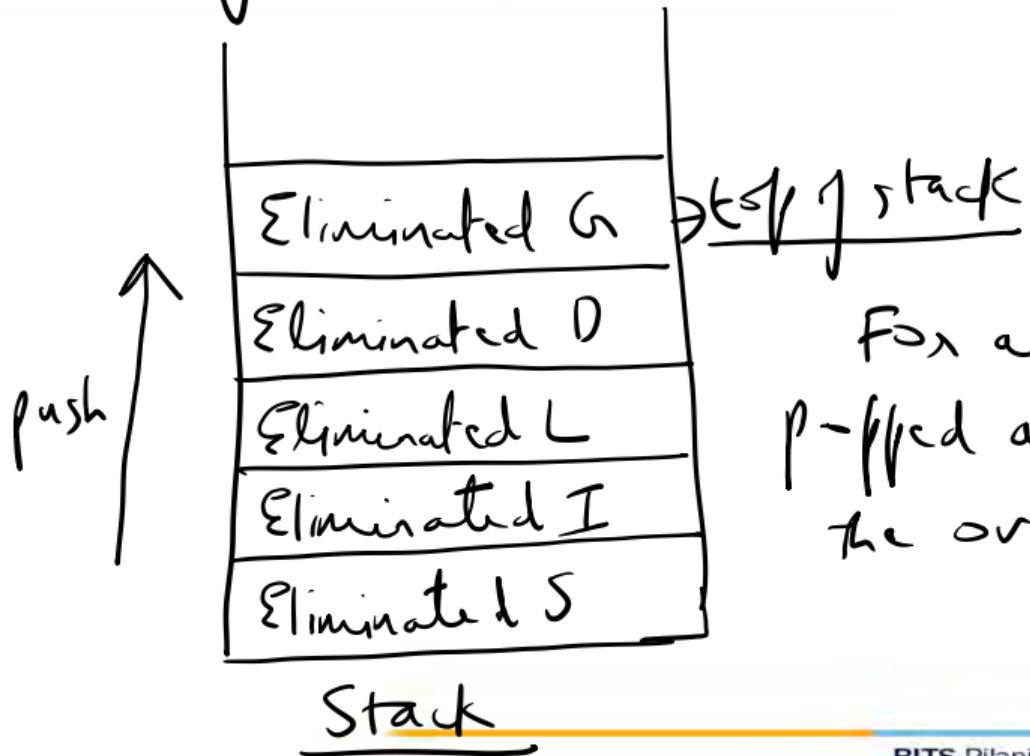
- Determine a conditional maximizing value – their maximizing value given the values of the variables that have not yet been eliminated.
- Pick the value of the final variable.
- Then go back and pick the values of the other variables accordingly.
- For the last variable eliminated X , the factor for the value x contains the probability of the most likely assignment that contains $X = x$.
- This process is called **traceback** of the solution.

$$\hat{a} \rightarrow \max_b P(b/a)$$

MAP, Variable Elimination and Traceback

Step	Variable eliminated	Factors used	Intermediate factor	New factor	Traceback
1	S	$\varphi_S(S, I)$	$\psi_1(I, S)$	$T_1(I)$	$s^* = \arg \max_s \psi_1(i^*, s)$
2	I	$\varphi_I(I) \cdot \varphi_G(G, I, D) \cdot T_1(I)$	$\psi_2(G, I, D)$	$T_2(G, D)$	$i^* = \arg \max_i \psi_2(g^*, d^*, i)$
3	D	$\varphi_D(D) \cdot T_2(G, D)$	$\psi_3(G, D)$	$T_3(G)$	$= \arg \max_d \psi_3(g^*, d)$
4	L	$\varphi_L(L, G)$	$\psi_4(L, G)$	$T_4(G)$	$l^* = \arg \max_l \psi_4(g^*, l)$
5	G	$T_3(G) \cdot T_4(G)$	$\psi_5(G)$	$T_5(\theta)$	$g^* = \arg \max_g \psi_5(g)$

Think of it like a stack....



For assignment the stack is
popped and variables assigned in
the order $G^* \rightarrow D^* \rightarrow L^* \rightarrow I^* \rightarrow S^*$

MAP and Variable Elimination Algorithm

Procedure Max-Product-VE (

Φ , // Set of factors over X
 \prec // Ordering on X

)

1 Let X_1, \dots, X_k be an ordering of X such that

2 $X_i \prec X_j$ iff $i < j$

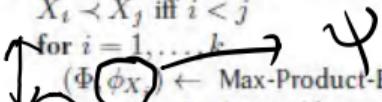
3 **for** $i = 1, \dots, k$

4 $(\Phi | \phi_{X_i}) \leftarrow \text{Max-Product-Eliminate-Var}(\Phi, X_i)$

5 $x^* \leftarrow \text{Traceback-MAP}(\{\phi_{X_i} : i = 1, \dots, k\})$

6 **return** x^*, Φ // Φ contains the probability of the MAP

$$X_1 = S, X_2 = I, X_3 = L, X_4 = D, X_5 = G$$



Procedure Max-Product-Eliminate-Var (

Φ , // Set of factors

Z // Variable to be eliminated

)

1 $\Phi' \leftarrow \{\phi \in \Phi : Z \in \text{Scope}[\phi]\}$

2 $\Phi'' \leftarrow \Phi - \Phi'$

3 $\psi \leftarrow \prod_{\phi \in \Phi'} \phi$

4 $\tau \leftarrow \max_Z \psi$

5 **return** $(\Phi'' \cup \{\tau\}) \psi$

Procedure Traceback-MAP (

$\{\phi_{X_i} : i = 1, \dots, k\}$

)

1 **for** $i = k, \dots, 1$

2 $u_i \leftarrow (x_{i+1}^*, \dots, x_k^*) (\text{Scope}[\phi_{X_i}] - \{X_i\})$

// The maximizing assignment to the variables eliminated after X_i

3 $x_i^* \leftarrow \arg \max_{x_i} \phi_{X_i}(x_i, u_i)$

// x_i^* is chosen so as to maximize the corresponding entry in the factor, relative to the previous choices u_i

4 **return** x^*

after X_i

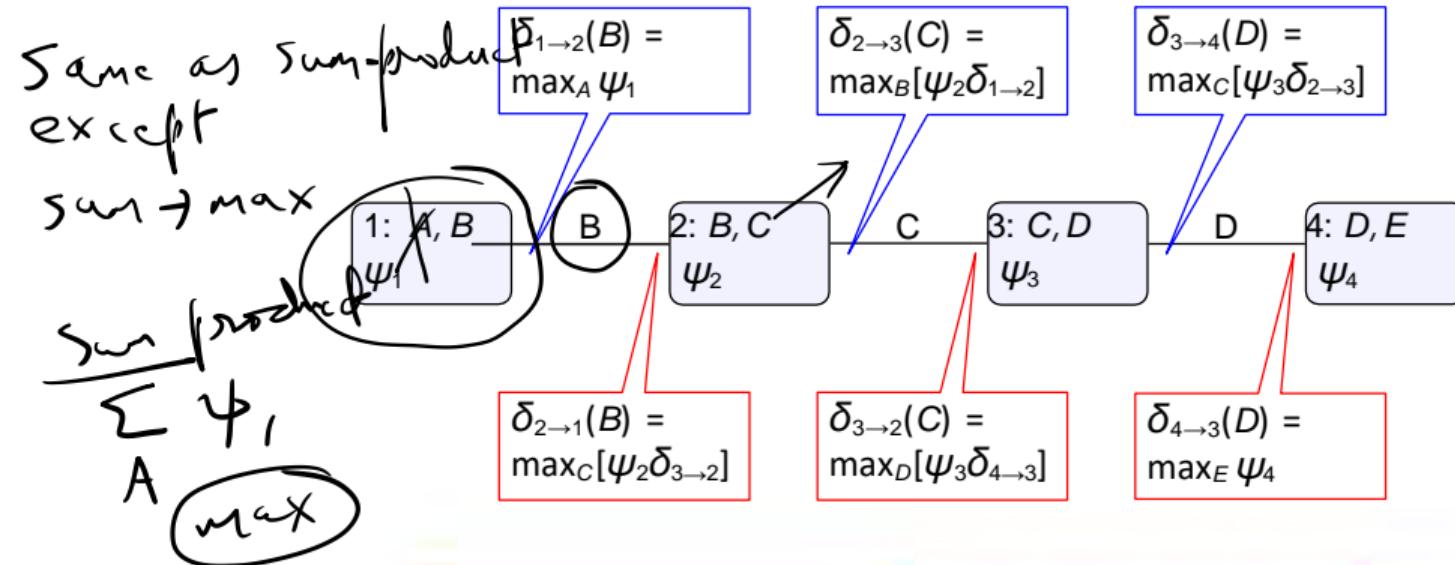
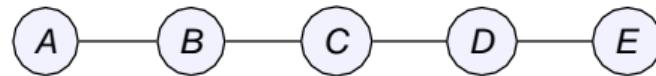
$X_i, X_{i+1}, X_{i+2}, \dots, X_k$



Table of Contents

- 1 Inferences
- 2 Maximum a Posteriori (MAP) Query
- 3 Max Product and Max Marginals
- 4 MAP and Variable Elimination
- 5 MAP using Belief Propagation

MAP using Belief Propagation



MAP using Belief Propagation

- An exact solution to the MAP problem via a variable elimination procedure is intractable.
- Use message passing procedures in cluster graphs to compute approximate max-marginals.
- These pseudo-max-marginals can be used for selecting an assignment.
- The task has two parts: computing the max-marginals and decoding them to extract a

MAP assignment.

$$\psi(C_i) = \psi_r \prod \delta_{k \rightarrow i}$$

$$\tau(S_{i,j}) = \max_{C_i - S_{i,j}} \psi(C_i)$$

MAP using Belief Propagation

- For each clique C_i , and each assignment c_i to C_i ,

$$\beta_i(c_i) = \text{MaxMarg}_{P_\Phi}(c_i)$$

↑

- Any two adjacent cliques must agree on their sepset. The cliques are said to be max-calibrated.

$$\max_{C_i - S_{i,j}} \beta_i = \max_{C_j - S_{i,j}} \beta_j = \mu_{ij}(S_{i,j})$$

- The beliefs in a clique tree resulting from an upward and downward pass of the max-product clique tree algorithm are max-calibrated.

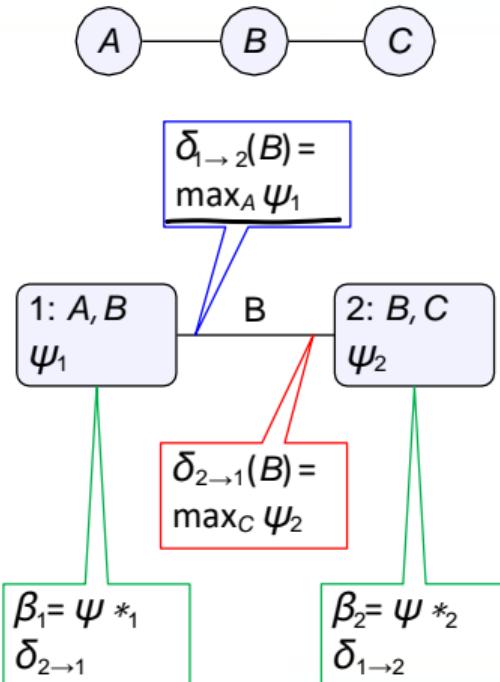
MAP using Belief Propagation

- The assignment ξ^* has the local optimal assignment ξ^* given a max-calibrated set of beliefs $\beta_i(C_i)$, if for each clique

$$\xi^*(C_i) \in \arg \max_{c_i} \beta(c_i)$$

- The task of finding a locally optimal assignment ξ^* given a max-calibrated set of beliefs is called the decoding task.

MAP + BP – Most Likely Assignment – Example



ψ_1	$a^1 b^1$	3	
	$a^2 b^1$	-1	
	$a^1 b^2$	0	
	$a^2 b^2$	1	

ψ_2	$b^1 c^1$	4	
	$b^1 c^2$	-1	
	$b^2 c^1$	1	
	$b^2 c^2$	2	

$\delta_{1 \rightarrow 2}$	b^1	3	
	b^2	1	

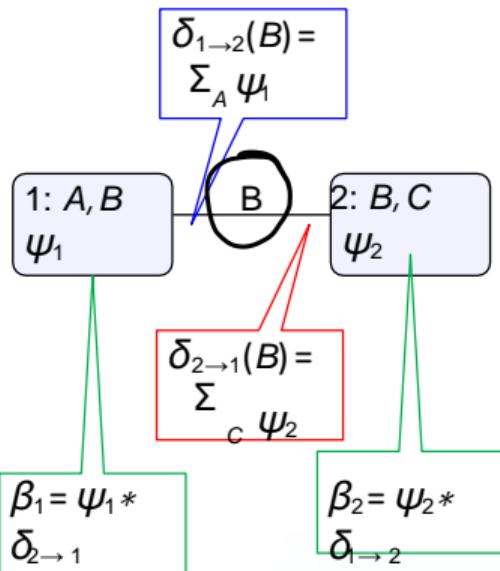
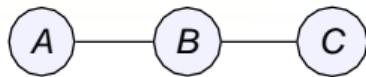
$\delta_{2 \rightarrow 1}$	b^1	4	
	b^2	2	

β_1	$a^1 b^1$	$3 * 4 = 12$	
	$a^2 b^1$	$-1 * 4 = -4$	
	$a^1 b^2$	$0 * 2 = 0$	
	$a^2 b^2$	$1 * 2 = 2$	

β_2	$b^1 c^1$	$4 * 3 = 12$	
	$b^1 c^2$	$-1 * 3 = -3$	
	$b^2 c^1$	$1 * 1 = 1$	
	$b^2 c^2$	$2 * 1 = 2$	

Most Likely assignment = (a^1, b^1, c^1)

MAP + BP – Calibration – Example



β_1	$a^1 b^1$	$3 * 4 = 12$
	$a^2 b^1$	$-1 * 4 = -4$
	$a^1 b^2$	$0 * 2 = 0$
	$a^2 b^2$	$1 * 2 = 2$

β_2	$b^1 c^1$	$4 * 3 = 12$
	$b^1 c^2$	$-1 * 3 = -3$
	$b^2 c^1$	$1 * 1 = 1$
	$b^2 c^2$	$2 * 1 = 2$

$$\max_A \beta_1 \begin{array}{|c|c|} \hline b^1 & 12 \\ \hline b^2 & 2 \\ \hline \end{array} = \max_C \beta_2 \begin{array}{|c|c|} \hline b^1 & 12 \\ \hline b^2 & 2 \\ \hline \end{array}$$



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 12: APPROXIMATE INFERENCE

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in

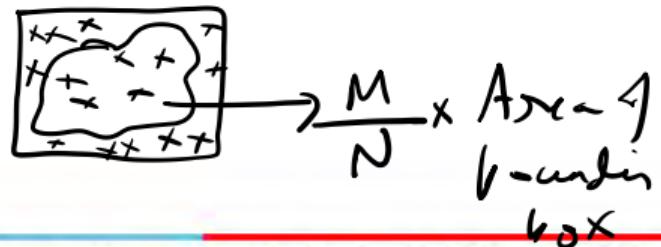
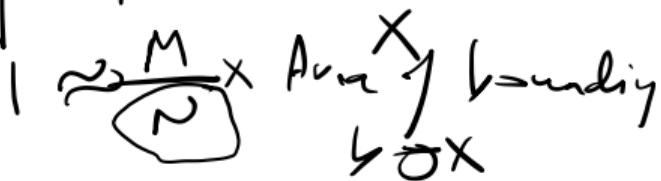
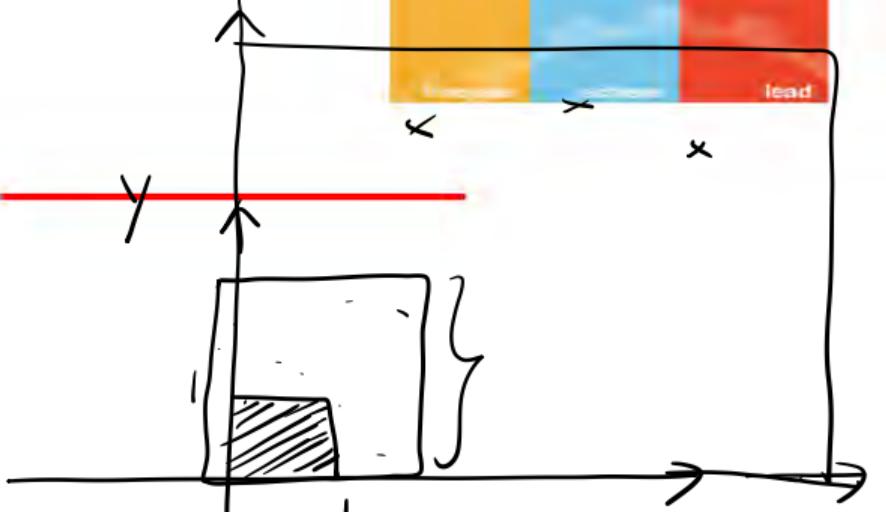
Table of Contents

0 + 0 + 2

1 Approximate Inference

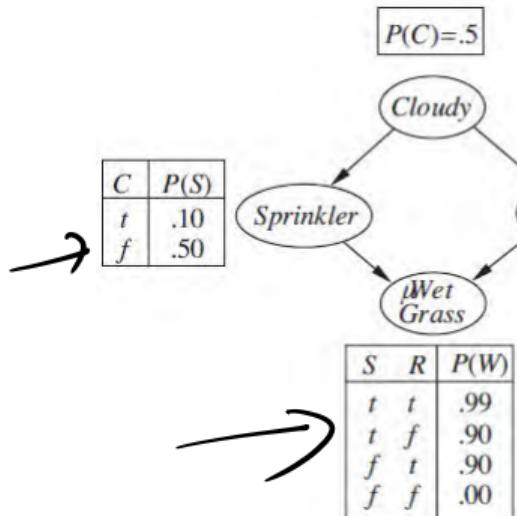
2 Propagation-Based Approximation

3 Markov Chain Monte Carlo Simulation

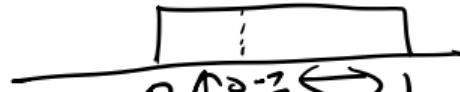


Approximate Inference Methods

- Given the intractability of exact inference in large, multiply connected networks, it is essential to consider approximate inference methods.



$$u = \frac{d \nu \text{ and } 48}{}$$



$$P(X=0) = 0.3 \quad X=0 \quad X=1$$

Approximate Inference Methods



- Provide approximate answers whose accuracy depends on the number of samples generated.
- Used to estimate quantities that are difficult to calculate exactly.
- Sampling can be applied to compute of posterior probabilities.
- Two families of sampling algorithms:
 - 1 Direct sampling
 - 2 Markov chain sampling

Direct Sampling in Bayesian Networks

48()

- The primitive element in any sampling algorithm is the generation of samples from a known **probability distribution**.
- Given a source of random numbers uniformly distributed in the range $[0, 1]$, it is a simple matter to sample any distribution on a single variable, whether discrete or continuous.
- PRIOR-SAMPLE generates samples from the prior joint distribution specified by the network.

function PRIOR-SAMPLE(*bn*) **returns** an event sampled from the prior specified by *bn*
inputs: *bn*, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \dots, X_n)$

x \leftarrow an event with n elements

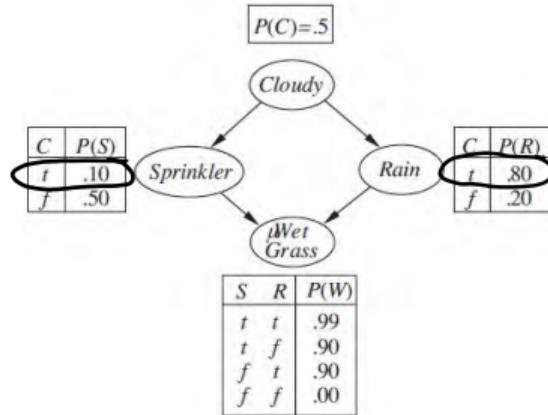
foreach variable X_i **in** X_1, \dots, X_n **do**

x[i] \leftarrow a random sample from $\mathbf{P}(X_i \mid \text{parents}(X_i))$

return **x**

Direct Sampling - Example

- Assume ordering [Cloudy, Sprinkler, Rain, WetGrass]
- Sample from $P(\text{Cloudy}) = < 0.5, 0.5 >$.
- Sampled value is true.
- Sample from $P(\text{Sprinkler} | \text{Cloudy} = \text{true}) = < 0.1, 0.9 >$. Sampled value is false.
- Sample from $P(\text{Rain} | \text{Cloudy} = \text{true}) = < 0.8, 0.2 >$.
- Sampled value is true.
- Sample from $P(\text{WetGrass} | \text{Sprinkler} = \text{false}, \text{Rain} = \text{true}) = < 0.9, 0.1 >$. Sampled value is true.
- PRIOR-SAMPLE returns the event [true, false, true, true].

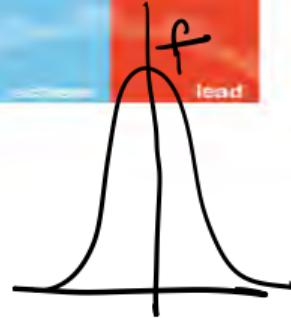


$$F^{-1}(u) = y$$

Rejection Sampling in Bayesian Networks



$$\int f(x) dx \stackrel{?}{=} \int df$$
$$\int f(x) dx = F(t) \stackrel{?}{=} \text{cdf}$$



- Produce samples from a hard-to-sample distribution given an easy-to-sample distribution.
- Used to compute conditional probabilities $P(X | e)$.
- Rejection sampling produces a consistent estimate of the true probability.

