NUS National University of Singapore | School of Computing

CS5340
Uncertainty Modeling in AI

Lecture 3:
Bayesian Networks
(Directed Graphical Models)

"independence diagrams" or "conditional independence diagrams"

Harold Soh

AY 2022/23

Semester 2

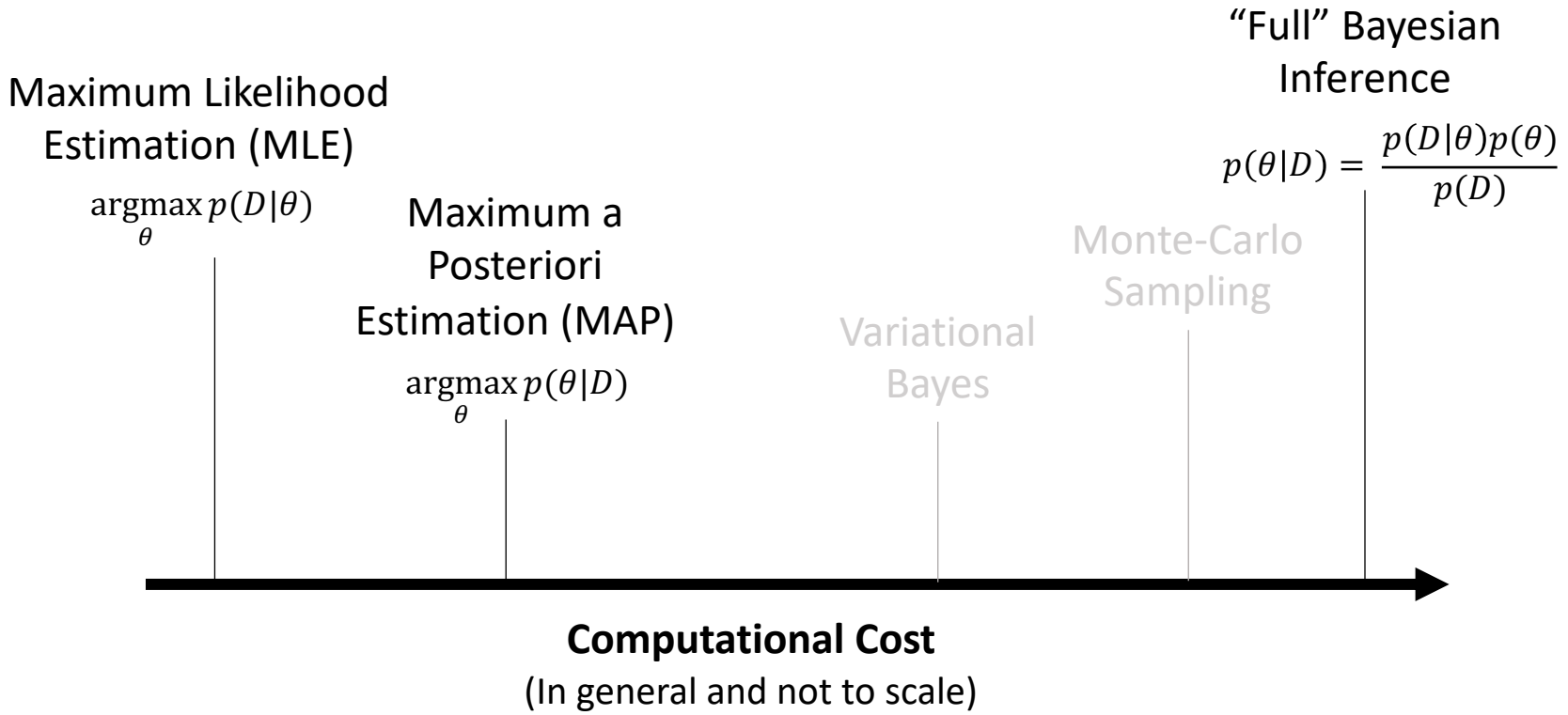
NUS National University of Singapore | School of Computing

Recap from Lecture 2

MLE, MAP, Bayes Inference, and Exponential Families

Learning Parameters

- Common approaches to learn the unknown parameters θ from a set of given data $\mathcal{D} = \{x[1], \ldots, x[N]\}$:



Exponential Family

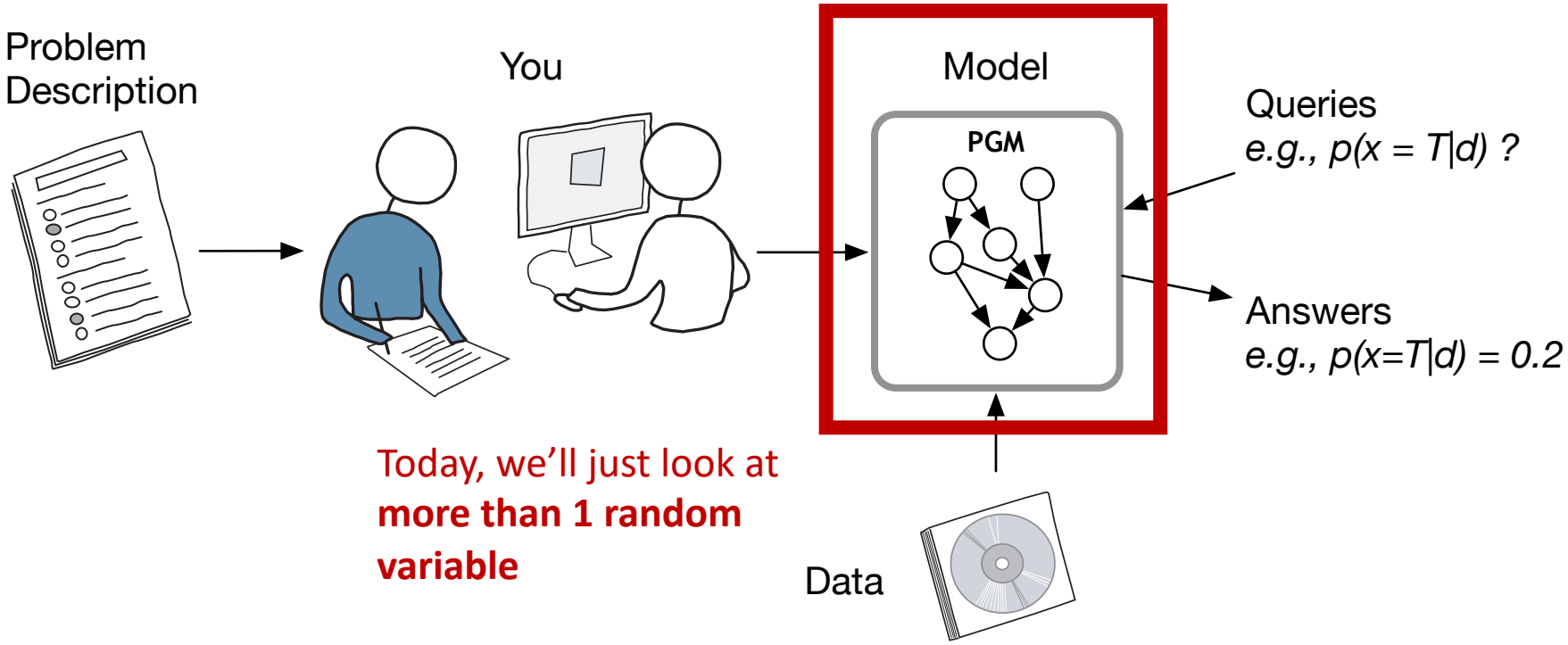
- An exponential family (ExpFam) is a set of probability distributions $\{p_\theta : \theta \in \Theta\}$ with the form

$$p_\theta(x) = \frac{h(x)\exp[\eta(\theta)^\top s(x)]}{Z(\theta)}$$

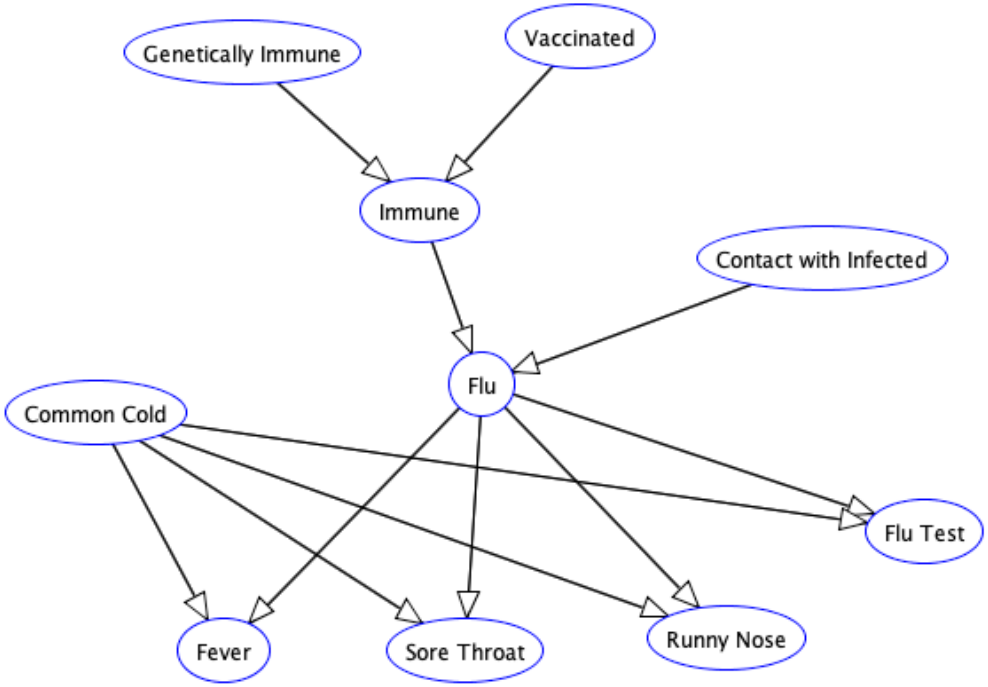
- where:
  - $\theta \in \Theta \subseteq \mathbb{R}^k, x \in \mathbb{R}^d$
  - Natural parameters: $\eta(\theta): \Theta \to \mathbb{R}^m$
  - Sufficient statistics: $s(x): \mathbb{R}^d \to \mathbb{R}^m$
  - Base Measure (Support and scaling): $h(x): \mathbb{R}^d \to [0, \infty)$
  - Partition function: $\mathrm{Z}(\theta): \Theta \to [0, \infty)$

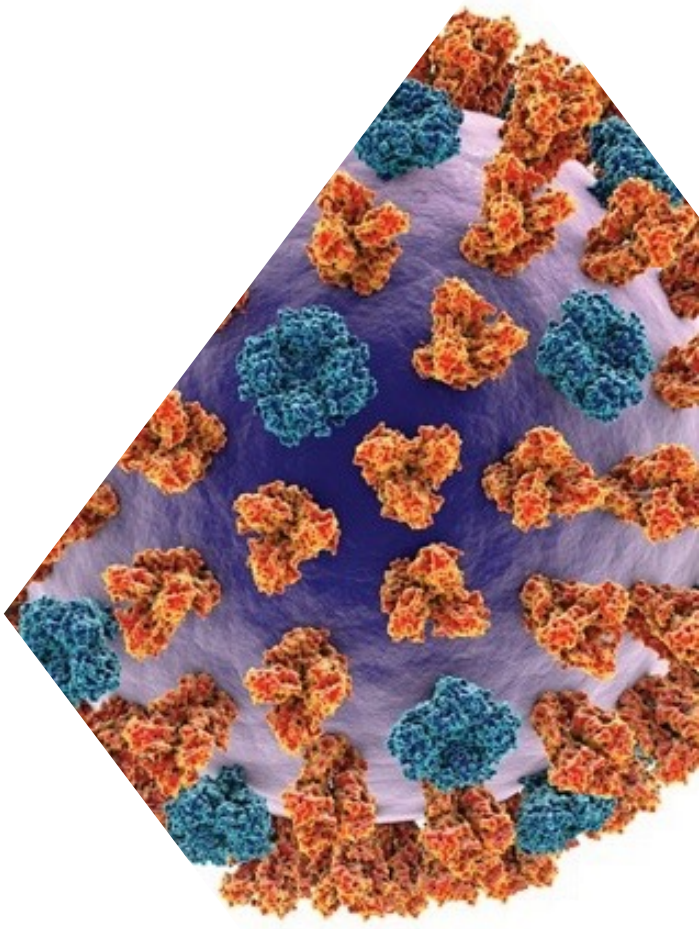
CS5340 in a nutshell

CS5340 is about how to "**represent**" and "**reason**" with uncertainty in a computer.



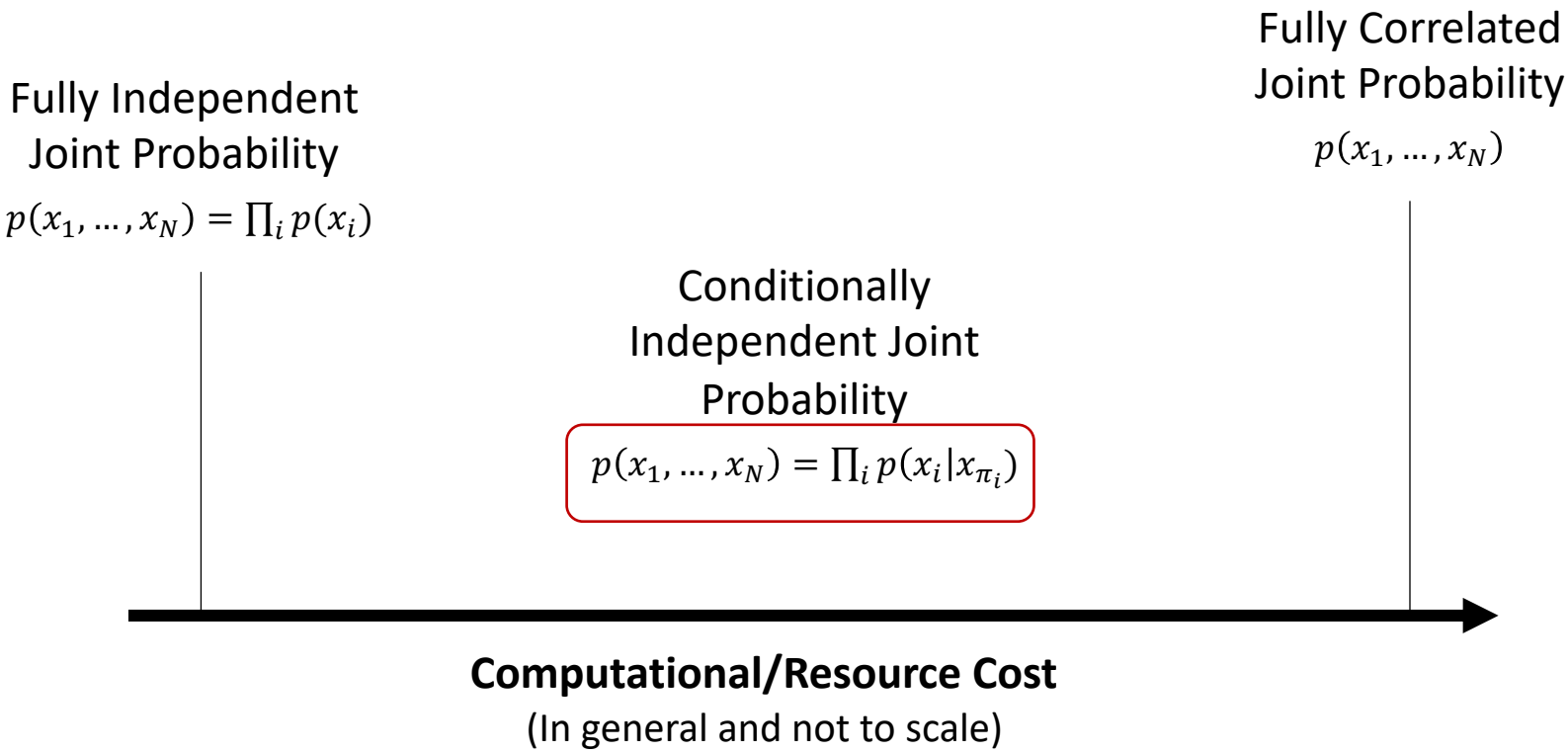
Generative (Causal) Modeling of Relationships between Variables





Representing the Joint Probability

- Random variables $X = \{X_1, X_2, \ldots, X_N\}$



Course Schedule

Week	Date	Lecture Topic	Tutorial Topic
1	12 Jan	Introduction to Uncertainty Modeling + Probability Basics	Introduction
2	19 Jan	Simple Probabilistic Models	Probability Basics
3	26 Jan	Bayesian networks (Directed graphical models)	More Basic Probability
4	2 Feb	Markov random Fields (Undirected graphical models)	DGM modelling and d-separation
5	9 Feb	Variable elimination and belief propagation	MRF + Sum/Max Product
6	16 Feb	Factor graph and the junction tree algorithm	**Quiz 1**
-	-	RECESS WEEK	
7	2 Mar	Mixture Models and Expectation Maximization (EM)	Linear Gaussian Models
8	9 Mar	Hidden Markov Models (HMM)	Probabilistic PCA
9	**16** Mar	Monte-Carlo Inference (Sampling)	Linear Gaussian Dynamical System
10	23 Mar	Variational Inference	MCMC + Sequential VAE
11	30 Mar	Inference and Decision-Making (Special Topic)	**Quiz 2**
12	6 Apr	Gaussian Processes (Special Topic)	Wellness Day
13	13 Apr	**Project Presentations**	Closing

Learning Outcomes

- Students should be able to:

1. Explain the concepts of conditional independence.
2. Use the Bayesian network to represent conditional independence in joint distributions.
3. Describe d-separation using the three canonical 3-node graph.
4. Explain the theoretical foundations of Bayes Nets: Independence-Maps (I-maps), I-equivalence, faithfulness, soundness, completeness, and Perfect-Maps.
5. Describe and use additional DGM notation (plates, points).
6. Explain the Bayes net structure of linear/ridge regression and Naïve Bayes.
7. Extra (in Appendix): Describe the Bayes Ball algorithm for finding d-separation.

Acknowledgements

- A lot of slides and content of this lecture are adapted from:

1. “An introduction to probabilistic graphical models”, Michael I. Jordan, 2002 http://people.eecs.berkeley.edu/~jordan/prelims/chapter2.pdf (Section 2.1)
2. "Probabilistic graphical models", Koller and Friedman (Chapter 3)
3. "Pattern recognition and machine learning", Christopher Bishop (Chapter 8, Section 8.1 and 8.2).
4. "Machine learning - a probabilistic approach", Kevin Murphy (Chapter 10)
5. Slide from Dr. Lee Gim Hee

Conditional Independence

Modeling and Inference

From 1 to N Random Variables

- In the previous lecture, we have looked at fitting probability models (learning), and predictive density (inference).
- But we have looked at the case of only ONE random variable, i.e. $p(x|\theta)$
- How about a joint probability with N random variables, i.e. $p(x_1, \ldots x_N|\theta)$?

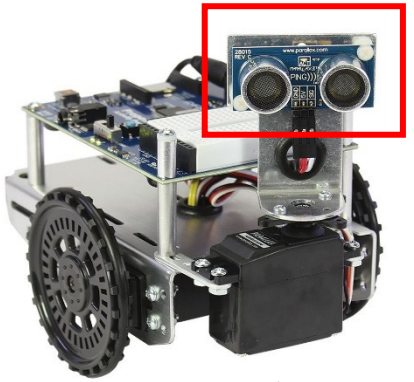
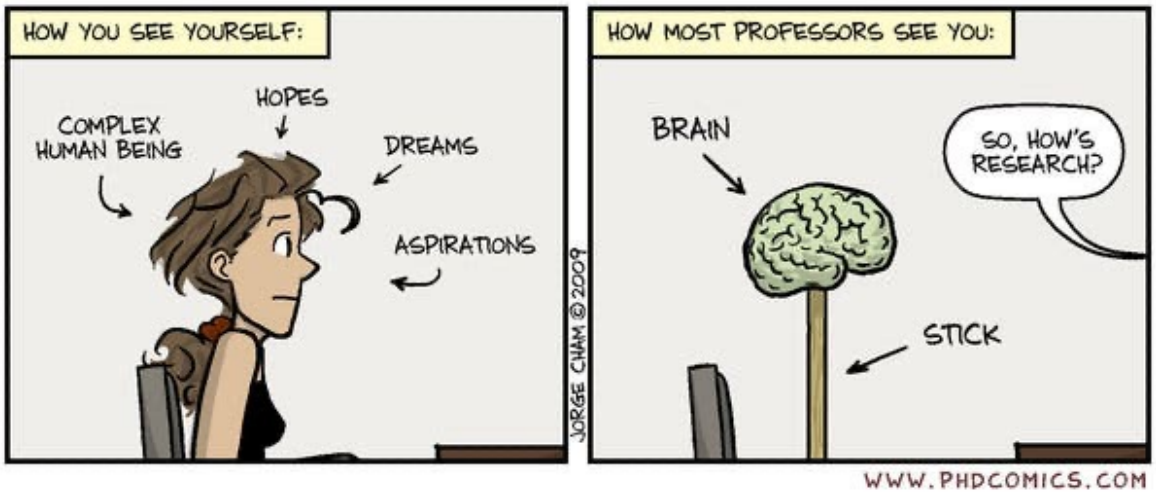


Image credit: https://www.parallax.com/product/910-28015a

Example: Research Output

- Let's consider a model for research output for a given student.
- First, some random variables:
  - Paper accepted! (X_6)
  - Novel idea (X_5)
  - Github code (X_4)
  - Background knowledge (X_3)
  - Technical skill (X_2)
  - Taken CS5340 (X_1)



Example: Research Output

- How to represent the joint probability $p(x_1, \ldots, x_6)$?
- Consider each random variable is *binary*.

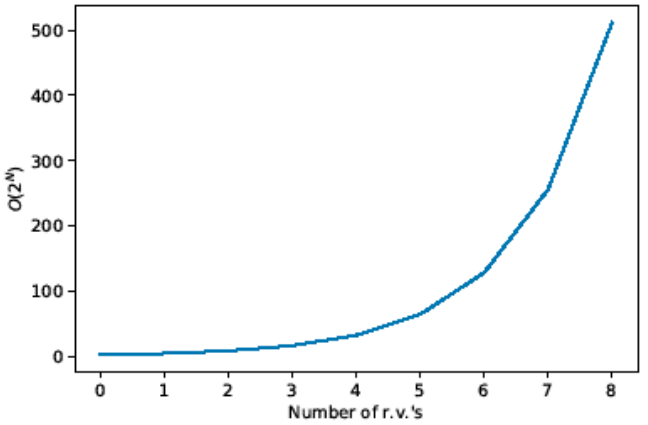
Example: Research Output

- How to represent the joint probability $p(x_1, \ldots, x_6)$?
- Consider each random variable is *binary*.
- $p(x_1, \ldots, x_6)$ can be written as a probability table.
- Q: How many parameters are needed?

Example: Research Output

- How to represent the joint probability $p(x_1, \ldots, x_6)$?
- Consider each random variable is *binary*.
- $p(x_1, \ldots, x_6)$ can be written as a probability table.
- Q: How many parameters are needed?

$$2^N - 1 = O(2^N)$$



Example: Research Output

- How to represent the joint probability $p(x_1, \dots, x_6)$?
- Alternative factorized representation

$$p(x_1, \dots, x_6) = p(x_6|x_5, x_4, x_3, x_2, x_1)p(x_5|x_4, x_3, x_2, x_1) \\ p(x_4|x_3, x_2, x_1)p(x_3|x_2, x_1)p(x_2|x_1)p(x_1)$$

- Consider each random variable is *binary*.
- So, each $p(x_i|x_j, \dots)$ can be written as a conditional probability table.
- Q: How many parameters are needed?

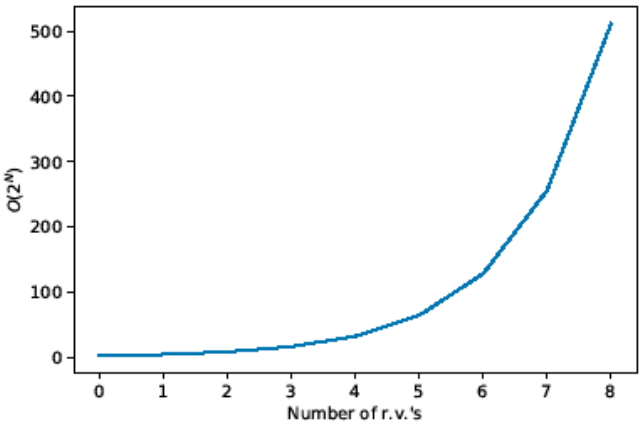
Example: Research Output

$$p(x_1, \dots, x_6) = p(x_6|x_5, x_4, x_3, x_2, x_1)p(x_5|x_4, x_3, x_2, x_1)$$
$$p(x_4|x_3, x_2, x_1)p(x_3|x_2, x_1)p(x_2|x_1)p(x_1)$$

Example: Research Output

$$p(x_1, \ldots, x_6) = p(x_6|x_5, x_4, x_3, x_2, x_1)p(x_5|x_4, x_3, x_2, x_1)$$
$$p(x_4|x_3, x_2, x_1)p(x_3|x_2, x_1)p(x_2|x_1)p(x_1)$$

$p(x_1)$ needs 2 parameters
$p(x_2|x_1)$ needs 4 parameters
$p(x_3|x_2, x_1)$ needs 8 parameters
…
$p(x_i|x_{i-1}, \ldots, x_1)$ needs $O(2^i)$ parameters

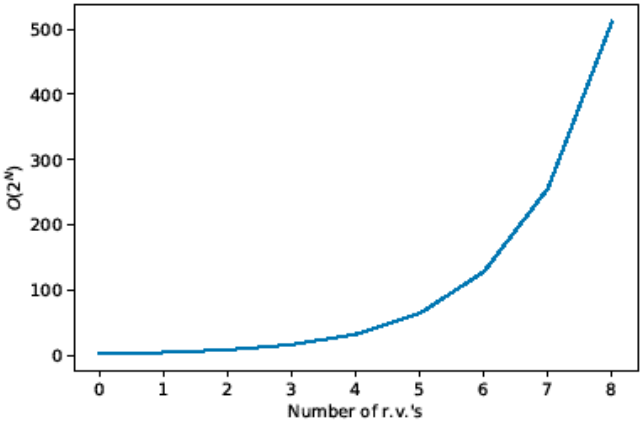


So in total, we need $\sum_{i=1}^{N} O(2^i) = O(2^N)$ parameters

Note: you *actually* need fewer parameters per CPT than stated above. Why?

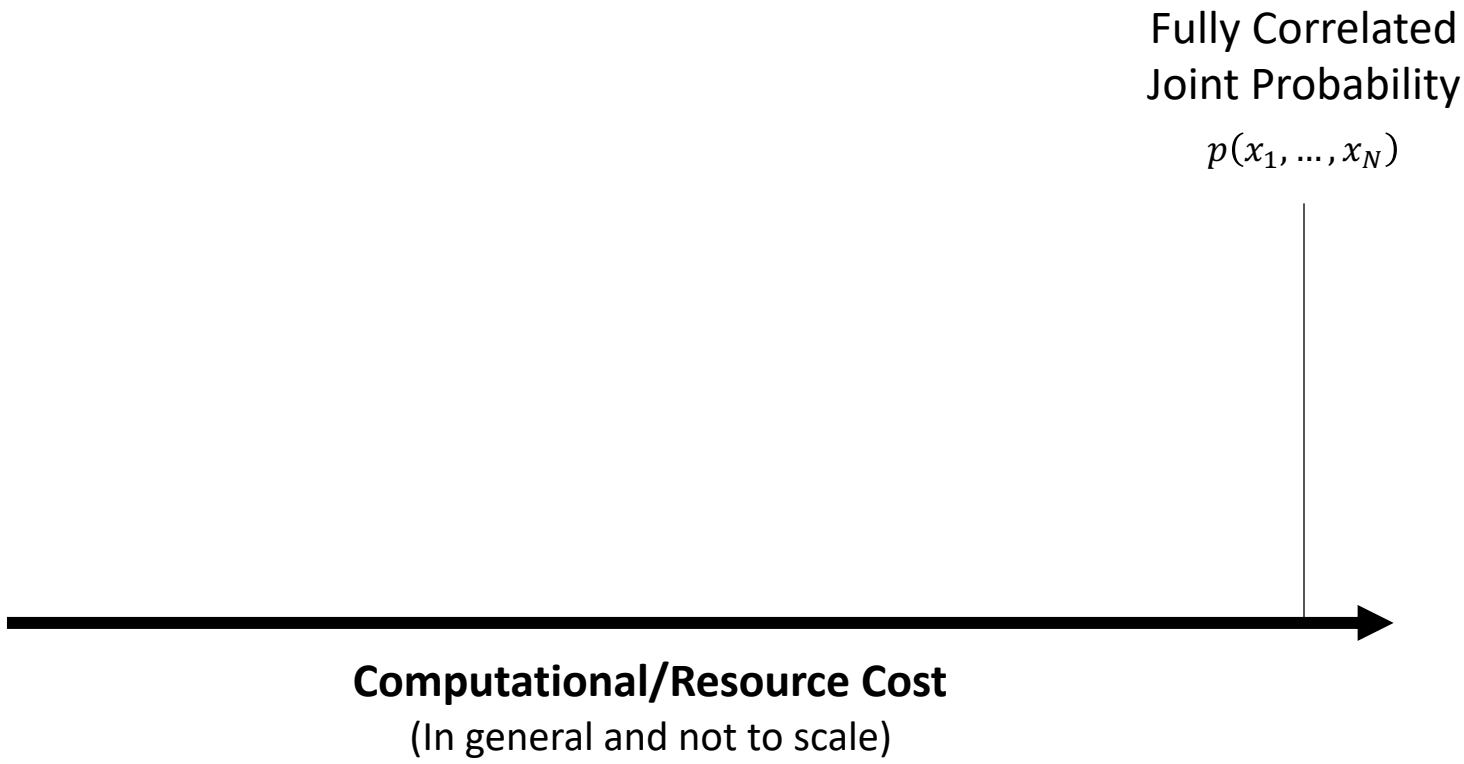
Representational and Inference Constraints

- Assume N discrete random variables $x_1, \ldots, x_N$, where $x_i \in \{1, \ldots, K\}$.
- In general, we need $O(K^N)$ parameters to represent the joint distribution $p(x_1, \ldots, x_N)$.
- Inference becomes intractable when N is large, and a huge amount of data is needed to learn all parameters.