Rejection Sampling in Bayesian Networks

- First, it generates samples from the prior distribution specified by the network. Then, it rejects all those that do not match the evidence. Finally, the estimate $\hat{P}(X | e)$ is obtained by counting how often $X = x$ occurs in the remaining samples.

```

function REJECTION-SAMPLING( $X, e, bn, N$ ) returns an estimate of  $P(X|e)$ 
  inputs:  $X$ , the query variable
     $e$ , observed values for variables  $E$ 
     $bn$ , a Bayesian network
     $N$ , the total number of samples to be generated
  local variables:  $N$ , a vector of counts for each value of  $X$ , initially zero
  for  $j = 1$  to  $N$  do
     $x \leftarrow \text{PRIOR-SAMPLE}(bn)$ 
    if  $x$  is consistent with  $e$  then
       $N[x] \leftarrow N[x] + 1$  where  $x$  is the value of  $X$  in  $x$ 
  return NORMALIZE( $N$ )

```

10000
950 samples survived
 $N(x)$
950

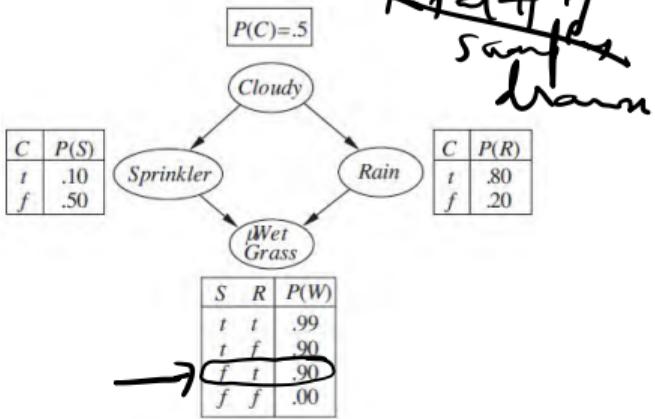
Rejection Sampling - Example

$$P(Q/e) = \frac{P(Q, e)}{P(e)}$$

samples
 # Q and e
 = ~~# total # Q~~
 # Q samples
 having e
total # samples
~~total # samples~~

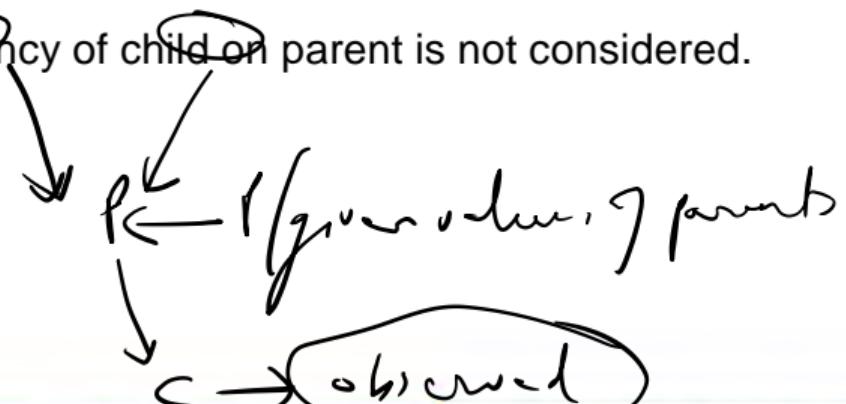
- Assume that we wish to estimate $P(\text{Rain} / \text{Sprinkler} = \text{true})$, using 100 samples.
- Of the 100 that we generate, suppose that 73 have $\text{Sprinkler} = \text{false}$ and are rejected, while 27 have $\text{Sprinkler} = \text{true}$;
- Of the 27 accepted samples, 8 have $\text{Rain} = \text{true}$ and 19 have $\text{Rain} = \text{false}$. Hence,
 $P(\text{Rain} / \text{Sprinkler} = \text{true}) \approx \text{Normalize}(< 8, 19 >) = < 0.296, 0.704 >.$

$$< \frac{8}{8+19} \mid \frac{19}{8+19} >$$



Rejection Sampling - Issues

- Rejection sampling is expensive, as it rejects many samples.
- Unusable for complex problems.
- When the child is observed, the dependency of child on parent is not considered.



Likelihood Weighting in Bayesian Networks

- Likelihood weighting avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence e .
- It is a particular instance of the general statistical technique of **importance sampling**, tailored for inference in Bayesian networks.
- LIKELIHOOD WEIGHTING fixes the values for the evidence variables E and samples only the nonevidence variables. This guarantees that each event generated is consistent with the evidence.
- Each event is weighted by the likelihood that the event accords to the evidence, as measured by the product of the conditional probabilities for each evidence variable, given its parents. Events in which the actual evidence appears unlikely should be given less weight.

Likelihood Weighting in Bayesian Networks

```

function LIKELIHOOD-WEIGHTING( $X, e, bn, N$ ) returns an estimate of  $P(X|e)$ 
  inputs:  $X$ , the query variable
     $e$ , observed values for variables  $E$ 
     $bn$ , a Bayesian network specifying joint distribution  $P(X_1, \dots, X_n)$ 
     $N$ , the total number of samples to be generated
  local variables:  $\mathbf{W}$ , a vector of weighted counts for each value of  $X$ , initially zero
  for  $j = 1$  to  $N$  do
     $\mathbf{x}, w \leftarrow \text{WEIGHTED-SAMPLE}(bn, e)$ 
     $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$  where  $x$  is the value of  $X$  in  $\mathbf{x}$ 
  return NORMALIZE( $\mathbf{W}$ )

```

```

function WEIGHTED-SAMPLE( $bn, e$ ) returns an event and a weight
   $w \leftarrow 1; \mathbf{x} \leftarrow \text{an event with } n \text{ elements initialized from } e$ 
  foreach variable  $X_i$  in  $X_1, \dots, X_n$  do
    if  $X_i$  is an evidence variable with value  $x_i$  in  $e$ 
      then  $w \leftarrow w \times P(X_i = x_i | \text{parents}(X_i))$ 
    else  $\mathbf{x}[i] \leftarrow \text{a random sample from } P(X_i | \text{parents}(X_i))$ 
  return  $\mathbf{x}, w$ 

```

Likelihood Weighting - Example

- Assume Query is $P(\text{Rain} / \text{Cloudy} = \text{true}, \text{WetGrass} = \text{true})$ and the ordering is $[\text{Cloudy}, \text{Sprinkler}, \text{Rain}, \text{WetGrass}]$.
- First, the weight w is set to 1.0.
- Then an event is generated:

- Cloudy is an evidence variable with value true. Therefore,

$$w \leftarrow w \times P(\text{Cloudy} = \text{true}) = 0.5$$

- Sprinkler is not an evidence variable, so sample from

$$P(\text{Sprinkler} / \text{Cloudy} = \text{true}) = <0.1, 0.9>; \text{ suppose this returns } \text{false.}$$

- Sample from $P(\text{Rain} / \text{Cloudy} = \text{true}) = <0.8, 0.2>$; suppose this returns true.

- WetGrass is an evidence variable with value true. Therefore,

$$w \leftarrow w \times P(\text{WetGrass} = \text{true} / \text{Sprinkler} = \text{false}, \text{Rain} = \text{true}) = 0.45. = 0.5 \times 0.9$$

- WEIGHTED-SAMPLE returns the event $[\text{true}, \text{false}, \text{true}, \text{true}]$ with weight 0.45, and this is tallied under Rain = true.

$$\frac{1}{M} \sum_{i=1}^M f(x_i) \underbrace{p(x_i)}_{\text{Probability}} \Delta x; M$$

$$\int f(x) p(x) dx$$

Some theory about Importance Sampling

We would like to estimate the expectation of some function f with respect to a distribution $p(x)$

$$E_p(f) = \frac{1}{M} \sum_{m=1}^M f(x_m)$$

What if p is not known or is very difficult to sample from?

We use another distribution q and adjust for it

Some theory about Importance Sampling

$$E_{P(x)}[f(x)] = E_{Q(x)} \left[f(x) \frac{P(x)}{Q(x)} \right] = \int f(x) \frac{P(x)}{Q(x)} Q(x) dx$$

We use the standard estimator for expectations relative to Q . Let $D = \{x[1], x[2], \dots, x[M]\}$ be a set of samples drawn from Q

$$\hat{E}_D(f) = \frac{1}{M} \sum_{m=1}^M f(x[m]) \frac{P(x[m])}{Q(x[m])}$$

Some theory about Importance Sampling

This is the unnormalized Importance Sampling estimator

Proposition 12.1: For data sets D / sampled from ϕ
 we have $E_D \left(\hat{E}_D(F) \right) = E_{\phi(x)} [f(x) w(x)] = E_{P(x)} [f(x)]$

Proof: $E_P \left[\frac{1}{M} \sum_{m=1}^M f(x[m]) \frac{P(x[m])}{Q(x[m])} \right] = \frac{1}{M} \sum_{m=1}^M E \left(f(x[m]) \frac{P(x[m])}{Q(x[m])} \right)$

Some theory on importance sampling

We need to take the expectation over the joint distribution on M-samples. The probability of selecting a particular sample $x^c[1], x^c[2] \dots x^c[M]$ is $\prod_{i=1}^M Q(x^c[i])$ and therefore

$$E_D(\hat{D}(f)) = \frac{1}{M} \sum_{x^c[1]} \sum_{x^c[2]} \dots \sum_{x^c[M]} \left(\frac{f(x^c[1])P(x^c[1])}{Q(x^c[1])} + \dots + \frac{f(x^c[M])P(x^c[M])}{Q(x^c[M])} \right) \prod_{i=1}^M Q(x^c[i])$$

We can rewrite this expression as in the next slide

Some theory on importance Sampling

$$\begin{aligned}
 &= \frac{1}{M} \sum_{x[1]} \sum_{x[2]} \dots \sum_{x[M]} f(x[i]) P(x[i]) \prod_{i=1, i \neq 1}^M Q(x[i]) \\
 &\quad + \frac{1}{M} \sum_{x[1]} \sum_{x[2]} \dots \sum_{x[M]} f(x[2]) P(x[2]) \prod_{i=1, i \neq 2}^M Q(x[i]) \\
 &\quad + \dots + \frac{1}{M} \sum_{x[1]} \sum_{x[2]} \dots \sum_{x[M]} f(x[M]) P(x[M]) \prod_{i=1, i \neq M}^M Q(x[i])
 \end{aligned}$$

There are M terms in the above sum which are all very similar \rightarrow they each evaluate to the same quantity.

Some theory on importance sampling

Each term can be simplified to a term like the following

$$\left[\sum_{x[1]} \frac{1}{M} f(x[1]) p(x[1]) \right] \underbrace{\left[\sum_{x[2]} \sum_{x[3]} \dots \sum_{x[M]} \prod_{i=2}^M Q(x[i]) \right]}$$

this evaluates to 1

We are left with $\sum_{x[1]} \frac{1}{M} f(x[1]) p(x[1]) = \frac{1}{M} E_p(x)(f(x))$

Note: this derivation is strictly valid for discrete distributions

Thus we have $E_p(x)(f(x))$ overall

$$\text{Var}(X+Y) = \text{Var}(X) + \text{Var}(Y)$$

$$\text{Var}(kX) = k^2 \text{Var}(X)$$



Some theory about Importance Sampling

Letting $\hat{E}_D[f] = \hat{E}_D[f]$ we see that

the distribution of $\hat{E}_D[f]$ is $N(0, \frac{\sigma_Q^2}{M})$ where

$$\sigma_Q^2 = E_Q(x) [(f(x)w(x))^2] - E_Q(x) [f(x)w(x)]^2 = \text{Var}(f(x)w(x))$$

Why is this true?

$$\sigma_Q^2 = \text{Var}(\hat{E}_D(f)) = \frac{1}{M^2} M \cdot \text{Var}[f(x)w(x)] \text{ since we}$$

$\hat{E}_D(f)$ is the average of M independent samples $f(x_i)w(x_i)$

$$\sum_x P_B(x, e) = P_B(e) \neq 1$$



Normalized Importance Sampling

We do not have access to a distribution $P(x)$, only $\tilde{P}(x)$ which is not normalized

$$Z P(x) = \tilde{P}(x) \text{ where } Z \text{ is a normalizing constant}$$

In a Bayesian network $P(x)$ could be the posterior distribution $P_B(x|e)$ and $\tilde{P}(x)$ could be $P_B(x, e)$

Let us define $\omega(x) = \frac{\tilde{P}(x)}{q(x)}$

$$\begin{aligned} \omega(x) &= \frac{P(e) P_B(x|e)}{q(x)} \\ &= P_B(x, e) \end{aligned}$$

Normalized Importance Sampling

$$E_Q[\omega(x)] = \int \frac{\tilde{P}(x)}{Q(x)} Q(x) dx \quad \text{or} \quad \sum \frac{\tilde{P}(x)}{Q(x)} Q(x)$$

$$= \int \tilde{P}(x) dx \quad \text{or} \quad \sum \tilde{P}(x) \rightarrow \text{both are equal to } z$$

$$\begin{aligned} E_{P(x)}[f(x)] &= \sum_x f(x) P(x) = \sum_x f(x) \frac{P(x)}{Q(x)} Q(x) \\ &= \frac{1}{z} \sum_x \frac{f(x) \tilde{P}(x)}{Q(x)} Q(x) = \frac{1}{z} \underbrace{E_{Q(x)}[f(x) \omega(x)]}_{\boxed{\quad}} \end{aligned}$$

$$w(x) = \frac{p(x)}{q(x)}$$

Normalized Importance Sampling

$$E_p(x)[f(x)] = \sum_{m=1}^M \frac{f(x[m]) w(x[m])}{\sum_{m=1}^M w(x[m])} \rightarrow ?$$

where

$x[1], x[2], \dots, x[m]$ are all samples drawn

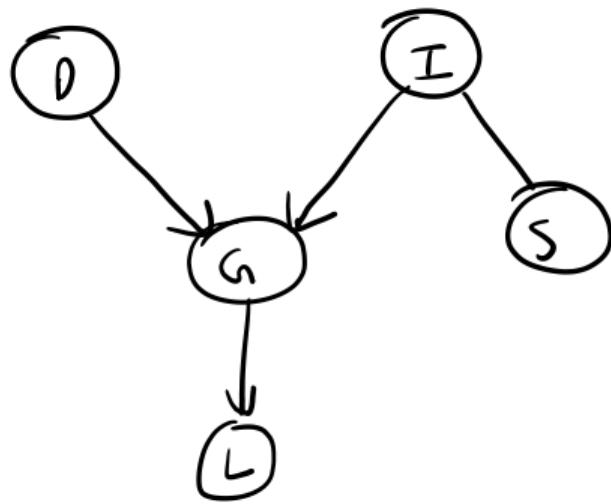
from $q(x)$

The bias of the normalized importance sampling estimator is not zero

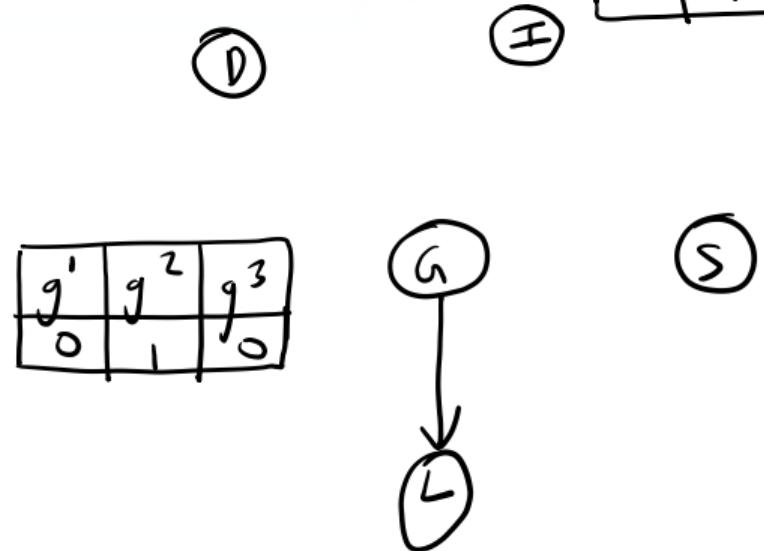
$\text{G}_1 \rightarrow \text{I}$



Link to Likelihood Weighting



Original network B



Mutilated Network $B_{z=3}$

link to Likelihood Weighting

Let ξ be a sample generated by the Likelihood weighting algorithm and let w be its weight. Then the distribution over ξ is as defined by the network

$$\text{B}_{Z=2} \quad \text{and} \quad w(\xi) = \frac{p_B(\xi)}{p_{B_{Z=2}}(\xi)} = \frac{\tilde{p}}{q}$$

Proof: Note that here $\underline{p}_B \rightarrow \tilde{p}$ in the previous slides
 and $p_{B_{Z=2}}(\xi) \rightarrow q$

Link to Likelihood Weighting

Proof Continued

The only difference between sampling from $P_{B_{Z=2}}(\xi)$ and $P_B(\xi)$ concerns what happens when we draw a sample $z_i \in \text{Evidence}_{\text{Var}(y_i/\text{parent}(z_i))}$

In $P_{B_{Z=2}}(\xi)$, we draw this sample z_i with probability

$= 1$
In $P_B(\xi)$ we draw this sample with probability $P_B(z_i/\text{parent}(z_i))$

Link to Likelihood weighting

The weight in the Algorithm gets multiplied by

$$\frac{P_B(z_i | \text{parent}(z_i))}{1}$$

when z_i is not in evidence, $P_B(z_i | \text{parent}(z_i)) = P_{B_{z=z}}(z_i | \text{parent}(z_i))$

so we have $\frac{1}{1}$.

Multiplying the weights together, we get $\omega(\xi) = \frac{P_B(\xi)}{P_{B_{z=z}}(\xi)}$

Sampling-based Approximate Methods

- The methods using instantiations are generally known as **particle-based methods**.
- Each instantiation is known as a **particle**.
- Either create particles using a deterministic process, or sample particles from some distribution.
- Complete assignments to all of the network variables is commonly known as **full particles**.
- Disadvantage of full particle is that each particle covers only a very small part of the space.
- A collapsed particle specifies an assignment w only to some subset of the variables W , associating with it the conditional distribution $P(X/w)$.
- Assignments only to a subset $P(X/w)$ of variables of the network representing the conditional probability are commonly known as **collapsed particles**.



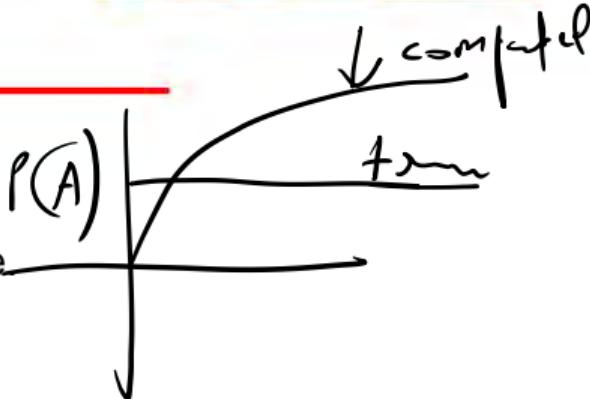
Table of Contents

1 Approximate Inference

2 Propagation-Based Approximation

3 Markov Chain Monte Carlo Simulation

Propagation-Based Approximation



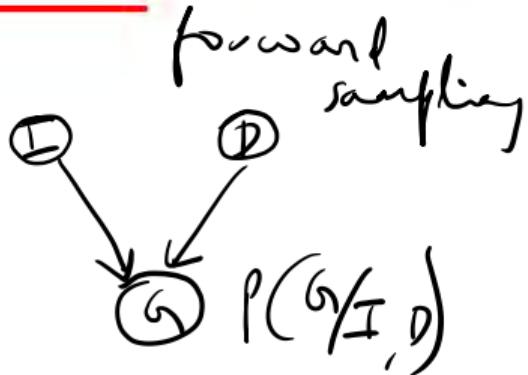
- Use the same message propagation as in exact inference.
- Use Cluster graph instead of clique tree.
- Use **Loopy belief propagation.**
- Message propagation process may not converge in two passes, since information from one pass will circulate and affect the next round.
- In some cases, the propagation of beliefs may not converge at all.
- Use Bethe cluster graph to eliminate the loops.

Table of Contents

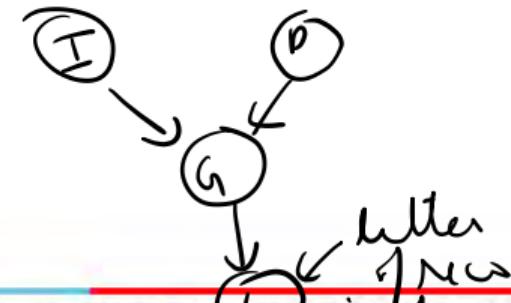
1 Approximate Inference

2 Propagation-Based Approximation

3 Markov Chain Monte Carlo Simulation



$$P(G) = P(G|I)$$



Markov Chain Monte Carlo Methods

- Generate a **sequence** of samples.
- Instead of generating each sample from scratch, MCMC algorithms generate each sample by making a random change to the preceding sample.
- Think of an MCMC algorithm as being in a particular **current state** specifying a value for every variable and generating a **next state** by making random changes to the current state.
- Markov chain methods apply equally well to directed and to undirected models.
- Works for both Bayesian and Markov networks.
- A particular form of MCMC is called **Gibbs sampling**, which is especially well suited for Bayesian networks.



Gibbs Sampling Algorithm

$$p_b(x|e) \rightarrow \text{posterior distribution}$$

lead

- Starts out by generating a sample of the unobserved variables from some initial distribution.
- Starting from initial sample, iterate over each of the unobserved variables, sampling a new value for each variable given our current sample for all other variables.
- This allows information to **flow** across the network as each variable is sampled.
- The Gibbs sampling algorithm for Bayesian networks starts with an arbitrary state (with the evidence variables fixed at their observed values) and generates a next state by randomly sampling a value for one of the nonevidence variables X_i .
- The sampling for X_i is done conditioned on the current values of the variables in the Markov blanket of X_i .
- The algorithm therefore wanders randomly around the state space flipping one variable at a time, but keeping the evidence variables fixed.

Gibbs Sampling Algorithm

function GIBBS-ASK(X, \mathbf{e}, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$

local variables: \mathbf{N} , a vector of counts for each value of X , initially zero
 \mathbf{Z} , the nonevidence variables in bn
 \mathbf{x} , the current state of the network, initially copied from \mathbf{e}

initialize \mathbf{x} with random values for the variables in \mathbf{Z}

for $j = 1$ to N **do**

for each Z_i in \mathbf{Z} **do**

$P(Z_i / \text{all other variables})$

set the value of Z_i in \mathbf{x} by sampling from $P(Z_i | mb(Z_i))$

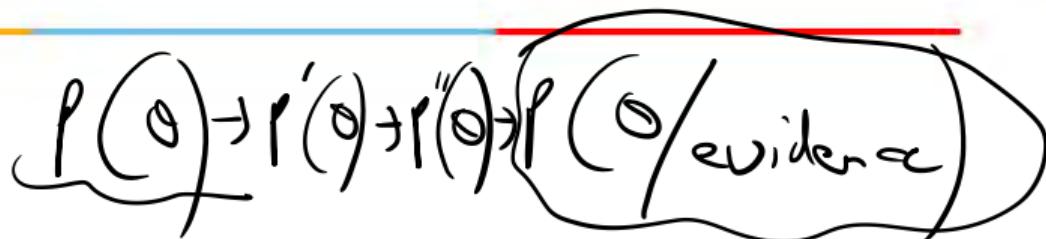
$\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in \mathbf{x}

return NORMALIZE(\mathbf{N})

Gibbs Sampling - Example

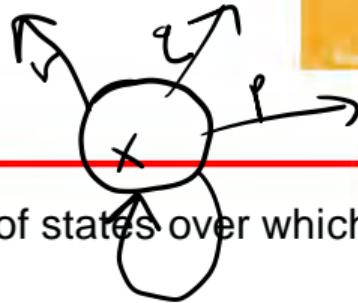
- Assume Query is $P(\text{Rain} | \text{Cloudy} = \text{true}, \text{WetGrass} = \text{true})$ and the ordering is $[\text{Cloudy}, \text{Sprinkler}, \text{Rain}, \text{WetGrass}]$.
- The evidence variables *Sprinkler* and *WetGrass* are fixed to their observed values,
- The nonevidence variables *Cloudy* and *Rain* are initialized randomly; say to *true* and *false* respectively. The initial state is $[\text{true}, \text{true}, \text{false}, \text{true}]$.
- The nonevidence variables are sampled repeatedly in an arbitrary order.
 - ① *Cloudy* is sampled from $P(\text{Cloudy} | \text{Sprinkler} = \text{true}, \text{Rain} = \text{false})$. Suppose the result is *Cloudy = false*. Then the new current state is $[\text{false}, \text{true}, \text{false}, \text{true}]$. new sample
 - ② *Rain* is sampled from $P(\text{Rain} | \text{Cloudy} = \text{false}, \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$. Suppose this yields *Rain = true*. The new current state is $[\text{false}, \text{true}, \text{true}, \text{true}]$. new
- Each state visited during this process is a sample that contributes to the estimate for the query variable *Rain*.
- If the process visits 20 states where *Rain is true* and 60 states where *Rain is false*, then $\text{Normalize}(< 20, 60 >) = < 0.25, 0.75 >$.

Markov Chain Monte Carlo (MCMC)



- MCMC framework generates samples from the posterior distribution, where we cannot efficiently sample from the posterior directly.
- Construct an iterative process that gradually samples from distributions that are closer and closer to the posterior.
- The number of iterations is to be determined.

Markov Chain



- A Markov chain is defined in terms of a graph of states over which the sampling algorithm takes a random walk.

Definition

Markov Chain defines a probabilistic transition model $T(x \rightarrow x^j)$ over a state x

$$\forall x : \sum_{x^j} T(x \rightarrow x^j) = 1$$

The **transition model** T specifies for each pair of state x, x^j the probability $T(x \rightarrow x^j)$ of going from x to x^j . This transition probability applies whenever the chain is in state x .

- MCMC is defined on **homogeneous** systems, where the system dynamics do not change over time.

$$P(x_j \text{ at time } t+1) \leq P(x \text{ at } t \& x_j \text{ at time } t+1)$$

Markov Chain

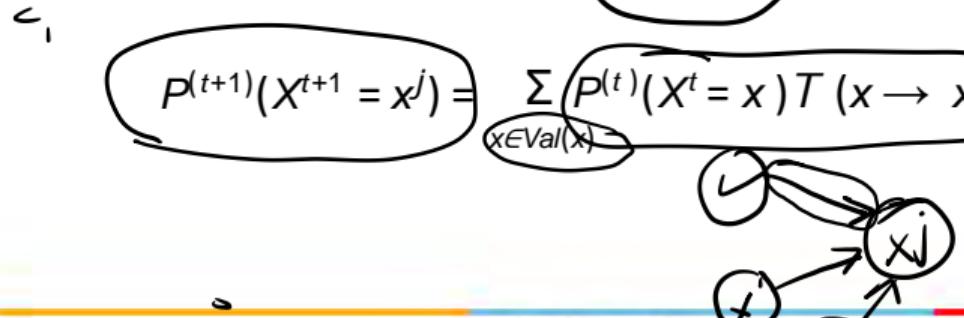


lead

$$\sum P\left(\frac{x_j}{x \text{ at time } t}\right) = 1$$

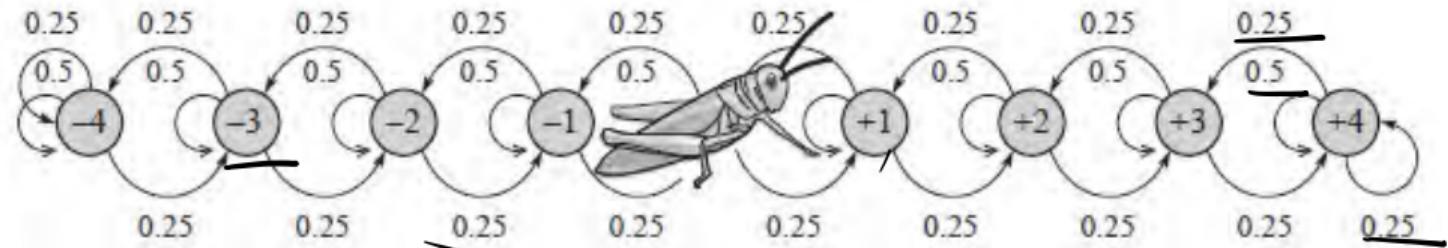
- Visualize the state space as a graph, with probability-weighted directed edges corresponding to transitions between different states.
- Generate a random sequence of states $x^{(0)}, x^{(1)}, x^{(2)}, \dots$
- The state of the process at step t is random variable $X^{(t)}$.
- Initial state $X^{(0)}$ is distributed according to some initial state distribution $P^{(0)}(X^{(0)})$.
- Subsequent states are defined by distributions $P^{(1)}(X^{(1)}), P^{(2)}(X^{(2)}), \dots$

$$P^{(t+1)}(X^{t+1} = x^j) = \sum_{x \in Val(X)} P^{(t)}(X^t = x) T(x \rightarrow x^j)$$



Grasshopper Example

$p^t \rightarrow$ probability dist over
all states at time t



Time	-2	-1	0	1	2
$P^{(0)}$	0	0	1	0	0
$P^{(1)}$	0	<u>0.25</u>	<u>0.5</u>	<u>0.25</u>	0
$P^{(2)}$	<u>0.25^2</u>	<u>$2 * 0.5 * 0.25$</u>	<u>$0.5^2 + 2 * 0.25^2$</u>	<u>$2 * 0.5 * 0.25^2$</u>	<u>0.25^2</u>
	$=0.0625$	$=0.25$	$=0.375$	$=0.25$	$=0.0625$

Grasshopper Example

$$P^2(-1) = P'(-1)T(-1 \rightarrow -1) + P'(1)T(1 \rightarrow -1) + P'(0)T(0 \rightarrow -1)$$

(all other probabilities in P' are 0)

$$\text{Now } T(1 \rightarrow -1) = 0$$

$$\begin{aligned} \therefore P^2(-1) &= (0.5)(0.25) + (0.5)(0.25) \\ &= \underline{\underline{0.25}} \end{aligned}$$

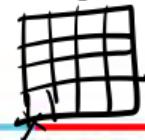
Markov Chain Monte Carlo (MCMC)

- Markov chain Monte carlo (MCMC) sampling is a process that mirrors the dynamics of the Markov chain.
- The sample $X^{(t)}$ is drawn from the distribution $P^{(t)}$.
- Find out whether $P^{(t)}$ converges, and if so, to what limit.

$$P^t(X^t = x)$$

$$P^{(t+1)}(X^{t+1} = x^j) = \sum_{x \in Val(x)} P^{(t)}(X^t = x) T(x \rightarrow x^j)$$

- Apply on uniform distribution.
- Expected time required for a chain to reach boundaries of interval $[-K, K]$ is K^2 steps.



Markov Chain Monte Carlo (MCMC)

Algorithm 12.5 Generating a Markov chain trajectory

Procedure MCMC-Sample (

$P^{(0)}(\mathbf{X})$, // Initial state distribution

\mathcal{T} , // Markov chain transition model

T // Number of time steps

)

1 Sample $\mathbf{x}^{(0)}$ from $P^{(0)}(\mathbf{X})$

2 **for** $t = 1, \dots, T$

3 Sample $\mathbf{x}^{(t)}$ from $\mathcal{T}(\mathbf{x}^{(t-1)} \rightarrow \mathbf{X})$

4 **return** $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(T)}$

Stationary Distributions

- As MCMC process converges, $P^{(t+1)}$ will be close to $P^{(t)}$.

$$P^{(t)}(x^j) \approx P^{(t+1)}(X^{t+1} = x^j) = \sum_{x \in \text{Val}(x)} P^{(t)}(X^t = x)^T (x \rightarrow x^j)$$

The resulting distribution is $\pi(x)$.

- At equilibrium, for any state x^j , $P^{(t+1)}$ will be almost equal to $P^{(t)}$.
- Example : Grasshopper Markov Chain

Stationary Distributions

Definition (Stationary Distribution)

A distribution $\pi(x)$ is a stationary distribution for a Markov chain T if it satisfies

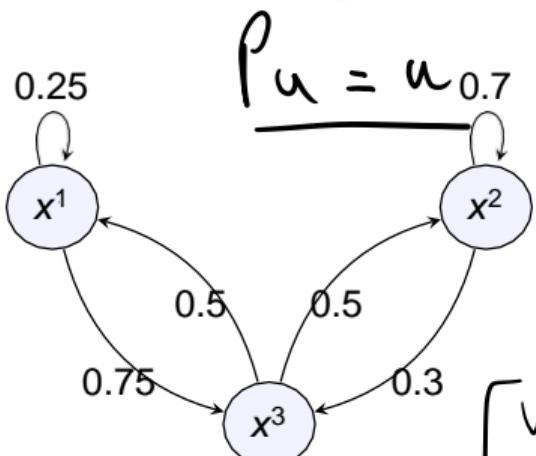
$$\pi(X = x^j) = \sum_{x \in Val(x)} \pi(X = x) T(x \rightarrow x^j)$$

A stationary distribution is also called an invariant distribution.

Stationary Distribution - Example

$\lambda = 1$ is an eigenvalue
of the transition matrix

$$u_{k+1} = P u_k$$



$$P u = u$$

- Less uniform distribution.

$$u_1 \text{ to be an eigenvector of } \lambda \text{ of the transition matrix}$$

$$\begin{aligned}\Pi(x^1) &= 0.25\Pi(x^1) + 0.5\Pi(x^3) \\ \Pi(x^2) &= 0.7\Pi(x^2) + 0.5\Pi(x^3) \\ \Pi(x^3) &= 0.75\Pi(x^1) + 0.3\Pi(x^2)\end{aligned}$$

$$\begin{aligned}P^2 u &= P(P u) = P u = u \\ \Pi(x^1) + \Pi(x^2) + \Pi(x^3) &= 1\end{aligned}$$

- Unique solution

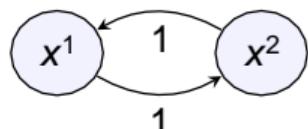
$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0.25 & 0 & 0.5 \\ 0 & 0.7 & 0.5 \\ 0.75 & 0.3 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$\begin{bmatrix} 0.25 & 0 & 0.5 \\ 0 & 0.7 & 0.5 \\ 0.75 & 0.3 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Periodic Markov Chains

$$P^t(x^1) = 0 \text{ if } t \text{ is even}$$

$$P^t(x^2) = 1 \text{ if } t \text{ is odd}$$



$$P^t(x^1) = 1 \text{ if } t \text{ is even}$$

$$P^t(x^2) = 0 \text{ if } t \text{ is odd}$$

Markov Chain with two states x^1 and x^2

$$T(x^1 \rightarrow x^2) = 1$$

$$T(x^2 \rightarrow x^1) = 1$$

Let $P^{(0)}(x^1) = 1$

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

$$P^{(t)}(x^1) = 1 \text{ if } t \text{ is even}$$

$$P^{(t)}(x^2) = 1 \text{ if } t \text{ is odd}$$

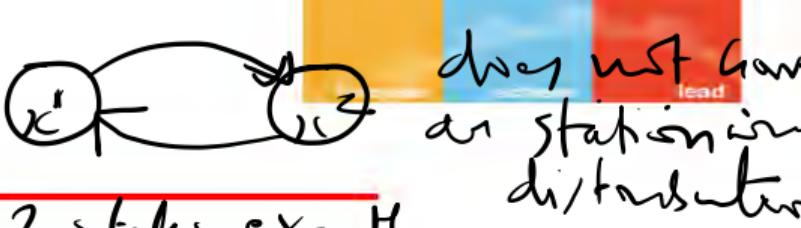
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

No convergence. So not a stationary distribution.

Markov chains that exhibit a fixed cyclic behavior, are called **periodic Markov chains**.

$$= \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

Regular Markov Chains



$x^1 \rightarrow x^2$ in 2 steps exactly

$x^1 \rightarrow x^2$ in 2 steps exactly → no

- A Markov chain is said to be regular if there exists some number k such that for every $x, x^j \in Val(x)$, the probability of getting from x to x^j in exactly k steps is >0 .
- Example: Grasshopper Markov chain $\rightarrow k=9$.

$= k, \text{ not } \leq k$

Theorem

If a finite state Markov chain T is regular, then it has a unique stationary distribution.

- Two conditions that together guarantee regularity.
 - Every state are connected. x^j can be reached from x with a probability > 0 .
 - For each state, there is a self-transition.

Using Markov Chains

$$\frac{\sum_{i=1}^M f(x[i])}{M} = E_p[f(x)]$$

how long should we wait
before we can draw
sample?

- Goal: Compute $P(x \in S)$
 -) P is too hard to sample from directly.
- Construct a regular Markov Chain T whose unique stationary distribution is P .
- Sample $x^{(0)}$ from some arbitrary distribution Q .
- For $t = 0, 1, 2, \dots$
 -) Generate $x^{(t+1)}$ from $T(x^{(t)} \rightarrow x^t)$.

$$x^0 \rightarrow x^1 \rightarrow x^2 \rightarrow \dots$$

transition matrix at what point can we
be sure that we are
Sampling from the stationary dist

Using Samples

$$\rho^{t+1} \approx \rho^t$$

[probability distributions
do not change with
time]

- Once the MCMC finds equilibrium, all samples $x^{(t)}$ are from stationary distribution $\pi(x)$.
- Collect and use samples $x^{(t)}$.
- Ideally collect every 100^{th} sample, as nearby samples are correlated.

here the correlations becomes very small



MCMC Algorithm I

- we are talking about c different Markov chains
- For $c = 1, 2, \dots, C$
 -) Sample $x^{(0)}$ from arbitrary distribution Q (or Gibbs distribution can be used).
 - Repeat until equilibrium
 -) For $c = 1, 2, \dots, C$
 - ↳ Generate $x^{(c,t+1)}$ from $T(x^{(c,t)} \rightarrow x^i)$.
 -) Check for equilibrium
 -) $t := t + 1$

MCMC Algorithm II

- Repeat until sufficient samples

-) Dataset $D := \text{empty}$
-) For $c = 1, 2, \dots, C$ *We have C different Markov chains*
 - ↳ Generate $x^{(c,t+1)}$ from $T(x^{(c,t)} \rightarrow x^i)$.
 - ↳ $D := D \cup x^{(c,t+1)}$
-) $t := t + 1$

- Compute Expectation

-) Let $D := \{x[1], x[2], \dots, x[m]\}$
-) Estimate expectation

$$\hat{E}_P[f] = \frac{1}{M} \sum_{i=1}^M f(x[m])$$



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 13: LEARNING

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



TABLE OF CONTENTS

1 LEARNING

2 PARAMETER ESTIMATION IN BAYESIAN NETWORKS

3 MLE FOR BAYESIAN NETWORKS

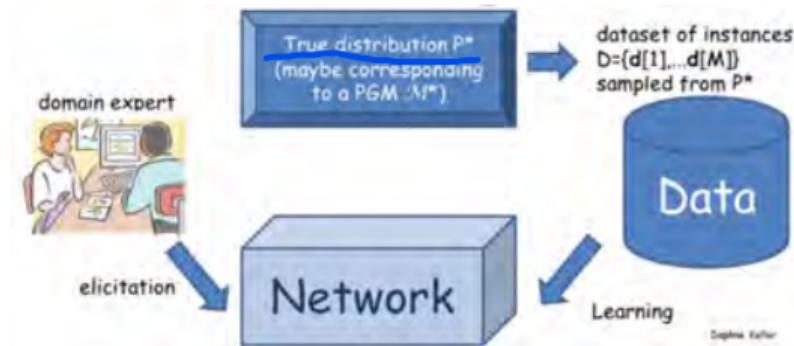
4 BAYESIAN PARAMETER ESTIMATION

MOTIVATION

- For Inference or predictions, the starting point was the PGM. The structure and parameters were part of the input.
- How to acquire a model?
 - ▶ Construct the network by hand, with the help of an expert – “manual” network construction
 - ▶ Learn a model using a set of examples generated from the distribution we wish to model.
- Predictions of structured objects; sequences, graphs, trees
- Incorporate prior knowledge into model.
- Learning a single model for multiple tasks.
- Framework for knowledge discovery.

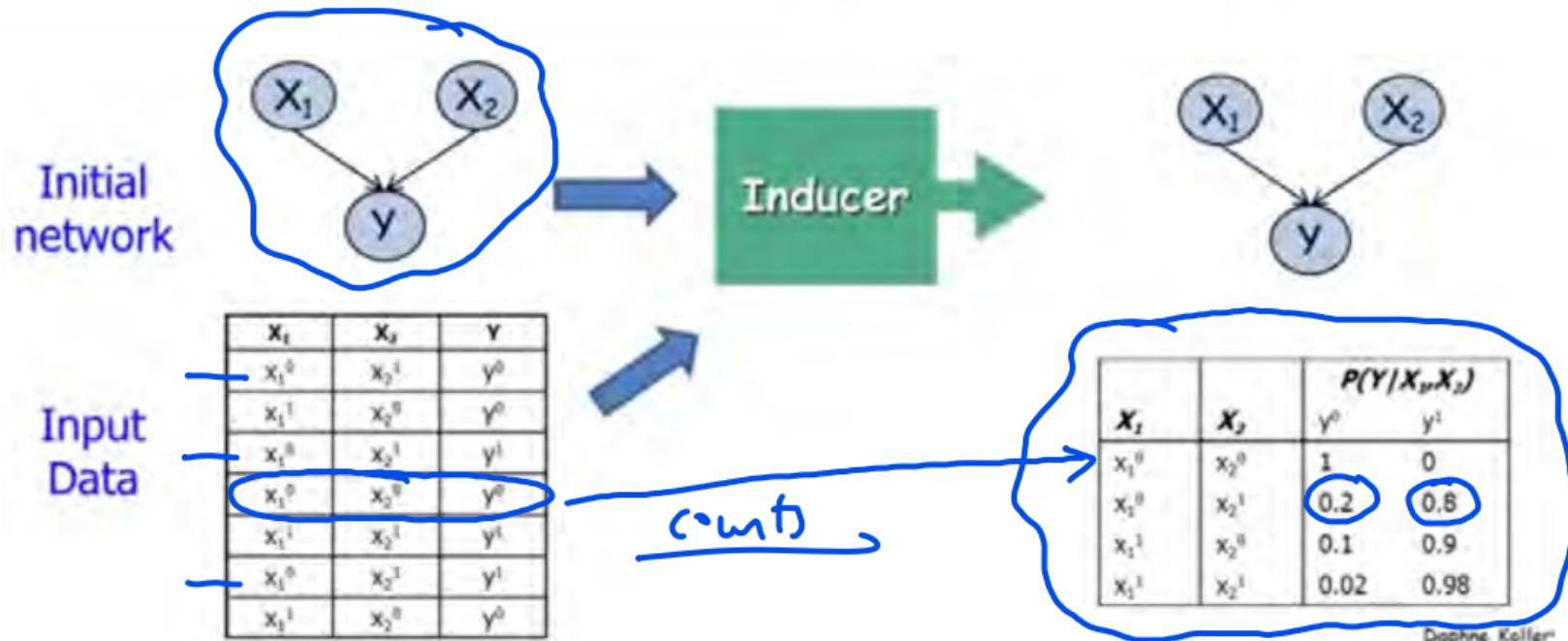
MODEL LEARNING

- The task of constructing a model from a set of instances is generally called **model learning**.
- Goal: Learn a model \tilde{M} from a family of models that defines a distribution $P_{\tilde{M}}$.



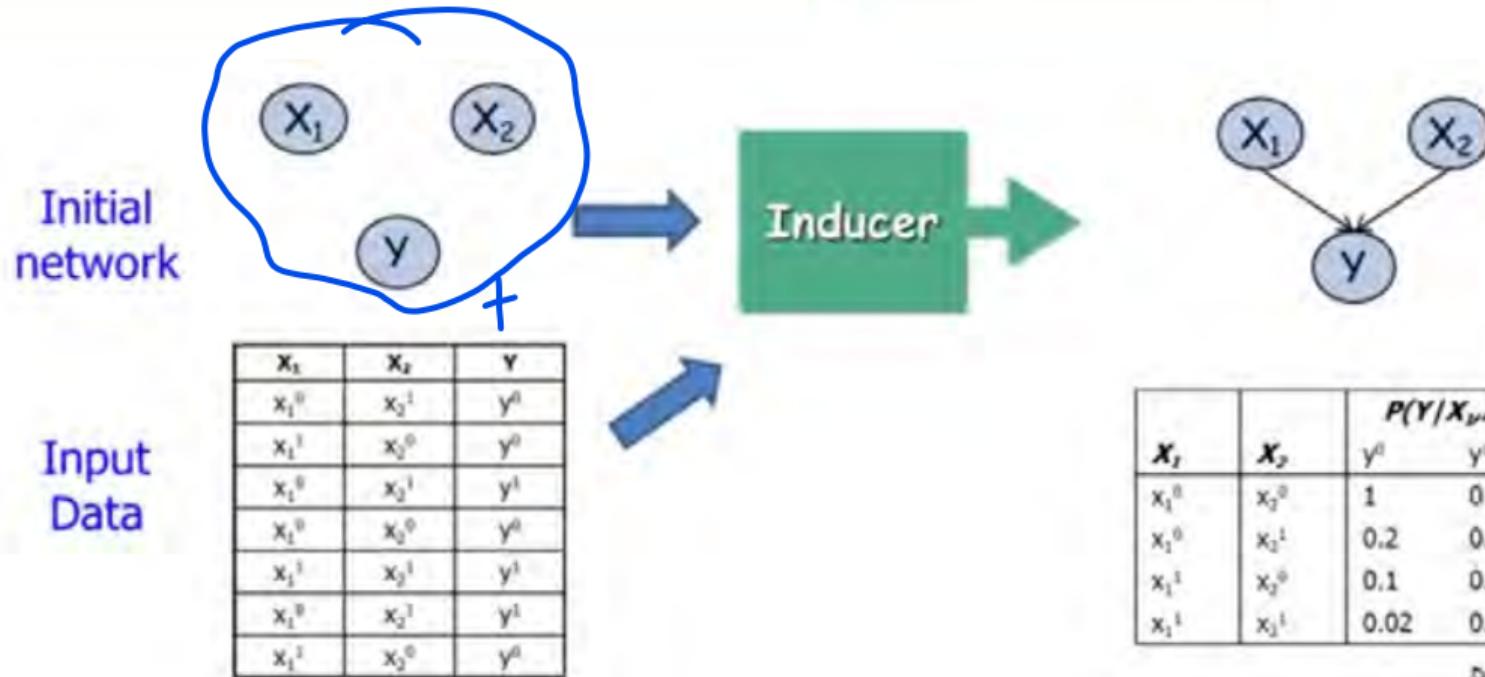
- Amount of data is insufficient, esp for high dimensional distributions. Select \tilde{M} so as to construct the “best” approximation to M^* .