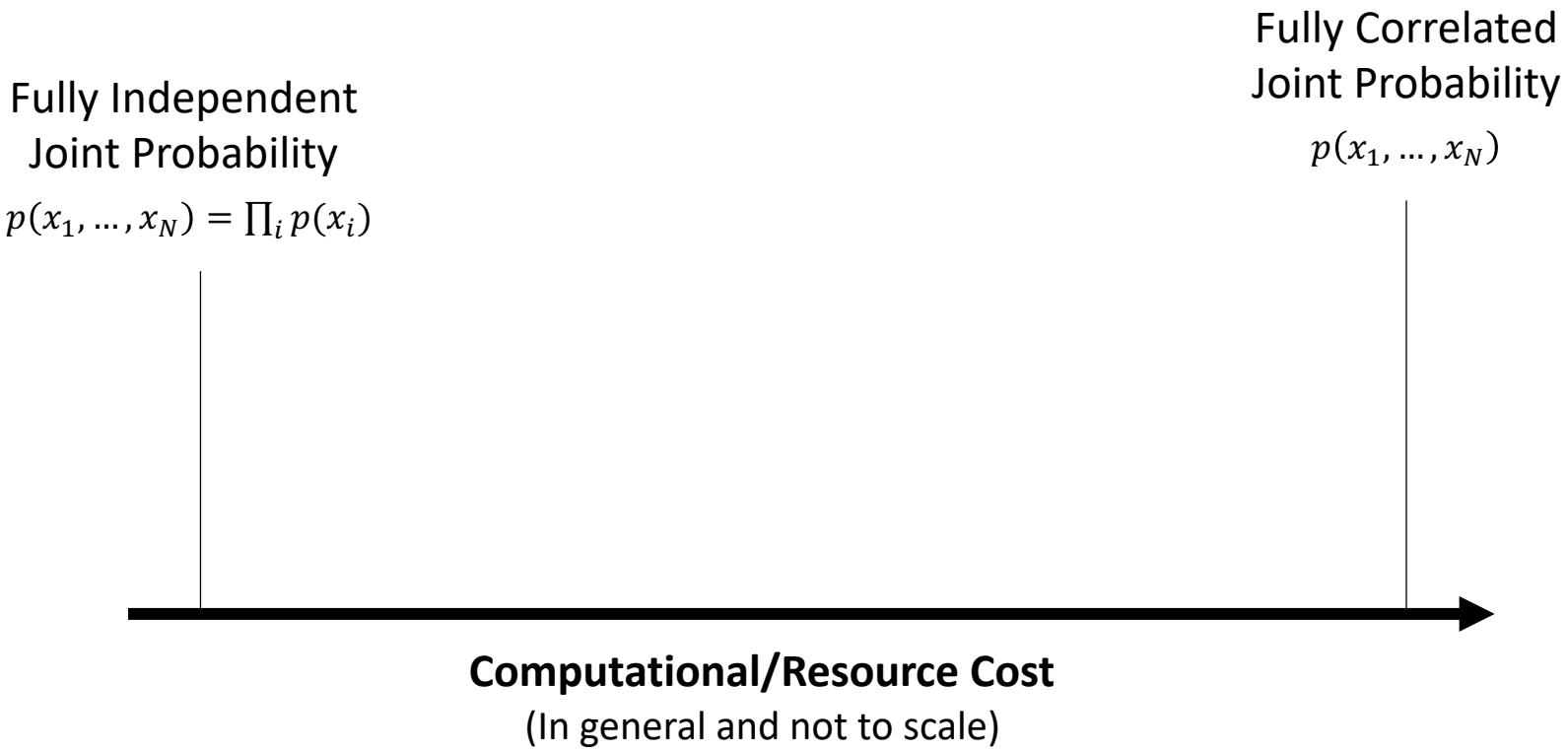
Representing the Joint Probability

- Random variables $X = \{X_1, X_2, \ldots, X_N\}$



Representing the Joint Probability

- Random variables $X = \{X_1, X_2, \ldots, X_N\}$



Assume Complete Independence

Easy solution?

- Assume all random variables are independent
- Reduces the number of parameters to $O(NK)$.

$$p(x_1, \ldots, x_N \,|\theta) = \prod_{i=1}^{N} p(x_i|\theta_i)$$

- Inference becomes tractable products of $p(x_i|\theta_i)$, and smaller amount of data is needed to learn all parameters.

Example: Research Output

- How to represent the joint probability $p(x_1, \dots, x_6)$?
- Let's write out the factorization.

$$p(x_1, \dots, x_6) = p(x_6)p(x_5)p(x_4)p(x_3)p(x_2)p(x_1)$$

- Consider each random variable is *binary*.
- So, each $p(x_i)$ can be written as a conditional probability table.
- Q: How many parameters are needed? **12.**

(or 6 if consider constraints)

But are the r.v.'s really independent?

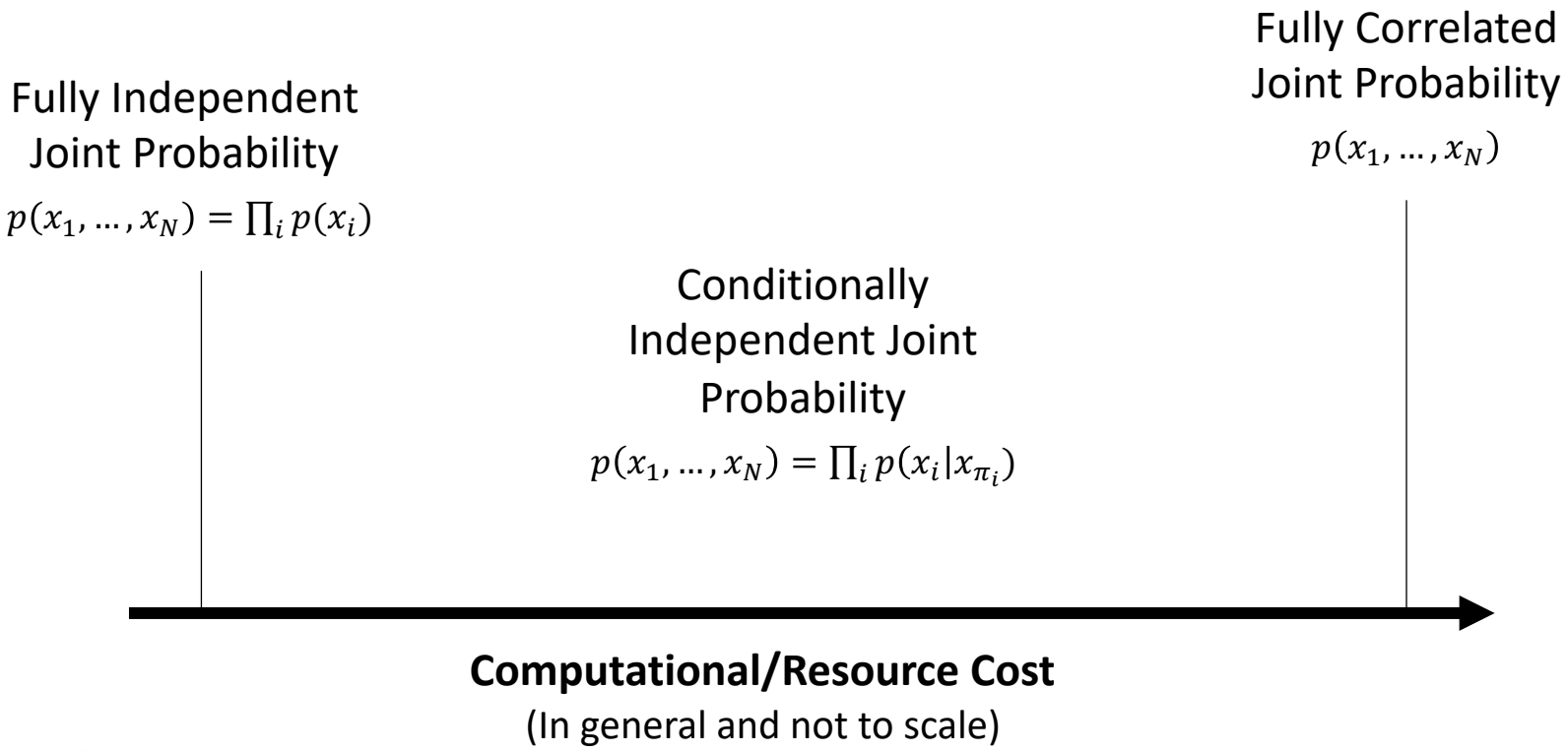
- For our running example:
  - Paper accepted! (X_6)
  - Novel idea (X_5)
  - Github code (X_4)
  - Background knowledge (X_3)
  - Technical skill (X_2)
  - Taken CS5340 (X_1)
- Is your technical skill unrelated to your ability to write code?
- Is having taken (and aced) CS5340 unrelated to our background knowledge in Uncertainty Modelling?

Conditional Independence

- Note that:
  - Real-world random variables are unlikely to be fully independent.
  - Fully correlated joint distributions can become intractable.
- Compromise by assuming an intermediate degree of dependency among the random variables.

Representing the Joint Probability

- Random variables $X = \{X_1, X_2, \ldots, X_N\}$



Conditional Independence

- **Definition:** Two random variables X_A and X_C are conditionally independent given X_B, denoted

$$X_A \perp X_C \mid X_B$$

if and only if:

$$p(x_A, x_C|x_B) = p(x_A|x_B)p(x_C|x_B)$$

- Or alternatively:

$$p(x_A|x_B, x_C) = p(x_A|x_B), \qquad \forall\, X_B : p(x_B) > 0$$

- That is, learning the values of X_C does not change prediction of X_A once we know the value of X_B.

Example: Research Output

- Let's consider a model for research output for a given student.
- First, some random variables:
  - Paper accepted! (X_6)
  - Novel idea (X_5)
  - Github code (X_4)
  - Background knowledge (X_3)
  - Technical skill (X_2)
  - Taken CS5340 (X_1)

Bayesian Networks: Joint Probability

- Locality of the parent-child relationship is used to construct economical representations of the joint distribution.
- The parent-child represents conditional independence:

$$p(x_i|x_{\pi_i})$$

- Joint probability can be read off the graph as the product of all local conditional independence:

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i|x_{\pi_i})$$

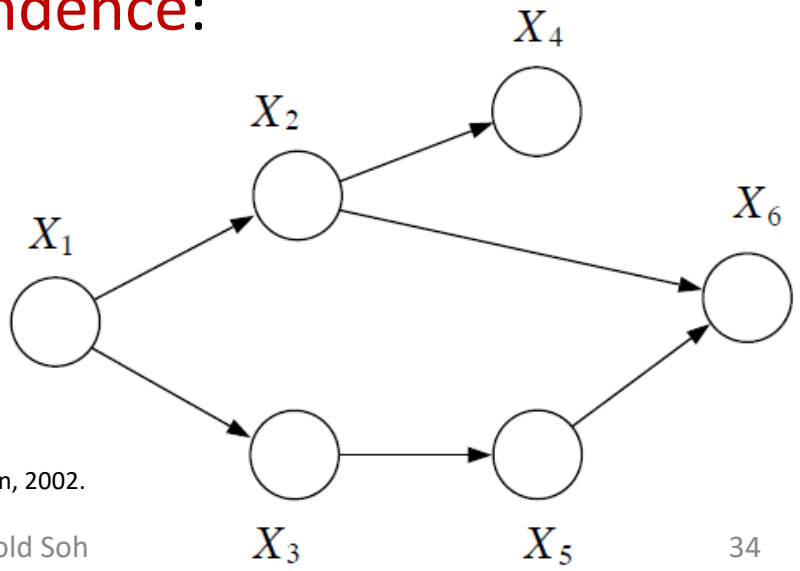
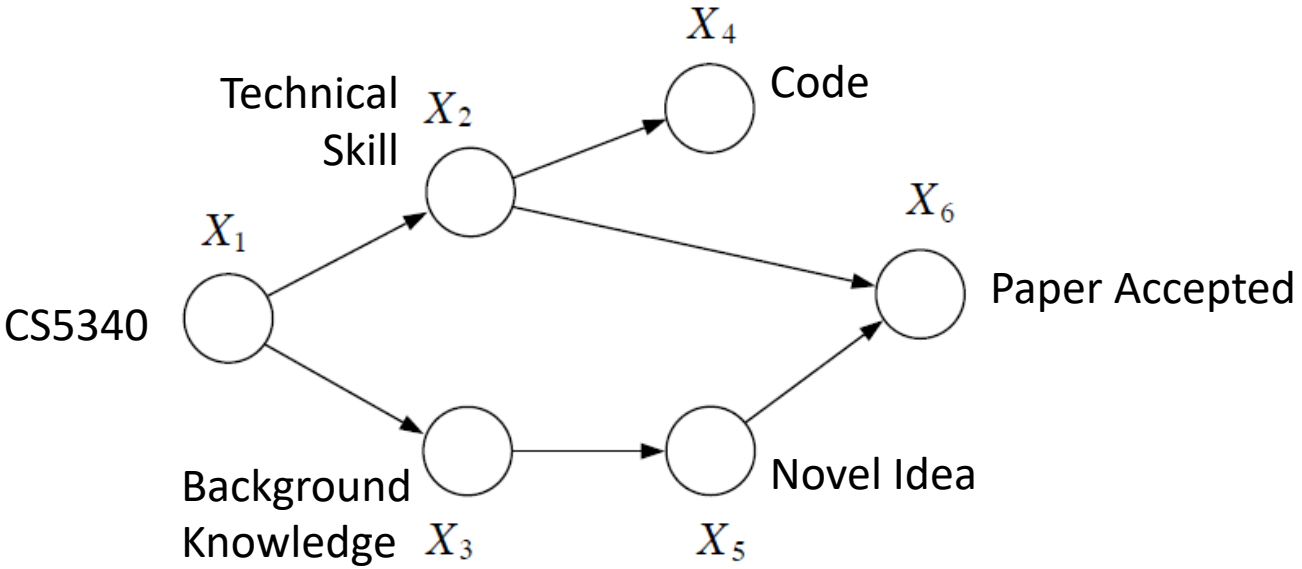


Image source: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Example: Research Output

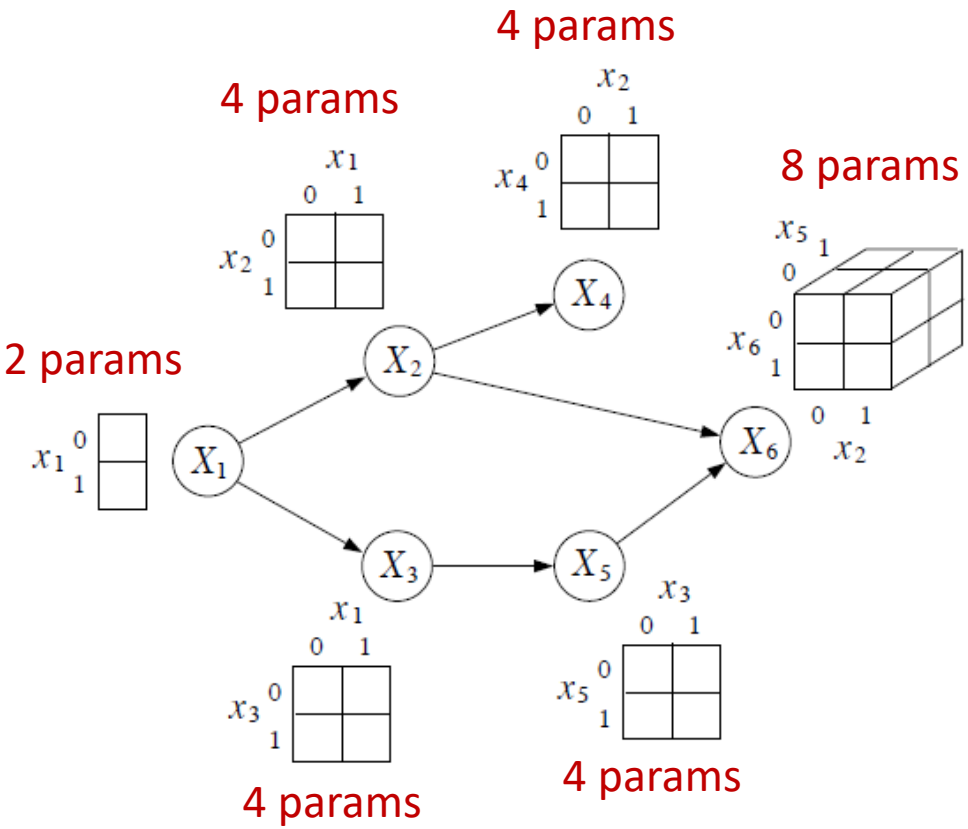


$$p(x_1, \ldots, x_6) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)p(x_5|x_3)p(x_6|x_2, x_5)$$

Parameter Reduction

Example:

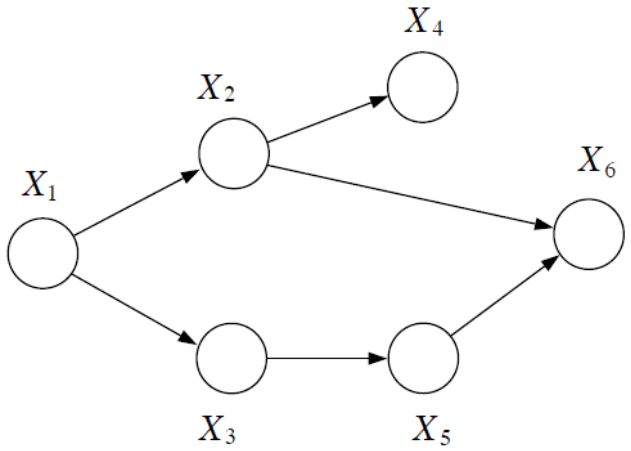
Binary state random variable $x_i \in \{0,1\}$.



- Total parameters = 26.
- Total parameters needed for fully dependent joint probability = $2^6 = 64$.

You actually need fewer than 26 parameters. Why?

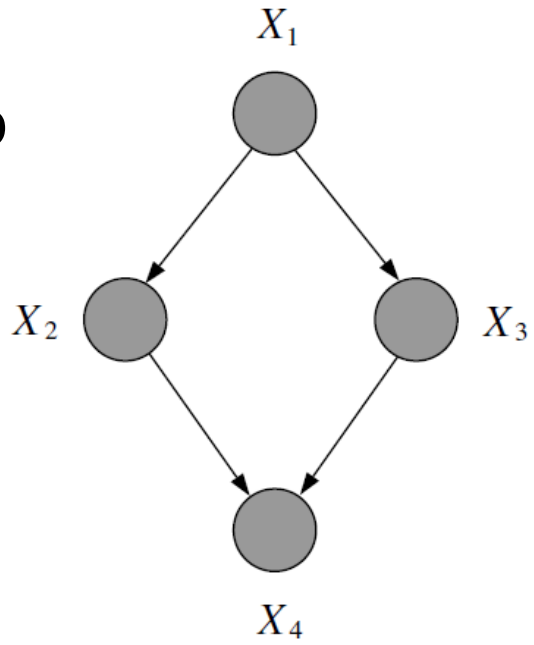
Bayesian Networks: Parameter Reduction



- Let m_i denote the number of parents of node X_i, and each node takes on K values.
- The conditional probability associated with X_i can be represented with a table of size K^{m_i+1}.
- Reduction of parameters needed to represent the joint probability: from $O(K^N)$ to $O(K^{m+1})$, $m \ll N$.

Learning with conditional independence: MLE

- Let's first consider a simple example to illustrate the basic idea.
- Assume "complete data".
- The model shown in the figure has the following joint probability distribution:



$$p(x \mid \theta) = p(x_1 \mid \theta_1)p(x_2 \mid x_1, \theta_2)p(x_3 \mid x_1, \theta_3)p(x_4 \mid x_2, x_3, \theta_4).$$

Image Source: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Learning with conditional independence: MLE

- Finding the maximum log-likelihood of each parameter θ_i can be carried out independently!

$$p(x \mid \theta) = p(x_1 \mid \theta_1)p(x_2 \mid x_1, \theta_2)p(x_3 \mid x_1, \theta_3)p(x_4 \mid x_2, x_3, \theta_4).$$

Taking the log of $p(x|\theta)$ converts the product into sums

$$\log p(x|\theta) = \log p(x_1|\theta_1) + \log p(x_2|x_1, \theta_2) + \log p(x_3|x_1, \theta_3) + \log(x_4|x_2, x_3, \theta_4)$$

Maximum log-likelihood of θ_1:

$$\underset{\theta_1}{\operatorname{argmax}} \log p(x|\theta) = \underset{\theta_1}{\operatorname{argmax}} \{\log p(x_1|\theta_1) + \log p(x_2|x_1, \theta_2) + \log p(x_3|x_1, \theta_3) + \log(x_4|x_2, x_3, \theta_4)\}$$

$$= \underset{\theta_1}{\operatorname{argmax}} \log p(x_1|\theta_1)$$

independent of θ_1, hence can be removed

How about for MAP?

- In similar vein, we can solve for the Maximum a Posterior (MAP) of each parameter separately:

$$\underset{\theta_i}{\operatorname{argmax}} \log p(\theta|x) = \underset{\theta_i}{\operatorname{argmax}} \log p(x|\theta)p(\theta)$$

$$= \underset{\theta_i}{\operatorname{argmax}} \{\log p(x_i|x_{\pi_i}, \theta_i) + \log p(\theta_i)\}$$

where $\theta = (\theta_1, \ldots, \theta_M)$

Representing the Joint Probability

- Random variables $X = \{X_1, X_2, \ldots, X_N\}$

Fully Independent Joint Probability

$p(x_1, \ldots, x_N) = \prod_i p(x_i)$

Conditionally Independent Joint Probability

$p(x_1, \ldots, x_N) = \prod_i p(x_i | x_{\pi_i})$

Fully Correlated Joint Probability

$p(x_1, \ldots, x_N)$

Computational/Resource Cost

(In general and not to scale)

NUS National University of Singapore | School of Computing

Bayesian Networks

A visual language for describing assumed conditional independences

A.M. TURING CENTENARY CELEBRATION WEBCAST

MORE ACM AWARDS

A.M. TURING AWARD

A.M. TURING AWARD WINNERS BY...

ALPHABETICAL LISTING | YEAR OF THE AWARD | RESEARCH SUBJECT

FATHERS OF THE DEEP LEARNING REVOLUTION RECEIVE ACM A.M. TURING AWARD

Bengio, Hinton, and LeCun Ushered in Major Breakthroughs in Artificial Intelligence

Yann LeCun

ACM named Yoshua Bengio, Geoffrey Hinton, and Yann LeCun recipients of the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing. Bengio is Professor at the University of Montreal and Scientific Director at Mila, Quebec's Artificial Intelligene Engineering Fellow of Google, Chie Institute, and University Professor

amturing.acm.org

MORE ACM AWARDS

A.M. TURING AWARD

A.M. TURING AWARD WINNERS BY...

ALPHABETICAL LISTING | YEAR OF THE AWARD | RESEARCH SUBJECT

Photo-Essay

BIRTH:

September 4, 1936, Tel Aviv.

EDUCATION:

B.S., Electrical Engineering (Technion, 1960); M.S., Electronics (Newark College of Engineering, 1961); M.S., Physics (Rutgers University, 1965); Ph.D., Electrical Engineering (Polytechnic Institute of Brooklyn, 1965).

Display a menu for "https://amturing.acm.org/bysubject.cfm"

JUDEA PEARL

DL

United States – 2011

CITATION

For fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning.

SHORT ANNOTATED BIBLIOGRAPHY | ACM TURING AWARD LECTURE VIDEO | RESEARCH SUBJECTS | ADDITIONAL MATERIALS

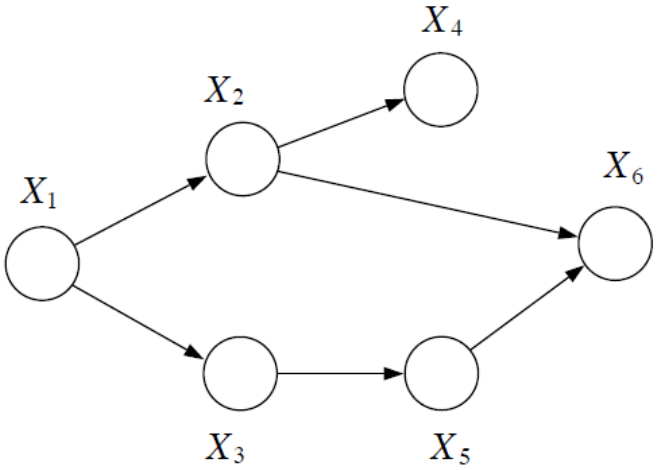
Judea Pearl created the representational and computational foundation for the processing of information under uncertainty.

He is credited with the invention of *Bayesian networks*, a mathematical formalism for defining complex probability models, as well as the principal algorithms used for inference in these models. This work not only revolutionized the field of artificial intelligence but also became an important tool for many other branches of engineering and the natural sciences. He later created a mathematical framework for causal inference that has had significant impact in the social sciences.

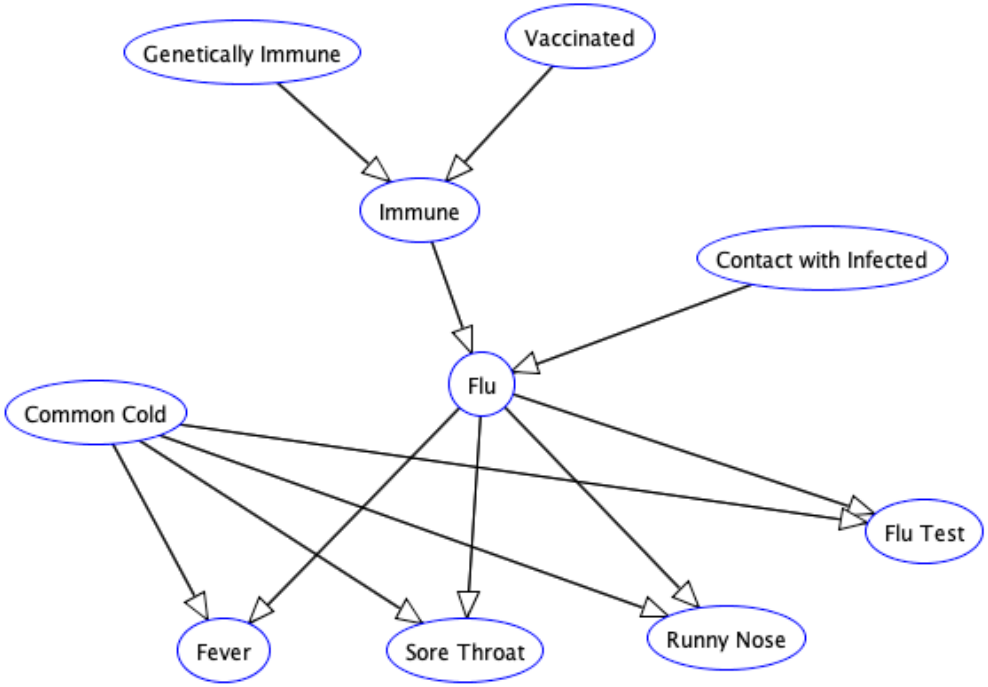
Judea Pearl was born on September 4, 1936, in Tel Aviv, which was at that time administered under the British Mandate for Palestine. He grew up in Bnei Brak, a Biblical town his grandfather went to reestablish in 1924. In 1956, after serving in the Israeli army and joining a Kibbutz, Judea decided to study engineering. He attended

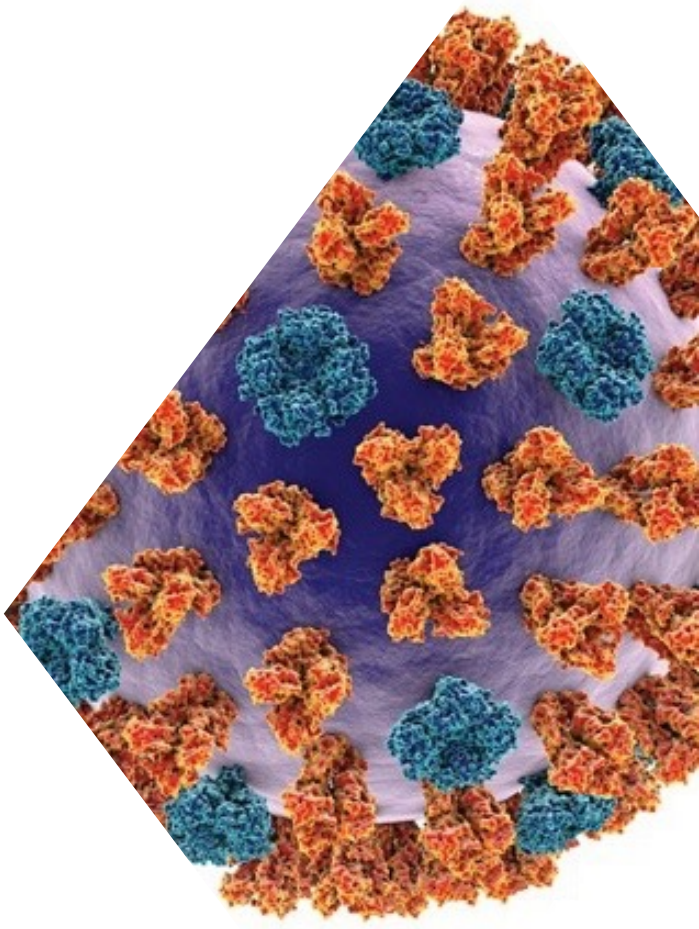
Why care about Bayesian Networks?

- Visualizes the structure of the probabilistic model.
- Provides insights into the properties of the model, particularly the conditional independence properties.
- Complex calculations can be expressed as graphical manipulations.



Generative (Causal) Modeling of Relationships between Variables





But 

!

- Some **misconceptions** about Bayesian networks.
  - ✘ The arrows *always* indicate *dependence*.
  - ✘ Every network represents a *unique* probability distribution.
  - ✘ Observations *always* results in independence between random variables.

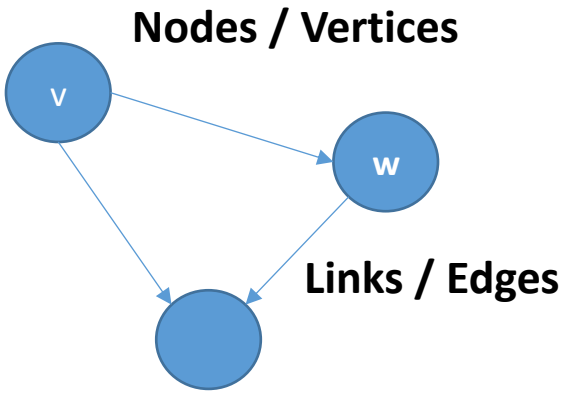
- These are incorrect and lead to errors!
- We will learn how to read Bayes Nets correctly.

From CS2040S:
Data Structures and Algorithms

- Adapted from my other course.
- Imagine you're a 1st year undergrad again. ☺

Directed Graphs

- Graph $\mathcal{G} = \langle V, E \rangle$ ("a tuple of two sets")
  - V is a set of nodes
  - E is a set of edges
    - $E \subseteq \{ (v, w) : v, w \in V \}$
    - (v, w) indicates an edge from $v \rightarrow w$

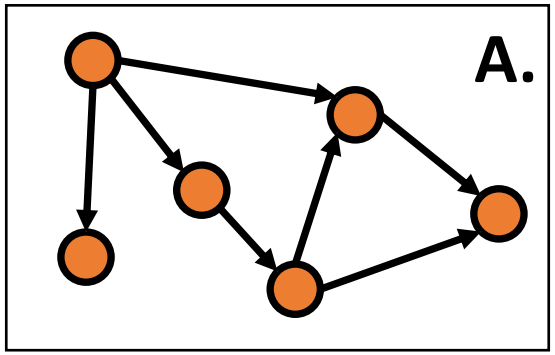


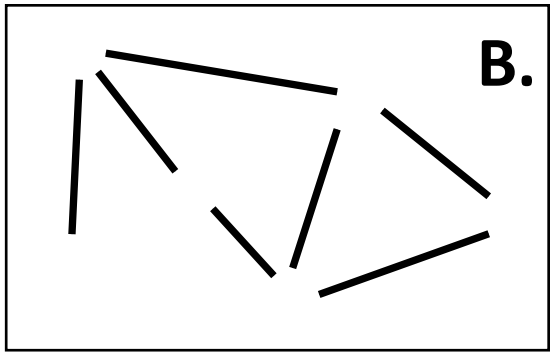
- **Simple Graph:**
  - $e = (v, w)$ for $v \neq w$ ("no self loops")
  - $\forall e_1, e_2 \in E : e_1 \neq e_2$ ("only one edge per pair of nodes")

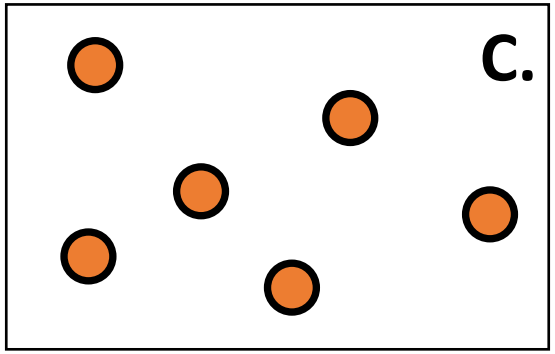
which of these is **not** a graph?

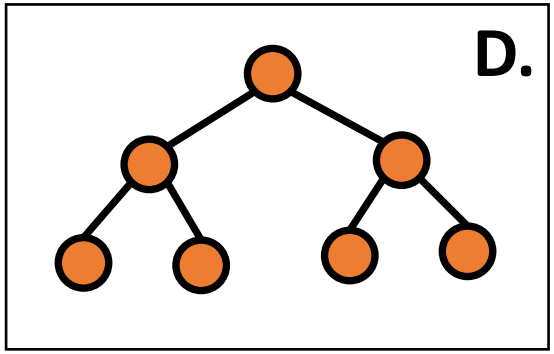
Poll Everywhere

https://bit.ly/2LvG9bq









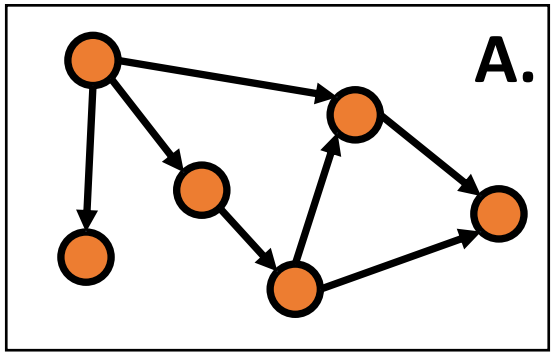
Which is **not** a graph?

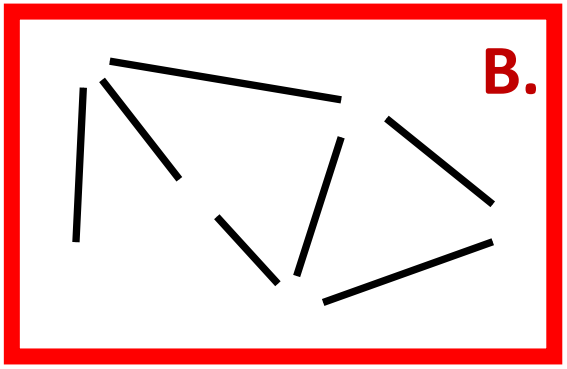
A. A
B. B
C. C
D. D
E. C & B
F. All are graphs
G. None are graphs

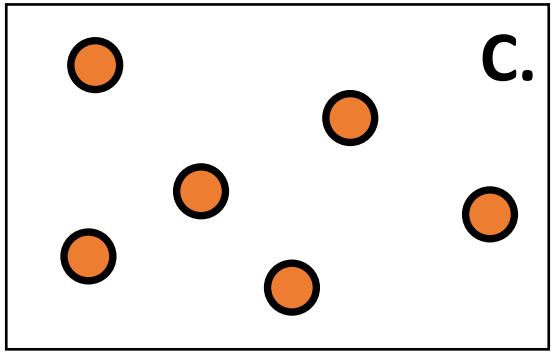
which of these is **not** a graph?

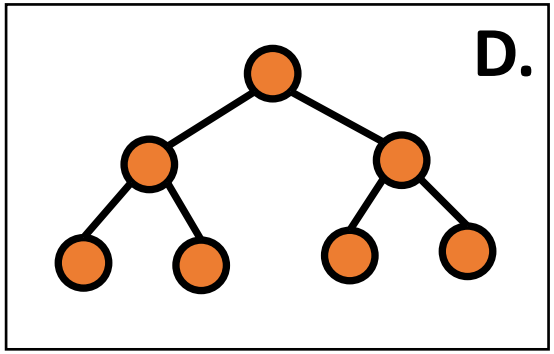
Poll Everywhere

https://bit.ly/2LvG9bq









Which is **not** a graph?

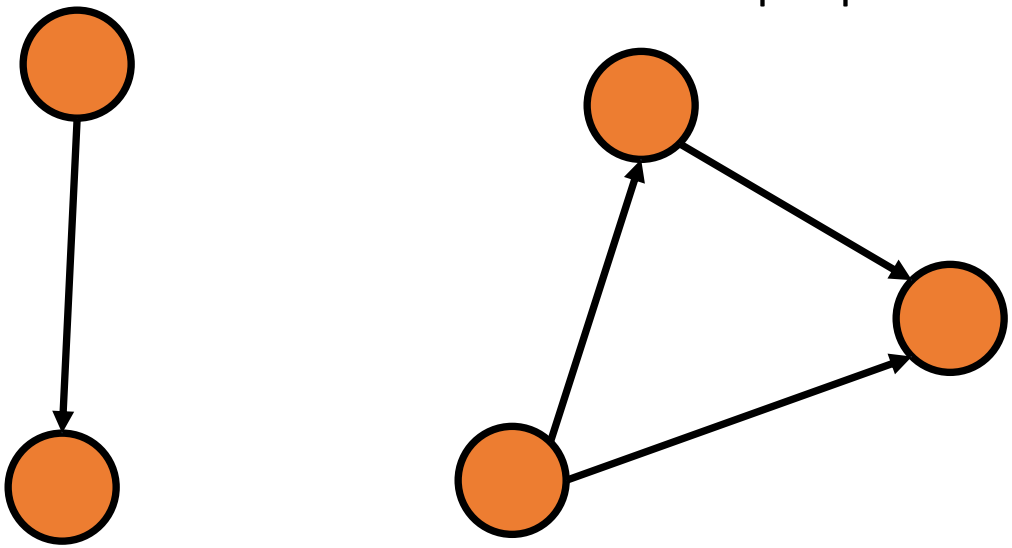
A. A
B. B
C. C
D. D
E. C & B
F. All are graphs
G. None are graphs

is this a simple graph?

Simple Graph:

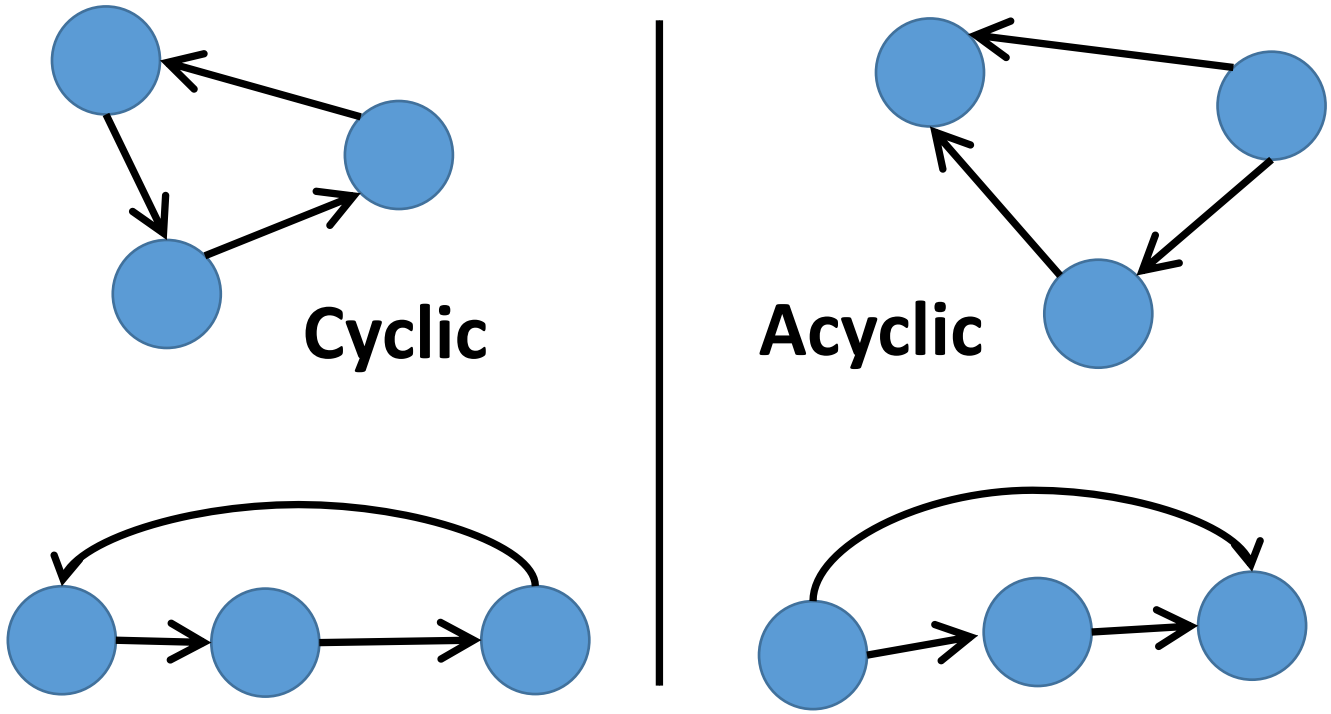
$e = (v, w)$ for $v \neq w$ ("no self loops")

$\forall e_1, e_2 \in E : e_1 \neq e_2$ ("only one edge per pair of nodes")



Yes, it is. But it is not **connected**.

Directed Acyclic Graph (DAG)



Back to CS5340 …

Bayes Nets (BN) and I-maps

- **Definition** (*Bayesian Network*) A Bayesian network is a tuple $B = (G, P)$ where P factorizes according to G and where P is specified as a set of conditional probability distributions (CPDs) associated with G's nodes.

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_{\pi_i})$$

Bayesian Networks: Definitions

- A Bayesian network is a DAG $\mathcal{G}$
- Each node is associated with a random variable X_i

Example of a Directed Graphical Model (DGM), i.e. Bayesian Network:

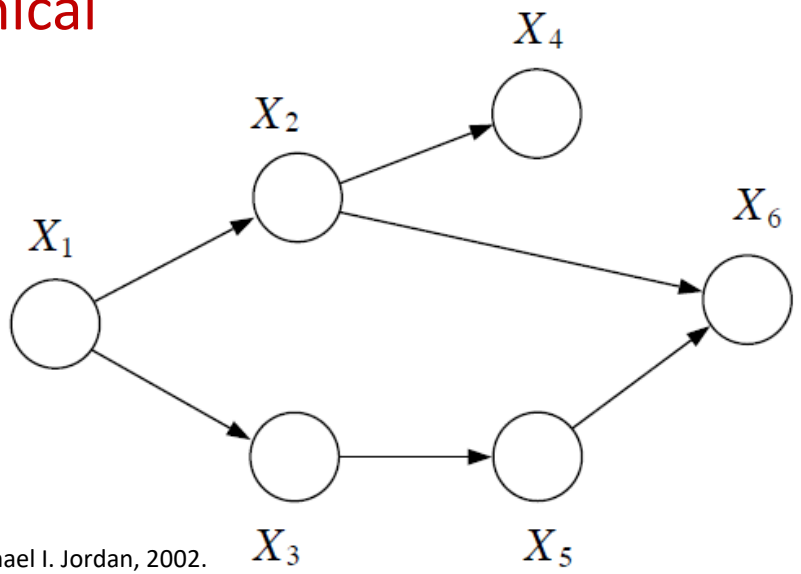


Image modified from: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Bayesian Networks: Definitions

- A Bayesian network is a DAG $\mathcal{G}$
- Each node is associated with a random variable X_i
- Shaded node refers to an observed variable.

Example of a Directed Graphical Model (DGM), i.e. Bayesian Network:

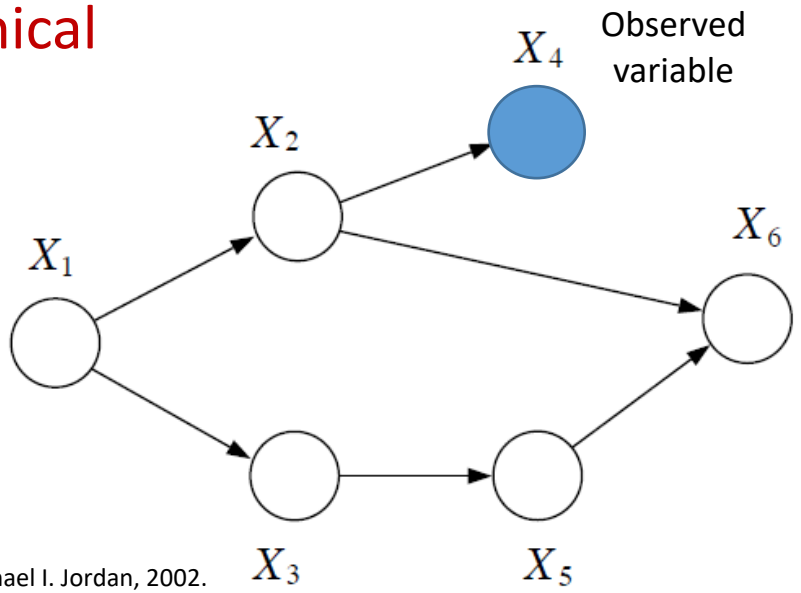


Image modified from: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Bayesian Networks: Definitions

- Each node $i \in \mathcal{V}$ has a set of parent nodes π_i, which can be the empty set.
- Let X_{π_i} represent all the random variables that are parents to the random variable X_i.

Example:

$$X_{\pi_1} = \emptyset, \quad X_{\pi_2} = \{X_1\}$$
$$X_{\pi_3} = \{X_1\}, \; X_{\pi_4} = \{X_2\}$$
$$X_{\pi_5} = \{X_3\}, \; X_{\pi_6} = \{X_2, X_5\}$$

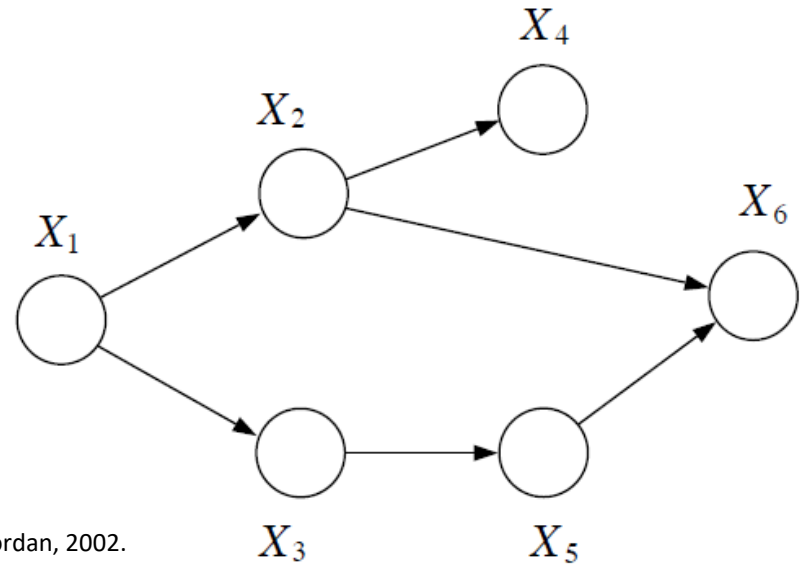


Image source: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

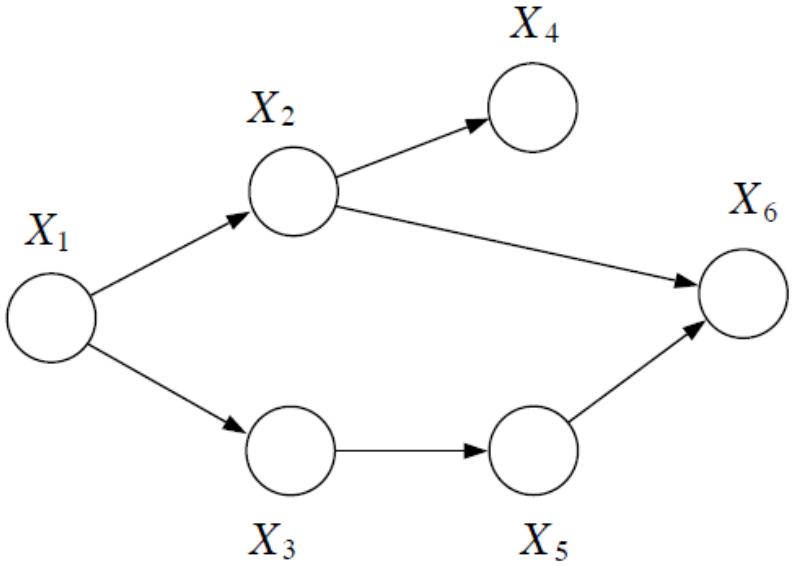
Topological Ordering

- An ordering of the nodes $X_1, X_2, \ldots, X_N$ is a topological ordering relative to $\mathcal{G}$ if whenever we have $X_i \rightarrow X_j$ (i.e., $(i, j) \in E$), then $i < j$

Remark: A topological ordering is only possible iff the graph is a DAG.

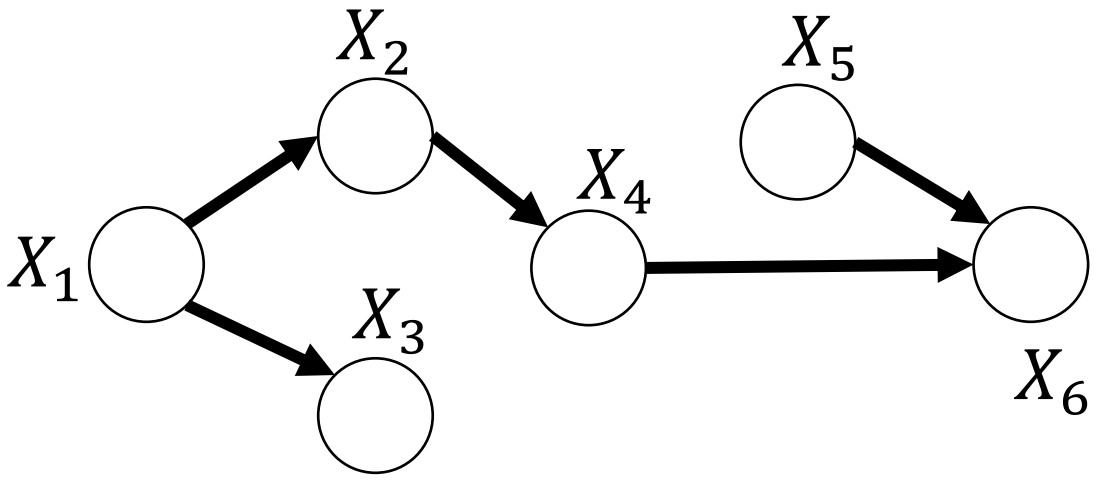
Question: Are topological orderings *unique*?

No!



Paths and Trails

- **Path**: $X_1, X_2, \ldots, X_k$ form a path in the graph $\mathcal{G}$ if for every $i = 1, \ldots, k-1$, we have $X_i \rightarrow X_{i+1}$
- **Trails:** $X_1, X_2, \ldots, X_k$ form a trail in the graph $\mathcal{G}$ if for every $i = 1, \ldots, k-1$, we have either $X_i \rightarrow X_{i+1}$ **or** $X_{i+1} \rightarrow X_i$



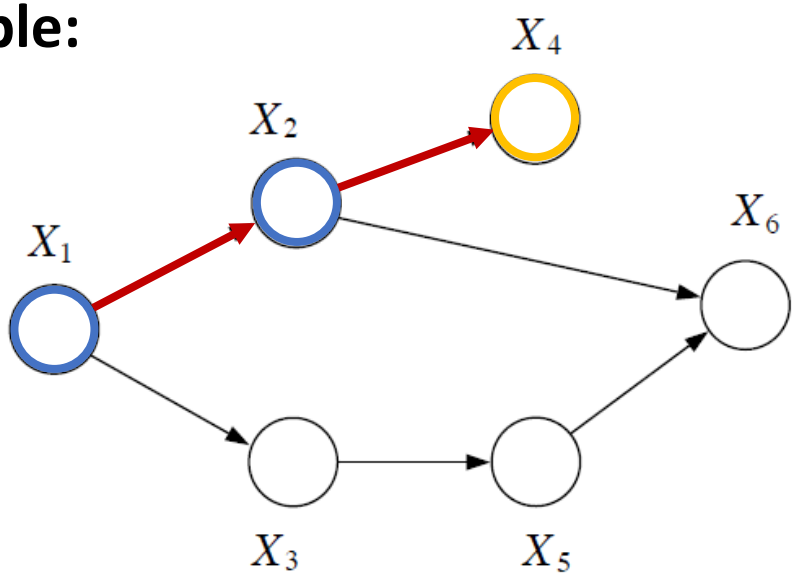
Questions:

- Is there a *path* from X_1 to X_6 ? **Yes**
- If there a *path* from X_6 to X_3? **No**
- Is there a *trail* from X_6 to X_3? **Yes**

Bayesian Networks: Definitions

- Ancestors are the parents, grand-parents, etc. of a node.
- The ancestors of t is the set of node s that connect to t via a directed path: $\text{anc}(t) \triangleq \{s: s \rightsquigarrow t\}$.

Example:



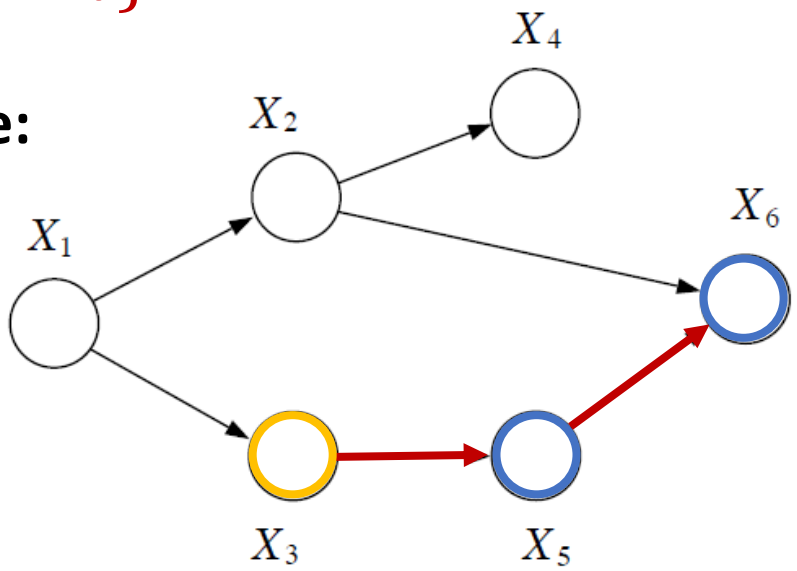
$$\text{anc}(X_4) = \{X_1, X_2\}$$

Image modified from: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Bayesian Networks: Definitions

- Descendants are the children, grand-children, etc of a node.
- Descendants of s is the set of nodes that can be reached via directed paths from s: $\text{desc}(s) \triangleq \{t: s \rightsquigarrow t\}$.

Example:



$$\text{desc}(X_3) = \{X_5, X_6\}$$

Image modified from: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Local Markov Assumption

Local Markov Assumption

- Local Markov assumption: Each random variable X_i is independent of its non-descendants $X_{\text{nonDesc}(X_i)}$ given its parents X_{π_i}.

- The following set of basic conditional independence statements can be associated to the DGM:

$$\{X_i \perp (X_{\text{nonDesc}(X_i)} \setminus X_{\pi_i}) \mid X_{\pi_i}\}$$

Example:

Non-descendants of X_2 are outlined blue.

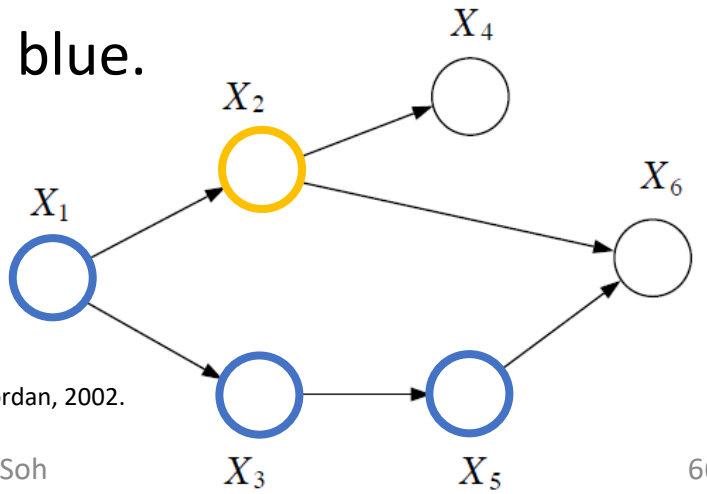
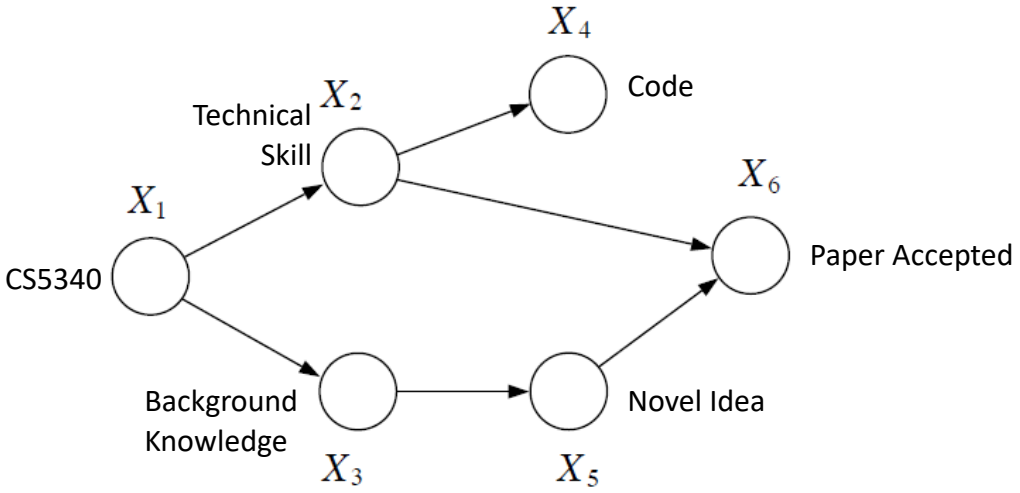


Image modified from: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Local Markov Assumption

Example:

We have the following set of basic conditional independence from the given Bayesian network.