KNOWN STRUCTURE COMPLETE DATA



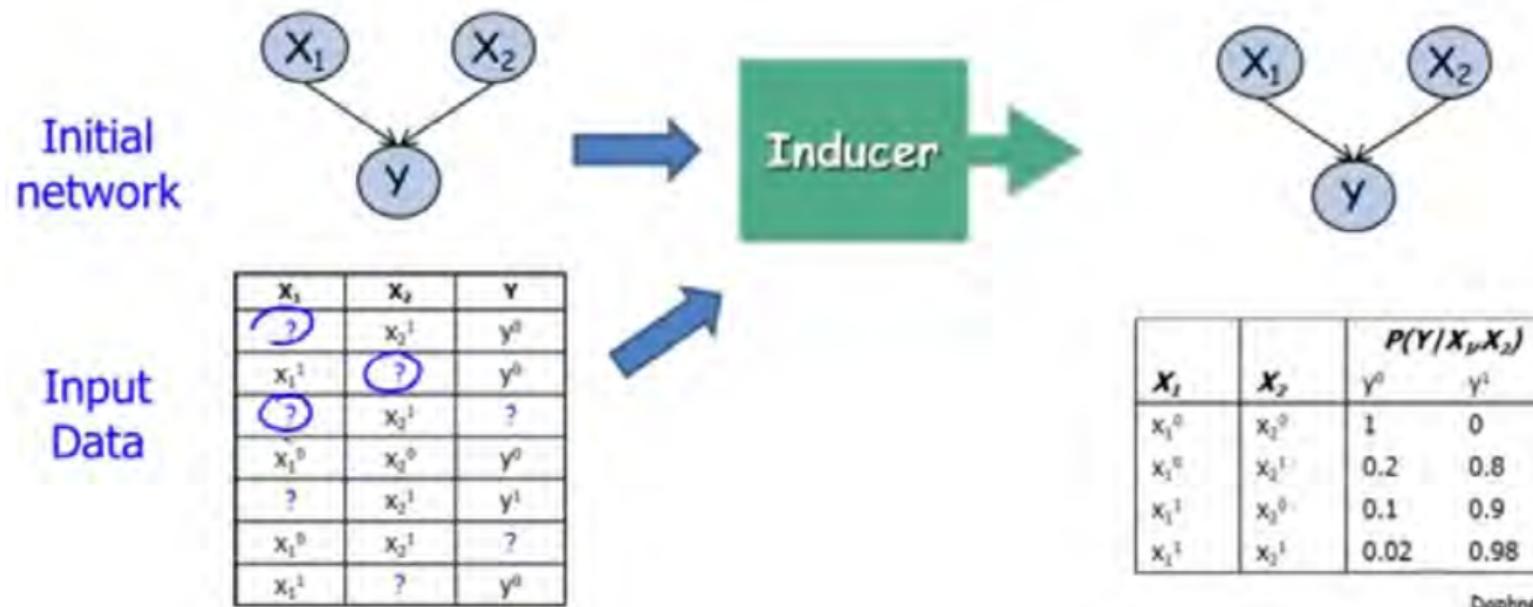
Daphne Koller

UNKNOWN STRUCTURE COMPLETE DATA

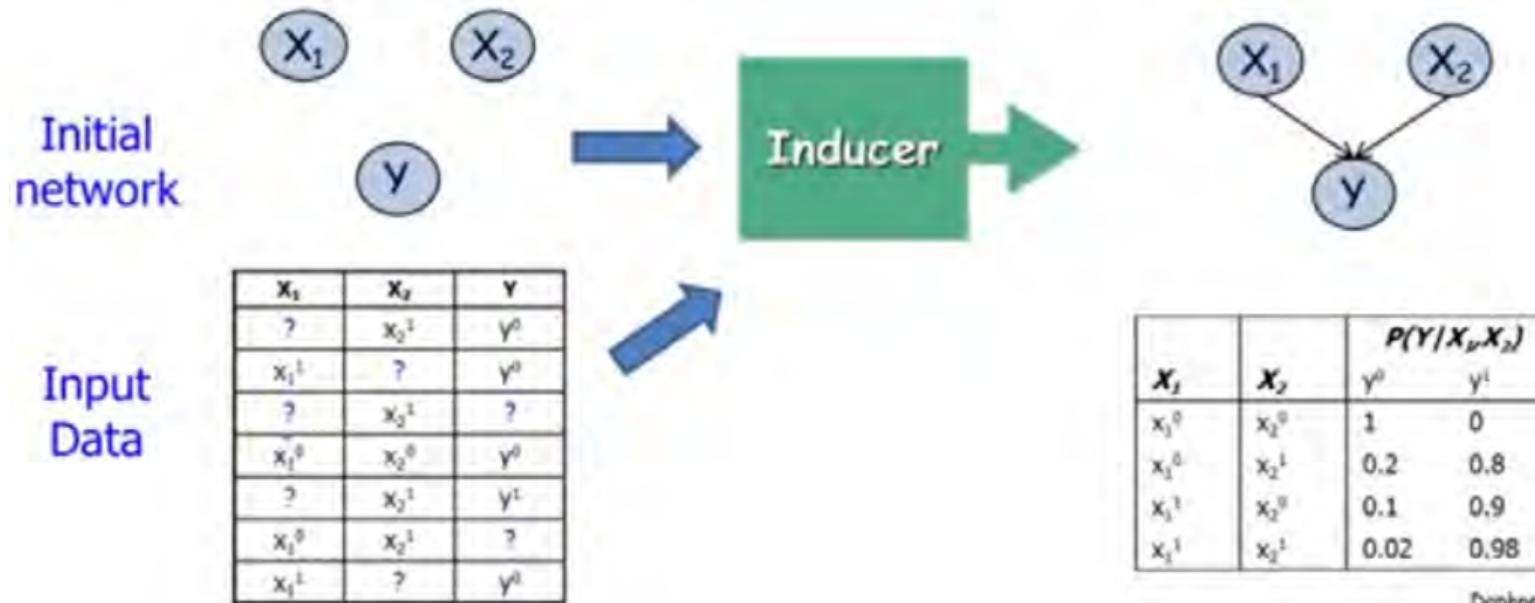


Daphne Koller

KNOWN STRUCTURE INCOMPLETE DATA

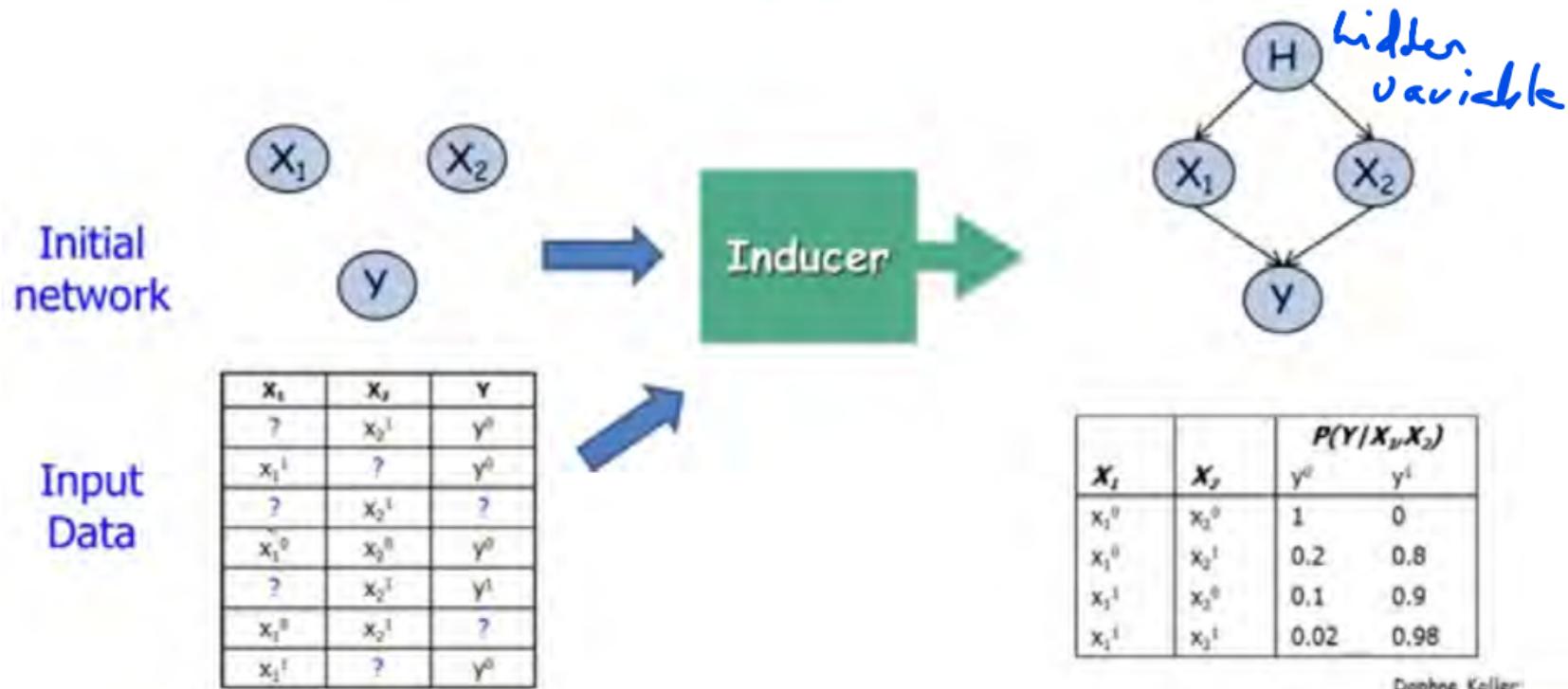


UNKNOWN STRUCTURE INCOMPLETE DATA

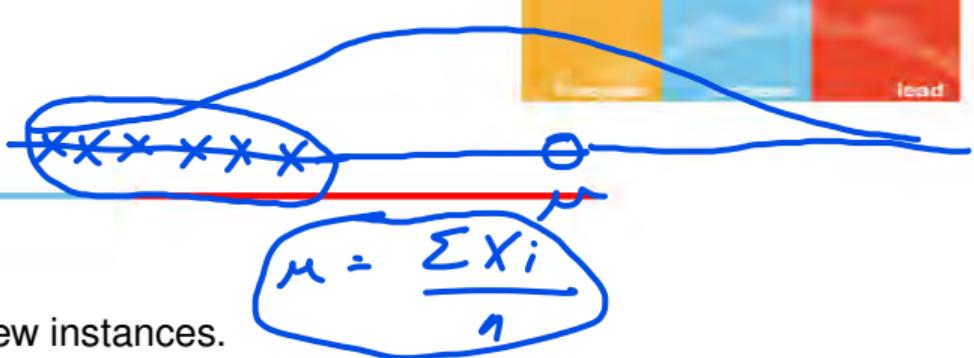


Daphne Koller

LATENT (HIDDEN) VARIABLES INCOMPLETE DATA



GOALS OF LEARNING I



- Density estimation:
 - ▶ Answer general queries about new instances.
 - ▶ Metric: Training set likelihood

$$P(\mathcal{D} : \mathcal{M}) = \prod_m P(d[m] : \mathcal{M})$$

- ▶ Care about new data: generalization performance – Evaluate on test set likelihood $P(\mathcal{D}' : \mathcal{M})$
- ▶ Minimize the expected loss :

$$\mathbb{E}_{\mathcal{D}}[\text{loss}(d : \mathcal{M})] = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \text{loss}(d : \mathcal{M})$$

GOALS OF LEARNING II

- Prediction or Classification Task:

- Specific prediction task on new instances.
- Select a MAP assignment to predict a set of variables \mathbf{Y} , given a set of observed variables \mathbf{X} .
 $\arg\max_{\mathbf{Y}} P(\mathbf{Y} | \mathbf{e})$
- Eg: Text document classification, Image segmentation, speech recognition
- Select model to optimize

★ likelihood

$$\prod_m P(d[m] : \mathcal{M})$$

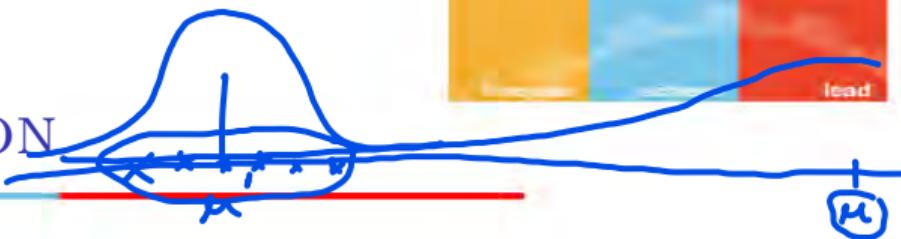
★ conditional likelihood

$$\prod_m P(y[m] | x[m] : \mathcal{M})$$

GOALS OF LEARNING III

- Knowledge Discovery of \mathcal{M}^* :
 - ▶ Discover knowledge about P^* .
 - ▶ Distinguish direct and indirect dependencies.
 - ▶ Possibly directionality of edges.
 - ▶ Presence and location of hidden variables.
 - ▶ Reconstruct correct model \mathcal{M}^* .
 - ▶ Measure the success in terms of the model = differences between \mathcal{M}^* and $\tilde{\mathcal{M}}$.

LEARNING AS OPTIMIZATION



- **Hypothesis space** – a set of candidate models.
- **Objective function** – a criterion for quantifying our preference for different models.
- **Learning Task** – find a high-scoring model within the hypothesis space.
- Use data \mathcal{D} to define an **empirical distribution** $\hat{P}_{\mathcal{D}}$

$$\hat{P}_{\mathcal{D}}(A) = \frac{1}{M} \sum_m \mathbf{1}\{d[m] \in A\}$$

indicator Function

count

- The probability of the event A is simply the fraction of training examples that satisfy A .
- Use of the empirical log-loss (or log-likelihood) as the objective.
- This type of objective tends to over-fit the learned model to the training data . Use regularization, cross-validation techniques.

GENERATIVE TRAINING

- Perform a particular task such as predicting \mathbf{Y} from \mathbf{X} .
- Goal: get \tilde{M} close to overall joint distribution $P^*(\mathbf{Y}, \mathbf{X})$.
- Model trained to generate all the variables.
- Naive Bayes model
- Higher bias
- Encode independence assumption about feature variables \mathbf{X} .
- Defines $\tilde{P}(\mathbf{Y}, \mathbf{X})$ and induces $\tilde{P}(\mathbf{Y} | \mathbf{X})$ and $\tilde{P}(\mathbf{X})$ using the same overall model for both.
- Works better when learning from limited amounts of data.

high bias
(like regression)

$$P(x_1, x_2, \dots, x_n | y) = P(x_1 | y) P(x_2 | y) \dots$$

assumption

DISCRIMINATIVE TRAINING

- Goal: get $\tilde{P}(\mathbf{Y} \mid \mathbf{X})$ to be close to $P^*(\mathbf{Y} \mid \mathbf{X})$
- Undirected model
- Train a conditional random field (CRF)
- Model directly encodes a conditional distribution $P(\mathbf{Y} \mid \mathbf{X})$.
- Encode independence assumptions about \mathbf{Y} and their dependence on \mathbf{X} .
- Find a good fit only to $P^*(\mathbf{Y} \mid \mathbf{X})$ without containing the same model to provide a good fit to $P^*(\mathbf{X})$.

LEARNING TASKS

Input to learning tasks

- Prior knowledge about $\tilde{\mathcal{M}}$
- Set of \mathcal{D} of data instances $\{d[1], \dots, d[M]\}$ which are sample IID from P^* .

Output of learning tasks

- Model $\tilde{\mathcal{M}}$ with structure and parameters.

3 Axes

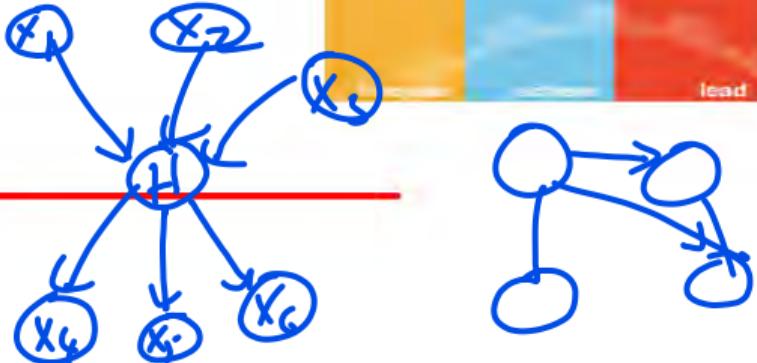
- ① The type of graphical model we are trying to learn – a Bayesian network or a Markov network.
- ② Hypothesis space
- ③ Data observability

LEARNING TASKS

Hypothesis space

- Given a graph structure, and learn only (some of) the parameters.
- We may not know the structure, and we have to learn both parameters and structure from the data.
- We may not even know the complete set of variables over which the distribution P^* is defined. We may only observe some subset of the variables in the domain and possibly be unaware of others.

LEARNING TASKS



Data Observability

- The data are **complete**, or **fully observed**, so that each of the training instances $d[m]$ is a full instantiation to all of the variables in \mathcal{X}^* .
- The data are **incomplete**, or **partially observed**, so that, in each training instance, some variables are not observed.
- The data contain **hidden variables** whose value is never observed in any training instance. The inclusion of a hidden variable in the network can greatly simplify the structure, reducing the complexity of the network that needs to be learned.



TABLE OF CONTENTS

1 LEARNING

2 PARAMETER ESTIMATION IN BAYESIAN NETWORKS

3 MLE FOR BAYESIAN NETWORKS

4 BAYESIAN PARAMETER ESTIMATION

PARAMETER ESTIMATION IN BAYESIAN NETWORKS

- Network structure is fixed.
- Data-set \mathcal{D} consists of fully observed data instances.

$$\mathcal{D} = \{d[1], \dots, d[M]\}$$

- Two approaches
 - ① Maximum likelihood estimation
 - ② Bayesian estimation

BIASED COIN EXAMPLE

- Head and tails outcome are controlled by a parameter θ .
- θ = frequency of heads in the coin tosses.

$$\max_{\theta} P(D|\theta)$$

$$P(X = H) = \theta$$

$$P(X = T) = 1 - \theta$$



- The distribution P is Bernoulli distribution.
- The data-set \mathcal{D} is sampled IID from P .

$$\mathcal{D} = \{x[1], \dots, x[M]\}$$

- ▶ Tosses are independent of each other.
- ▶ Tosses are sampled from the same distribution.
- Goal: Take \mathcal{D} and reconstruct θ .

BIASED COIN EXAMPLE

$x[1]$ and $x[2]$ are independent
given θ

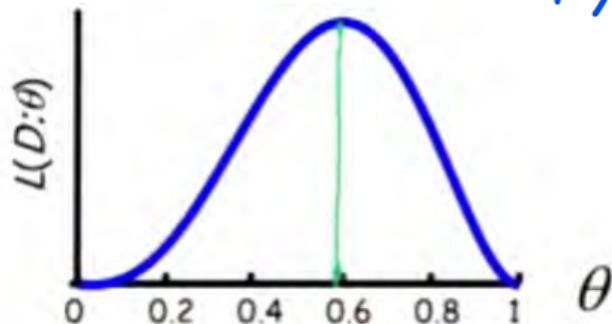
$$P(x[1], x[2] | \theta) = P(x[1] | \theta) P(x[2] | \theta)$$

- Suppose we observe: H, T, T, H, H

$$P(X[1] = H) = \theta$$

$$P(X[2] = T) = (1 - \theta) \quad \text{IID samples}$$

$$\begin{aligned} P(< H, T, T, H, H >: \theta) &= \underline{\theta(1 - \theta)(1 - \theta)\theta\theta} \\ &= \underline{\theta^3(1 - \theta)^2} \end{aligned}$$



$$\max P(\theta_1, \theta_2, \dots, \theta_N | \theta) = \max P(\theta_1 | \theta) I(\theta_2 | \theta) \dots I(\theta_N | \theta)$$

BIASED COIN EXAMPLE

- The probability of the data changes as a function of θ .
- Define the **likelihood function**:

$$\mathcal{L}(\theta : \langle H, T, T, H, H \rangle) = P(\langle H, T, T, H, H \rangle : \theta) = \theta^3(1 - \theta)^2$$

- Use the likelihood function as the measure of quality for different parameter values.
- Select the parameter value that maximizes the likelihood; this value is called the **maximum likelihood estimator (MLE)**.
- Observations: M_H heads and M_T tails .
- M_H and M_T are sufficient statistics.

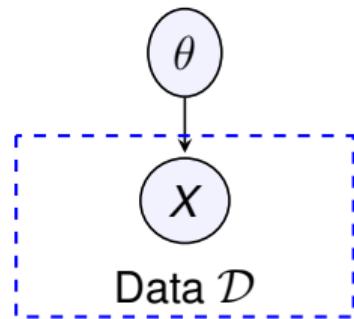
$$\theta = \frac{3}{5}$$

$$\hat{\theta} = \frac{M_H}{M_H + M_T} = \frac{3}{5} = 0.6$$

$\max_{\theta} \log \text{likelihood}$
 $\log(\theta^3(1-\theta)^2)$
 $3\log\theta + 2\log(1-\theta)$
 $\rightarrow \frac{3}{\theta} + \frac{2}{1-\theta}(-1) = 0$

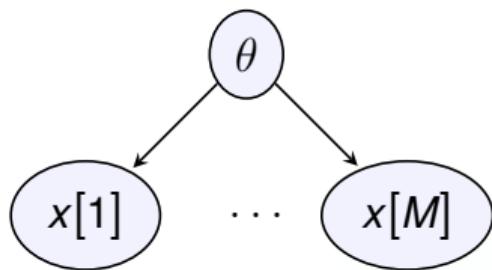
BERNOULLI DISTRIBUTION AS PGM

- Distribution



$$P(x[m] | \theta) = \begin{cases} \theta & x[m] = x^1 \\ 1 - \theta & x[m] = x^0 \end{cases}$$

- Hypothesis space



- Likelihood function

$\Theta = [0, 1] : \sum_i \theta_i = 1$

Condition and independent

$P(x[1], x[2], \dots, x[M] | \theta) = \prod_{i=1}^M P(x[i] | \theta)$

$\mathcal{L}(\theta : \mathcal{D}) = P(\mathcal{D} : \theta) = \prod_{m=1}^M P(x[m] | \theta) = \theta^{M_1} (1 - \theta)^{M_0}$

BERNOULLI DISTRIBUTION AS PGM

- Observations: M_1 and M_0
- The counts M_1 and M_0 give a compact distribution of the likelihood. These are called **sufficient statistics**.
- M_1 and M_0 are sufficient statistics.
- Find θ that maximizes likelihood:

$$\underline{\mathcal{L}(\theta : \mathcal{D})} = P(\mathcal{D} : \theta) = \underline{\theta^{M_1}} (1 - \theta)^{M_0}$$

- Equivalently maximize log-likelihood:

$$\underline{\ell(\theta : \mathcal{D})} = M_1 \log \theta + M_0 \log(1 - \theta)$$

- Differentiate log-likelihood and solve for θ as

$$\hat{\theta} = \frac{M_1}{M_1 + M_0}$$

SUFFICIENT STATISTICS

- Sufficient statistic is a function of the data that summarizes the relevant information for computing the likelihood.