$X_1 \perp \emptyset \mid \emptyset,$

$X_2 \perp \{X_3, X_5, \} \mid X_1,$

$X_3 \perp \{X_2, X_4\} \mid X_1,$

$X_4 \perp \{X_1, X_3, X_5, X_6\} \mid X_2,$

$X_5 \perp \{X_1, X_2, X_4\} \mid X_3,$

$X_6 \perp \{X_1, X_3, X_4\} \mid \{X_2, X_5\}$

Image adapted from: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Local Markov Assumption

- The basic conditional independence statements in the DGM give rise to a set of conditional probabilities:

$$\{X_i \perp (X_{\text{nonDesc}(x_i)} \setminus X_{\pi_i}) \mid X_{\pi_i}\}$$

$$p(x_i|x_{\pi_i}), \qquad i = 1, \ldots, N$$

- $p(x_i|x_{\pi_i})$ is defined locally according to the parent-child relationship specified by the DGM.

Bayesian Networks: Joint Probability

- Locality of the parent-child relationship is used to construct economical representations of the joint distribution.
- The parent-child represents conditional independence:

$$p(x_i|x_{\pi_i})$$

- Joint probability can be read off the graph as the product of all local conditional independence:

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i|x_{\pi_i})$$

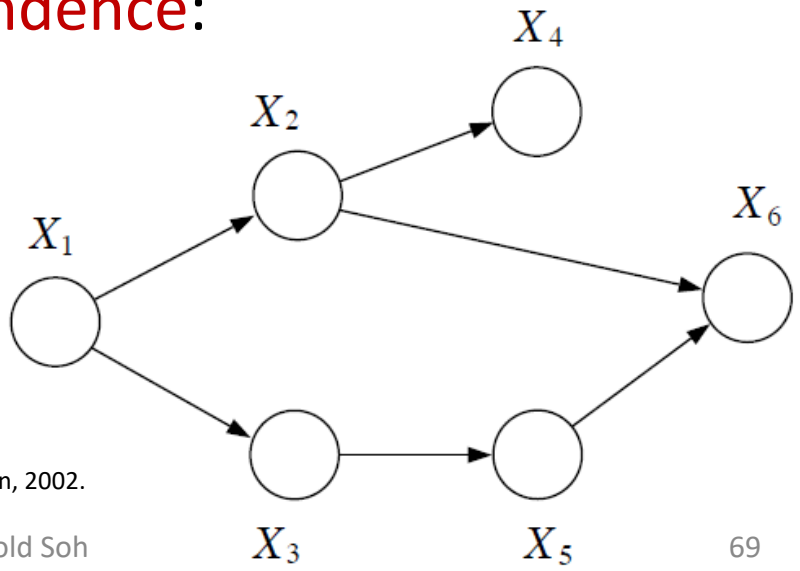


Image source: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Bayesian Networks: Joint Probability

Proof Sketch:

$$p(x_1, \ldots, x_N) = p(x_1) \prod_{i=2}^{N} p(x_i | x_{x_1, \ldots, x_{i-1}})$$ (chain rule)

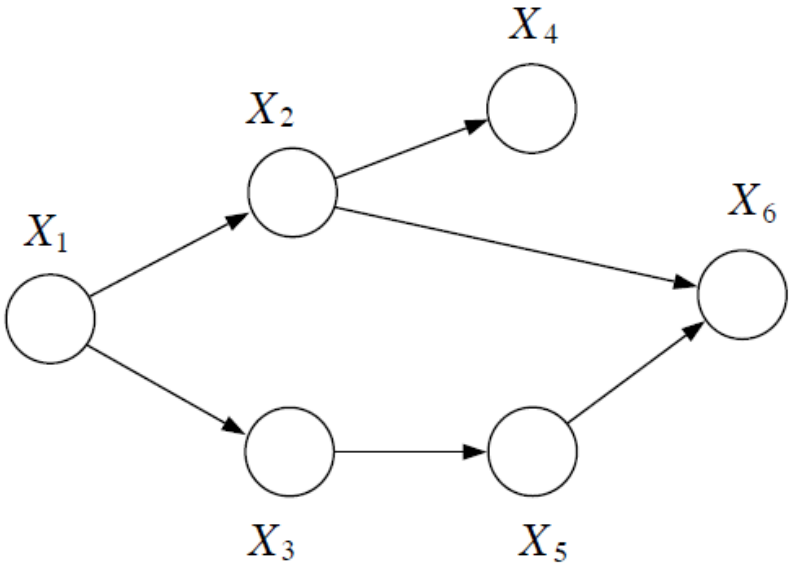
$$\{X_i \perp (X_{\text{nonDesc}(x_i)} \setminus X_{\pi_i}) \mid X_{\pi_i}\}$$

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_{\pi_i})$$

(Assuming topological ordering of DGM)

Bayesian Networks: Joint Probability

Example:



$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p\left(x_i \middle| x_{\pi_i}\right)$$

$$p(x_1, \ldots, x_6) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)p(x_5|x_3)p(x_6|x_2, x_5)$$

Image source: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Bayes Nets (BN) and I-maps

- **Definition** (*Bayesian Network*) A Bayesian network is a tuple $B = (G, P)$ where P factorizes according to G and where P is specified as a set of conditional probability distributions (CPDs) associated with G's nodes.

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_{\pi_i})$$

Independence-Maps (I-Maps)

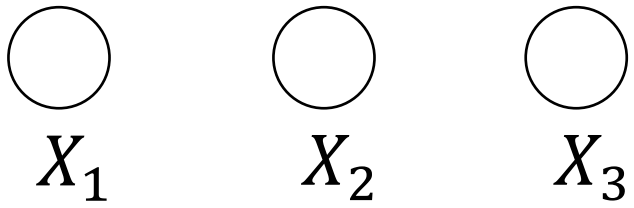
- **Definition** (*Independence set*) Let P be a distribution over $\mathcal{X}$. Define $\mathcal{I}(P)$ as the set of independence assertions of the form $(X \perp Y \mid Z)$ that hold in P.

- **Definition** (*Independence map*) Let G be associated with independence assertions $\mathcal{I}(G)$. G is an I-map for P if $\mathcal{I}(G) \subseteq \mathcal{I}(P)$

Factorization and Graphs

$$p(x_1, \dots, x_N) = \prod_{i=1}^{N} p(x_i | x_{\pi_i})$$

- Question:
  - Assume $p(x_1, \dots, x_3) = p(x_1)p(x_2)p(x_3)$
  - Is the probability distribution above consistent with this graph? (is the graph an I-map for p above?)

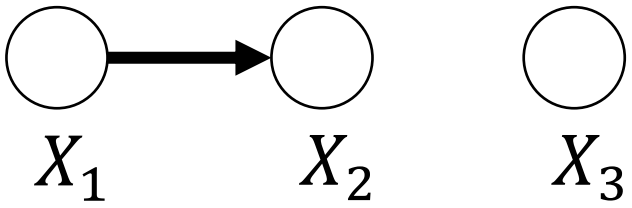


Yes!

Factorization and Graphs

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_{\pi_i})$$

- Question:
  - Assume $p(x_1, \ldots, x_3) = p(x_1)p(x_2)p(x_3)$
  - Is the probability distribution above consistent with this graph? (is the graph an I-map for p above?)

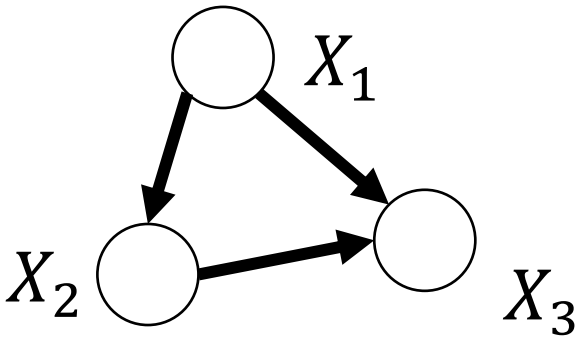


Yes!

Factorization and Graphs

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_{\pi_i})$$

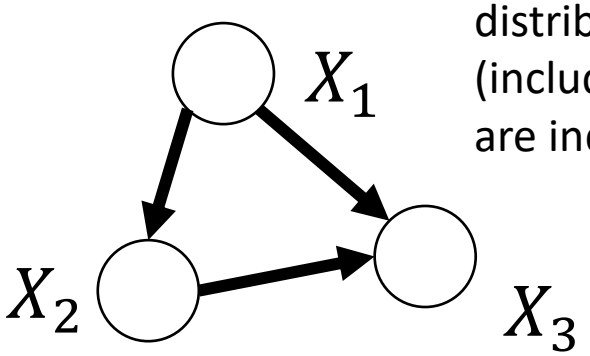
- Question:
  - Assume $p(x_1, \ldots, x_3) = p(x_1)p(x_2)p(x_3)$
  - Is the probability distribution above consistent with this graph? (is the graph an I-map for p above?)



Yes!

Remember!

- A DAG represents conditional independences
- It does **not** assert conditional dependence!
  - Interprete arrows as "*possible* dependence"
- E.g., a fully connected DAG can represent *any* distribution over its random variables.



This graph can represent *any* distribution over X_1, X_2, X_3 (including those where the r.v.'s are independent)

From Earlier Today…

- Some **misconceptions** about Bayesian networks.
  - ✗ The arrows *always* indicate *dependence*.
  - ✗ Every network represents a *unique* probability distribution.
  - ✗ Observations *always* results in independence between random variables.
- These are incorrect and lead to errors!
- We will learn how to read Bayes Nets correctly.

NUS National University of Singapore | School of Computing

Analyzing Bayesian Networks

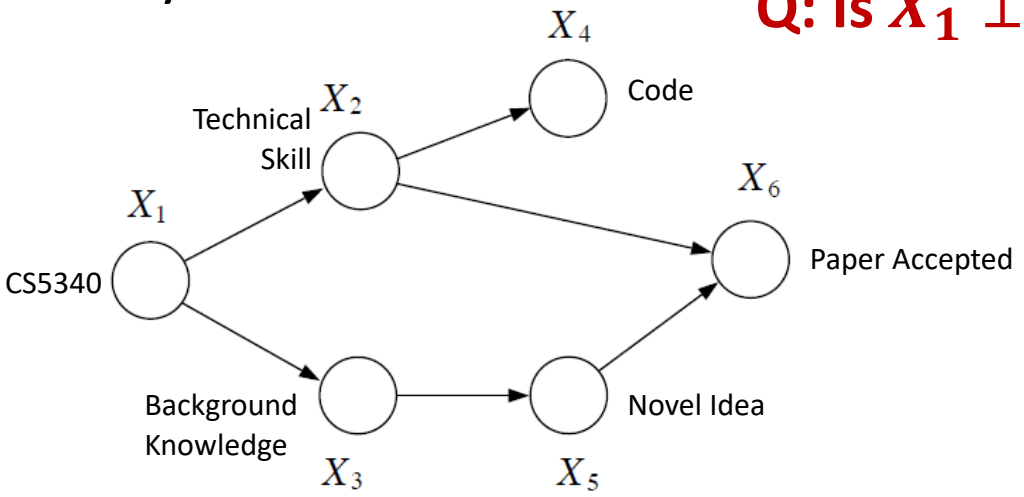
Canonical 3-node graphs, blocking, and d-separation

Any extra conditional independences?

Example:

We have the following set of basic conditional independence from the given Bayesian network.

Q: Is $X_1 \perp X_6 \mid \{X_2, X_3\}$?



$X_1 \perp \emptyset \mid \emptyset,$

$X_2 \perp \{X_3, X_5, \} \mid X_1,$

$X_3 \perp \{X_2, X_4\} \mid X_1,$

$X_4 \perp \{X_1, X_3, X_5, X_6\} \mid X_2,$

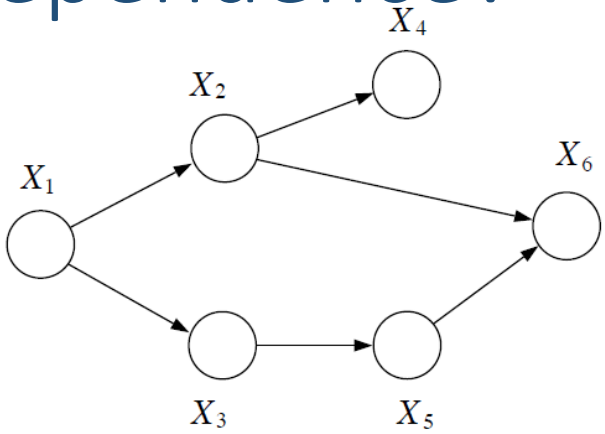
$X_5 \perp \{X_1, X_2, X_4\} \mid X_3,$

$X_6 \perp \{X_1, X_3, X_4\} \mid \{X_2, X_5\}$

Image adapted from: "An introduction to probabilistic graphical models", Michael I. Jordan, 2002.

Additional Conditional Independence?

It turns out $X_1 \perp X_6 \mid \{X_2, X_3\}$ is also a conditional independence, but not ***directly*** observed from the parent-child relation.



Why?:

Strategy: Check if $p(x_1|x_2, x_3, x_6) = p(x_1|x_2, x_3)$?

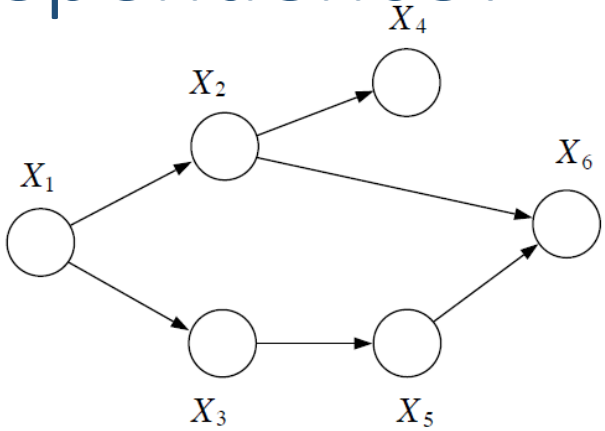
Recall that:

$$p(x_1|x_2, x_3, x_6) = \frac{p(x_1, x_2, x_3, x_6)}{p(x_2, x_3, x_6)}$$

Given the joint $p(x_1, x_2, \ldots, x_5, x_6)$, how do we find $p(x_1, x_2, x_3, x_6)$ and $p(x_2, x_3, x_6)$?

Additional Conditional Independence?

It turns out $X_1 \perp X_6 \mid \{X_2, X_3\}$ is also a conditional independence, but not ***directly*** observed from the parent-child relation.



Why? (continued):

$$p(x_1, x_2, x_3, x_6) = \sum_{x_4} \sum_{x_5} p(\mathrm{x_1})\mathrm{p}(\mathrm{x_2}|\mathrm{x_1})\mathrm{p}(\mathrm{x_3}|\mathrm{x_1})\mathrm{p}(\mathrm{x_4}|\mathrm{x_2})\mathrm{p}(\mathrm{x_5}|\mathrm{x_3})\mathrm{p}(\mathrm{x_6}|\mathrm{x_2}, \mathrm{x_5})$$

$$= p(\mathrm{x_1})\mathrm{p}(\mathrm{x_2}|\mathrm{x_1})\mathrm{p}(\mathrm{x_3}|\mathrm{x_1}) \sum_{x_4} \mathrm{p}(\mathrm{x_4}|\mathrm{x_2}) \sum_{x_5} \mathrm{p}(\mathrm{x_5}|\mathrm{x_3})\mathrm{p}(\mathrm{x_6}|\mathrm{x_2}, \mathrm{x_5}) \quad \sum_{x_4} \mathrm{p}(\mathrm{x_4}|\mathrm{x_2}) = 1$$

$$= p(\mathrm{x_1})\mathrm{p}(\mathrm{x_2}|\mathrm{x_1})\mathrm{p}(\mathrm{x_3}|\mathrm{x_1}) \sum_{x_5} \mathrm{p}(\mathrm{x_5}|\mathrm{x_3})\mathrm{p}(\mathrm{x_6}|\mathrm{x_2}, \mathrm{x_5})$$

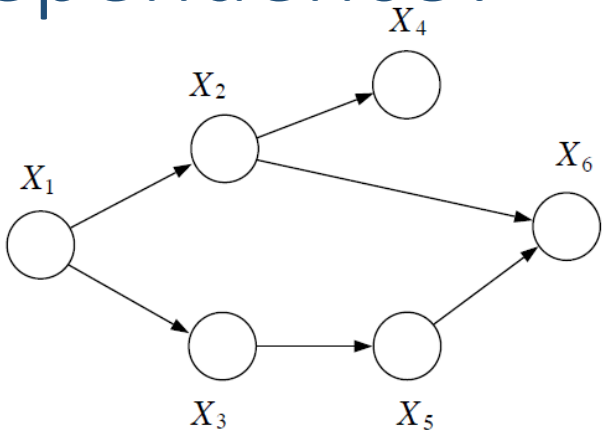
$$p(x_2, x_3, x_6) = \sum_{x_1} \sum_{x_4} \sum_{x_5} p(\mathrm{x_1})\mathrm{p}(\mathrm{x_2}|\mathrm{x_1})\mathrm{p}(\mathrm{x_3}|\mathrm{x_1})\mathrm{p}(\mathrm{x_4}|\mathrm{x_2})\mathrm{p}(\mathrm{x_5}|\mathrm{x_3})\mathrm{p}(\mathrm{x_6}|\mathrm{x_2}, \mathrm{x_5})$$

$$= \sum_{x_1} p(\mathrm{x_1})\mathrm{p}(\mathrm{x_2}|\mathrm{x_1})\mathrm{p}(\mathrm{x_3}|\mathrm{x_1}) \sum_{x_4} \mathrm{p}(\mathrm{x_4}|\mathrm{x_2}) \sum_{x_5} \mathrm{p}(\mathrm{x_5}|\mathrm{x_3})\mathrm{p}(\mathrm{x_6}|\mathrm{x_2}, \mathrm{x_5}) \quad \sum_{x_4} \mathrm{p}(\mathrm{x_4}|\mathrm{x_2}) = 1$$

$$= \sum_{x_1} p(\mathrm{x_1})\mathrm{p}(\mathrm{x_2}|\mathrm{x_1})\mathrm{p}(\mathrm{x_3}|\mathrm{x_1}) \sum_{x_5} \mathrm{p}(\mathrm{x_5}|\mathrm{x_3})\mathrm{p}(\mathrm{x_6}|\mathrm{x_2}, \mathrm{x_5})$$

Additional Conditional Independence?

It turns out $X_1 \perp X_6 \mid \{X_2, X_3\}$ is also a conditional independence, but not ***directly*** observed from the parent-child relation.



Why? (continued):

$$p(x_1, x_2, x_3, x_6) = p(\mathrm{x}_1)\mathrm{p}(\mathrm{x}_2|\mathrm{x}_1)\mathrm{p}(\mathrm{x}_3|\mathrm{x}_1)\sum_{x_5} \mathrm{p}(\mathrm{x}_5|\mathrm{x}_3)\mathrm{p}(\mathrm{x}_6|\mathrm{x}_2, \mathrm{x}_5)$$

$$p(x_2, x_3, x_6) = \sum_{x_1} p(\mathrm{x}_1)\mathrm{p}(\mathrm{x}_2|\mathrm{x}_1)\mathrm{p}(\mathrm{x}_3|\mathrm{x}_1)\sum_{x_5} \mathrm{p}(\mathrm{x}_5|\mathrm{x}_3)\mathrm{p}(\mathrm{x}_6|\mathrm{x}_2, \mathrm{x}_5)$$

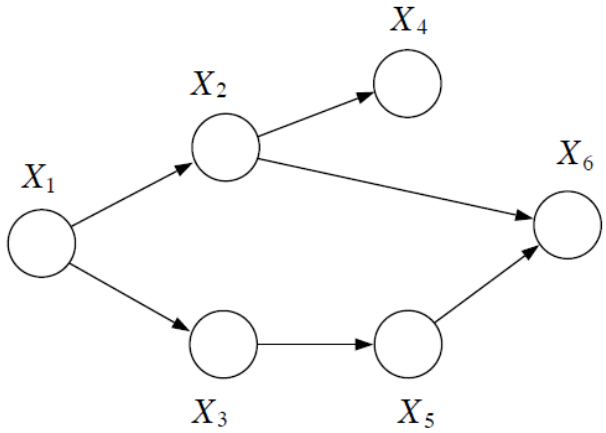
$$p(x_1|x_2, x_3, x_6) = \frac{p(x_1, x_2, x_3, x_6)}{p(x_2, x_3, x_6)}$$

$$= \frac{p(\mathrm{x}_1)\mathrm{p}(\mathrm{x}_2|\mathrm{x}_1)\mathrm{p}(\mathrm{x}_3|\mathrm{x}_1)\sum_{x_5} \mathrm{p}(\mathrm{x}_5|\mathrm{x}_3)\mathrm{p}(\mathrm{x}_6|\mathrm{x}_2, \mathrm{x}_5)}{\sum_{x_1} p(\mathrm{x}_1)\mathrm{p}(\mathrm{x}_2|\mathrm{x}_1)\mathrm{p}(\mathrm{x}_3|\mathrm{x}_1)\sum_{x_5} \mathrm{p}(\mathrm{x}_5|\mathrm{x}_3)\mathrm{p}(\mathrm{x}_6|\mathrm{x}_2, \mathrm{x}_5)}$$

$$= \frac{p(\mathrm{x}_1, x_2, x_3)}{\sum_{x_1} p(\mathrm{x}_1, x_2, x_3)} = \frac{p(\mathrm{x}_1, x_2, x_3)}{p(x_2, x_3)} = p(x_1|x_2, x_3)$$

Additional Conditional Independence?

It turns out $X_1 \perp X_6 \mid \{X_2, X_3\}$ is also a conditional independence, but not directly observed from the parent-child relation.



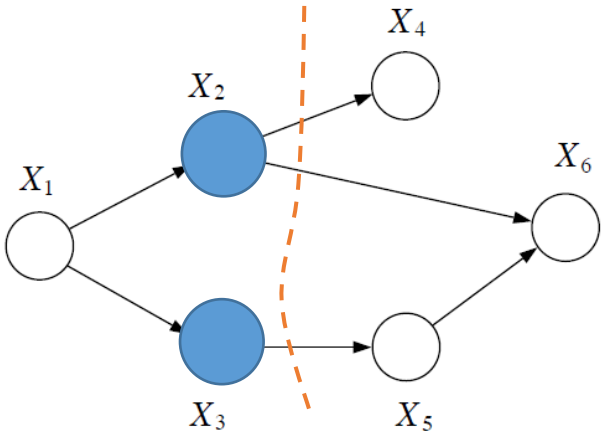
Why? (continued):

$p(x_1|x_2, x_3, x_6) = p(x_1|x_2, x_3)$ **Done!**

Question: Can we write all other conditional independencies by just inspecting the DGM without going through the complicated derivation?

Additional Conditional Independence?

It turns out $X_1 \perp X_6 \mid \{X_2, X_3\}$ is also a conditional independence, but not directly observed from the parent-child relation.



Why? (continued):

$$p(x_1|x_2, x_3, x_6) = p(x_1|x_2, x_3)$$ **Done!**

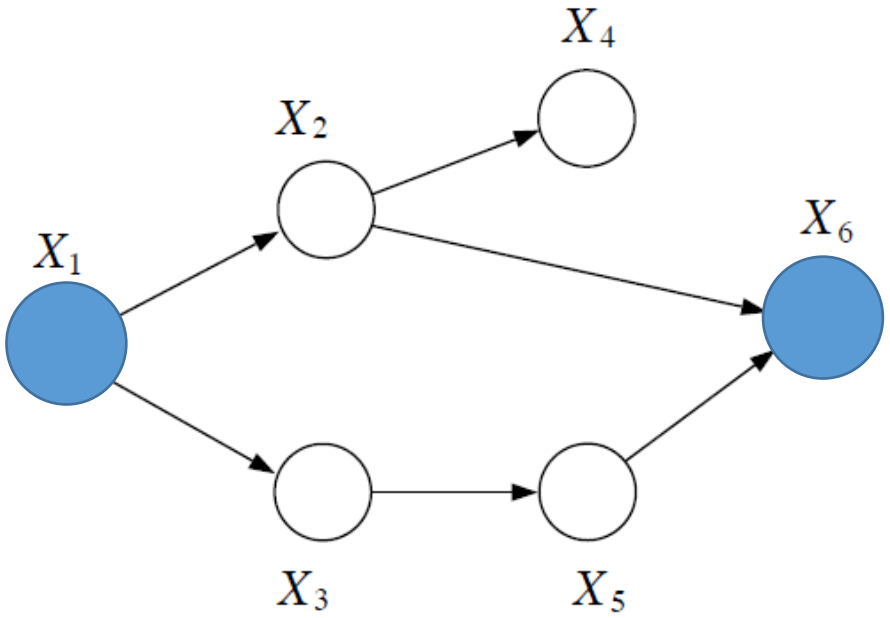
The nodes $\{X_2, X_3\}$ "block" X_1 from X_6.

Question: Can we write all other conditional independencies by just inspecting the DGM without going through the complicated derivations?

Answer: Yes, observe that the nodes $\{X_2, X_3\}$ "block" all paths from X_1 to X_6. This suggests the notion of graph separation for inferring conditional independence.

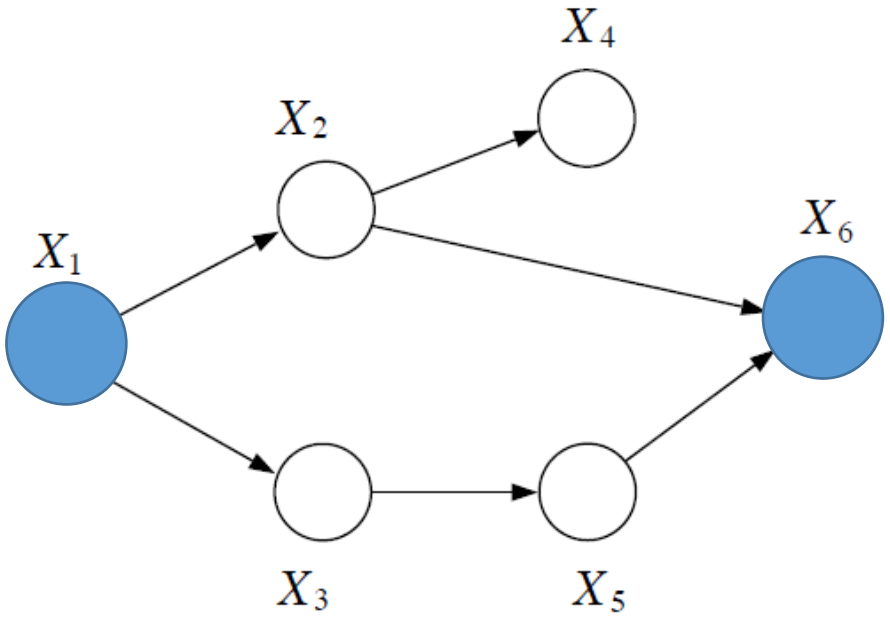
Question: Blocked or not blocked?

- Is true that $X_2 \perp X_3 | \{X_1, X_6\}$?



Question: Blocked or not blocked?

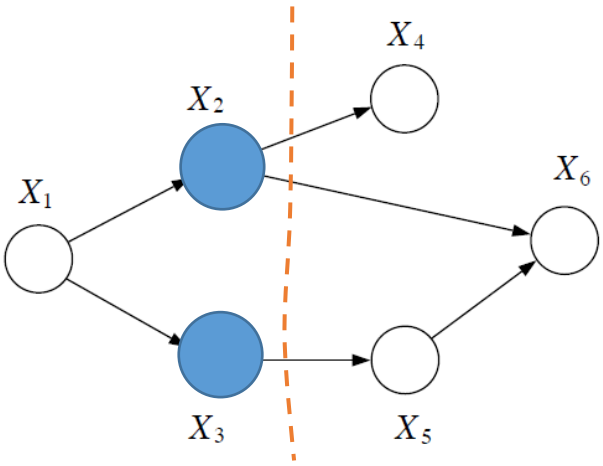
- Is true that $X_2 \perp X_3 | \{X_1, X_6\}$?



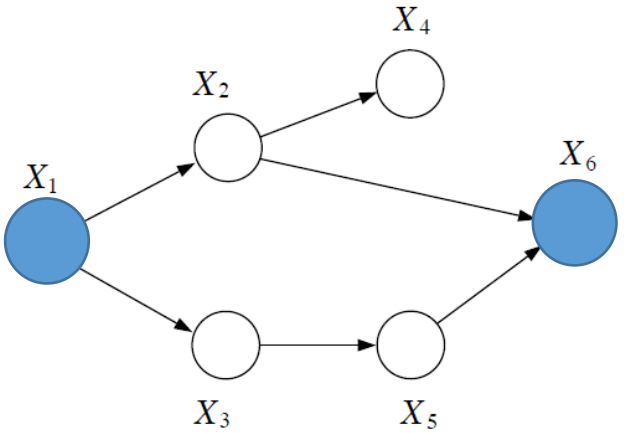
X_2 is **NOT independent** of X_3 given $\{X_1, X_6\}$.

Additional Conditional Independence?

- We have to be careful in making the notion of "blocking".
- X_2 is NOT independent of X_3 given X_1 and X_6 as would be suggested by a naïve interpretation of "blocking".
- Precise definition of "blocking" has to be done through the "three canonical 3-node graphs", and "d-separation".

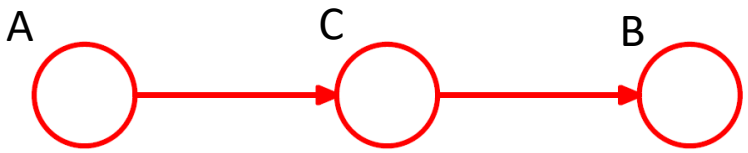


The nodes $\{X_2, X_3\}$ **"block"** X_1 from X_6.