DEFINITION (SUFFICIENT STATISTICS)

A function $\tau(\xi)$ from instances \mathcal{X} to \mathbb{R}^ℓ is a sufficient statistic, if for any two data sets \mathcal{D} and \mathcal{D}' and any $\theta \in \Theta$

data vector

$$\text{If } \sum_{\xi[m] \in \mathcal{D}} \tau(\xi[m]) = \sum_{\xi'[m] \in \mathcal{D}'} \tau(\xi'[m])$$

then $\mathcal{L}(\theta : \mathcal{D}) = \mathcal{L}(\theta : \mathcal{D}')$

\downarrow
 $[\quad] \leftarrow \mathbb{R}^\ell$
 \leftarrow
 $[\quad] \leftarrow \mathbb{R}^\ell$

- The tuple $\sum_{\xi[m] \in \mathcal{D}} \tau(\xi[m])$ is referred to as sufficient statistics of the data-set \mathcal{D} .

MULTINOMIAL DISTRIBUTION AS PGM

- Multinomial variable X takes the values x^1, \dots, x^K .

- Distribution :

$$P(X : \theta) = \theta_k \quad \text{if } x = x^k$$

- Hypothesis space:

$$\Theta = \left\{ \theta \in [0, 1]^K : \sum_i \theta_i = 1 \right\}$$

- Sufficient Statistics:

$$\tau(x^k) = \underbrace{0, \dots, 0}_{k-1}, 1, \underbrace{0, \dots, 0}_{n-k}$$

k dimensional vector.

} Symmetric dice
for example

MULTINOMIAL DISTRIBUTION AS PGM

- Likelihood function:

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = 0 \quad \frac{\partial \mathcal{L}}{\partial \theta_2} = 0$$

$$\mathcal{L}(\theta : \mathcal{D}) = \prod_k \theta_k^{M[k]}$$

- Maximum Likelihood Estimation:

$$\theta_1^{M[1]} \theta_2^{M[2]}$$

$$\theta_1^{M[1]} (1 - \theta_1)^{M[2]}$$

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = 0$$

$$\mathcal{L}(\theta : \mathcal{D}) = \max_{\theta \in \Theta} \mathcal{L}(\theta : \mathcal{D})$$

$$\hat{\theta} = \frac{M[k]}{M}$$

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = 0 \quad \frac{\partial \mathcal{L}}{\partial \theta_2} = 0 \quad \dots$$



MLE SUMMARY

- MLE is a simple principle for estimating or parameter selection given a data-set \mathcal{D} .
- Likelihood function uniquely determined by sufficient statistic that summarize \mathcal{D} .
- MLE has a closed form solution for many parametric distributions.

TABLE OF CONTENTS

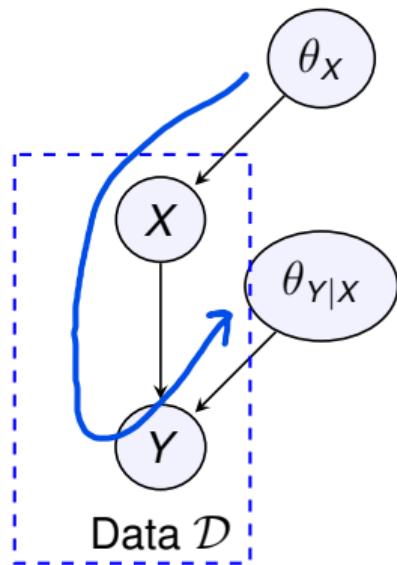
1 LEARNING

2 PARAMETER ESTIMATION IN BAYESIAN NETWORKS

3 MLE FOR BAYESIAN NETWORKS

4 BAYESIAN PARAMETER ESTIMATION

MLE FOR BAYESIAN NETWORKS



- A network consisting of two binary variables.
- Each assignment or training instance is given by $< x[m], y[m] >$.
- The network is parametrized by θ , which defines the set of parameters for all the CPDs in the network.
- Parameters for X : $\underline{\theta_{x^1}}$ and $\underline{\theta_{x^0}}$
- Parameters for Y : $\underline{\theta_{Y|X}} = \underline{\theta_{Y|x^1}} \cup \underline{\theta_{Y|x^0}}$

$$\theta_{Y|x^1} = \{\theta_{y^1|x^1}, \theta_{y^0|x^1}\}$$

$$\theta_{Y|x^0} = \{\theta_{y^1|x^0}, \theta_{y^0|x^0}\}$$

MLE FOR BAYESIAN NETWORKS

- Goal: Maximize the likelihood function.

$$\mathcal{L}(\theta : \mathcal{D}) = \prod_{m=1}^M P(x[m], y[m] | \theta)$$

$$f(\theta_1, \theta_2) = g_1(\theta_1, \theta_2) \times h_1(\theta_1, \theta_2)$$

$$= \prod_m \underbrace{P(x[m] : \theta)}_{\text{no dependence on } \theta_{Y|X}} \underbrace{P(y[m] | x[m] : \theta)}_{\theta_{Y|X}}$$

$$= \left[\prod_m \underbrace{P(x[m] : \theta_X)}_{\theta_X} \right] \left[\prod_m \underbrace{P(y[m] | x[m] : \theta_{Y|X})}_{\theta_{Y|X}} \right]$$

$$= \left[\prod_m \underbrace{P(x[m] : \theta_X)}_{\theta_X} \right] \left[\prod_{m: x[m] = x^0} \underbrace{P(y[m] | x[m] : \theta_{Y|x^0})}_{\theta_{Y|x^0}} \right] \left[\prod_{m: x[m] = x^1} \underbrace{P(y[m] | x[m] : \theta_{Y|x^1})}_{\theta_{Y|x^1}} \right]$$

- The likelihood function decomposes into local likelihood terms, each one depends only on the parameters for that variable's CPD.

$$\max_{\mathbf{x}, \mathbf{y}} F(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{x}, \mathbf{y}} g(\mathbf{x})h(\mathbf{y}) = \left[\max_{\mathbf{x}} g(\mathbf{x}) \right] \left[\max_{\mathbf{y}} h(\mathbf{y}) \right]$$

MLE FOR BAYESIAN NETWORKS

- Goal: Maximize the likelihood function.

$$\begin{aligned}
 \mathcal{L}(\theta : \mathcal{D}) &= \prod_m P_G(x[m], y[m] : \theta) \\
 &= \prod_m \prod_i P(x_i[m] | pa_{X_i}[m] : \theta) \\
 &= \prod_i \left[\prod_m P(x_i[m] | pa_{X_i}[m] : \theta) \right] \\
 &= \prod_i L_i(\theta_{X_i|Pa_{X_i}} : \mathcal{D})
 \end{aligned}$$

Factorization

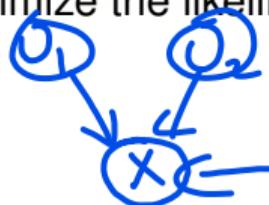
all samples

$\rightarrow \cdot x[1], y[1]$
 $\rightarrow \cdot x[2], y[2]$
 $\rightarrow \cdot$

- When the parameter sets $\theta_{X_i|Pa_{X_i}}$ are disjoint, then MLE can be computed by maximizing each local likelihood separately. The likelihood decomposes as a product of independent terms, one for each CPD in the network.

MLE FOR BAYESIAN NETWORKS FOR TABLE CPDs

- Goal: Maximize the likelihood function.



$$\mathcal{L}_x(\theta_{x|u} : \mathcal{D}) = \prod_m \theta_{x[m]|u[m]}$$

$\dots - u_1 - u_2 \times$
 o o o

- MLE parameters

optimal

$$\hat{\theta}_{x|u} = \frac{M[u, x]}{M[u]}$$

for $(x = x / u = u)$
 $P(x = x | u = u) = P_u(x = x / u = u)$

$M[u, x] = \text{frequency of } x[m] = x \text{ and } u[m] = u \text{ in } \mathcal{D}$

$$M[u] = \sum_x M[u, x]$$

MLE FOR BAYESIAN NETWORK SUMMARY

- For Bayesian Network with disjoint sets of parameters in CPDs, likelihood decomposes as a product of local likelihood functions, one per variable.
- For table CPDs, local likelihood functions further decomposes as a product of likelihood for multinomials, one for each parent combination.

TABLE OF CONTENTS

- 1 LEARNING
- 2 PARAMETER ESTIMATION IN BAYESIAN NETWORKS
- 3 MLE FOR BAYESIAN NETWORKS
- 4 BAYESIAN PARAMETER ESTIMATION



DRAWBACK OF MLE

$$P(\text{thumbstack} = H) = \frac{1}{3}$$
$$= \frac{T}{3} = \frac{2}{3}$$

$\frac{1}{3}$ $\frac{2}{3}$
bias, tail

MLE does not distinguish between

- a biased coin and unbiased coin. thumbstack and coin
- 10 tosses and 1,000,000 tosses of the coin.

Another approach – Bayesian Estimation.

$$P(\text{coin} = H) = \frac{1}{2}$$

JOINT PROBABILISTIC MODEL

$$\frac{P(X = H)}{\text{prior belief } F}$$

$$P(X = H | D) = P(X = H)$$

- If θ is unknown, tosses are not marginally independent.
- Each toss tells us something about θ and about the probability of next toss.
- Tosses are conditionally independent given θ .
- Treat θ as a random variable.

$$P(X = H | D, \theta) = P_{\theta}(X = H | \theta)$$

PARAMETER ESTIMATION AS PGM



Distribution $P(x[m] | \theta) = \begin{cases} \theta & x[m] = x^1 \\ 1 - \theta & x[m] = x^0 \end{cases}$

Prior distribution

$$P(\theta) \in [0, 1]$$

Uniform Prior

$$\underline{P(\theta) = 1}$$

PARAMETER ESTIMATION AS PGM

- Joint Distribution or Likelihood

$$P(x[1], \dots, x[M] | \theta)$$

$$\underline{P(x[1], \dots, x[M], \theta)} = P(x[1], \dots, x[M])P(\theta)$$

$$= P(\theta) \prod_{m=1}^M P(x[m] | \theta)$$

$$= P(\theta) \theta^{M[1]} (1 - \theta)^{M[0]}$$

$P(\theta | \theta)$ = likelihood

- Posterior Distribution

$$\underline{P(\theta)} \quad \underline{P(\theta | \theta)} \quad P(\theta | x[1], \dots, x[M]) = \frac{\underline{P(x[1], \dots, x[M] | \theta)} P(\theta)}{\underline{P(x[1], \dots, x[M])}} \rightarrow \text{prior}$$

Posterior \propto likelihood \times prior

handwritten notes

PARAMETER ESTIMATION AS PGM

$$P(\gamma_3) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

- To predict the next value $X[M + 1]$; integrate the posterior over θ .

→ how did we get this?

$$P(x[M + 1] | x[1], \dots, x[M]) = \int P(x[M + 1] | \theta)P(\theta | x[1], \dots, x[M])d\theta$$

$$P(X[M + 1] = x^1 | x[1], \dots, x[M]) = \int_0^1 \theta^{x^1} (1 - \theta)^{M[0]} d\theta$$

$$\begin{aligned} P(x[M + 1] = x^1 | x[1], \dots, x[M]) &= \frac{P(x[1], \dots, x[M])}{\int_0^1 \theta^{x^1} (1 - \theta)^{M[0]} d\theta} \\ &= \frac{M[1] + 1}{M + 2} \end{aligned}$$

- As the number of samples grows, the Bayesian estimator and the MLE estimator converge to the same value.

More Detailed Derivation

$$P(x[M+1], D) = \int P(x[M+1], D, \theta) d\theta$$

where $D = x[1], x[2], \dots, x[M]$

$$P(x[M+1], D) = \int P(x[M+1]/D, \theta) p(\theta, \phi) d\theta$$

$$p(\theta) P(x[M+1]/D) = \int P(x[M+1]/D, \phi) p(\phi/D) p(D) d\phi$$

$$P(x \in [M+1] / D) = \int P(x \in [M+1] / \emptyset) P(\emptyset / D) d\emptyset$$

Since $P(x \in [M+1] / D, \emptyset) = P(x \in [M+1] / \emptyset)$

PRIORS AND POSTERIORS

- Observe a training set $\mathcal{D} = x[1] \dots x[M]$
- M IID samples of random variable \mathcal{X} from an unknown distribution $P^*(\mathcal{X})$.
- Assume a parametric model $P(\xi | \theta)$ with a parameter space Θ .
- Treat θ as a random variable.
- Joint distribution

$$P(\mathcal{D}, \theta) = \underline{P(\mathcal{D} | \theta)} \underline{P(\theta)}$$

- The first term is the **likelihood function**. Compactly described by using sufficient statistics.
- The second term is the **prior distribution** over the possible values in Θ . Captures initial uncertainty about the parameters.



PRIORS AND POSTERIORS

- Derive the **posterior probability** over parameters

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D} | \theta)P(\theta)}{P(\mathcal{D})}$$

- Marginal likelihood**

$$P(\mathcal{D}) = \int_{\Theta} P(\mathcal{D} | \theta)P(\theta)d\theta$$

PRIORS AND POSTERIORS

- For a multinomial distribution, parameter space Θ is a space of

$$\theta = \langle \theta_1, \dots, \theta_K \rangle$$

$$\sum_k \theta_k = 1$$

- Likelihood function

$$\mathcal{L}(\theta : \mathcal{D}) = \prod_k \theta_k^{M[k]}$$

- Prior is assumed to follow Dirichlet distribution.

DIRICHLET DISTRIBUTION

$$\langle \theta_1, \theta_2, \dots, \theta_K \rangle$$

DEFINITION (DIRICHLET DISTRIBUTION)

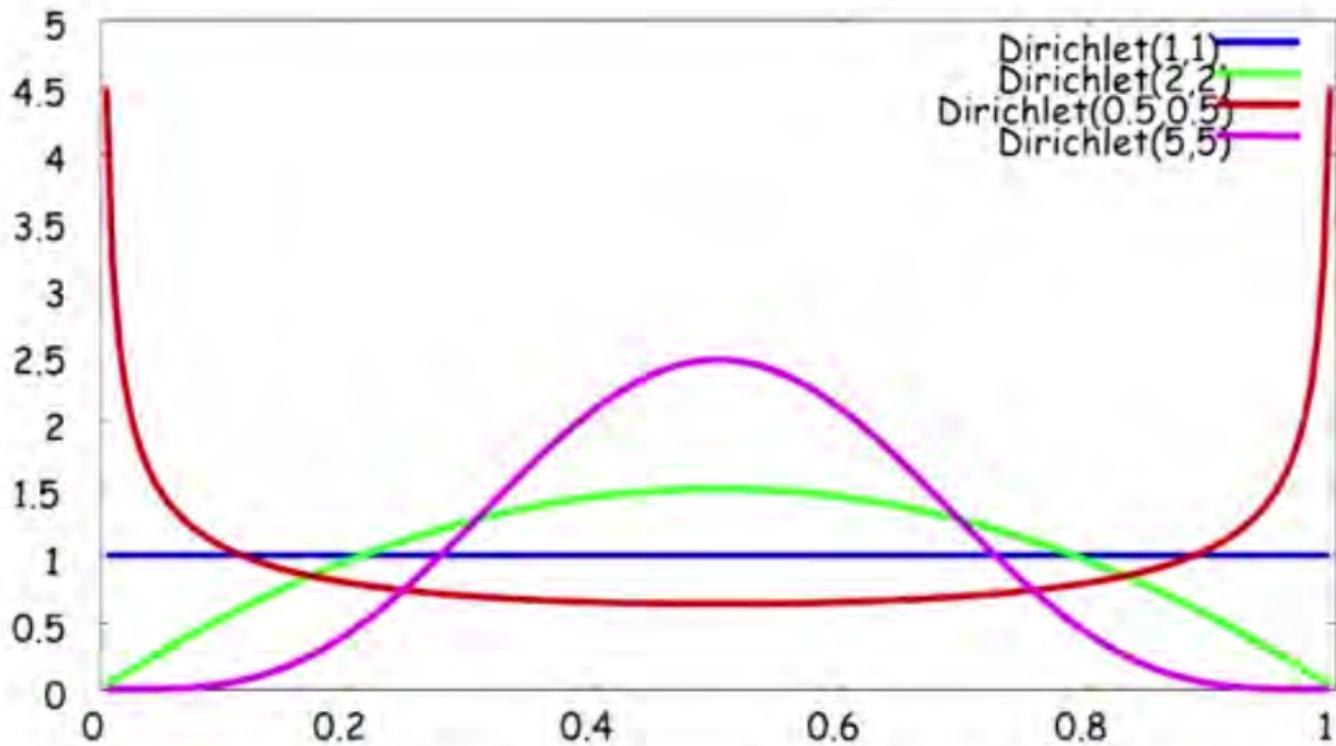
A Dirichlet distribution is specified by a set of hyper-parameters $\langle \underline{\alpha}_1, \dots, \underline{\alpha}_K \rangle$

$$\underline{\theta} \sim Dirichlet(\underline{\alpha}_1, \dots, \underline{\alpha}_K) \quad \text{if} \quad P(\theta) \propto \prod_k \theta_k^{\underline{\alpha}_k - 1}$$

- If $P(\theta)$ is Dirichlet distribution with hyper-parameters $\langle \underline{\alpha}_1, \dots, \underline{\alpha}_K \rangle$ and $\underline{\alpha} = \sum_j \underline{\alpha}_j$, then

$$\mathbb{E}[\theta_k] = \frac{\alpha_k}{\alpha}$$

DIRICHLET DISTRIBUTION



Daphne Koller

DIRICHLET PRIORS AND POSTERIORS

- Posterior distribution

$$P(\theta | \mathcal{D}) \propto P(\mathcal{D} | \theta) P(\theta)$$

Diagram illustrating the posterior distribution formula:

- The term $P(\theta | \mathcal{D})$ is circled and labeled "Dirichlet".
- The term $P(\mathcal{D} | \theta)$ is circled and labeled "multinomial".
- The term $P(\theta)$ is circled and labeled "Dirichlet".

- Likelihood function – multinomial

$$\underline{P(\mathcal{D} | \theta)} = \mathcal{L}(\theta : \mathcal{D}) = \prod_k \theta_k^{M[k]}$$

- Prior distribution – Dirichlet

$$P(\theta) \propto \prod_k \theta_k^{\alpha_k - 1}$$

- Posterior distribution – Dirichlet

DIRICHLET PRIORS AND POSTERIORS

- If Prior $P(\theta)$ is Dirichlet and the Likelihood $P(D | \theta)$ is multinomial, then the Posterior $P(\theta|D)$ is also Dirichlet.

Prior = Dirichlet($\underline{\alpha_1}, \dots, \underline{\alpha_K}$)

Data counts = M_1, \dots, M_K

Posterior = Dirichlet($\underline{\alpha_1 + M_1}, \dots, \underline{\alpha_K + M_K}$)

$$\int \frac{P(\theta|\underline{\alpha}) P(\underline{\theta})}{(\prod \theta_i^{\alpha_i})} (\underline{\theta_1^{\alpha_1}, \theta_2^{\alpha_2}, \dots, \theta_K^{\alpha_K}}) d\underline{\theta}$$

- Prior and Posterior have the same form of distribution.
- Dirichlet is a conjugate pair for multinomial.

BAYESIAN PREDICTION

- To predict for a new sample

$$P(x[M+1]) = \frac{\int g(x) f(x) dx}{\int g(x) dx}$$

$$\mathbb{E}(g(x)) = \int g(x) f(x) dx$$

$$P(x[M+1] | \mathcal{D}) = \int P(x[M+1] | \mathcal{D}, \theta) P(\theta | \mathcal{D}) d\theta$$

$$= \int P(x[M+1] | \theta) P(\theta | \mathcal{D}) d\theta$$

$$= \mathbb{E}_{P(\theta | \mathcal{D})} [P(x[M+1] | \theta)]$$

$$P(x[M+1] = x^k | \mathcal{D}) = \frac{M[k] + \alpha_k}{M + \alpha}$$

Θ_k

$$\mathbb{E}[\Theta_k] = \frac{\alpha_k}{\alpha}$$

- Equivalent sample size = $\alpha = \alpha_1 + \dots + \alpha_k$
- Larger α implies more confidence in our prior.

$$\mathbb{E}[\Theta_k] =$$

EXAMPLE

- For a given binomial data with uniform distribution for parameter θ , it is observed that

$$(M[1], M[0]) = (5, 2) \quad \text{uniform} = \text{Dirichlet}(1, 1)$$

Predict $P(X[8] = 1)$ using MLE and Bayesian prediction.

- MLE

$$P(X[8] = 1) = \frac{M_1}{M} = \frac{5}{7} = 0.71$$

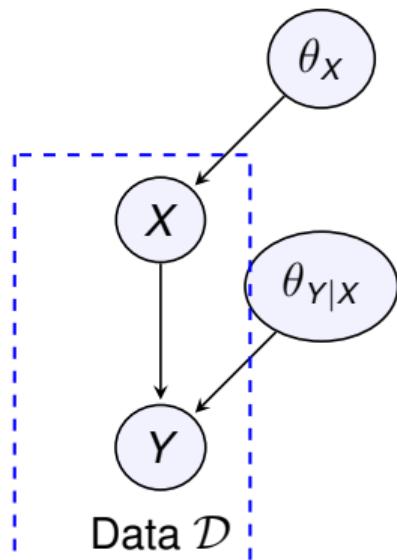
- Bayesian prediction

$$\begin{aligned} P(X[8] = 1) &= \frac{\alpha_1 + M_1}{\alpha + M} \\ &= \frac{1 + 5}{2 + 7} = \frac{6}{9} = 0.66 \end{aligned}$$

BAYESIAN ESTIMATION SUMMARY

- Bayesian Learning treats parameters as random variables.
- Dirichlet distribution as conjugate pair of multinomial distribution.
 - ▶ Posterior has the same form as prior.
 - ▶ Can be updated in closed form using sufficient statistic from data.
- Bayesian prediction combines sufficient statistic from imaginary Dirichlet sample and real data samples.
- Dirichlet hyper-parameters determine both the prior beliefs and their strengths.
- Bayesian Learning is robust in sparse data regime in terms of its generalization ability.

BAYESIAN ESTIMATION ON BAYESIAN NETWORK



- A network consisting of two binary variables.
- Training data consists of M observations given by $\langle x[m], y[m] \rangle$
- Unknown parameters θ_X and $\theta_{Y|X}$
- Instances are independent given the unknown parameters.
- $\langle x[m], y[m] \rangle$ are d-separated from $\langle x[m'], y[m'] \rangle$ once we know the parameter variables.
- Assume that the priors for the individual parameters variables are apriori independent. That is, we believe that knowing the value of one parameter tells us nothing about another.

BAYESIAN ESTIMATION ON BAYESIAN NETWORK

- The Bayesian network have parameters

$$\theta = (\theta_{X_1|Pa_{X_1}}, \dots, \theta_{X_n|Pa_{X_n}})$$

- Prior distribution satisfies **global parameter independence** if

$$P(\theta) = \prod_i P(\theta_{X_i|Pa_{X_i}})$$

- If the complete data $\langle x[m], y[m] \rangle$ are observed for all m , then parameters θ_X and $\theta_{Y|X}$ are d-separated.

$$P(\theta_X, \theta_{Y|X} | \mathcal{D}) = P(\theta_X | \mathcal{D})P(\theta_{Y|X} | \mathcal{D})$$

- Given the data set \mathcal{D} , determine the posterior over θ_X independently of the posterior over $\theta_{Y|X}$.