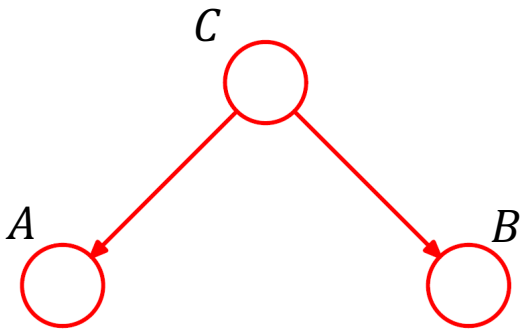


X_2 is **NOT independent** of X_3 given $\{X_1, X_6\}$.

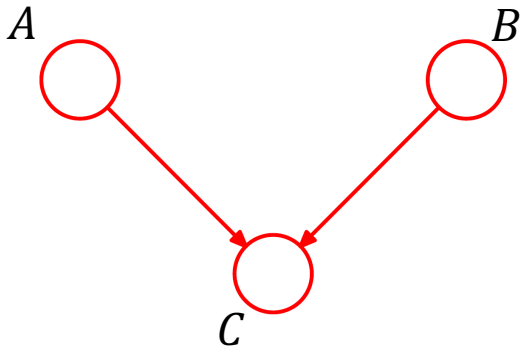
The Canonical 3-node graphs



Head-Tail (wrt C)
(Chain/Causal-trail)



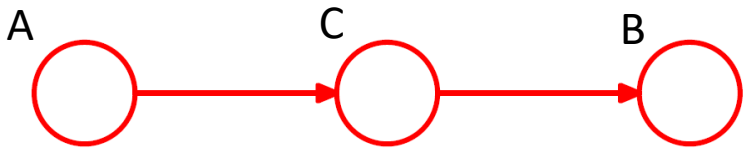
Tail-Tail (wrt C)
(Tent/Common cause)



Head-Head (wrt C)
(V-structure/Collider/Common Effect)

Head-Tail

1.



Joint distribution corresponding to this graph:

$$p(a, b, c) = p(a)p(c|a)p(b|c)$$

If none of the variables are observed, we can see that *A* and *B* are NOT independent by marginalizing over *C*:

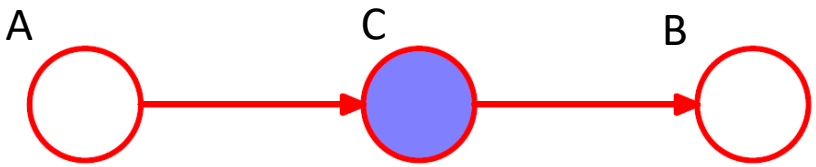
$$p(a, b) = p(a) \sum_c p(c|a)p(b|c) = p(a)p(b|a)$$

which *in general* does not factorize into $p(a)p(b)$, and so, in general,

$$A \not\perp B \mid \emptyset$$

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Head-Tail



If we condition on node *C*, using Bayes' theorem together with the joint distribution, we get:

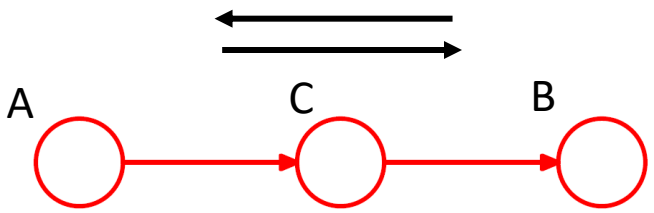
$$
\begin{aligned}
p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\
&= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\
&= p(a|c)p(b|c) \qquad \text{(Bayes rule)}
\end{aligned}
$$

which shows the conditional independence property:

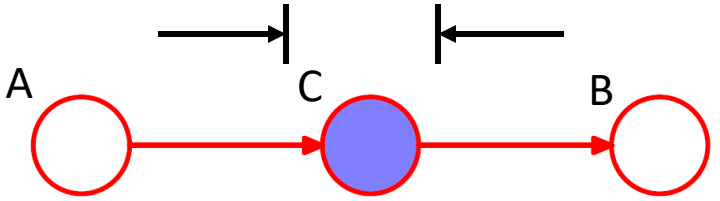
$$A \perp B \mid C$$

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Head-Tail



$A \not\perp B \mid \emptyset$

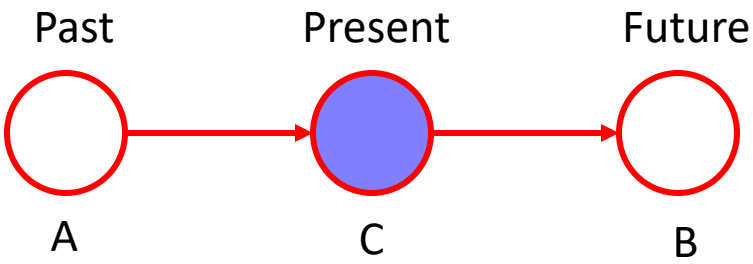


$A \perp B \mid C$

- The node *C* is said to be *head-to-tail* with respect to the path from node *A* to node *B*.
- Such a path connects nodes *A* and *B* and renders them dependent.
- The observation of *C* 'blocks' the path from *A* to *B* and so we obtain the conditional independence property.

Head-Tail

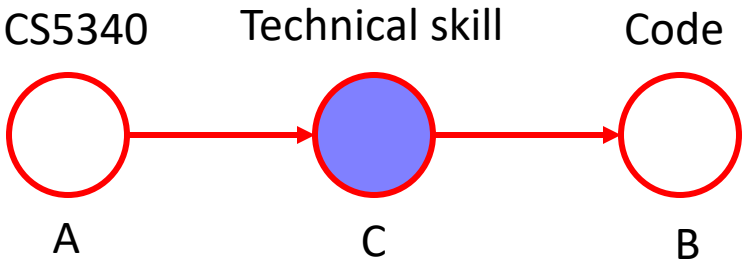
Intuitive interpretation:



- The conditional independence $A \perp B \mid C$ translates into the statement: "the past is independent of the future given the present".
- This is an example of a simple classical Markov Chain.

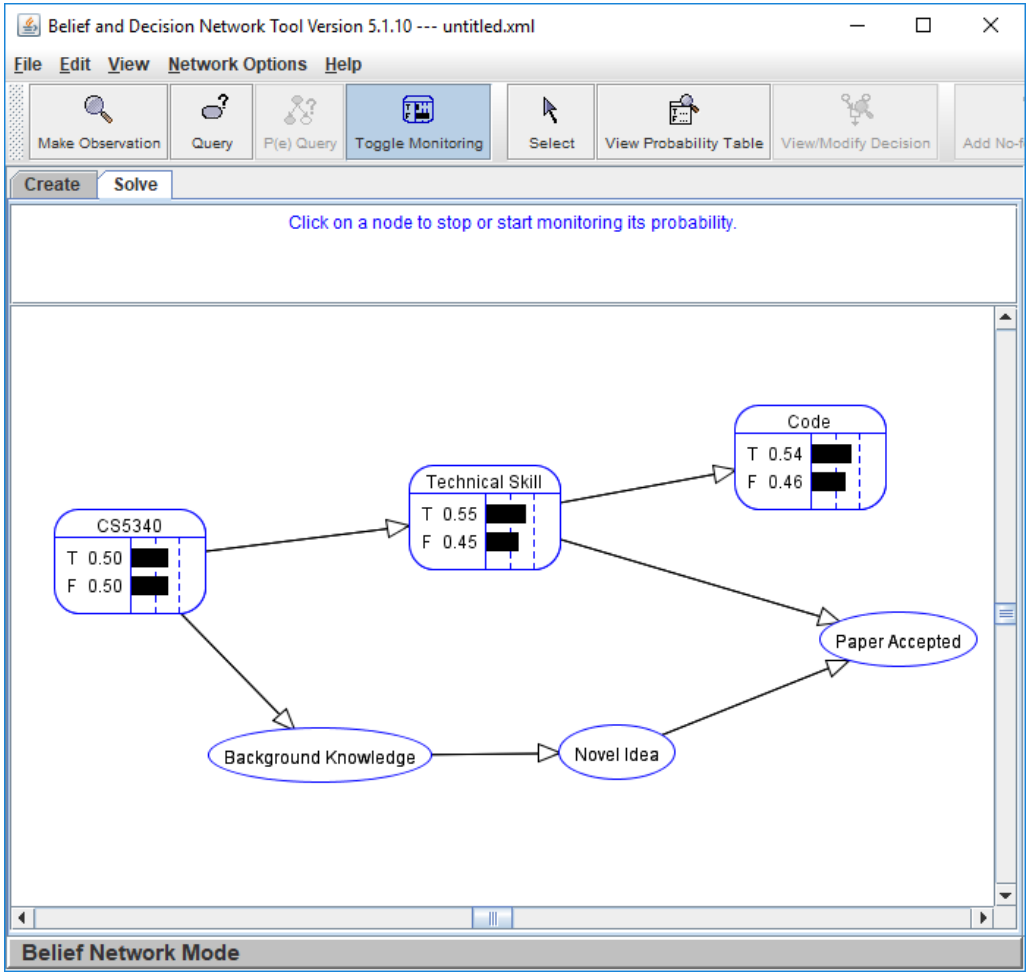
Head-Tail

Intuitive interpretation:



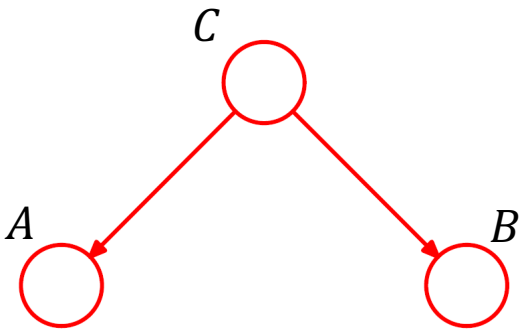
- The conditional independence $A \perp B \mid C$ translates into the statement: "whether a person can produce good code is independent of whether he/she took CS5330 given his technical skill".

Head-Tail Demo



Tail-Tail

2.



Joint distribution corresponding to this graph:

$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

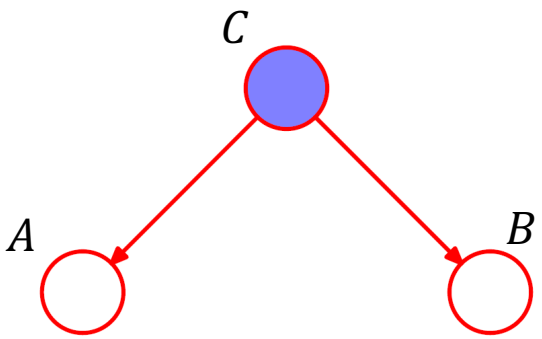
If none of the variables are observed, we can see that *A* and *B* are NOT independent by marginalizing both sides over *C*:

$$p(a, b) = \sum_c p(a|c)p(b|c)p(c)$$

which in general does not factorize into $p(a)p(b)$, and so, in general

$$A \not\perp B \mid \emptyset$$

Tail-Tail



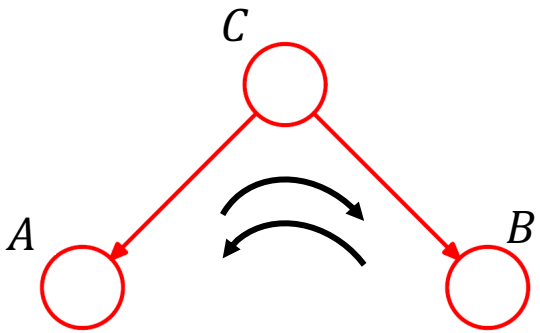
If we condition on node *C*, we can easily write down the conditional distribution of *A* and *B*, given *C*, in the form:

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned}$$

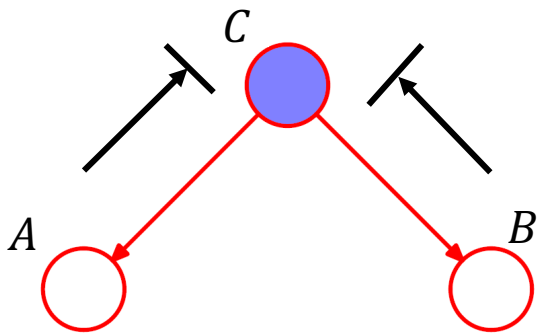
which shows the conditional independence property:

$$A \perp B \mid C$$

Tail-Tail



$A \not\perp B \mid \emptyset$

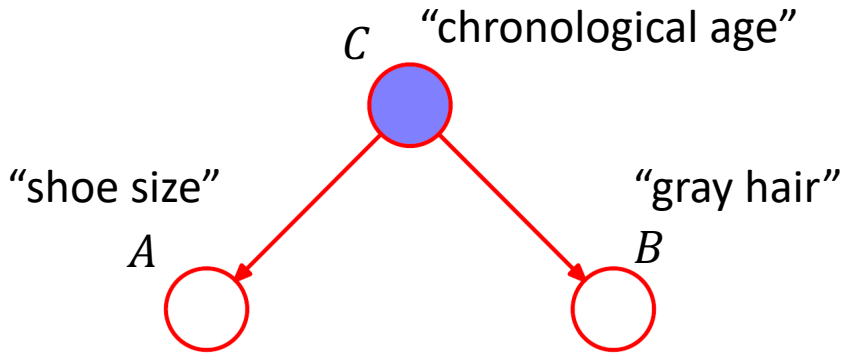


$A \perp B \mid C$

- The node *C* is said to be *tail-to-tail* with respect to the path from node *A* to *B.*
- Such a path connects nodes *A* and *B* and renders them dependent.
- The observation of *C* 'blocks' the path from *A* to *B,* and we obtain the conditional independence property.

Tail-Tail

Intuitive interpretation:

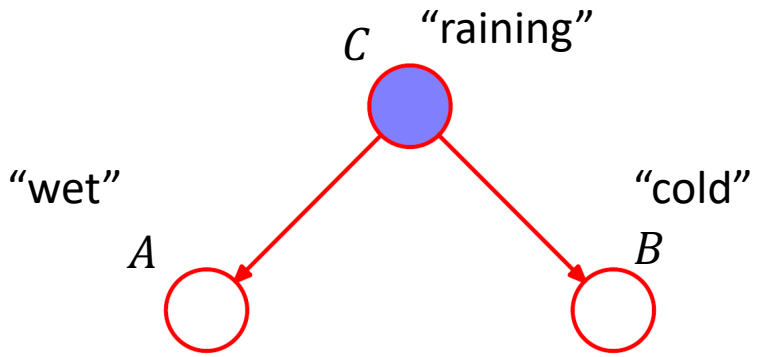


- Given the age of a person, there is no further relationship between the size of his feet and the amount of gray hair he has.
- We say that the variable *C* "explains" all of the observed dependence between *A* and *B*.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Tail-Tail

Intuitive interpretation:

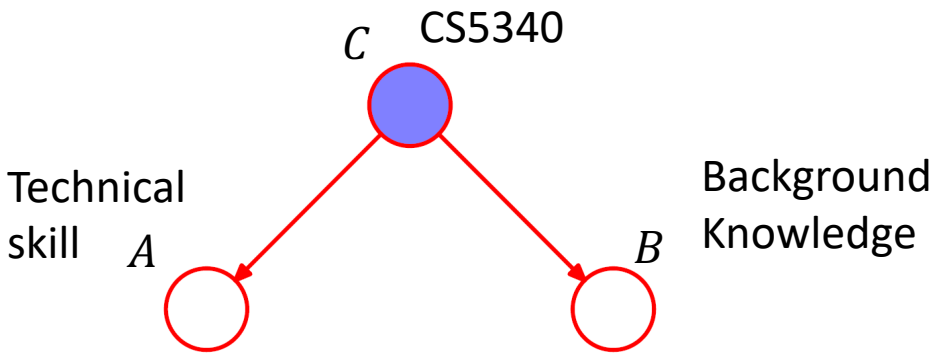


- Given that is raining, there is no further relationship between the how wet it is outside and how cold it is.
- We say that the variable *C* "explains" all of the observed dependence between *A* and *B*.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Tail-Tail

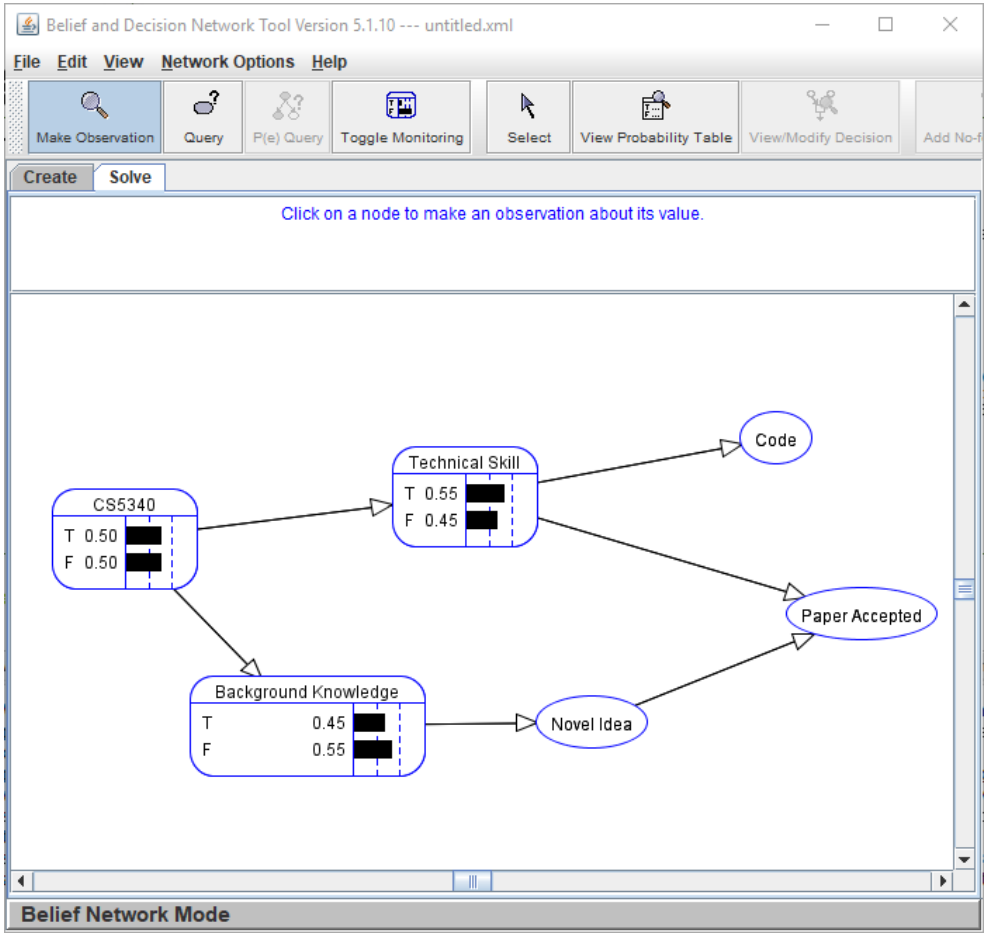
Intuitive interpretation:



- Given that a student took CS5340, there is no further relationship between his knowledge of PGMs and his technical skill
- We say that the variable *C* "explains" all of the observed dependence between *A* and *B*.

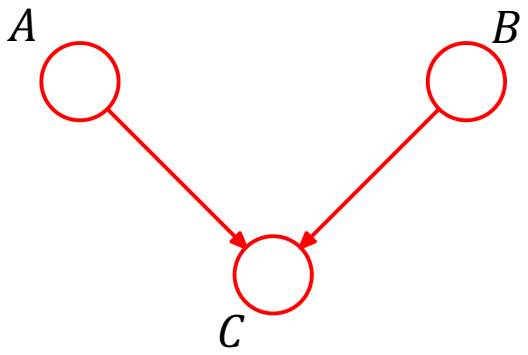
Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Tail-Tail Demo



Head-Head

3.



Joint distribution corresponding to this graph:

$$p(a, b, c) = p(a)p(b)p(c|a, b)$$

If none of the variables are observed, marginalizing both sides over *c* we obtain:

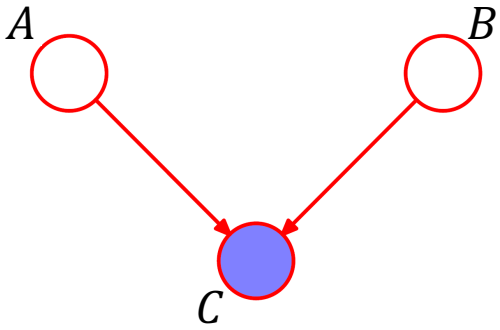
$$p(a, b) = p(a)p(b)$$

A and *B* are independent with no variables observed, in contrast to the two cases:

$$A \perp B \mid \emptyset$$

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Head-Head



If we condition on node *C*, the conditional distribution of *A* and *B* is given by:

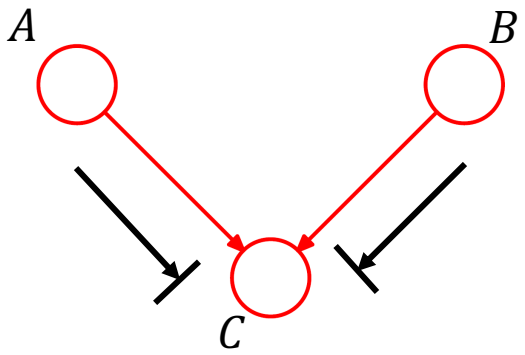
$$p(a,b|c) = \frac{p(a,b,c)}{p(c)} = \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

which in general does not factorize into the product $p(a)p(b)$, and so, in general

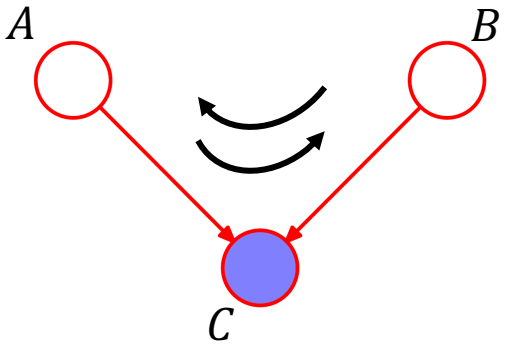
$$A \not\perp B \mid C$$

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Head-Head



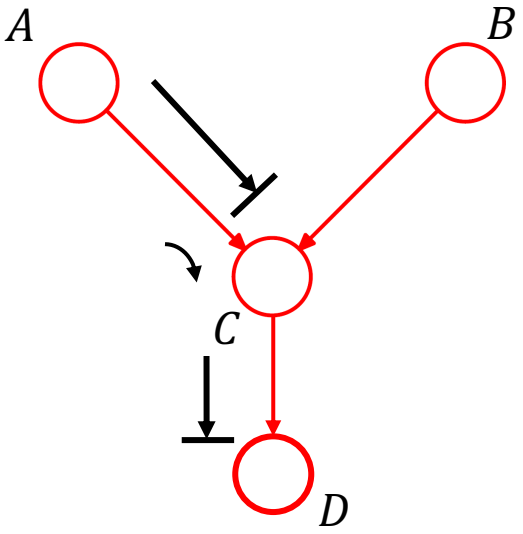
$A \perp B \mid \emptyset$



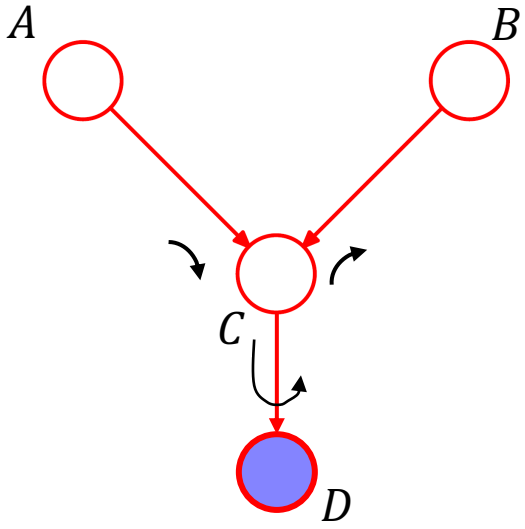
$A \not\perp B \mid C$

- Node *C* is *head-to-head* with respect to the path from *A* to *B*, also known as the "v-structure".
- When node *C* is unobserved, it "blocks" the path, and the variables *A* and *B* are independent.
- However, conditioning on *C* "unblocks" the path and renders *A* and *B* dependent.

Head-Head



$A \perp B \mid \emptyset$

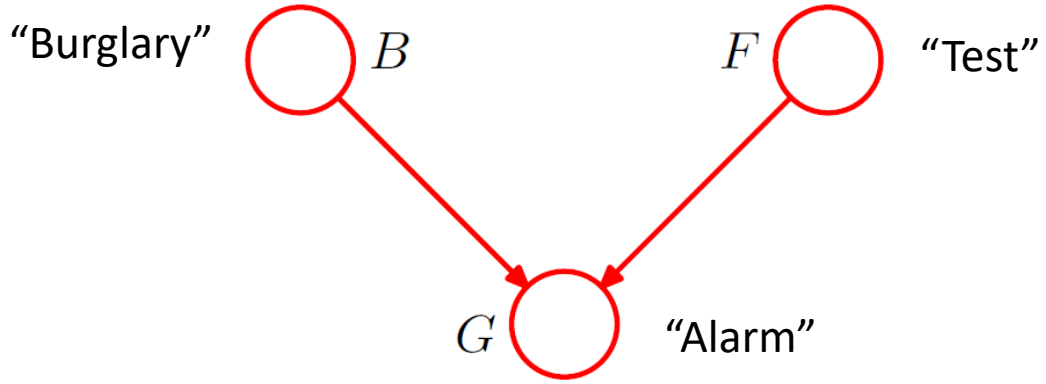


$A \not\perp B \mid D$

- The observation of any descendent node of *C* "unblocks" the path from *A* to *B*.

Head-Head

Intuitive interpretation:

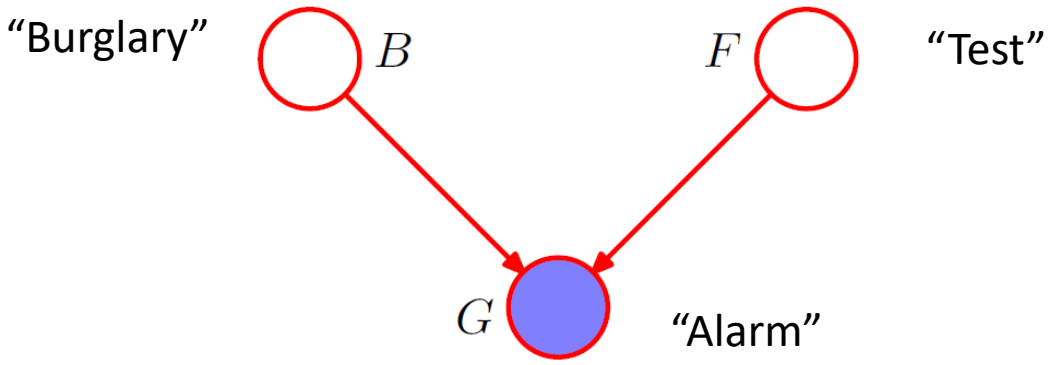


- No relation between the burglary and test if alarm is not checked.
- This implies conditional independence of "burglary" and "Test" when "Alarm" is not observed.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Head-Head

Intuitive interpretation:

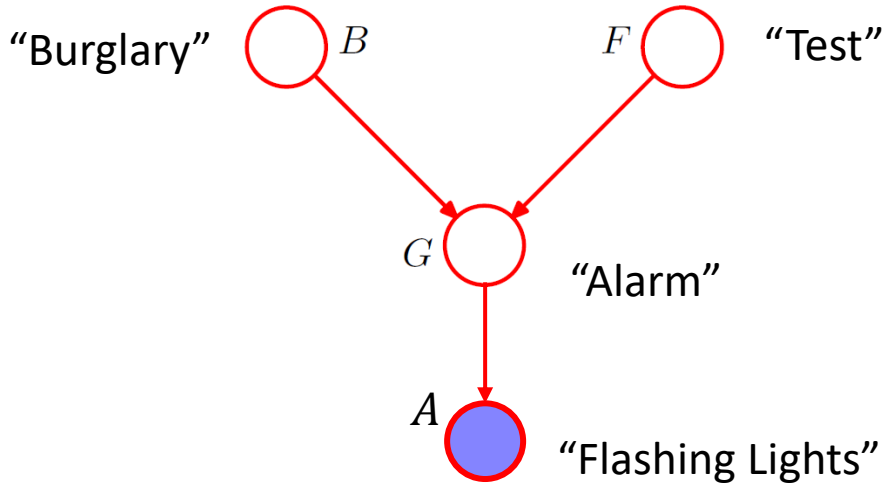


- Suppose the alarm goes off. Knowing that a test is occurring lowers our belief that we are being burgled.
- Burglary and test are now no longer independent.
- This is know as the "explaining-away" effect.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

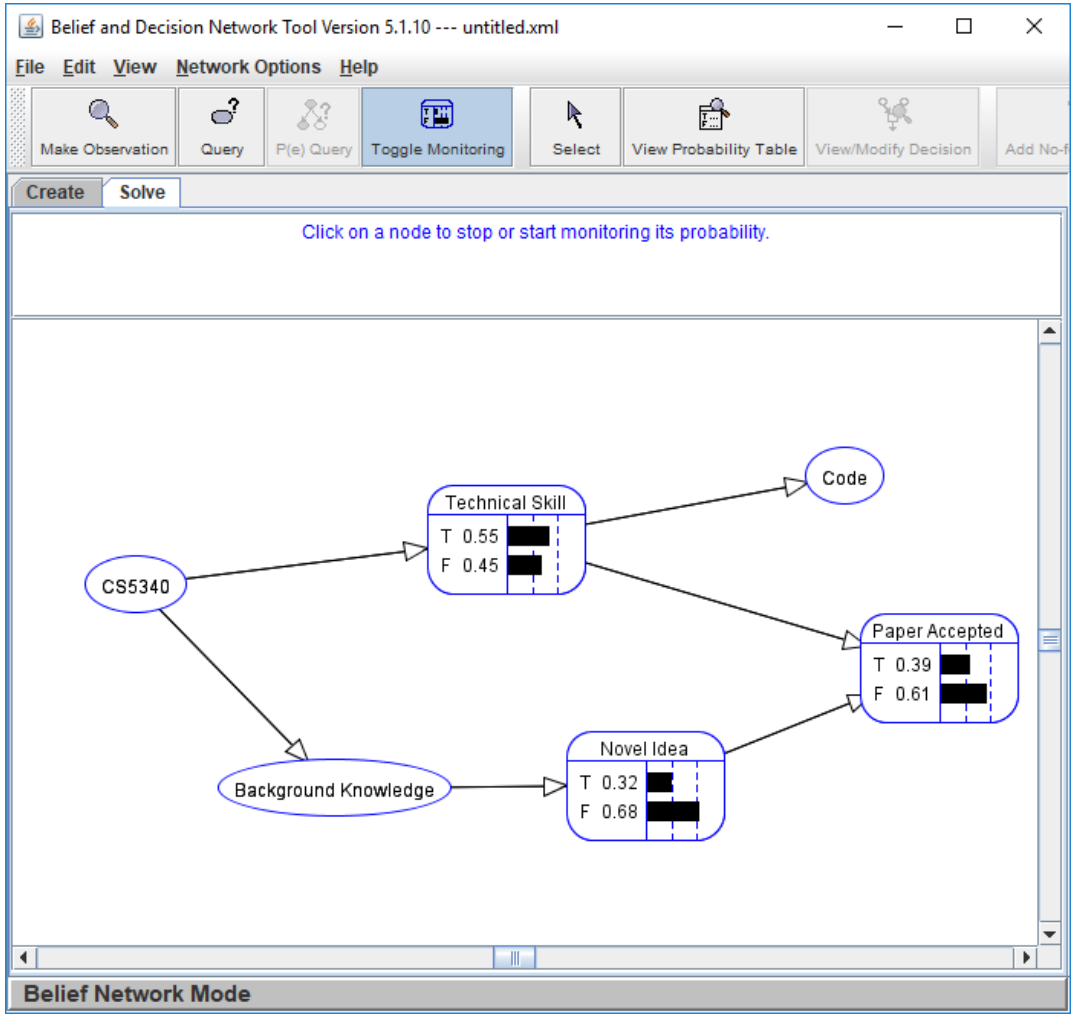
Head-Head

Intuitive interpretation:

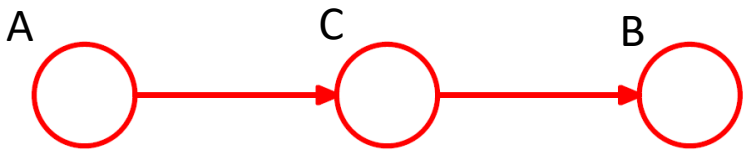


- Lights will flash when alarm goes off.
- Suppose the light starts to flash, and we know that there is a test going on.
- Knowing that there is a test lowers our belief that a burglary is also occuring, i.e. burglary and test are now no longer independent.