BAYESIAN ESTIMATION ON BAYESIAN NETWORK

- Let X have parents U . Then the prior $P(\theta_{X|U})$ satisfies **local parameter independence** if

$$P(\theta_{X|U}) = \prod_u P(\theta_{X|u})$$

- If $P(\theta)$ satisfies global and local parameter independence then,

$$P(\theta | \mathcal{D}) = \prod_i \prod_{Pa_{X_i}} P(\theta_{X_i | Pa_{X_i}} | \mathcal{D})$$

BAYESIAN ESTIMATION ON BAYESIAN NETWORK

- Multinomial Parameters $\theta_{X|u}$
- $P(\theta_{X|u})$
 - ▶ Dirichlet prior with hyper-parameters $\alpha_{x^1|u}, \dots, \alpha_{x^K|u}$
- $P(\theta_{X|u} | \mathcal{D})$
 - ▶ Dirichlet posterior with hyper-parameters $\alpha_{x^1|u} + M[u, x^1], \dots, \alpha_{x^K|u} + M[u, x^K]$

SUMMARY

- In Bayesian networks, if parameters are independent apriori, then they are also independent in the posterior.
- For multinomial Bayesian networks, estimation uses sufficient statistic $M[x, u]$ (counts).

$$\text{MLE} \quad \hat{\theta}_{X|u} = \frac{M[X, u]}{M[u]}$$

$$\text{BL} \quad P(x \mid U, \mathcal{D}) = \frac{\alpha_{X,u} + M[x, u]}{\alpha_u + M[u]}$$

- Bayesian Learning requires a choice of prior.

EXAMPLE

Given the following data and the structure that $X \rightarrow Y$, learn the parameters using MLE.

X	Y
0	1
0	1
1	0
1	1
1	0
1	1
1	0
0	1
0	1

$$\hat{\theta}_{X|u} = \frac{M[X, u]}{M[u]}$$

$$\theta_X = P(X) = \begin{bmatrix} 4/9 & 5/9 \end{bmatrix}$$

$$\theta_{Y|X} = P(Y | X) = \begin{bmatrix} 0/4 & 4/4 \\ 3/5 & 2/5 \end{bmatrix}$$



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 14: LEARNING

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in



TABLE OF CONTENTS

- ① STRUCTURE LEARNING IN BAYESIAN NETWORKS
- ② SCORE-BASED LEARNING
- ③ BAYESIAN SCORE
- ④ TREE STRUCTURED NETWORK



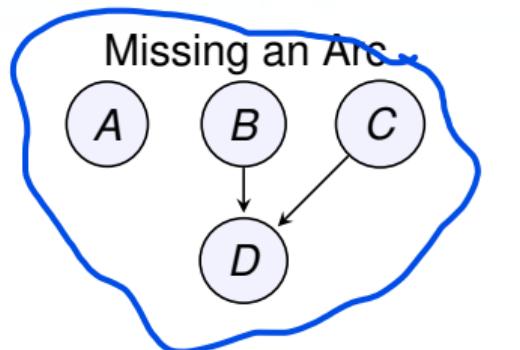
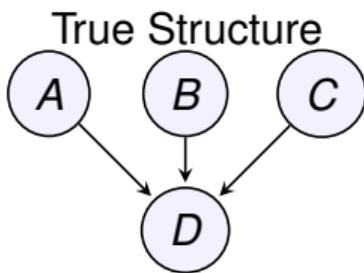
STRUCTURE LEARNING IN BAYESIAN NETWORKS

- We do not know the structure of the Bayesian network in advance.
- The data are generated IID from an underlying distribution $P^*(\mathcal{X})$.
- $P^*(\mathcal{X})$ is induced by some Bayesian network G^* over \mathcal{X} .

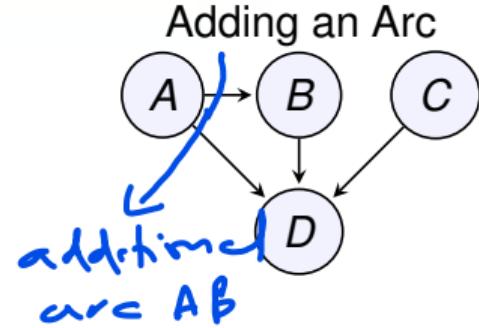
WHY STRUCTURE LEARNING?

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

STRUCTURE – WHICH ONE?



- Incorrect independencies
- Correct P^* cannot be learned
- Could generalize better



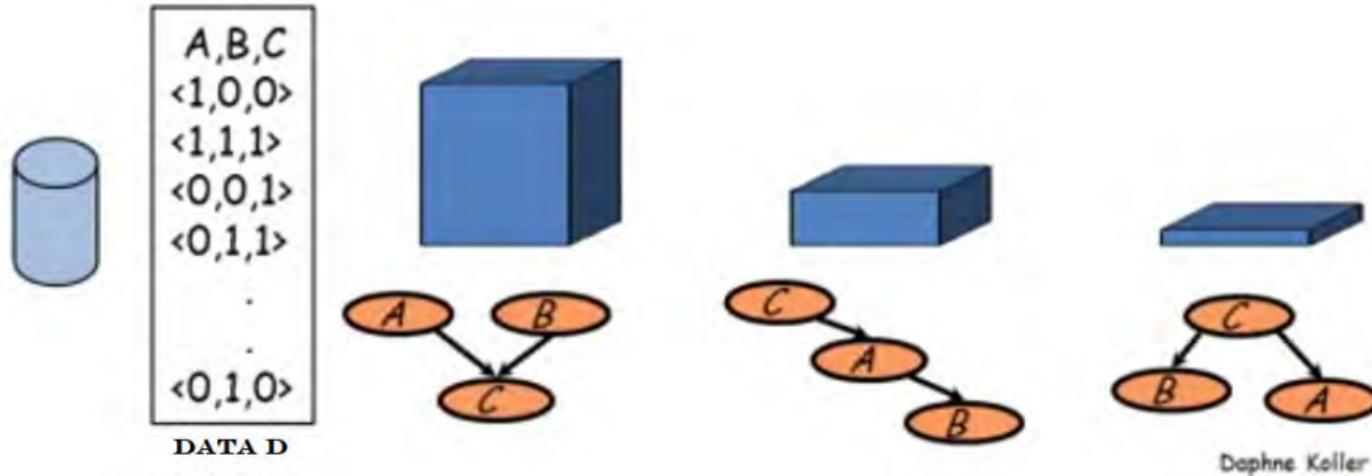
- Spurious dependencies
- Cannot correctly learn P^*
- More number of parameters are learned
- Worse generalization

TABLE OF CONTENTS

- 1 STRUCTURE LEARNING IN BAYESIAN NETWORKS
- 2 SCORE-BASED LEARNING *vs Constraint-based learning
some independence tests
can go wrong*
- 3 BAYESIAN SCORE
- 4 TREE STRUCTURED NETWORK

SCORE-BASED LEARNING

- Define **scoring function** that evaluates how well a structure matches the data.



- Search for a structure that maximizes the score. This converts the learning problem into an optimization problem.

LIKELIHOOD STRUCTURE SCORE

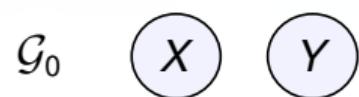
- Simplest score
- Find the graph and parameters that maximize the likelihood of the data.

$$\text{score}_L(\mathcal{G} : \mathcal{D}) = \ell((\hat{\theta}, \mathcal{G}) : \mathcal{D})$$

- $\hat{\theta}$ = MLE of parameters given \mathcal{G} and \mathcal{D}

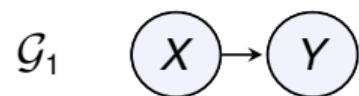
maximum likelihood of
data for the given
graph \mathcal{G}

EXAMPLE



$$G_0 \quad score_L(G_0 : \mathcal{D}) = \sum_m (\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]})$$

Handwritten notes: $\sum_{m=1}^M \log \hat{\theta}_{x[m]} \hat{\theta}_{y[m]}$



$$G_1 \quad score_L(G_1 : \mathcal{D}) = \sum_m (\log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]})$$

Handwritten notes: $\log \left(\prod_{m=1}^M \hat{\theta}_{x[m]} \hat{\theta}_{y[m]|x[m]} \right)$



EXAMPLE

$$\underline{\text{score}_L(\mathcal{G}_1 : \mathcal{D}) - \text{score}_L(\mathcal{G}_0 : \mathcal{D})} = \sum_m (\log \hat{\theta}_{y[m] | x[m]}) - \sum_m (\log \hat{\theta}_{y[m]})$$

$$= \sum_{x,y} M[x, y] \log \hat{\theta}_{y|x} - \sum_y M[y] \log \hat{\theta}_y$$

$$= M \sum_{x,y} \hat{P}(x, y) \log \hat{P}(y | x) - M \sum_y \hat{P}(y) \log \hat{P}(y)$$

$$= M \left(\sum_{x,y} \hat{P}(x, y) \log \hat{P}(y | x) - \sum_{x,y} \hat{P}(x, y) \log \hat{P}(y) \right)$$

$$= M \sum_{x,y} \hat{P}(x, y) \log \frac{\hat{P}(x, y)}{\hat{P}(x)\hat{P}(y)}$$

$$= M \cdot I_{\hat{P}}(X; Y)$$

$M[x, y] = [P(x, y)]M$
sufficient statistics
empirical distribution $\hat{P}(x, y)$

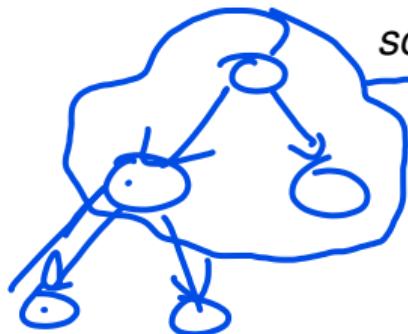
$$\sum_x \hat{P}(x, y) = \hat{P}(y)$$

$$\hat{P}(y | x) = \frac{\hat{P}(x, y)}{\hat{P}(x)}$$

samples times Mutual Information

LIKELIHOOD STRUCTURE SCORE

- In general, the Likelihood Structure Score decomposes as number of samples M times the difference between the mutual information and the sum of entropies of variables X .



$$\text{score}_L(\mathcal{G} : \mathcal{D}) = M \sum_{i=1}^n I_{\hat{P}}(X_i; Pa_{X_i}^{\mathcal{G}}) - M \sum_i H_{\hat{P}}(X_i)$$

$$I_{\hat{P}}(X; Y) = \sum_{x,y} \hat{P}(x,y) \log \frac{\hat{P}(x,y)}{\hat{P}(x)\hat{P}(y)}$$

$$H_{\hat{P}}(X) = - \sum_x P(x) \log P(x)$$

- The score is higher if X_i is correlated with its parents. Helps in identifying the Parent-Child relationship and place the nodes accordingly.

LIMITATIONS OF LIKELIHOOD SCORE

$$\text{score}_L(\mathcal{G}_1 : \mathcal{D}) - \text{score}_L(\mathcal{G}_0 : \mathcal{D}) = M \cdot I_{\hat{P}}(X; Y)$$

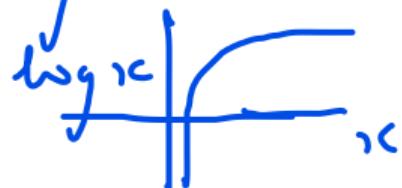
- Positive difference suggests that \mathcal{G}_1 has to be chosen.
 - Negative difference implies that \mathcal{G}_0 has to be chosen.
 - Mutual Information is always positive. $I_{\hat{P}}(X; Y) \geq 0$
 - Mutual Information $I_{\hat{P}}(X; Y) = 0$ iff X and Y are independent.
 - In empirical distribution \hat{P} , due to statistical fluctuations while taking samples, $I_{\hat{P}}(X; Y) > 0$ almost always.
 - Score is maximized for fully connected network.
 - Over-fits the data
- why is this always true?*

why is mutual information always non-negative?

$$I(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

\log is a concave function since

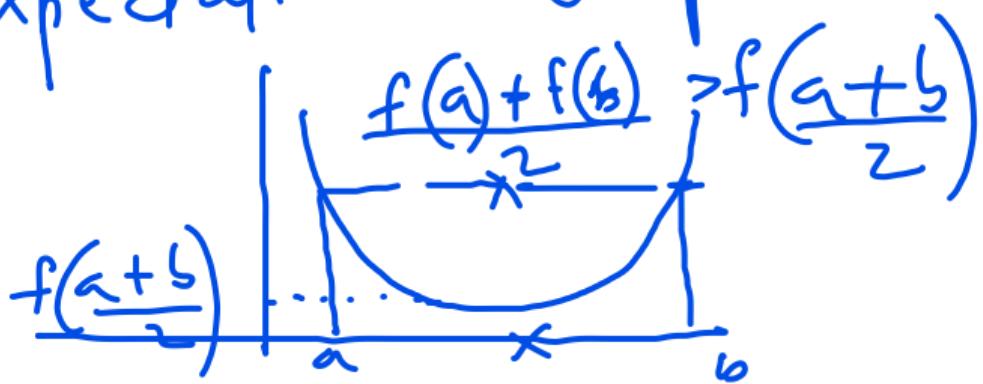
$$\frac{d^2}{dx^2} \log x = -\frac{1}{x^2} < 0$$



- \log is a convex function

We can apply Jensen's inequality on
 $-\log$, which is a convex function

$E(F(X)) \geq F(E(X))$ where X is
a random variable, and F is a convex
function, E is expectation over $\{x\}$ of X



$$\begin{aligned}
 I(x, y) &\geq -\log \sum_{x \in} \sum_y p(x, y) \frac{p(x)p(y)}{p(x, y)} \\
 &\geq -\log \sum_{x \in} \sum_y p(x)p(y) \\
 &\geq -\log \left(\left(\sum_{x \in} p(x) \right) \left(\sum_y p(y) \right) \right) \\
 &\geq -\log 1 \geq 0
 \end{aligned}$$

AVOID OVER-FITTING

- Restrict the hypothesis space
 - ▶ Restrict # parents – common strategy
 - ▶ Restrict # parameters
- Score that penalize complexity
 - ▶ Explicitly penalize model complexity
 - ▶ Use Bayesian score that averages over all possible parameter values.
 -



TABLE OF CONTENTS

- ① STRUCTURE LEARNING IN BAYESIAN NETWORKS
- ② SCORE-BASED LEARNING
- ③ BAYESIAN SCORE
- ④ TREE STRUCTURED NETWORK

BAYESIAN SCORE

$$\mathcal{L}(\hat{\theta}; \mathcal{G}) \quad \text{Pr}(\hat{\theta})$$

- Bayesian score is derived from Bayesian paradigm – Any uncertainty can be represented by using a probability distribution for it.
- Uncertainty over graph can be represented using a prior over graph.
- Uncertainty over parameters can be represented using a prior over parameters.

BAYESIAN SCORE

- Optimization problem – find a graph that maximizes a probability of \mathcal{G} over \mathcal{D}

$$P(\mathcal{G} | \mathcal{D}) = \frac{P(\mathcal{D} | \mathcal{G})P(\mathcal{G})}{P(\mathcal{D})}$$

$P(\mathcal{D} | \mathcal{G})$ – Marginal likelihood

$P(\mathcal{G})$ – Prior over structures

$P(\mathcal{D})$ – Marginal probability of Data – normalizing constant (ignored)

BAYESIAN SCORE

- Taking log on both sides, Bayesian Score

$$\text{score}_B(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D} | \mathcal{G}) + \log P(\mathcal{G})$$

$\text{score}_B(\mathcal{G} : \mathcal{D})$ – Bayesian score

$\log P(\mathcal{D} | \mathcal{G})$ – Marginal likelihood of Data given graph \mathcal{G}

$\log P(\mathcal{G})$ – Prior over structures – less significant

MARGINAL LIKELIHOOD OF DATA GIVEN GRAPH

- As $M \rightarrow \infty$ a network with Dirichlet prior satisfies

$$\log P(\mathcal{D} | \mathcal{G}) = \ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D}) - \frac{\log M}{2} \text{score}_{\text{BIC}}(\mathcal{G} : \mathcal{D}) + \mathcal{O}(1)$$

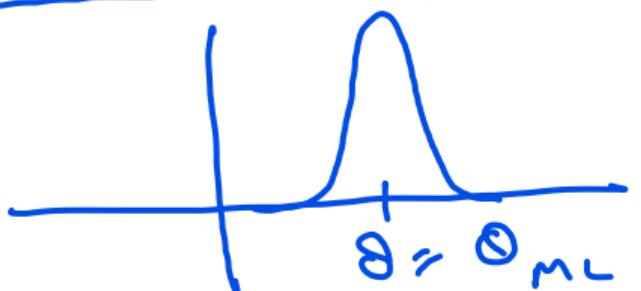
$\text{score}_{\text{BIC}}(\mathcal{G} : \mathcal{D})$

- Marginal likelihood is the Bayesian Information Criterion (BIC) score.
- $\ell(\hat{\theta}_{\mathcal{G}} : \mathcal{D})$ – Log likelihood score with Maximum likelihood parameters $\hat{\theta}_{\mathcal{G}}$
- M – # training data instances
- $\text{Dim}[\mathcal{G}]$ – # independent parameters = # entries in the distribution - 1

$$\text{Pr}(\theta|G) = \int \text{Pr}(P|\theta, G) \text{Pr}(\theta|G) d\theta$$

$\int \text{Pr}(\theta|G) d\theta$
 $\text{Pr}(\theta|G)$

$\text{Pr}(P|\theta, G) \text{Pr}(\theta|G)$ can be approximated as
 a function that peaks at $\theta = \hat{\theta}_{ML} =$
max likelihood parameters



We can use second-order approximation
(Taylor's series) and write

$$\log \left(\Pr(\theta | \phi_{1,2}) \Pr(\phi_{1,2}) \right) = \log \left(\Pr(\theta | \theta_{MC}, g) \Pr \left(\frac{\theta_{MC}}{g} \right) \right) - \frac{1}{2} (\theta - \theta_{MC})^T H (\theta - \theta_{MC})$$
$$g(\theta) = g(\theta_{MC}) e^{-\frac{1}{2} (\theta - \theta_{MC})^T H (\theta - \theta_{MC})}$$

where $H = \begin{bmatrix} \frac{\partial^2 \ln g(\theta)}{\partial \theta \partial \theta} & \\ & \ddots \end{bmatrix}$ a $d \times d$ matrix
 [Fisher Information matrix]

$P(\theta|G) = \int g(\theta) d\theta$ is a Gaussian
 integral can be approximated as

$$\frac{(2\pi)^{d/2}}{|H|^{\frac{1}{2}}} P_v(\theta|\theta_{MC}) \text{pr}(\theta_{MC}|G)$$

Taking $\log P(\theta|G)$ and approximate $|H| = M^d$
 + get the result.

More specifically, $\int_{-\infty}^{\infty} e^{-\frac{1}{2}x^T A x} dx$

$$= \frac{(2\pi)^{n/2}}{|A|^{1/2}}$$

Here $A = H$ and $n=d$, and we have a constant $\Pr(D|\theta_{MC}, G) \Pr(\theta_{MC}|G)$ in the integrand leading to the integral being equal to $\frac{(2\pi)^{d/2}}{|H|^{1/2}} \Pr(D|\theta_{MC}, G) \Pr(\theta_{MC}|G)$

$$\Pr(D|G) = \frac{(2\pi)^{d/2}}{|H|^{\frac{d}{2}}} \Pr(D|O_{MC}, G) \Pr(O_{MC}|G)$$

$$\begin{aligned} \log \Pr(D|G) &= \frac{d}{2} \log 2\pi - \frac{1}{2} \log |H| + \\ &\quad \boxed{\log (\Pr(D|O_{MC}, G) \Pr(O_{MC}|G))} \\ &= O(1) - \frac{1}{2} \log |H| + \ell(O_{MC}; G) \end{aligned}$$

We can take $|H| \approx M^d$ + $O(d)$ $H = \begin{bmatrix} M & M & \dots & M \end{bmatrix}_{d \times d}$

$$\log(\Pr(D|G)) = \ell(O_{MC}; G) - \frac{d}{2} \log M + O(1)$$

BAYESIAN SCORE

- As $M \rightarrow \infty$ Marginal Likelihood is equivalent to BIC score.
- That implies Bayesian score is equivalent to BIC score.
- The graph or its I-equivalent graph will have the highest score among all possible graphs.

A closer look at the formula:

$$\text{score}_{\text{BIC}}(G:D) = l(\hat{\theta}:G) - \frac{-\log M}{2} \dim(G)$$

We can rewrite $l(\hat{\theta}:G)$ as follows

$$l(\hat{\theta}:G) = M \sum_{i=1}^n I_p(x_i, p_{x_i}) - M \sum_{i=1}^n H_p(x_i) \\ - \frac{-\log M}{2} \dim(G)$$

$$score_{\text{BIC}}(G:D) = M \sum_{i=1}^n I_p^{\hat{P}}(X_i, \text{par}_i) - M \sum_{i=1}^n H_p^{\hat{P}}(X_i) - \frac{\log M \cdot \dim(G)}{2}$$

- Entropy terms do not depend on the graph and can be ignored
- The stronger the dependence between a variable and its parents, the higher the score
- The more complex the network, the lower the score
- Mutual information term grows linearly in M , complexity term grows logarithmically \Rightarrow more the data, more emphasis given to fitting data



TABLE OF CONTENTS

-
- 1 STRUCTURE LEARNING IN BAYESIAN NETWORKS
 - 2 SCORE-BASED LEARNING
 - 3 BAYESIAN SCORE
 - 4 TREE STRUCTURED NETWORK

OPTIMIZATION PROBLEM

- Input
 - ▶ Training data
 - ▶ Scoring function
 - ▶ Set of possible structures
- Output
 - ▶ Network that maximizes the score
- Key property for computation efficiency
 - ▶ Decomposability of score

$$\text{score}(\mathcal{G} : \mathcal{D}) = \sum_i \text{score}(X_i \mid \text{Pa}_{X_i}^{\mathcal{G}} : \mathcal{D})$$

LEARNING TREES / FORESTS

$$\begin{aligned}
 \underline{\text{score}(\mathcal{G} : \mathcal{D})} &= \sum_i \text{score}(X_i | \text{Pa}_{X_i}^{\mathcal{G}} : \mathcal{D}) \\
 &= \left[\sum_{i:p(i)>0} \text{score}(X_i | X_{p(i)} : \mathcal{D}) - \text{score}(X_i : \mathcal{D}) \right] + \sum_{i=1}^n \text{score}(X_i : \mathcal{D})
 \end{aligned}$$


- Score = sum of edge scores + constant.