Head-Head Demo

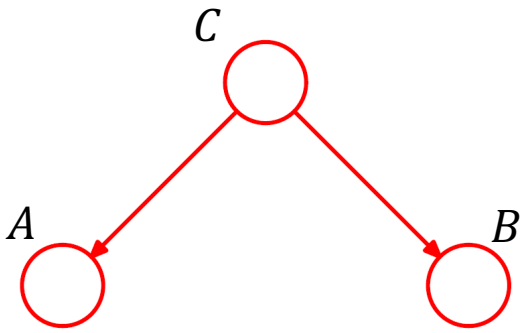


The Canonical 3-node graphs



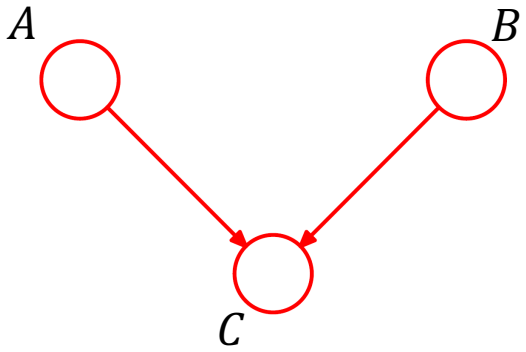
Head-Tail

(Chain/Causal-trail)



Tail-Tail

(Tent/Common cause)



Head-Head

(V-structure/Collider/Common Effect)

Graph Separation

- We have seen earlier that $A \perp B \mid C$ if all trails from nodes in set A are "blocked" from nodes in set B when all nodes from set C are observed.

- *A* is said to be d-separated from *B* by *C*, and the joint distribution over all of the variables in the graph will satisfy $A \perp B \mid C$.

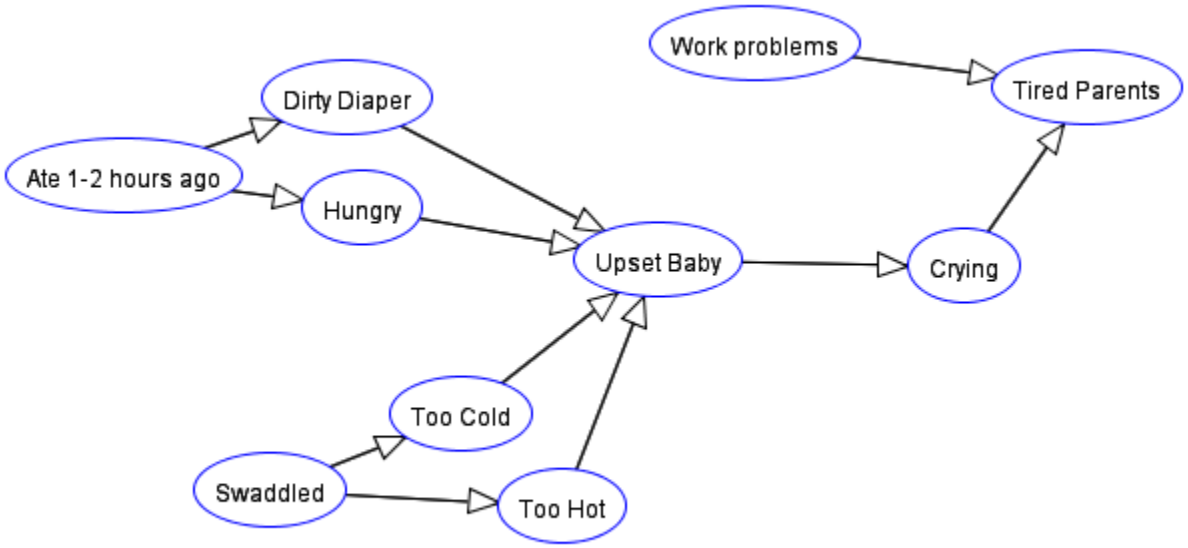
Graph Separation

- From the three canonical 3-node graphs, any trail is said to be "blocked" if it includes a node such that either:

  a) The arrows on the trail meet either head-to-tail or tail-to-tail at the node, and the node is in the set *C*, or

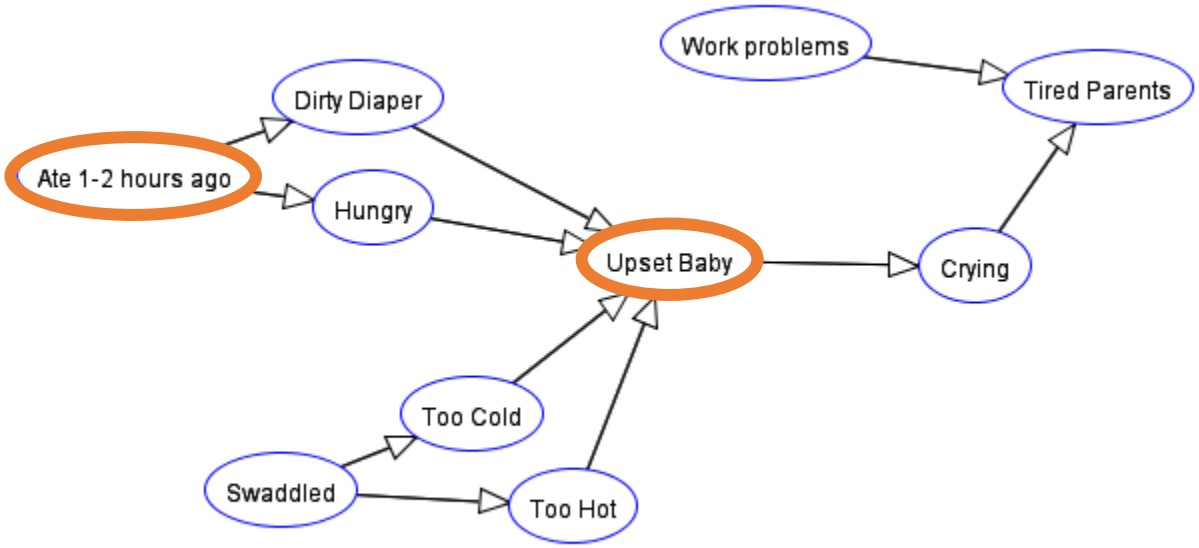
  b) The arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in the set *C*.

d-separation



d-separation

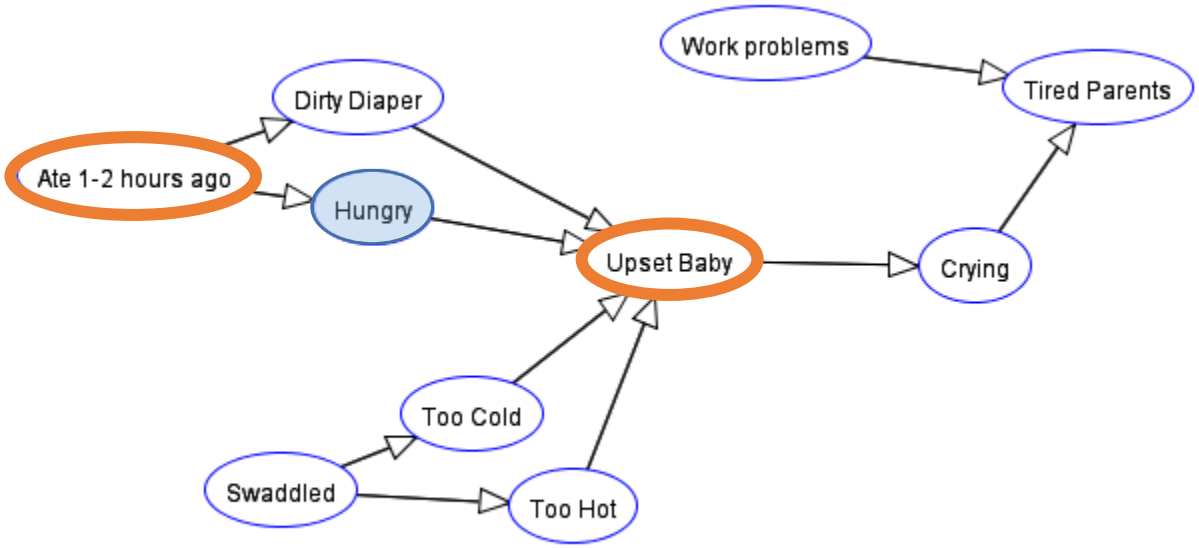
("Ate 1-2 Hours ago" ⊥ "Upset Baby" | Ø)?



False

d-separation

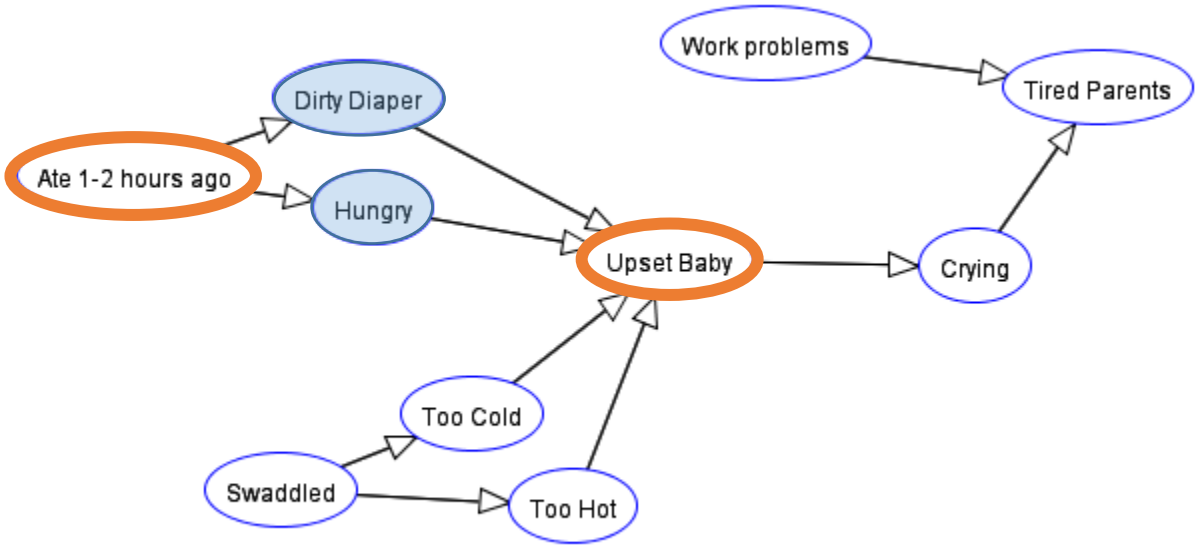
("Ate 1-2 Hours ago" ⊥ "Upset Baby" | "Hungry")?



False

d-separation

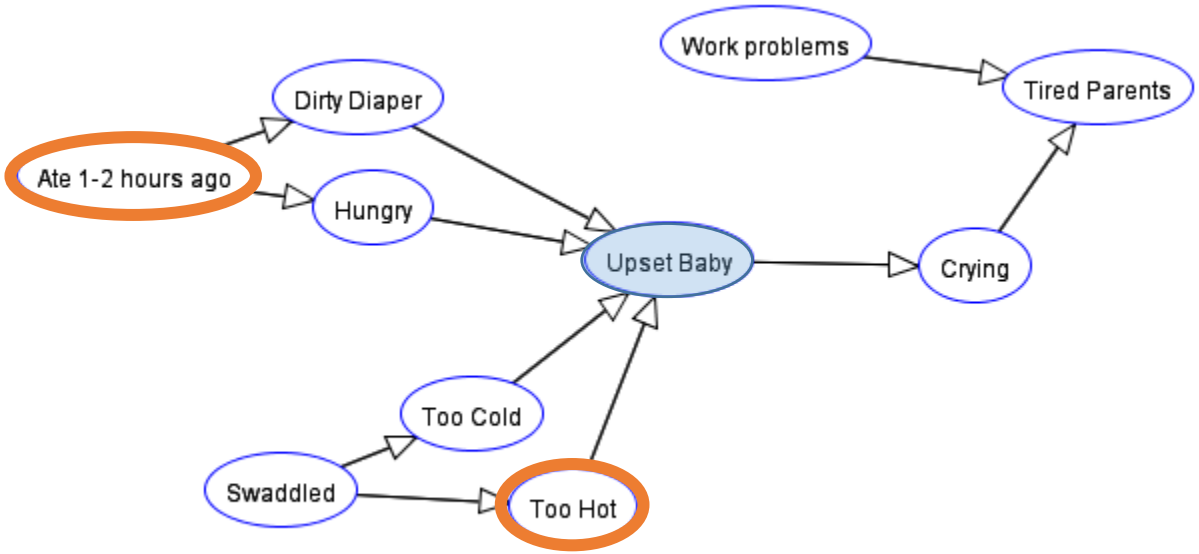
("Ate 1-2 Hours ago" ⊥ "Upset Baby" | "Hungry", "Dirty Diaper")?



True

d-separation

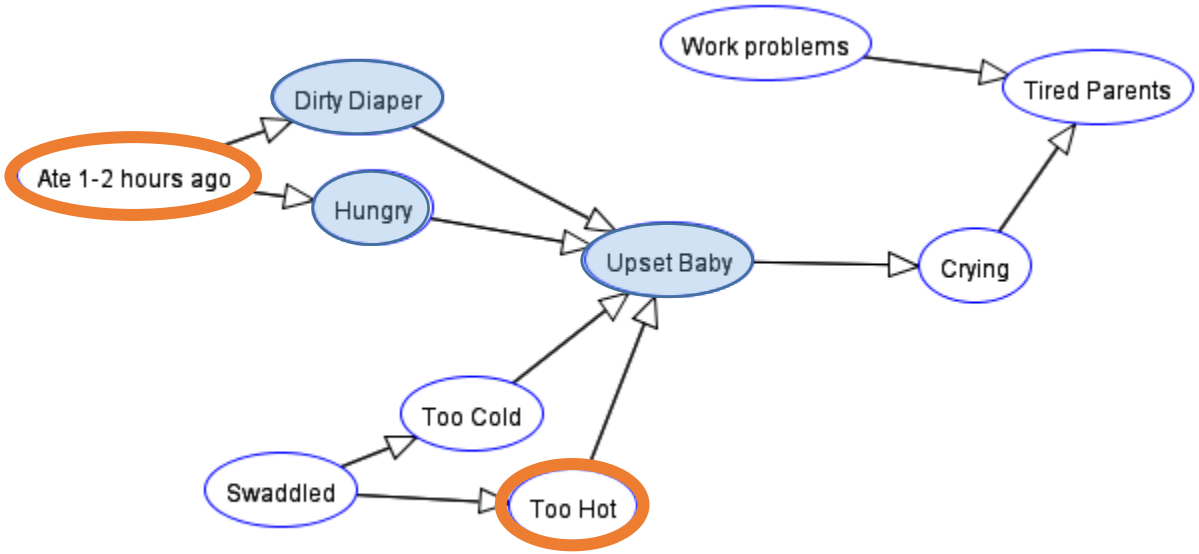
("Ate 1-2 Hours ago" ⊥ "Too Hot" | "Upset Baby")?



False

d-separation

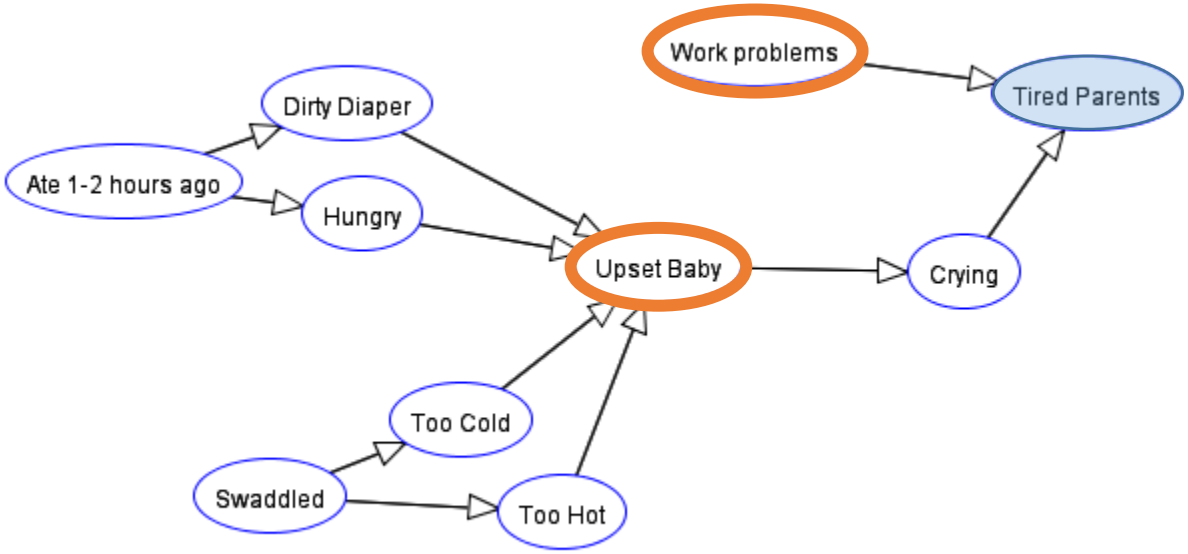
("Ate 1-2 Hours ago" ⊥ "Too Hot" | "Upset Baby", "Hungry", "Dirty Diaper")?



True

d-separation

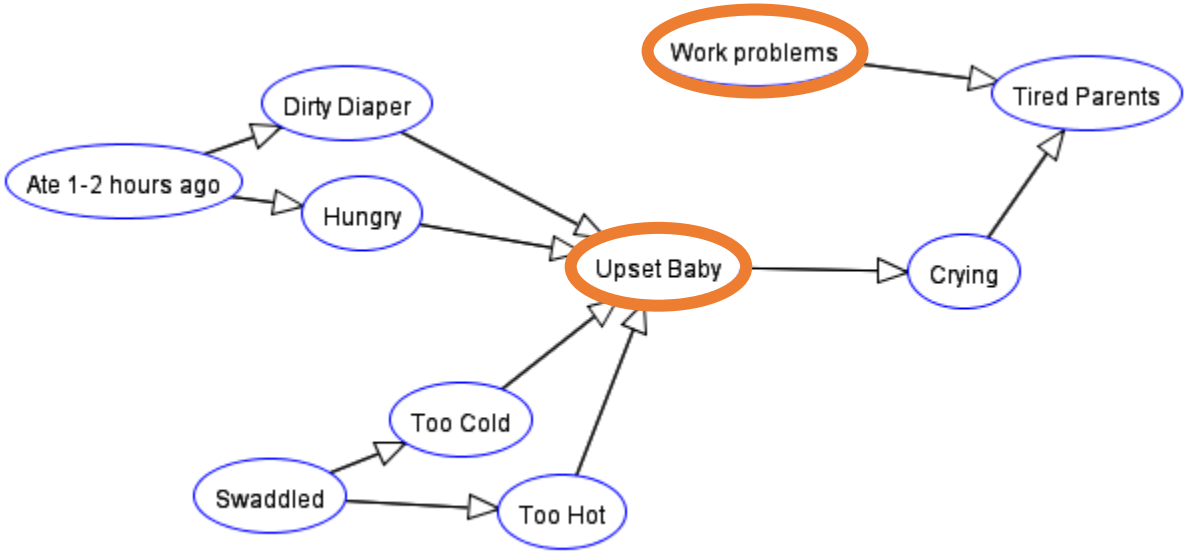
("Work Problems" ⊥ "Upset Baby" | "Tired Parents")?



False

d-separation

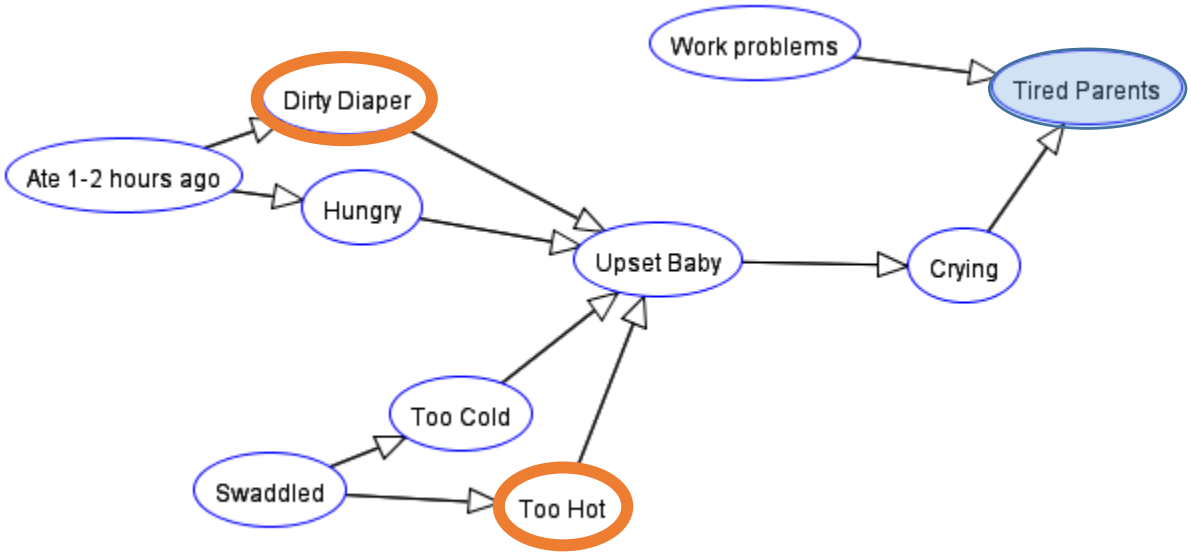
("Work Problems" ⊥ "Upset Baby" | Ø)?



True

d-separation

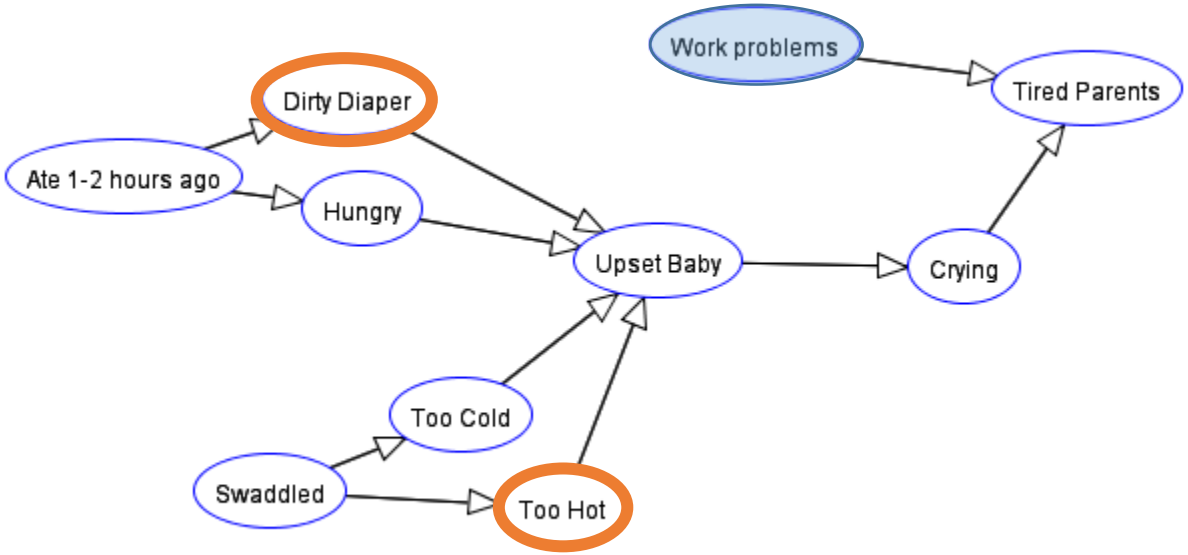
("Dirty Diaper" ⊥ "Too Hot" | "Tired Parents")?



False

d-separation

("Dirty Diaper" ⊥ "Too Hot" | "Work problems")?



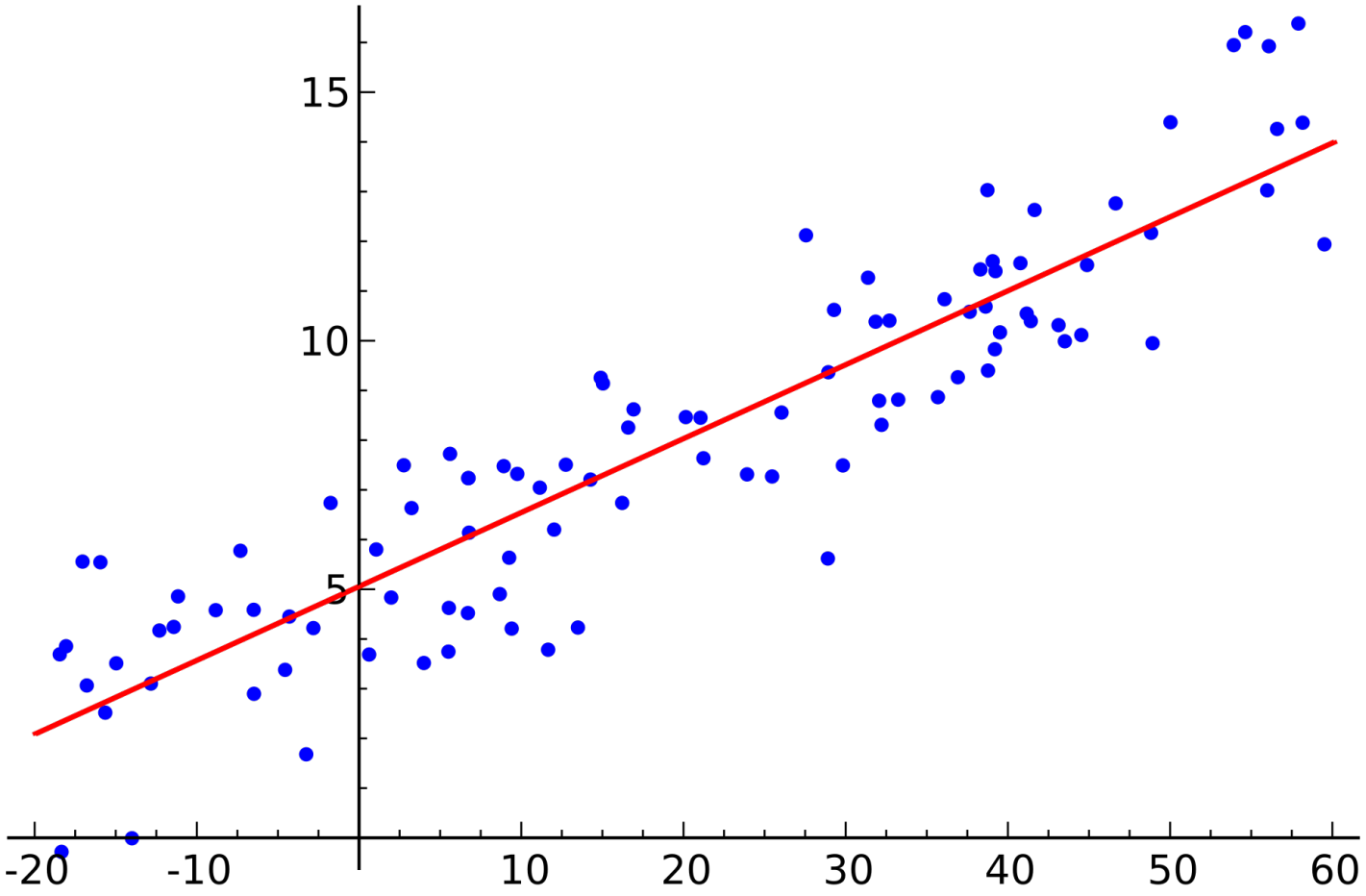
True

NUS National University of Singapore | School of Computing

Bayesian Network Examples

Linear/Nonlinear Regression, Regularization (Ridge Regression) and Naïve Bayes

Example 1: Linear Regression



Linear Regression

- Model for data point with index i:

$$Y[i] = \mathbf{w}^{\mathsf{T}}\mathbf{x}[i] + \epsilon[i]$$

where:

- $\mathbf{x}[i] = [\mathrm{x}[i]_1, \mathrm{x}[i]_2, \ldots, \mathrm{x}[i]_{\mathrm{D}}]^{\mathsf{T}}$ is a D-dimensional observed input vector
- $\mathbf{w} = [w_1, w_2, \ldots, w_D]^{\mathsf{T}}$ is a coefficient vector
- $\epsilon[i] \sim N(0, \sigma_n^2)$ is iid zero-mean Gaussian noise

Why Linear Regression?!?