ALGORITHM FOR LEARNING TREES / FORESTS

Algorithm

- Define unirected graph with nodes $\{1, \dots, n\}$
- Set $w(i \rightarrow j) = \max[score(X_j|X_i) - score(X_j), 0]$
- Find forest with maximal weights
 - ▶ Use MST algorithms like Prim's or Kruskal's
 - ▶ Remove edges with weight 0.
- Generates tree if likelihood score is used and a forest if BIC score is used.
- Network with only one parent.



GENERAL NETWORK

- For general trees, with # parents ≥ 2 , use
 - ▶ Heuristic hill climbing algorithm
 - ▶ Best first search
 - ▶ Simulated annealing

GREEDY HILL CLIMBING ALGORITHM

- Start with a given network
 - ▶ Empty network
 - ▶ Best tree
 - ▶ Random network
 - ▶ Prior knowledge
- At each iteration
 - ▶ Consider score of all possible changes like addition, removal and reversal of edges.
 - ▶ Apply change that most improves the score (greedy approach)
- Stop when no modification improves score. This gives local maxima.

GREEDY HILL CLIMBING ALGORITHM

- Pitfalls
 - ▶ Local maxima
 - ▶ Plateau – often in BN due to I-equivalent structures.
- To remove pitfalls
 - ▶ Use edge reversals
 - ▶ Random restarts
 - ★ Take some random steps and then start climbing again
 - ▶ Tabu lists
 - ★ Keep a list of most recent K steps and search cannot reverse any of these steps.

CHOW LIU ALGORITHM

Chow-Liu Algorithm

Step 1: For each pair of variables A,B, use data to estimate $P(A,B)$, $P(A)$, $P(B)$

Step 2: For each pair of variables A,B, calculate mutual information

$$I(A, B) = \sum_a \sum_b P(a, b) \log \frac{P(a, b)}{P(a)P(b)}$$

Step 3: Calculate the maximum spanning tree over the set of variables, using edge weights $I(A,B)$ (given N variables, this costs only $O(N^2)$ time)

Step 4: Add arrows to edges to form a directed-acyclic graph by picking an arbitrary node as root and directing edges outward from the root

Step 5: Learn the CPD's for this graph

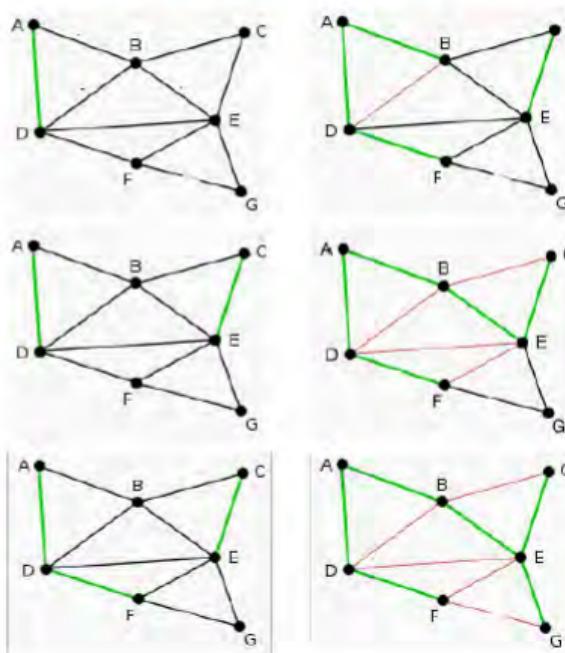
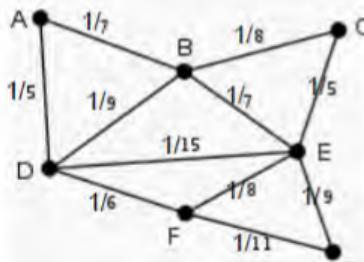
KRUSKAL'S ALGORITHM

Maximum Spanning Tree Algorithm

- Kruskal's Algorithm
 - Start with the empty graph and add edges one by one
 - As the next edge to add, choose one that
 - Is not in graph yet
 - Does not introduce a cycle. Has the maximum weight

CHOW LIU ALGORITHM

Chow-Liu algorithm example Greedy Algorithm to find Max-Spanning Tree





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODEL SESSION # 15: LEARNING - MARKOV NETWORK

SEETHA PARAMESWARAN
seetha.p@pilani.bits-pilani.ac.in

TABLE OF CONTENTS

-
- ① LEARNING IN MARKOV NETWORK
 - ② PARAMETER ESTIMATION IN MARKOV NETWORKS
 - ③ MLE FOR CRF
 - ④ MAP ESTIMATION FOR MRF AND CRF

CONDITIONAL RANDOM FIELD

- Random Variables X_1, \dots, X_n
- Gibbs Distribution $\Phi = \{\phi_1(D_1), \dots, \phi_k(D_k)\}$
- Un-normalized distribution

$$\tilde{P}_\Phi(X, Y) = \prod_{i=1}^k \phi_i(D_i)$$

$$Z_\Phi(X) = \sum_Y \tilde{P}_\Phi(X, Y)$$

- Partition function

- CRF

$$P_\Phi(Y | X) = \frac{1}{Z_\Phi(X)} \tilde{P}_\Phi(X, Y)$$

$$Z = \sum_{X, Y} \tilde{P}_\Phi(X, Y)$$

$\tilde{P}_\Phi(X, Y)$

unnormalised

partition function

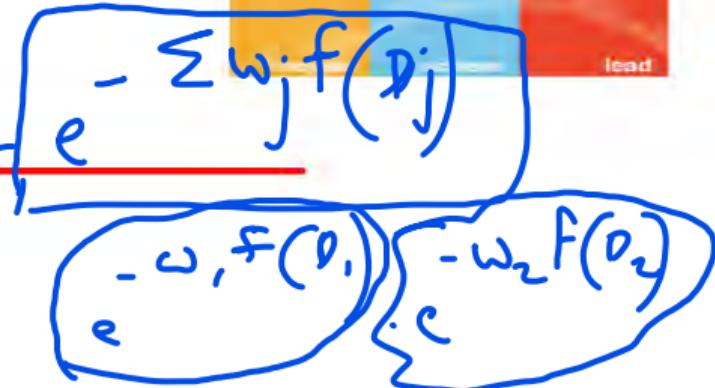
LOG-LINEAR REPRESENTATION

- Incorporate local structure in Markov Network

un-normalized

$$\tilde{P}_\phi(X, Y) = \prod_{i=1}^k \phi_i(D_i)$$

$$= \exp \left[- \sum_j w_j f_j(D_j) \right]$$



can have
multiple
indicator
functions on the

- w_j – coefficient of each feature f_j , which is used to represent the network ~~same~~ ^{scope}
- Any factor can be represented by a log-linear model by including all of the appropriate features.

EXAMPLE

factor table

-	-
01	-
10	-
11	-

- Binary random variables X_1 and X_2 with $\Phi(X_1, X_2) = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$
- To represent this as a log-linear model, use indicator functions

$$f(X_1=0, X_2=0) = a_{00}$$

$$\begin{aligned} f_{00} &= \mathbf{1}\{X_1 = 0, X_2 = 0\} \\ f_{01} &= \mathbf{1}\{X_1 = 0, X_2 = 1\} \\ f_{10} &= \mathbf{1}\{X_1 = 1, X_2 = 0\} \\ f_{11} &= \mathbf{1}\{X_1 = 1, X_2 = 1\} \end{aligned}$$

$f_{00}(x_1, x_2) = 1$
iff $x_1 = 0, x_2 = 0$
and 0 otherwise

EXAMPLE

$$e^{+\log \omega_{00}} = \underline{\omega_{00}} \quad \omega_{00} = -\log a_{00}$$

- Factors can be represented as

$$\Phi(X_1, X_2) = \exp \left[- \sum_{k,l} w_{kl} f^{kl}(X_1, X_2) \right]$$

$$w_{kl} = -\log a_{kl} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}$$

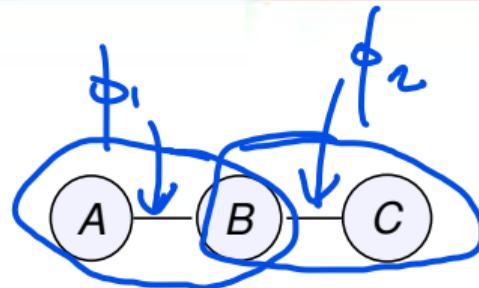
$$w_{kf} = \begin{bmatrix} -\log a_{00} & -\log a_{01} \\ -\log a_{10} & -\log a_{11} \end{bmatrix}$$



TABLE OF CONTENTS

- 1 LEARNING IN MARKOV NETWORK
- 2 PARAMETER ESTIMATION IN MARKOV NETWORKS
- 3 MLE FOR CRF
- 4 MAP ESTIMATION FOR MRF AND CRF

MAXIMUM LIKELIHOOD FOR MARKOV NETWORKS



$$Z = \sum_{A, B, C} \phi_1(A, B) \times \phi_2(B, C)$$

- Joint Distribution:

$$P_\Phi(A, B, C) = \frac{1}{Z} \underline{\phi_1(A, B)} \underline{\phi_2(B, C)}$$

$A[1], B[1], C[1]$
 $A[2], B[2], C[2]$, ... $A[m], B[m], C[m]$

MAXIMUM LIKELIHOOD FOR MARKOV NETWORKS

- Log-likelihood: $P(D) = P_{\phi}(A[1], B[1], C[1]) P_{\phi}(A[2], B[2], C[2]) \dots$

$$\ell(\theta : D) = \sum_m [\ln \phi_1(a[m], b[m]) + \ln \phi_2(b[m], c[m]) - \ln Z(\theta)]$$

- Using sufficient statistics:

$$\ell(\theta : D) = \sum_{a,b} M[a, b] \ln \phi_1(a, b) + \sum_{b,c} M[b, c] \ln \phi_2(b, c) - M \ln Z(\theta)$$

instances

- Partition function:

$$Z(\theta) = \sum_{a,b,c} \phi_1(a, b) \phi_2(b, c)$$

MAXIMUM LIKELIHOOD FOR MARKOV NETWORKS

$$P_\phi(A, B, C) = \sum_{a,b} M[a, b] \ln \phi_1(a, b) + \sum_{b,c} M[b, c] \ln \phi_2(b, c) - M \ln \sum_{a,b,c} \phi_1(a, b) \phi_2(b, c)$$

$M \ln z(0)$

- Partition function couples the parameters
 - ▶ No decomposition of likelihood
 - ▶ No closed form solution for optimization

MAXIMUM LIKELIHOOD FOR LOG-LINEAR MODELS

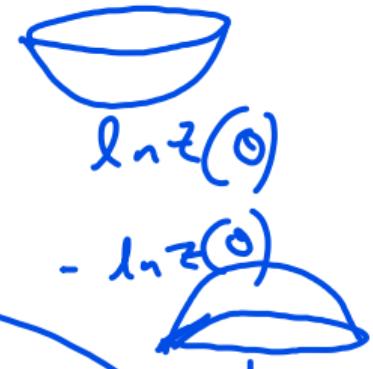
- Use Log-linear representation

$$\frac{P(X_1, \dots, X_n : \theta)}{P(\mathcal{D} : \theta)} = \frac{1}{Z(\theta)} \exp \left[- \sum_{i=1}^k \theta_i f_i(D_i) \right]$$

- Log-likelihood

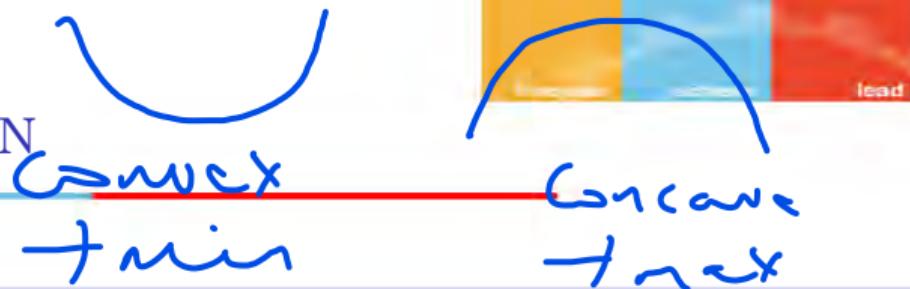
$$\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(x[m]) \right) - M \ln Z(\theta)$$

$$\ln Z(\theta) = \ln \sum_x \exp \left[- \sum_i \theta_i f_i(x) \right]$$



concave

LOG PARTITION FUNCTION



THEOREM

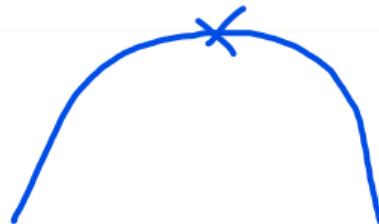
Hess.

$$\frac{\partial}{\partial \theta} \ln Z(\theta) = E_{\theta}[f_i]$$
$$\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln Z(\theta) = Cov_{\theta}[f_i, f_j]$$

$\ln Z(\theta)$ is convex
 $-\ln Z(\theta)$ is concave

- Hessian of $\ln Z(\theta)$ is a covariance matrix which is always positive semidefinite
- Log partition function is a Hessian; hence a convex function.
- Negation of log partition function is a concave function.

LOG LIKELIHOOD FUNCTION


$$\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(x[m]) \right) - M \ln Z(\theta)$$

= Linear function + Concave function

- Log likelihood function is a concave function.
- No local optima
- Easy to optimize using hill climbing or Gradient Ascent method (L-BFGS) to obtain global optima.

MAXIMUM LIKELIHOOD ESTIMATION

- Log likelihood function

*count of the feature i
in the given instance
of the data*

$$\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(x[m]) \right) - M \ln Z(\theta)$$

$$\frac{1}{M} \ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\frac{1}{M} \sum_m f_i(x[m]) \right) - \ln Z(\theta)$$

$\sum_m f_i(x[m])$

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta : \mathcal{D}) = E_{\mathcal{D}}[f_i(X)] - E_{\theta}[f_i]$$

$\ln Z(\theta)$

MAXIMUM LIKELIHOOD ESTIMATION

THEOREM

$\hat{\theta}$ is the Maximum Likelihood Estimate if and only if expectation in the data \mathcal{D} equals the expectation relative to the model for each and every feature.

$$\mathbb{E}_{\mathcal{D}}[f_i(X)] = \mathbb{E}_{\hat{\theta}}[f_i]$$

because $\frac{\partial}{\partial \theta_i} \frac{1}{m} \sum l(\theta; \mathcal{D}) = 0$ at optimal value

TABLE OF CONTENTS

- 1 LEARNING IN MARKOV NETWORK
- 2 PARAMETER ESTIMATION IN MARKOV NETWORKS
- 3 MLE FOR CRF
- 4 MAP ESTIMATION FOR MRF AND CRF

MAXIMUM LIKELIHOOD ESTIMATION FOR CRF

- CRF for target Y given evidence X

$$P_{\theta}(Y | X) = \frac{1}{Z(\theta)} \hat{P}_{\theta}(X, Y)$$

$$Z(\theta) = \sum_Y \hat{P}_{\theta}(X, Y)$$

$$\mathcal{D} = \{x[m], y[m]\}_{m=1}^M \quad M \text{ data instances}$$

MAXIMUM LIKELIHOOD ESTIMATION FOR CRF

- Log conditional likelihood

$$\ell_{Y|X}(\theta : \mathcal{D}) = \sum_{m=1}^M \ln P_\theta(y[m] | x[m], \theta)$$

$$\ell_{Y|X}(\theta : \mathcal{D}) = \sum_i \theta_i f_i(x[m], y[m]) - \ln Z_{x[m]}(\theta)$$

- First partial derivative

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell_{Y|X}(\theta : \mathcal{D}) = \frac{1}{M} \sum_{m=1}^M f_i(x[m], y[m]) - E[f_i(x[m], Y)]$$

What does
this mean?

MAXIMUM LIKELIHOOD ESTIMATION FOR CRF

- Requires inference for each data instance $x[m]$ at each gradient step.
- Requires M inference steps.
- More expensive.
- Likelihood function is concave; optimized using gradient ascent.



TABLE OF CONTENTS

- 1 LEARNING IN MARKOV NETWORK
- 2 PARAMETER ESTIMATION IN MARKOV NETWORKS
- 3 MLE FOR CRF
- 4 MAP ESTIMATION FOR MRF AND CRF

MAP ESTIMATION FOR MRF AND CRF

- MLE may over-fit the parameters to the training data.
- Hence use parameter prior to smooth out the estimates of the parameters.
- In MRF and CRF, the likelihood function cannot be maintained in closed form.
- For regularization, use MAP estimation.

GAUSSIAN PARAMETER PRIOR

- Define a Gaussian distribution over each parameter θ_i with zero mean and a variance σ^2 .

$$P(\theta : \sigma^2) = \prod_{i=1}^k \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-\theta^2}{2\sigma^2}\right]$$

LAPLACIAN PARAMETER PRIOR

- Define a Laplacian distribution over each parameter θ_i using β as the hyperparameter.

$$P(\theta : \beta) = \prod_{i=1}^k \frac{1}{2\beta} \exp\left[\frac{-|\theta|}{\beta}\right]$$



MAP ESTIMATION

- MAP Estimation

$$\arg \max_{\theta} P(\mathcal{D}, \theta) = \arg \max_{\theta} P(\mathcal{D} \mid \theta)P(\theta)$$

- Find the θ that maximizes the joint distribution $P(\mathcal{D}, \theta)$

$$\arg \max_{\theta} P(\mathcal{D}, \theta) = \arg \max_{\theta} [\ell(\theta : \mathcal{D}) + \log P(\theta)]$$



MAP ESTIMATION WITH GAUSSIAN PRIOR

- $\log P(\theta)$ is quadratic
- L2 regularization
- Many parameters are close to zero but not exactly zero.
- Dense – many $\theta \neq 0$



MAP ESTIMATION WITH LAPLACIAN PRIOR

- $\log P(\theta)$ is linear
- L1 regularization
- Push many parameters towards zero.
- Sparse – many $\theta \approx 0$



Thank You for the support and cooperation
for the entire course. :)



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

PROBABILISTIC GRAPHICAL MODELS SESSION # 16 : Some Problems

SRINATH NAIDU

srinath.naidu@pilani.bits-pilani.ac.in



The instructor is gratefully acknowledging
the authors who made their course
materials freely available online.

Problem 1

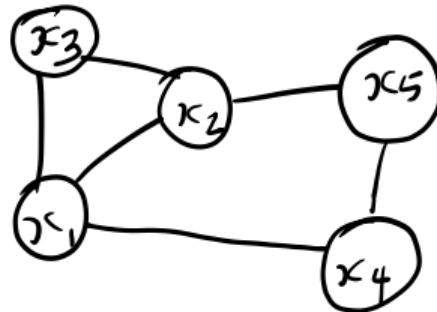
Consider the following Gibbs distribution

$$P(x_1, x_2, \dots, x_5) = \phi_1(x_1, x_2) \phi_2(x_1, x_3) \phi_3(x_1, x_4) \phi_4(x_2, x_3) \phi_5(x_4, x_5)$$

- 1) Visualise this distribution as an undirected graph
- 2) Do we have $x_3 \perp x_4 \mid x_1, x_2$?

Answer - Problem 1

We draw a vertex for each variable x_i . There is an edge between two variables if the variables occur together in a factor. This gives us



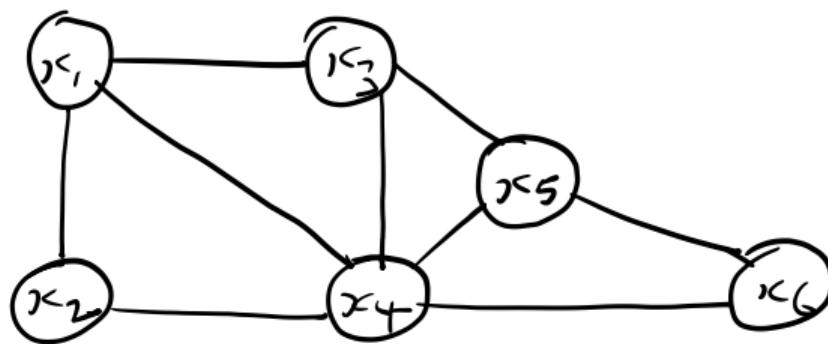
Answer - Problem 1

Given x_1 and x_2 there is no active path between x_3 and x_4 .

Therefore $x_3 \perp x_4 \mid x_1, x_2$

Problem 2

Consider the undirected graph G below:



- what is the set of Gibbs distributions induced by the graph?
- if P factorizes according to G , does $P(x_3/x_2, x_4) = P(x_3/x_4)$ hold?

Answer - Problem 2

a) The maximal cliques in the graph are (x_1, x_2, x_4) ,
 (x_1, x_3, x_4) , (x_3, x_4, x_5) , (x_4, x_5, x_6)

The Gibbs distribution induced by h is

$$p(x_1, x_2, x_3, x_4, x_5, x_6) \propto \phi_1(x_1, x_2, x_4) \phi_2(x_1, x_3, x_4) \\ \phi_3(x_3, x_4, x_5) \phi_4(x_4, x_5, x_6)$$

Answer - Problem 2

To answer part (b) we need to check if x_3 is conditionally independent of x_2 given x_4 .
In graph G , there exists an active path between x_3 and x_2 even when x_4 is observed. Therefore,

$$P(x_3 | x_2, x_4) \neq P(x_3 | x_4).$$

Problem 3

Assume that you have the following Markov
Blankets for all variables $x_1, x_2, x_3, x_4, y_1, y_2, \dots, y_4$

$$MB(x_1) = \{x_2, y_1\}, MB(x_2) = \{x_1, x_3, y_2\}, MB(x_3) = \{x_2, x_4, y_3\}$$

$$MB(x_4) = \{x_3, y_4\}, MB(y_1) = \{x_1\}, MB(y_2) = \{x_2\}$$

$$MB(y_3) = \{x_3\}, MB(y_4) = \{x_4\}$$

Let p be the corresponding pdf. How do we factorize?
(Assume $p \propto y + v_c$).

Answer - Problem 3

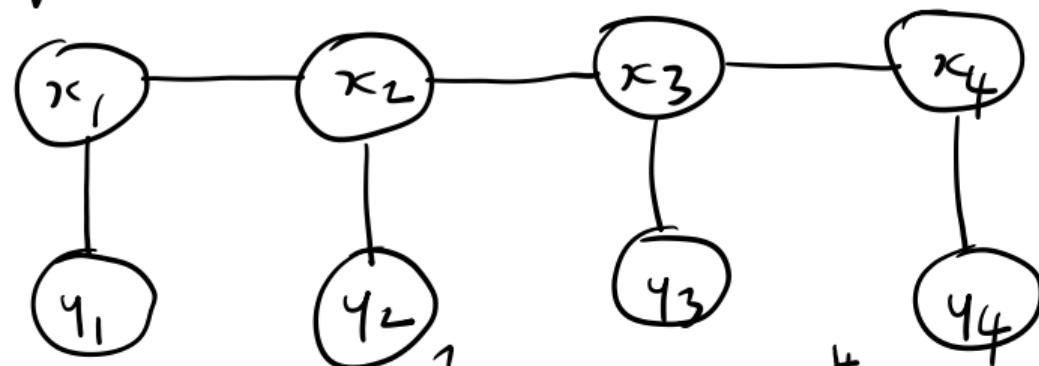
The key idea is that in undirected graphical models the Markov Blanket for a node is its set of neighbours.