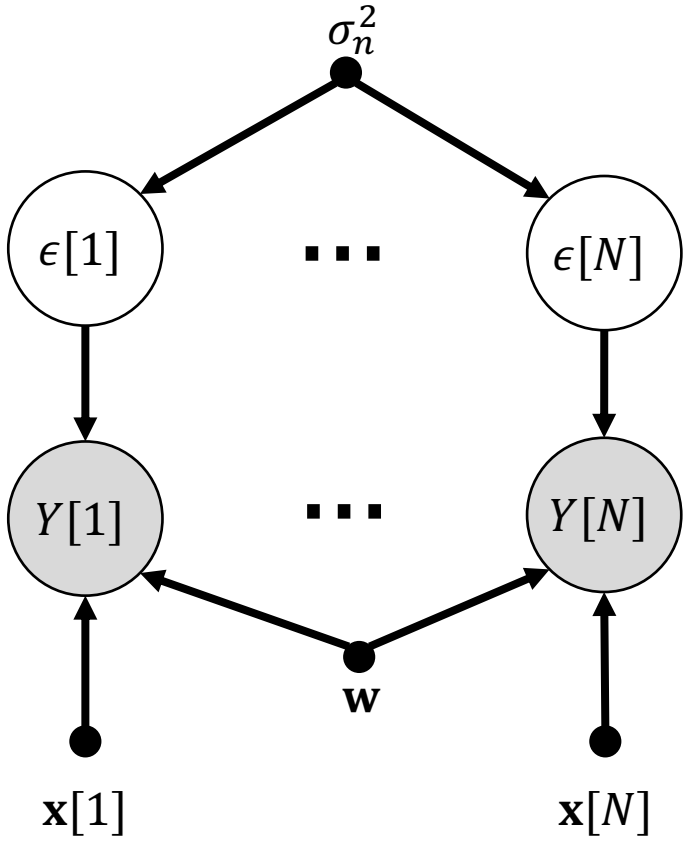
- Model for data point with index i:

$$Y[i] = \mathbf{w}^\top \mathbf{x}[i] + \epsilon[i]$$

Why Linear Regression?

- Basis function "trick"
- Let $\boldsymbol{\phi}(x)$ be some function that transforms x into another vector of "features"
- E.g.:
  - $\boldsymbol{\phi}(x) = [x, x^2, 1]^\top$
  - $\boldsymbol{\phi}(x) = [x^p, x^{p-1}, \ldots\ x^2, x, 1]^\top$
- Then, applying the linear model, we get:
  - $Y[i] = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}[i]) + \epsilon[i]$
  - For the examples above, this is *polynomial regression*.
- $\boldsymbol{\phi}(x)$ can be more complex:
  - E.g.: $\boldsymbol{\phi}(\boldsymbol{x}[i]) = h(\mathbf{A}\mathbf{x}[i])$ where h is an nonlinear "activation function" (What is this?)

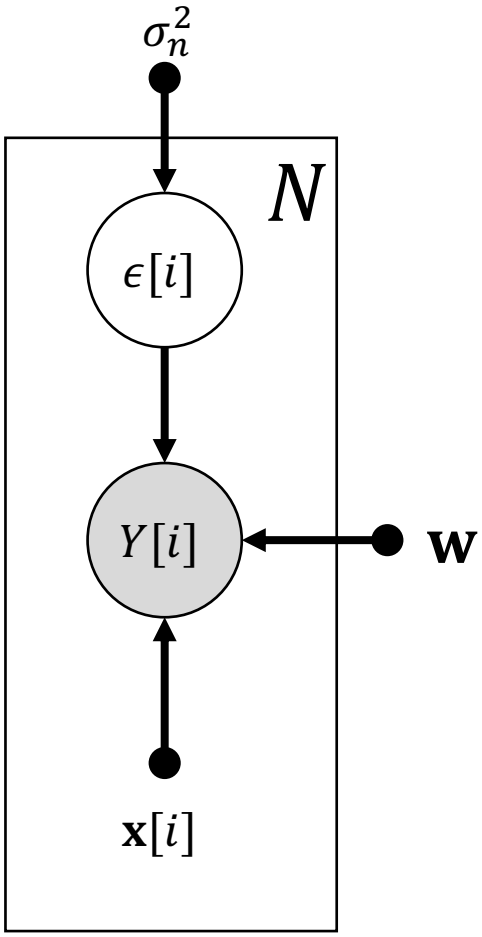
DGM for Linear Regression



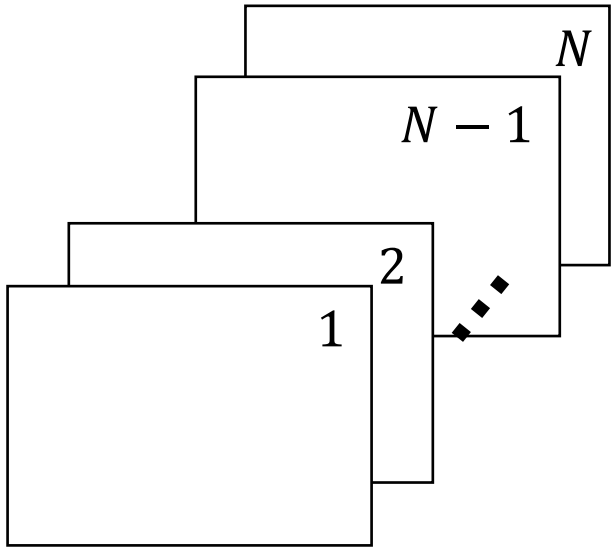
N datapoints
D feature dimensions

- Circles are our random variables
- Shaded circles are observed random variables
- Unshaded circles are unobserved ("latent" or "hidden")
- Filled circles (points) are deterministic parameters

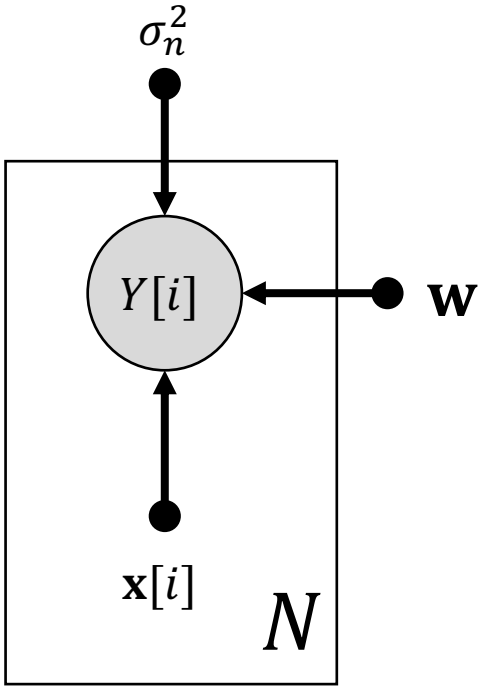
DGM for lazy people



- Plate notation: denotes the nodes are replicated N times

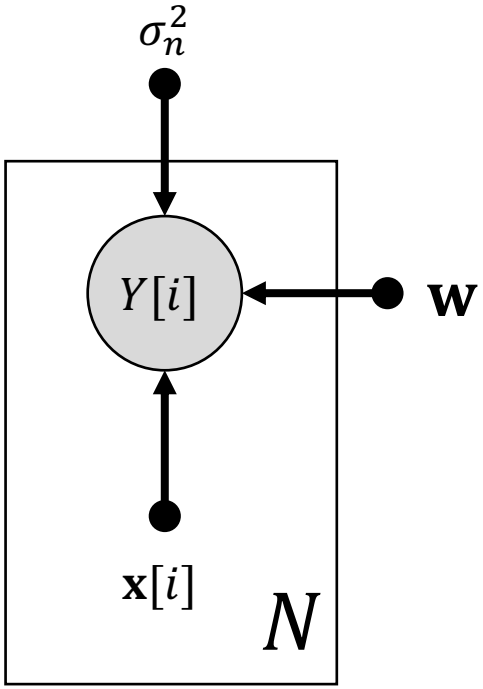


Simplified DGM for even lazier people (like me)



- We can do this.
- **Why?**

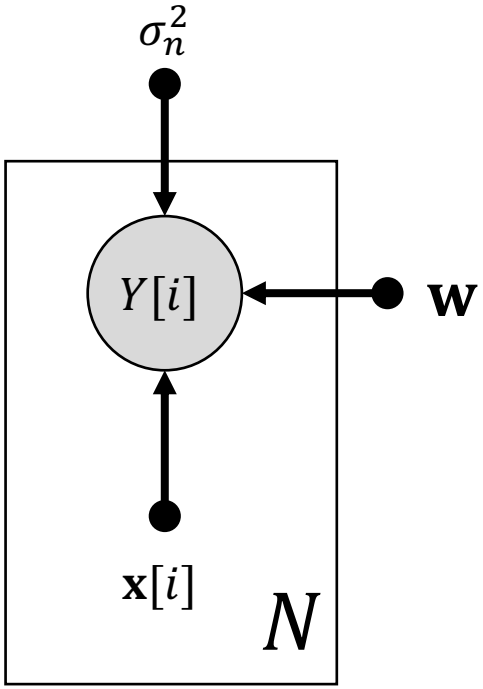
Simplified DGM for even lazier people (like me)



- We can do this.
- **Why?**
- We don't care about ϵ
- If $\epsilon \sim N(0, \sigma_n^2)$, then $Y \sim N(\mathbf{w}^\top \mathbf{x}, \sigma_n^2)$

  (Affine property of Gaussians)

DGM for Linear Regression



- What conditional independence assertions are there?

$$Y[i] \perp Y[i+1] \mid \mathbf{x}[i], \mathbf{w}, \sigma_n^2$$

- Write the factorization:

$$p(y[1], \ldots, y[N]) = \prod_{i}^{N} p(y[i] \mid \mathbf{w}^{\top}\mathbf{x}[i], \sigma_n^2)$$

Assume we know σ_n^2, how can we learn the unknown (deterministic) parameter $\mathbf{w}$?

Recall from Lecture 2: MLE Estimate

Approach 1: Maximum Likelihood Estimation (MLE)

$$\theta_{MLE} = \underset{\theta}{\text{argmax}}[p(x|\theta)]$$

$$= \underset{\theta}{\text{argmax}}\left[\prod_{i=1}^{N} p(x[i] \mid \theta)\right] \qquad \text{(iid)}$$

Likelihood given by pdf

$$p(x|\mu, \sigma^2) = \text{Norm}_x[\mu, \sigma^2] = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x-\mu)^2}{2\sigma^2}$$

MLE for Linear Regression

$$\theta_{MLE} = \text{argmax}_\theta \log p(D|\theta), \theta = \{\mathbf{w}\}$$

$$\log p(D|\theta) = \log \prod_i^N p(y[i]|\mathbf{w}^\top \mathbf{x}[i], \sigma_n^2)$$

$$= \sum_i^N \log \mathcal{N}(y[i]|\mathbf{w}^\top \mathbf{x}[i], \sigma_n^2)$$

$$\propto -\sum_i^N \frac{(y[i] - \mathbf{w}^\top \mathbf{x}[i])^2}{2\sigma_n^2}$$

$$\Rightarrow \text{argmin}_\theta \underbrace{\frac{1}{2}\sum_i^N (y[i] - \mathbf{w}^\top \mathbf{x}[i])^2}_{\mathcal{L}(\boldsymbol{w})}$$

MLE for Linear Regression (cont)

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$= \frac{1}{2}\mathbf{w}^\top(\mathbf{X}^\top\mathbf{X})\mathbf{w} - \mathbf{w}^\top(\mathbf{X}^\top\mathbf{y}) + \frac{1}{2}\mathbf{y}^\top\mathbf{y}$$

where

$$\mathbf{X} = \begin{bmatrix} -\mathbf{x}[1] - \\ -\mathbf{x}[2] - \\ \vdots \\ -\mathbf{x}[N] - \end{bmatrix} \text{ and } \mathbf{y} = \begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[N] \end{bmatrix}$$

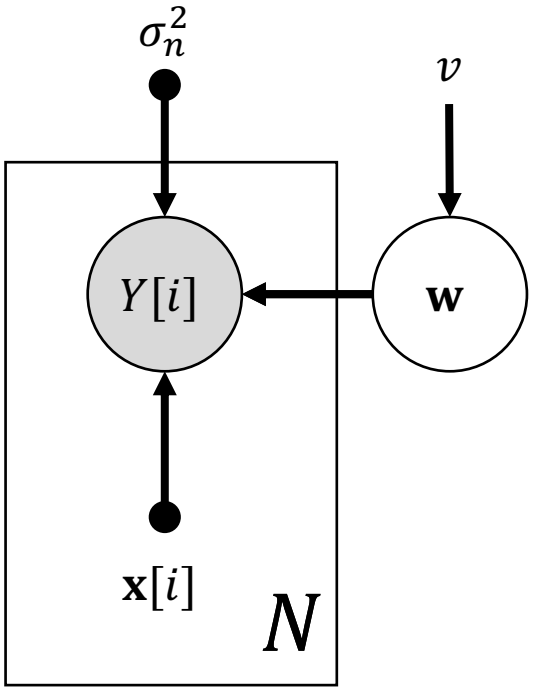
MLE for Linear Regression (cont)

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = (\mathbf{X}^\top \mathbf{X})\mathbf{w} - (\mathbf{X}^\top \mathbf{y}) = 0$$

$$\Rightarrow (\mathbf{X}^\top \mathbf{X})\mathbf{w} = (\mathbf{X}^\top \mathbf{y})$$

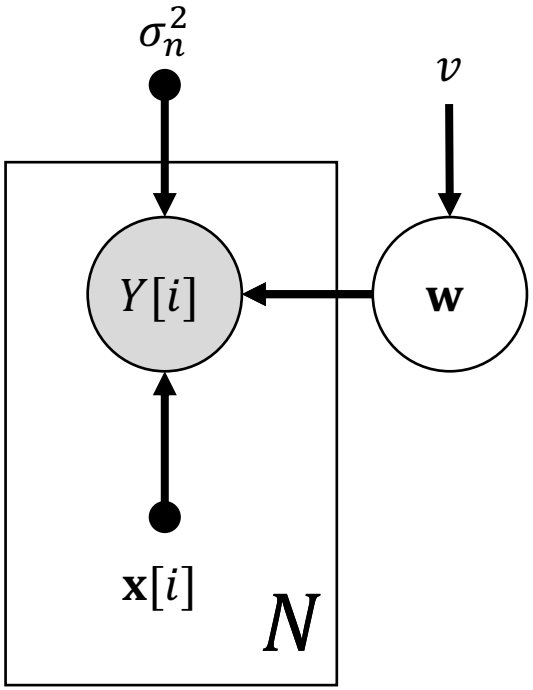
$$\Rightarrow \mathbf{w}_{\text{MLE}} = (\mathbf{X}^\top \mathbf{X})^{-1}(\mathbf{X}^\top \mathbf{y})$$

DGM for Bayesian Linear Regression



- Model uncertainty over $\mathbf{w}$
- The coefficient vector $\mathbf{w}$ is now a random variable with a prior $p(\mathbf{w}|v) = N(\mathbf{0}, v\mathbf{I})$

DGM for Bayesian Linear Regression

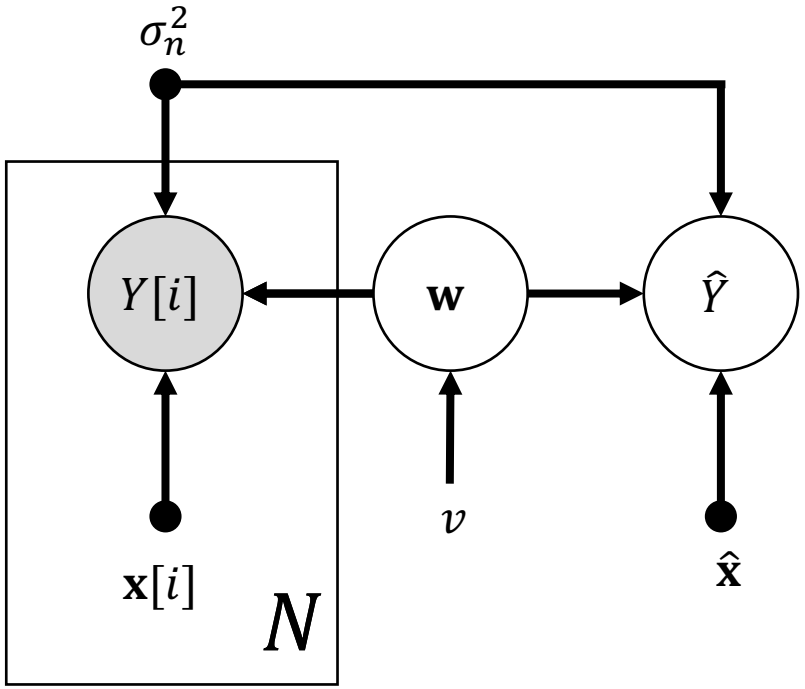


- Write the factorization:

$$p(y[1], \ldots, y[N], \mathbf{w}) = p(\mathbf{w}|v) \prod_i^N p(y[i]|\, \mathbf{w}^\top \mathbf{x}[i], \sigma_n^2)$$

Exercise: Assume we know σ_n^2, give the MAP solution for $\mathbf{w}$.

Predictive DGM



Example 2: Naive Bayes

- Generative Model
  - a "causal" model for "generating" or "constructing" the data.
  - Create a joint model $p(y, \mathbf{x}) = p(\mathbf{x}|y)p(y)$ [or $p(y|\mathbf{x})p(\mathbf{x})$]
- Consider a model for class $c \in \{1, \ldots, K\}$ given input features $\mathbf{x} = [x_1, x_2, \ldots, x_D]^\top$:

$$p(\mathbf{x}, c) = p(\mathbf{x}|c)p(c)$$

- Can be used to classify new data via Bayes rule:

$$p(c|\mathbf{x}) = \frac{p(\boldsymbol{x}|c)p(c)}{\sum_k p(\mathbf{x}|k)p(k)}$$

  - Return c which maximizes $p(c|\mathbf{x})$

$c = 0$ "orange"

$c = 1$ "apple"

Example

Features (X_i)	$C = 1$ ("Orange")	$C = 2$ ("Apple")
Color:Orange	1	0
Color:Red	0	1
Size	4.5cm	5.0cm
Sweet	1	1
HasSeeds	1	1
…		

Naïve Bayes: Assumptions

- class $c \in \{1, \ldots, K\}$ and input $\mathbf{x} = [x_1, x_2, \ldots, x_D]^\top$
- Naïve Bayes assumes that:

$$p(\mathbf{x}|c) = \prod_j p(x_j|c)$$

- A naïve assumption. Why?

Example

All features are independent (given class)!

$$p(\mathbf{x}|c) = \prod_i p(x_i|c)$$

Features (X_i)	$C = 1$ ("Orange")	$C = 2$ ("Apple")
Color:Orange	1	0
Color:Red	0	1
Size	4.5cm	5.0cm
Sweet	1	1
HasSeeds	1	1
…		

$$p(\text{Color: Orange, Color: Red, Size, Sweet, HasSeeds}|\ c) = p(\text{Color: Orange}|c)p(\text{Color: Red}|c)p(\text{Size}|c)p(\text{Sweet}|c)p(\text{HasSeeds}|c)$$

Naïve Bayes: Assumptions

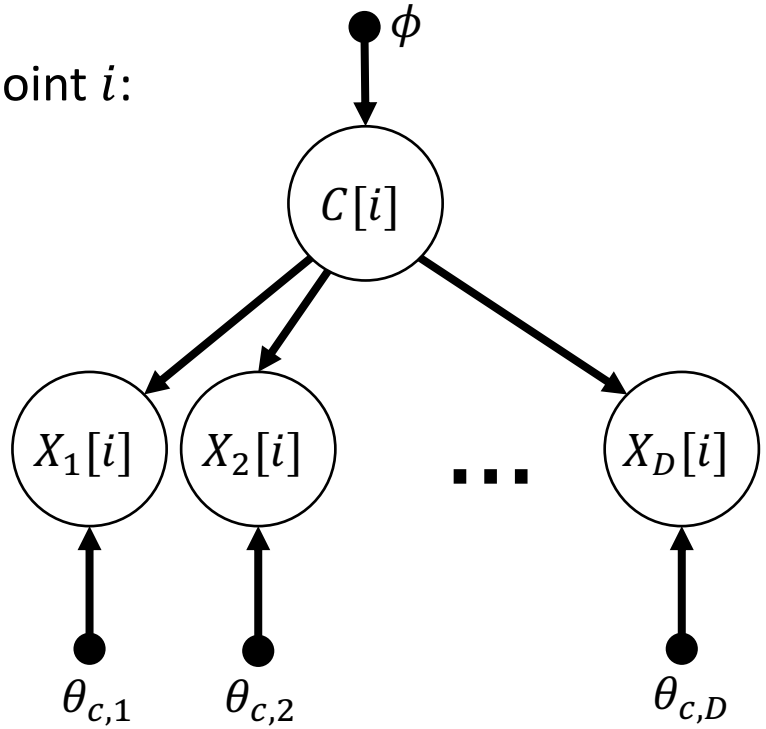
- Class $c \in \{1, \ldots, K\}$ and input $\mathbf{x} = [x_1, x_2, \ldots, x_D]^\top$
- Naïve Bayes assumes that:

$$p(\mathbf{x}|c) = \prod_j p(x_j|c)$$

- A naïve assumption. Why?
- But highly scalable! And can be competitive with other more complex methods (e.g., kernel methods)

DGM for Naïve Bayes

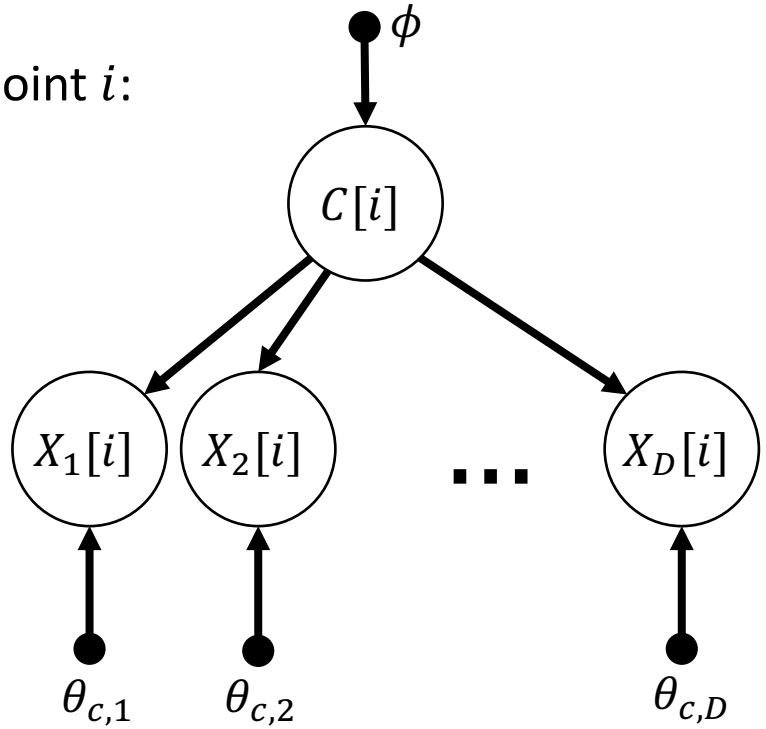
For each data point i:



- Assume data samples are iid.
- Filled dots are deterministic parameters
- If we have priors, we can make parameters random variables (open circles)

DGM for Naïve Bayes

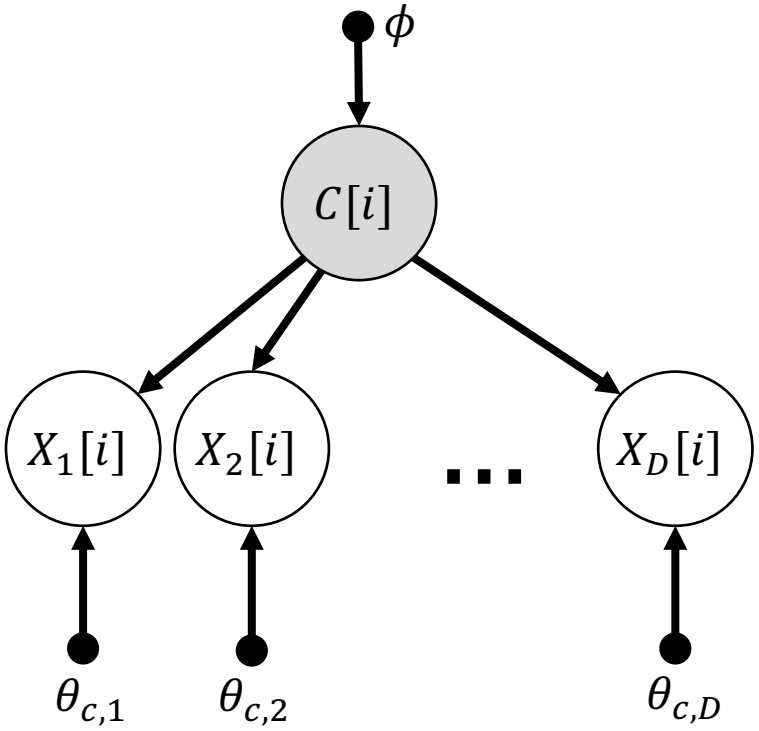
For each data point i:



- What conditional independence assertions can you read?

DGM for Naïve Bayes

For each data point i:

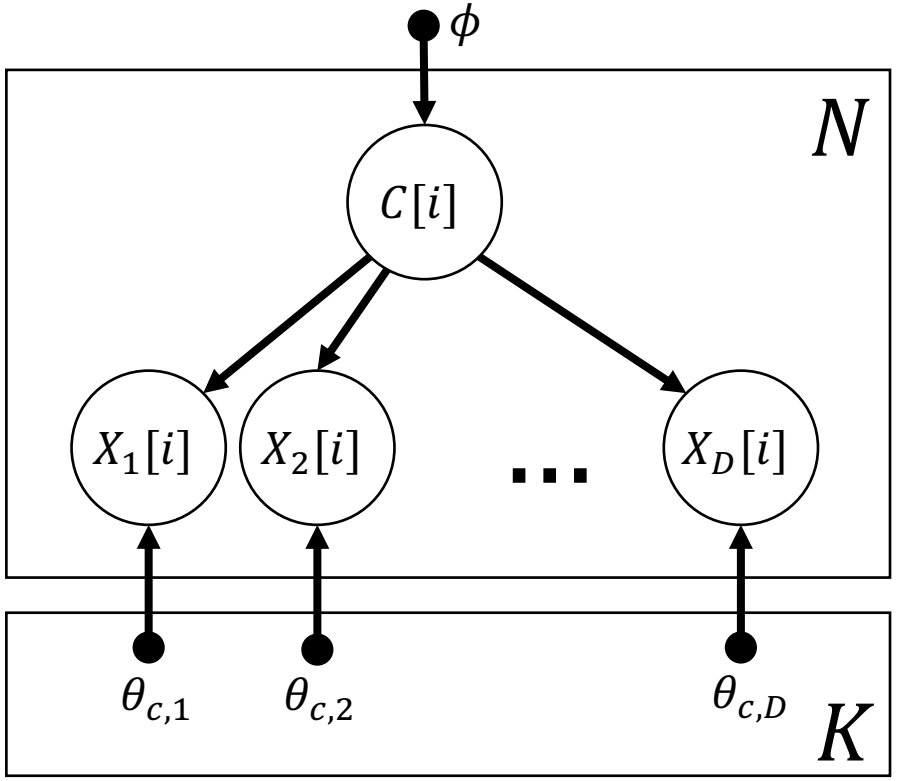


- Given class $C[i]$, all features are conditionally independent.
- Few parameters since we only need $p(x_j|c)$
- Given training samples, $(c_i, x_1, x_2, \ldots, x_D)$, we can learn each $\theta_{c,j}$ separately

$$p(\mathbf{x}, c|\phi, \boldsymbol{\theta}) = p(c|\phi)\prod_i p(x_j|c, \theta_{c,j})$$

DGM for Naïve Bayes

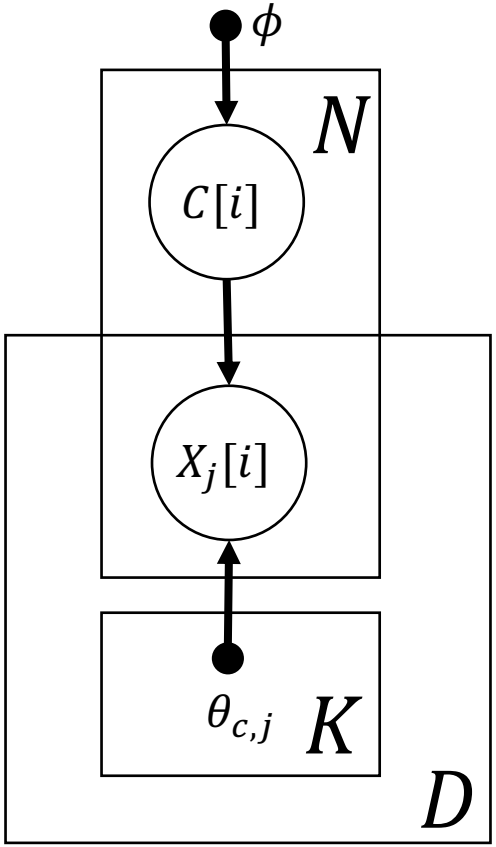
- Applying plate notation (for datapoints and classes):



N datapoints
K classes
D feature dimensions

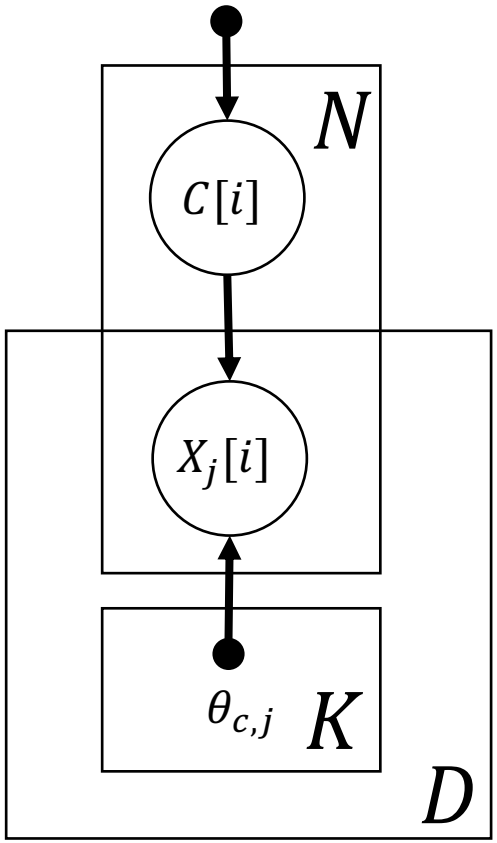
DGM for Naïve Bayes

- Applying nested plate notation (for dimensions):



N datapoints
K classes
D feature dimensions

Extra: Why does Naïve Bayes work so well?