Given all the Markov Blankets we know what local Markov property p must satisfy.

For positive distributions we have an equivalence between p satisfying the local Markov property and p factorizing over the graph.

Answer - Problem 3

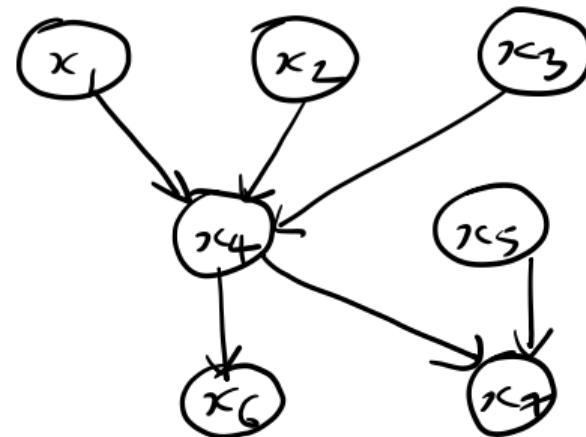
The graph satisfying the Markov Blanket relationships is



The factorization is $\frac{1}{Z} \prod_{i=1}^3 m(x_i, x_{i+1}) \prod_{i=1}^4 g(x_i, y_i)$

PrBLEM 4

For distributions that factorize over the graph below, find the minimal undirected I-map.



Answer - Problem 4

We construct a moralized graph where the unmarried parents of a vertex are married.

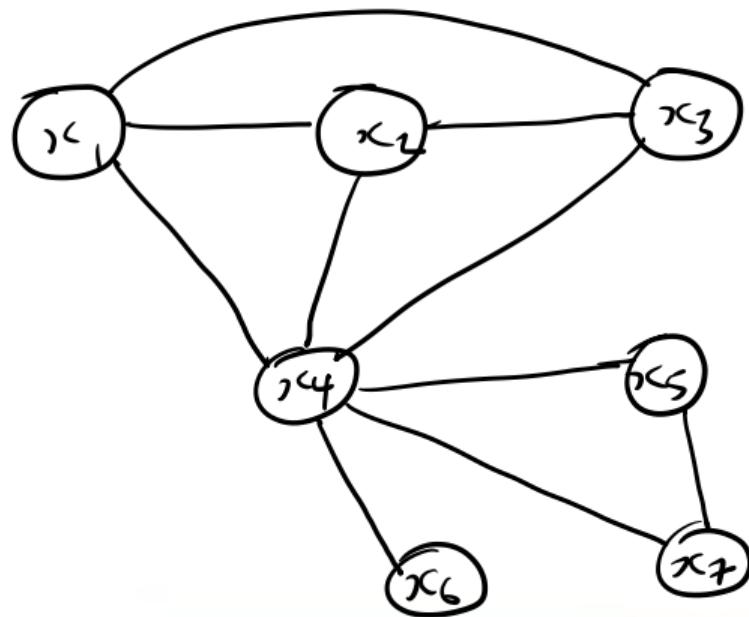
Why do we do this?



if y is specified, α and β are not independent in the directed graph

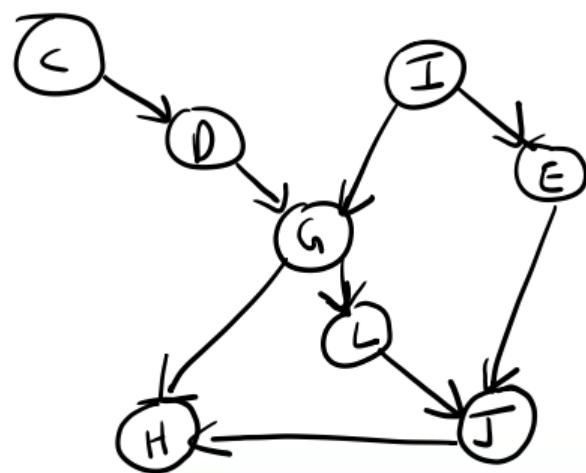
if y is specified
 α and β are independent

Answer - Problem 4



Problem 5

Consider a DAG G associated with a student's performance in some course



C = coherence

I = intelligence

D = difficulty

G = grade

L = letter of recommendation

J = job

E = exam score

H = happy

Problem 5

Construct a clique tree for this example with respect to an instance of variable elimination where ϕ is the set of conditional probabilities $p(x_i | pa_G(x_i))$ associated to G , $Z = \{C, D, I, G, E, L, H\}$ and $<$ is the ordering C, D, I, H, G, E, L

Answer- Problem 5

We first run variable elimination on this DAG for the set of factors $\Phi = \{P(C), P(D|C), P(I), P(G|D, I), P(H|G, I), P(J|E, C), P(H|G, J)\}$

and the ordering C, D, I, H, G, E, J . We then compute

$$(a) \psi_1(C, D) = P(C) P(D|C), \tau_1(D) = \sum_C \psi_1(C, D)$$

$$(b) \psi_2(G, D, I) = \tau_1(D) P(G|D, I), \tau_2(G, I) = \sum_D \psi_2(G, D, I)$$

Answer - Problem 5

- (c) $\psi_3(G, E, I) = \tau_2(G, I) P(I) P(E|I)$, $\bar{\tau}_3(G, E) = \sum_I \psi_3(G, E, I)$
- (d) $\psi_4(H, G, J) = P(H|G, J)$, $\bar{\tau}_4(G, J) = \sum_H \psi_4(H, G, J)$
- (e) $\psi_5(G, J, E, L) = \tau_4(G, J) \tau_3(G, E) P(L|G)$, $\bar{\tau}_5(J, E, L) = \sum_G \psi_5(G, J, E, L)$
- (f) $\psi_6(J, E, L) = \tau_5(J, E, L) P(J|E, L)$, $\bar{\tau}_6(J, L) = \sum_E \psi_6(J, E, L)$
- (g) $\psi_7(J, L) = \tau_6(J, L)$, $\bar{\tau}_7(J) = \sum_L \psi_7(J, L)$

Answer - Problem 5

We then create one clique for each φ_i consisting of the variables in $\text{Scope}(\varphi_i)$

(a) $C_1 = \{C, D\}$

(f) $C_6 = \{J, E, C\}$

(b) $C_2 = \{D, G, J\}$

(g) $C_7 = \{J, C\}$

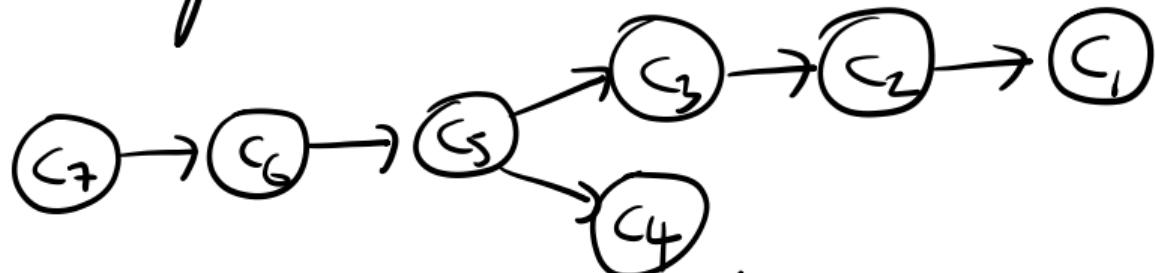
(c) $C_3 = \{G, E, I\}$

(d) $C_4 = \{H, G, J\}$

(e) $C_5 = \{G, J, E, C\}$

Answer - Problem 5

We then create a clique tree by connecting cliques c_i and c_j by the edges $c_i \leftarrow c_j$ if ψ_j is defined by τ_i . We get the following clique tree:



Here we specify a directed clique tree, taking c_7 as the root.

Problem 6

Let ϕ denote the set of factors in the previous problem. Consider the clique tree



$$\text{where } C_1 = \{C, D\}, C_2 = \{D, I, G\}, C_3 = \{G, E, I\},$$

$$C_4 = \{G, H, J\}, C_5 = \{G, J, L, E\}$$

Assign each factor a clique: $\alpha(P(C)) = C_1, \alpha(P(D/C)) = C_1,$
 $\alpha(P(G/I, D)) = C_2, \alpha(P(I)) = C_3, \alpha(P(E/I)) = C_3, \alpha(P(H/G, J)) = C_4$

Problem 6

$$\alpha(p(\ell|g)) = c_5, \alpha(p(j|\ell, e)) = c_5.$$

- (a) what are the initial potentials of the given clique tree?
- (b) With c_5 as root compute $p_v(c_j)$ using sum-product messaging.

Answer - Problem 6

(a) The initial potential ψ_i for digue C_i is obtained by multiplying together all the factors assigned to C_i .

$$\psi_1 = P(G) P(I/G)$$

$$\psi_2 = P(G/I, \rho)$$

$$\psi_3 = P(I) P(E/I)$$

$$\psi_4 = P(H/G, J)$$

$$\psi_5 = P(L/G) P(J/L, \vartheta)$$

Answer - Problem 6

(b) We fix $c_5 + b$ to be the root and recurrent edges of the clique tree $S =$ that C_5 is the root



The subsets are respectively $S_{1,2} = \{D\}$, $S_{2,3} = \{G, I\}$
 $S_{3,4} = \{G, E\}$ $S_{4,5} = \{G, J\}$

Answer - Problem 6

The message $s_{i \rightarrow j}$ is computed by summing out the variables in c_i that are not in $s_{i,j}$ which labels the edge between c_i and c_j

$$s_{1 \rightarrow 2}(D) = \sum_c \psi_1(c, D)$$

$$s_{2 \rightarrow 3}(G, I) = \sum_D \psi_2(G, I, D) s_{1 \rightarrow 2}(D)$$

$$s_{3 \rightarrow 5}(G, E) = \sum_I \psi_3(E, I) s_{2 \rightarrow 3}(G, I)$$

$$s_{4 \rightarrow 5}(G, J) = \sum_H \psi_4(G, H, J)$$

Answer - Problem 6

$$\text{Hence the beliefs } \beta_5(g, J, L, E) = \psi_5(g, J, L, E) \delta_{3 \rightarrow 5}(g, J) \\ \delta_{4 \rightarrow 5}(g, J)$$

We can only compute β at the root since it has received all its messages



Problem 7

A mouse moves along a tiled corridor with $2m$ tiles where $m > 1$. When $i \neq 1, 2m$, it moves to tile $i+1$ or $i-1$ with equal probability. From the boundary tiles it moves right or left with probability 1.

Each time the mouse moves to a tile $i \leq m$ or $i > m$ an electronic device outputs a signal L or R respectively. Can the generated sequence of signals L and R be described as a Markov chain with states L and R?

Answer - Problem 7

To see if the sequence is a Markov chain we need to check if the next state is determined only by the current state.

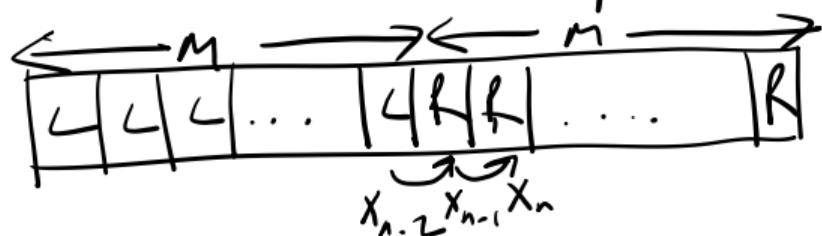


$$P(X_{n+1} = L \mid X_n = f, X_{n-1} = L) = \frac{1}{2} \quad \text{since we are on the border, going left}$$

will generate L and going right will generate f

Answer - Problem 7

What about $P(X_{n+1} = L \mid X_n = R, X_{n-1} = f, X_{n-2} = L)$?



We are now deep into R territory so that whether we move left or right from X_n we will not see a L.

$$\therefore P(X_{n+1} = L \mid X_n = R, X_{n-1} = f, X_{n-2} = L) = 0$$

Problem 8

Consider the same scenario as in Problem 7
except that the device outputs L or R when the
mouse moves to tile 1 or tile $2m$, and not when
 $i \leq m$ or $i \geq m$.

Can the generated sequence of L and R now be
described as a Markov chain with states L and R?

Answer - Problem 8

In this case the sequence of states can be described as a Markov chain.

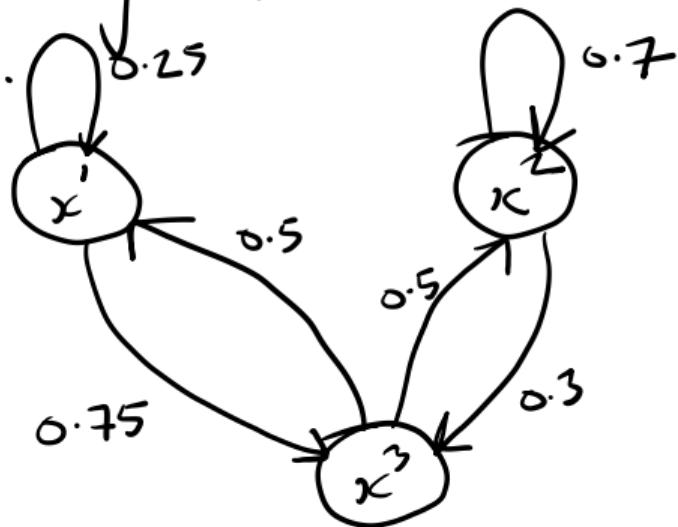
At any time t given that the mouse has just moved to position i , the probability of any event that concerns its position in future depends only on its current state

$$P(X_{n+1} = x_{n+1} / X_n = L, X_{n-1} = x_{n-1}, \dots, X_1 = x_1)$$

$$= f(x_{n+1} / X_n = L) \text{ and similarly } f(x_n = R)$$

Problem 9

Find the stationary distribution for the following Markov chain.



Answer - Problem 9

For a stationary distribution $p^t(x') \approx p^{t+1}(x')$

$$\text{where } p^{t+1}(x') = \sum_{x \in Val(x)} p^t(x) T(x \rightarrow x')$$

$$\left. \begin{array}{l} \pi(x') = 0.25\pi(x') + 0.5\pi(x^3) \\ \pi(x^2) = 0.7\pi(x^2) + 0.5\pi(x^3) \\ \pi(x^3) = 0.75\pi(x^1) + 0.3\pi(x^2) \end{array} \right\} \text{three transition equations}$$

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

Answer - Problem 9

We can set up the problem in Matrix terms as follows:

$$\begin{bmatrix} 0.25 & 0 & 0.5 \\ 0 & 0.7 & 0.5 \\ 0.75 & 0.3 & 0 \end{bmatrix} \begin{bmatrix} \pi(x^1) \\ \pi(x^2) \\ \pi(x^3) \end{bmatrix} = \begin{bmatrix} \pi(x^1) \\ \pi(x^2) \\ \pi(x^3) \end{bmatrix} \Rightarrow Ax = x$$

We need to fit the eigenvector to the matrix above which corresponds to eigenvalue 1. Verify that any matrix whose columns add up to 1 will have eigenvalue 1.

Answer - Problem 9

$$Ax = \lambda x \Rightarrow \det(A - \lambda I) = 0 \Rightarrow \det(A - I) = 0 \text{ for } \lambda = 1$$

$$(A - I)x = 0$$

$$\begin{bmatrix} -0.75 & 0 & 0.5 \\ 0 & -0.3 & 0.5 \\ 0.75 & 0.3 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

After Gaussian elimination
we get ...

Answer - Problem 9

$$\begin{bmatrix} -0.75 & 0 & 0.5 \\ 0 & -0.3 & 0.5 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

x_3 is a free variable; x_1 and x_2 are pivot variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2x_3/3 \\ 5x_3/3 \\ x_3 \end{bmatrix}$$

Answer - Problem 9

How do we decide the value of x_3 ?

$$\begin{bmatrix} \pi(x^1) \\ \pi(x^2) \\ \pi(x^3) \end{bmatrix} = \begin{bmatrix} 2\alpha/3 \\ 5\alpha/3 \\ \alpha \end{bmatrix} \quad \pi(x^1) + \pi(x^2) + \pi(x^3) = 1$$

$$\frac{7\alpha}{3} + \alpha = 1$$

$$\Rightarrow \alpha = \frac{3}{10}$$

$$\begin{bmatrix} \pi(x^1) \\ \pi(x^2) \\ \pi(x^3) \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.5 \\ 0.3 \end{bmatrix}$$

Problem 10

Build a Chow-Liu tree for the following empirical data

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

Answer - Problem 10

Obtain weights for every pair of vertices where
the weight = mutual information

For example

$$I_{A,B} = \sum_{A,B} p(A,B) \log_2 \frac{p(A,B)}{p(A)p(B)}$$

The various probabilities are determined
empirically

Answer - Problem 10

Let us calculate the edge weight between A and B.

$$I_{A,B} = \sum_m p(A, B) \log_2 \frac{p(A, B)}{p(A)p(B)}$$

$$\text{We have } p(A=0, B=0) = \frac{3}{5}, \quad p(A=0, B=1) = \frac{1}{5}$$

$$p(A=1, B=0) = \frac{1}{5}$$

$$p(A=0) = \frac{4}{5}$$

$$p(B=0) = \frac{4}{5}$$

Answer - Problem 1 =

$$\begin{aligned}
 I(A, B) &= P_{A,B}(0,0) \log_2 \frac{P_{AB}(0,0)}{P_A(0)P_B(0)} + P_{AB}(0,1) \log_2 \frac{P_{AB}(0,1)}{P_A(0)P_B(1)} \\
 &\quad + P_{AB}(1,0) \log_2 \frac{P_{AB}(1,0)}{P_A(1)P_B(0)} \\
 &= \frac{3}{5} \log_2 \frac{\frac{3}{5}}{\left(\frac{4}{5}\right)\left(\frac{4}{5}\right)} + \frac{1}{5} \log_2 \frac{\frac{1}{5}}{\left(\frac{4}{5}\right)\left(\frac{1}{5}\right)} + \frac{1}{5} \log_2 \frac{\frac{1}{5}}{\frac{4}{5} \frac{1}{5}} \\
 &= 0.07
 \end{aligned}$$

Answer - Problem 1.Similarly for $I_{A, C}$

$$P(A=0, C=0) = \frac{1}{5}$$

$$P(A=0, C=1) = \frac{3}{5}$$

$$P(A=1, C=0) = \frac{1}{5}$$

$$P(A=1, C=1) = \frac{0}{5}$$

$$P(A=0) = \frac{4}{5}$$

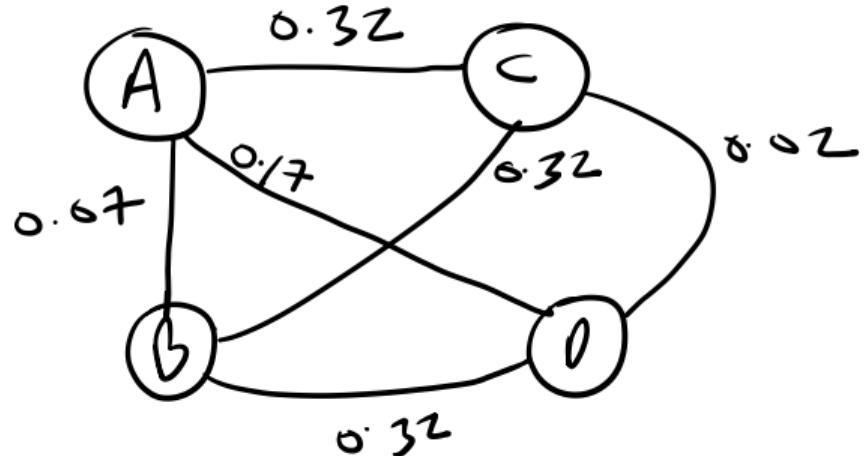
$$P(C=0) = \frac{2}{5}$$

Answer - Problem 10

$$\begin{aligned}
 I_{A,C} &= P_{AC}(0,0) \log_2 \frac{P_{AC}(0,0)}{P_A(0)P_C(0)} + P_{AC}(0,1) \log_2 \frac{P_{AC}(0,1)}{P_A(0)P_C(1)} \\
 &\quad + P_{AC}(1,0) \log_2 \frac{P_{AC}(1,0)}{P_A(1)P_C(0)} \\
 &= \frac{1}{5} \log_2 \frac{\frac{1}{5}}{\frac{8}{25}} + \frac{3}{5} \log_2 \frac{\frac{3}{5}}{\left(\frac{4}{5}\right)\left(\frac{3}{5}\right)} + \frac{1}{5} \log_2 \frac{\frac{1}{5}}{\left(\frac{1}{5}\right)\left(\frac{2}{5}\right)} \\
 &= 0.32
 \end{aligned}$$

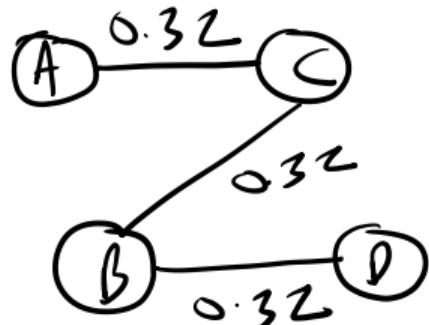
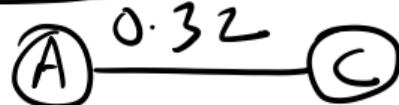
Answer - Problem 10

We get the following graph:



Answer- Problem 10

To construct maximum weight spanning tree:
start with



Each time we add the max weight edge that does not create a cycle

Problem 11

Prior information about the parameters of a biased coin with sample space $[x^1, x^2]$ suggests that the parameter vector (θ^1, θ^2) obeys a Dirichlet distribution with parameters 2 and 3 respectively. Let θ_1 stand for the outcome x^1 and θ_2 for the outcome x^2 . What is $P[X[1] = x^1]$? What is the smallest number of samples M that we need in order to conclude that $P(X[M+1] = x^1 | D) = \frac{1}{2}$ where $D = X[1], X[2] \dots X[M]$?

Answer - Problem 11

We have $P(X[1] = x^1) = \frac{\alpha_1}{\alpha_1 + \alpha_2} = \frac{2}{2+3} = \frac{2}{5}$

Further $P(X[M+1] = x^1 | D) = \frac{M[1] + \alpha_1}{M + \alpha_1 + \alpha_2}$ where $M = M[1] + M[2]$

We need $P(X[M+1] = x^1 | D) = \frac{1}{2}$

This means

$$\frac{M[1] + 2}{M[1] + M[2] + 5} = \frac{1}{2}$$

$$\Rightarrow 2M[1] + 4 = M[1] + M[2] + 5$$

Answer - Problem 11

This gives $M[1] = M[2] + 1$.

The smallest possible value for $M[2] = 0$ which gives

$$M[1] = 1$$

Thus $M = M[1] + M[2] = 1$ is the smallest number of samples we need in order to conclude that

$$P(X[M+1] = X) = \frac{1}{2}$$