- The conditional independence assumptions are strong in NB.
- Why does it work so well in practice?

	MNB	TWCNB	SVM
Industry Sector	0.582	0.923	0.934
20 Newsgroups	0.848	0.861	0.862
Reuters (micro)	0.739	0.844	0.887
Reuters (macro)	0.270	0.647	0.694

"Tackling the poor assumptions of Naive Bayes classifiers", *Rennie, J.; Shih, L.; Teevan, J.; Karger,* ICML 2003.

Extra: Why does Naïve Bayes work so well?

- "The Optimality of Naïve Bayes", H. Zhang, AAAI 2004
- **Key idea:** it works well when the local feature dependencies between the two classes "cancel out".
- Note: For more details and the proofs, please see the original paper.

NUS National University of Singapore | School of Computing

Theoretical Foundations

Independence-Maps, Soundness, Completeness, Faithfulness and Perfect Maps

Questions we want answers to

- Is the Bayesian Network correct/sound?
  - Does a conditional independence identified by d-separation always exist in the distribution?
- Is the Bayesian Network complete?
  - If a conditional independence exists in the distribution, can it always be detected by d-separation?
- How powerful are Bayesian Networks as a modeling language?
  - Can they exactly represent all conditional independencies for a given distribution?

Independence-Maps (I-Maps)

- **Definition** (*Independence set*) Let P be a distribution over $\mathcal{X}$. Define $\mathcal{I}(P)$ as the set of independence assertions of the form $(X \perp Y \mid Z)$ that hold in P.

- **Definition** (*Independence map*) Let G be associated with independence assertions $\mathcal{I}(G)$. G is an I-map for P if $\mathcal{I}(G) \subseteq \mathcal{I}(P)$

Bayes Nets (BN) and I-maps

- **Definition** (*Bayesian Network*) A Bayesian network is a tuple $B = (G, P)$ where P factorizes according to G and where P is specified as a a set of conditional probability distributions (CPDs) associated with G's nodes.

- **Theorem 4.1:** Given a graph G over a set of random variables $\mathcal{X} = \{X_1, X_2, \ldots, X_N\}$ and P be a joint distribution over the same space. If G is an I-map for P, then P factorizes according to G

Proof of Theorem 4.1

Let $\mathcal{G}$ be a BN structure over a set of random variables $\mathcal{X}$, and let P be a joint distribution over the same space. If $\mathcal{G}$ is an I-map for P, then P factorizes according to $\mathcal{G}$.

PROOF Assume, without loss of generality, that $X_1, \ldots, X_n$ is a *topological ordering* of the variables in $\mathcal{X}$ relative to $\mathcal{G}$ (see definition 2.19). As in our example, we first use the chain rule for probabilities:

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

Now, consider one of the factors $P(X_i \mid X_1, \ldots, X_{i-1})$. As $\mathcal{G}$ is an I-map for P, we have that $(X_i \perp \text{NonDescendants}_{X_i} \mid \text{Pa}_{X_i}^{\mathcal{G}}) \in \mathcal{I}(P)$. By assumption, all of X_i's parents are in the set $X_1, \ldots, X_{i-1}$. Furthermore, none of X_i's descendants can possibly be in the set. Hence,

$$\{X_1, \ldots, X_{i-1}\} = \text{Pa}_{X_i} \cup \boldsymbol{Z}$$

where $\boldsymbol{Z} \subseteq \text{NonDescendants}_{X_i}$. From the local independencies for X_i and from the decomposition property (equation (2.8)) it follows that $(X_i \perp \boldsymbol{Z} \mid \text{Pa}_{X_i})$. Hence, we have that

$$P(X_i \mid X_1, \ldots, X_{i-1}) = P(X_i \mid \text{Pa}_{X_i}).$$

Applying this transformation to all of the factors in the chain rule decomposition, the result follows. ■

From: Koller and Friedman, "Probabilistic Graphical Models: Principles and Techniques"

Bayes Nets (BN) and I-maps

- **Theorem 4.2:** Let P be a joint distribution over $\mathcal{X}$ and G be a Bayesian Network structure over $\mathcal{X}$. If P factorizes according to G, then the *local independence assertions* $\mathcal{I}_l(G) \subseteq \mathcal{I}(P)$.

  The local Markov independencies $\mathcal{I}_l(G)$ is the set of all basic conditional independence assertions of the form:

$$\{X_i \perp (X_{\text{nonDesc}(x_i)} \setminus X_{\pi_i}) \mid X_{\pi_i}\}$$

Thm 4.2 (Proof sketch)

assume P factorizes accord. to G

goal: show $\mathcal{I}_l(G) \subseteq \mathcal{I}(P)$

$$p(x_i \mid X \setminus x_i) = p(x_i \mid X_{\pi_i}) \quad \text{for all } x_i$$

Consider factor X_k

$$p(X_k \mid X \setminus X_k) = \frac{p(X_1, \ldots, X_N)}{p(\{X_1, \ldots, X_N\} \setminus X_k)}$$

$$= \frac{\prod_{i=1}^{N} p(x_i \mid X_{\pi_i})}{\sum_{x_k} \prod_{j=1}^{N} p(x_j \mid X_{\pi_j})} = \frac{p(X_k \mid X_{\pi_k}) \prod_{j \neq k} p(x_j \mid X_{\pi_j})}{\sum_{x_k} p(X_k \mid X_{\pi_k}) \prod_{j \neq k} p(x_j \mid X_{\pi_j})}$$

$$= p(X_k \mid X_{\pi_k})$$

$\Rightarrow (X_k \perp X_{\text{nondescs}(X_k)} \mid X_{\pi_k})$ given topological ordering. □

Connection to d-separation

- **Definition** (*Global Markov Independencies*) The set of all independencies that correspond to d-separation in graph G is the set of global Markov independencies:

$$\mathcal{I}(G) = \{(X \perp Y \mid Z) : \mathrm{dsep}_G(X; Y|Z)\}$$

Is d-separation sound and complete?

Soundness

- **Theorem 4.3** (*Soundness*) If a distribution P factorizes according to G, then $\mathcal{I}(G) \subseteq \mathcal{I}(P)$

- **Translation:** if two nodes are found to be d-separated given Z, they are *in fact* conditionally independent given Z in P.

Completeness

- Does d-separation find all conditional independencies?
  - If two variables X and Y are conditionally independent given Z, are they are d-separated?

- **Definition** (*faithful*): P is faithful to G if for any conditional independence $(X \perp Y \mid Z) \in \mathcal{I}(P)$ then $\mathrm{dsep}_G(X; Y|Z)$.
  - **Translation:** any independence in P is reflected as a d-separation in the graph G.

Completeness v1

- **Definition** (*Completeness?*) For any distribution P that factorizes over G, then P is faithful to G.
- Does this definition work?

Completeness v1

- **Definition** (*Completeness?*) For any distribution P that factorizes over G, then P is faithful to G.
- Does this definition work?
- No.
  - if P is faithful to G if for any conditional independence $(X \perp Y \mid Z) \in \mathcal{I}(P)$ then $\text{dsep}_G(X; Y|Z)$.
  - Consider ***contrapositive***: if two nodes X and Y are **not** d-separated, then they are **not** conditionally independent in *all* distributions P that factorize over G.

"Weak" Completeness

- **Theorem 4.4** (*"Weak" Completeness*) If $(X \perp Y \mid Z)$ in all distributions P that factorize over G, then $\mathrm{dsep}_G(X; Y|Z)$.
- **Contrapositive:** If X and Y are not d-separated in G, then X and Y are dependent in *some* distribution P that factorizes over G.

- **Theorem 4.5** (*"Almost" Completeness*) For *almost* all distributions P that factorize over G, we have $\mathcal{I}(P) = \mathcal{I}(G)$

Perfect Maps

- **Definition** (*Perfect Map*) A graph G is a perfect map for a probability distribution P if $\mathcal{I}(G) = \mathcal{I}(P)$.

Can *G* represent all the independencies for a given *P*?

- How "powerful" are Bayes Nets as a modeling language?
- Can we find a Bayes Net that represents all the conditional independencies in a given probability distribution P?

Perfect Maps

- **Definition** (*Perfect Map*) A graph G is a perfect map for a probability distribution P if $\mathcal{I}(G) = \mathcal{I}(P)$.

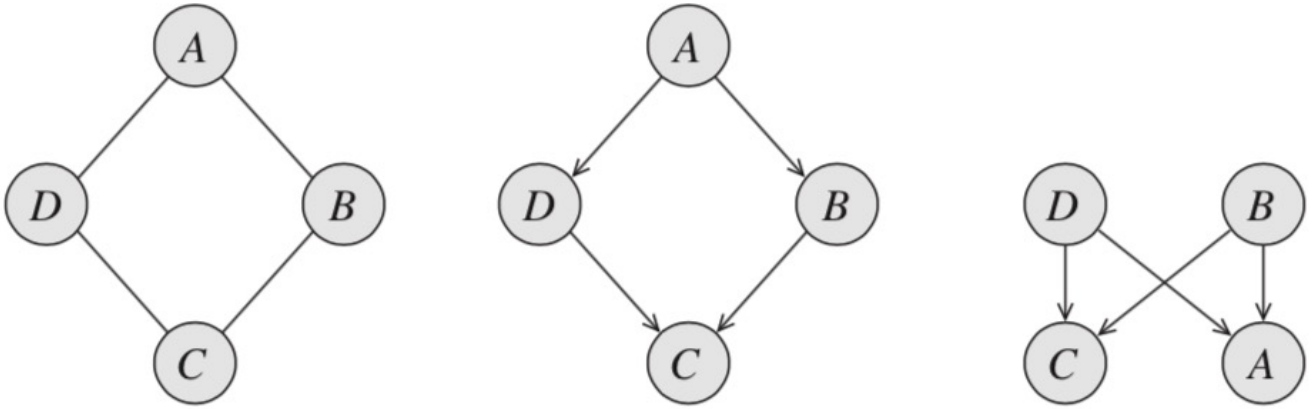
Does every distribution have a perfect map?

Perfect Maps

- Consider a distribution *P* with two conditional independence assertions: $\mathcal{I}(P)$ contains:
  - $(A \perp C \mid \{B, D\})$ and
  - $(B \perp D \mid \{A, C\})$

Can you find a perfect map for $\mathcal{I}(P)$?

Counter-example



- Can we find a Bayes Net that represents all the conditional independencies in a given probability distribution P?
- No. Bayes Nets cannot model certain sets of conditional independence assertions.
- Next week, we will learn about Markov random fields (MRFs) which use undirected links.

Questions we want answers to

- Is the Bayesian Network correct/sound?
  - Does a conditional independence identified by d-separation always exist in the distribution? **Yes.** ☺
- Is the Bayesian Network complete?
  - If a conditional independence exists in the distribution, can it always be detected by d-separation? **Almost Yes.** 😐
- How powerful are Bayesian Networks as a modeling language?
  - Can they exactly represent all conditional independencies for a given distribution? **No.** ☹

NUS National University of Singapore | School of Computing

Appendix

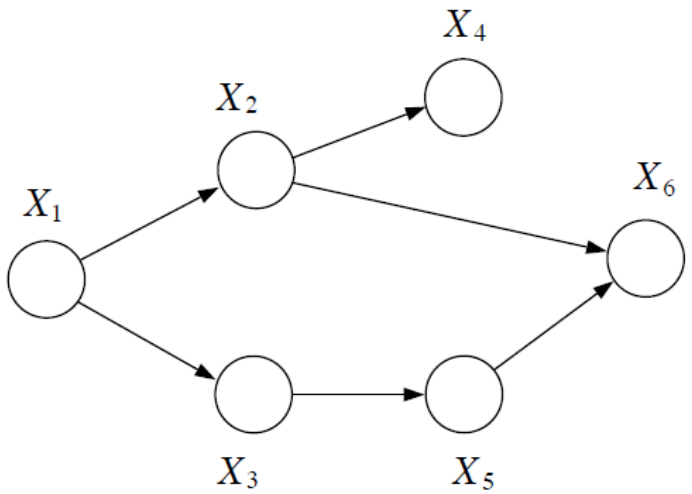
Extra Stuff

Bayesian Networks: Joint Probability

Example:

Let's verify that the basic sets of conditional independence are indeed represented in the joint probability:

$$p(x_1, \ldots, x_6) = p(x_1)\mathrm{p}(x_2|x_1)\mathrm{p}(x_3|x_1)\mathrm{p}(x_4|x_2)\mathrm{p}(x_5|x_3)\mathrm{p}(x_6|x_2, x_5)$$



$X_1 \perp \emptyset \mid \emptyset,$

$X_2 \perp \{X_3, X_5, \} \mid X_1,$

$X_3 \perp \{X_2, X_4\} \mid X_1,$

$X_4 \perp \{X_1, X_3, X_5, X_6\} \mid X_2,$

$X_5 \perp \{X_1, X_2, X_4\} \mid X_3,$

$X_6 \perp \{X_1, X_3, X_4\} \mid \{X_2, X_5\}$

Bayesian Networks: Joint Probability

Example:

Let's verify that X_1 and X_3 are independent of X_4 given X_2.

First we compute the marginal probability of $\{X_1, X_2, X_3, X_4\}$:

$$
\begin{aligned}
p(x_1, x_2, x_3, x_4) &= \sum_{x_5} \sum_{x_6} p(x_1, x_2, x_3, x_4, x_5, x_6) \\
&= \sum_{x_5} \sum_{x_6} p(x_1) p(x_2 \mid x_1) p(x_3 \mid x_1) p(x_4 \mid x_2) p(x_5 \mid x_3) p(x_6 \mid x_2, x_5) \\
&= p(x_1) p(x_2 \mid x_1) p(x_3 \mid x_1) p(x_4 \mid x_2) \sum_{x_5} p(x_5 \mid x_3) \sum_{x_6} p(x_6 \mid x_2, x_5) \\
&= p(x_1) p(x_2 \mid x_1) p(x_3 \mid x_1) p(x_4 \mid x_2),
\end{aligned}
$$

Bayesian Networks: Joint Probability

Example:

Let's verify that X_1 and X_3 are independent of X_4 given X_2.

Next we compute the marginal probability of $\{X_1, X_2, X_3\}$:

$$\begin{aligned} p(x_1, x_2, x_3) &= \sum_{x_4} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2) \\ &= p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1). \end{aligned}$$

Dividing the two marginal yields the desired conditional:

$$p(x_4 \mid x_1, x_2, x_3) = p(x_4 \mid x_2),$$

Which demonstrates the conditional independence relationship $X_4 \perp \{X_1, X_3\} \mid X_2$.

Question: Is RHS properly normalized?

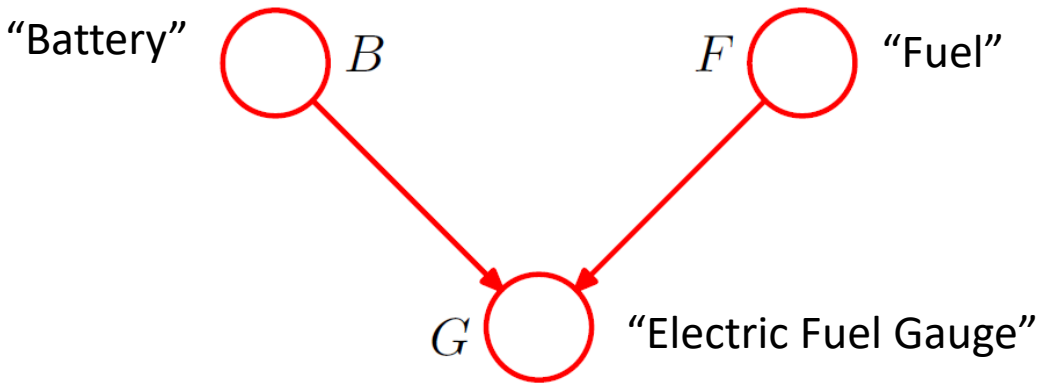
$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | x_{\pi_i})$$

- We claim that the above is true. But is the right hand side (RHS) normalized?
  - Does $\sum_{x_1} \sum_{x_2} \ldots \sum_{x_N} \prod_{i=1}^{N} p(x_i | x_{\pi_i}) = 1$?*
- If you think yes, prove it. If no, provide a counter-example.

*replace summations with integrals where necessary

Three Canonical 3-Node Graphs

Numerical Example:

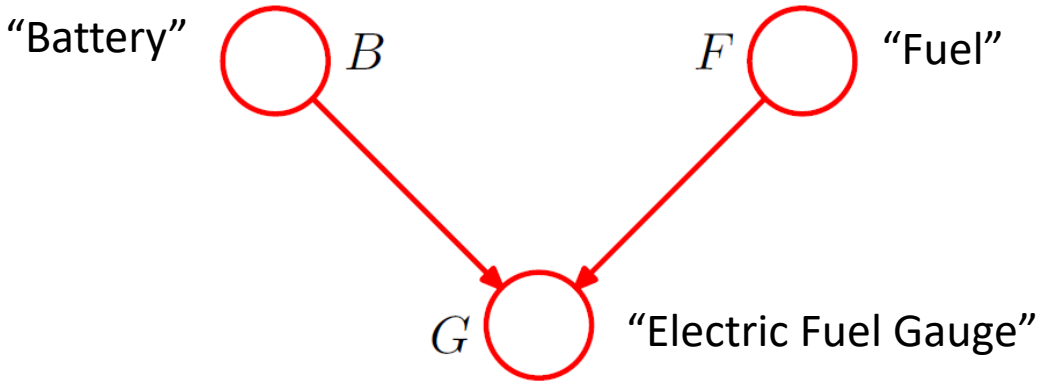


- *B* : battery state that is either charged (*B* = 1) or flat (*B* = 0).
- *F :* fuel tank state that is either full of fuel (*F* = 1) or empty (*F* = 0).
- *G* : electric fuel gauge state which indicates either full (*G* = 1) or empty (*G* = 0).

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Three Canonical 3-Node Graphs

Numerical Example:



Given:

$$p(B=1) = 0.9$$
$$p(F=1) = 0.9$$

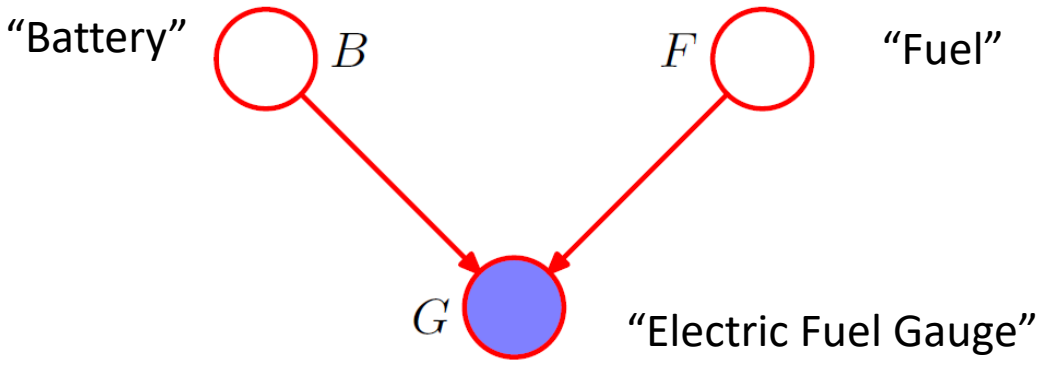
$$p(G=1|B=1,F=1) = 0.8$$
$$p(G=1|B=1,F=0) = 0.2$$
$$p(G=1|B=0,F=1) = 0.2$$
$$p(G=1|B=0,F=0) = 0.1$$

Before we observe any data, the prior probability of the fuel tank being empty is $p(F = 0) = 0.1$.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Three Canonical 3-Node Graphs

Numerical Example:



Suppose that we observe the fuel gauge and it reads empty, i.e., $G = 0$, we have:

$$p(F=0|G=0) = \frac{p(G=0|F=0)p(F=0)}{p(G=0)} \simeq 0.257$$

where

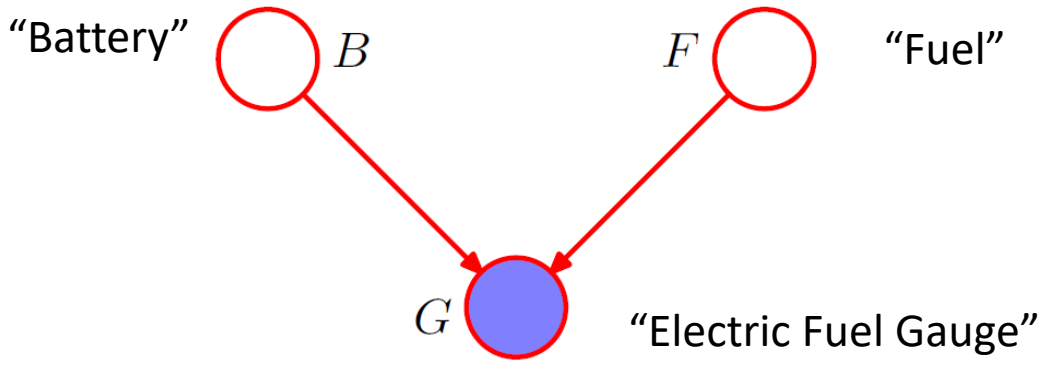
$$p(G=0) = \sum_{b\in\{0,1\}} \sum_{f\in\{0,1\}} p(G=0|B,F)p(B)p(F) = 0.315$$

$$p(G=0|F=0) = \sum_{b\in\{0,1\}} p(G=0|B,F=0)p(B) = 0.81$$

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Three Canonical 3-Node Graphs

Numerical Example:



Hence,

0.257 0.1

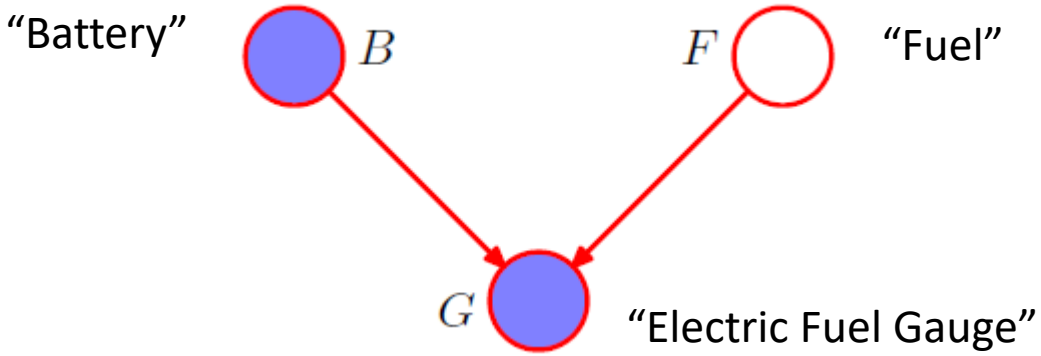
$$p(F = 0|G = 0) > p(F = 0)$$

Observing that the gauge reads empty makes it more likely that the tank is indeed empty, as we would intuitively expect.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Three Canonical 3-Node Graphs

Numerical Example:

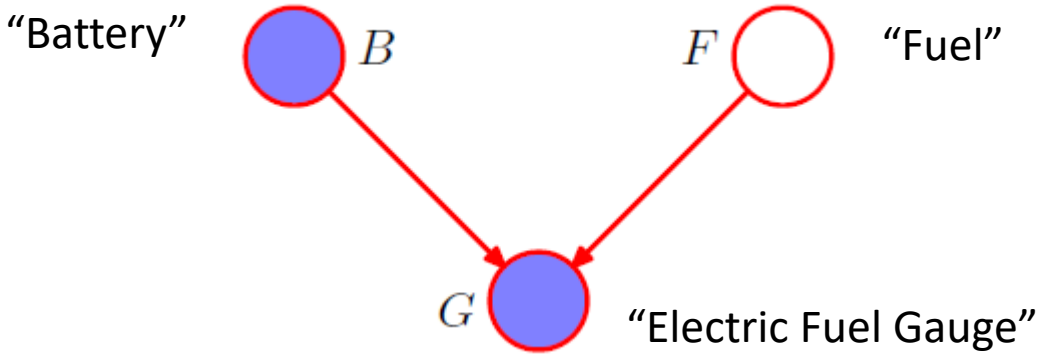


- If we also check the state of the battery and find that it is flat, i.e., $B = 0$.
- We have now observed the states of both fuel gauge and battery.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Three Canonical 3-Node Graphs

Numerical Example:



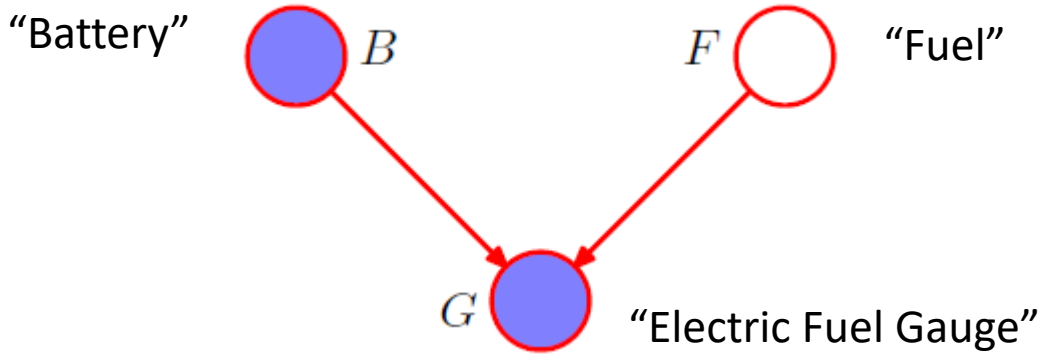
- Posterior probability that fuel tank is empty given observations of both fuel gauge and battery state is:

$$p(F=0|G=0,B=0) = \frac{p(G=0|B=0,F=0)p(F=0)}{\sum_{F\in\{0,1\}} p(G=0|B=0,F)p(F)} \simeq 0.111$$

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Three Canonical 3-Node Graphs

Numerical Example:



0.257 0.111

$$p(F = 0|G = 0) > p(F = 0|G = 0, B = 0)$$

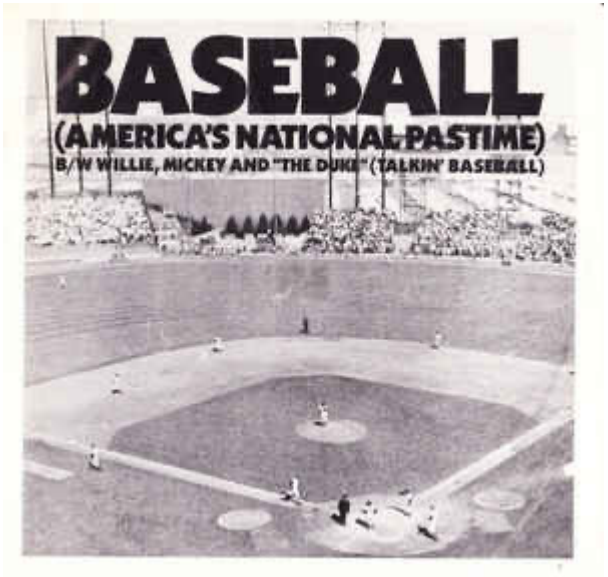
- Finding out that battery is flat *explains away* observation that the fuel gauge reads empty!

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

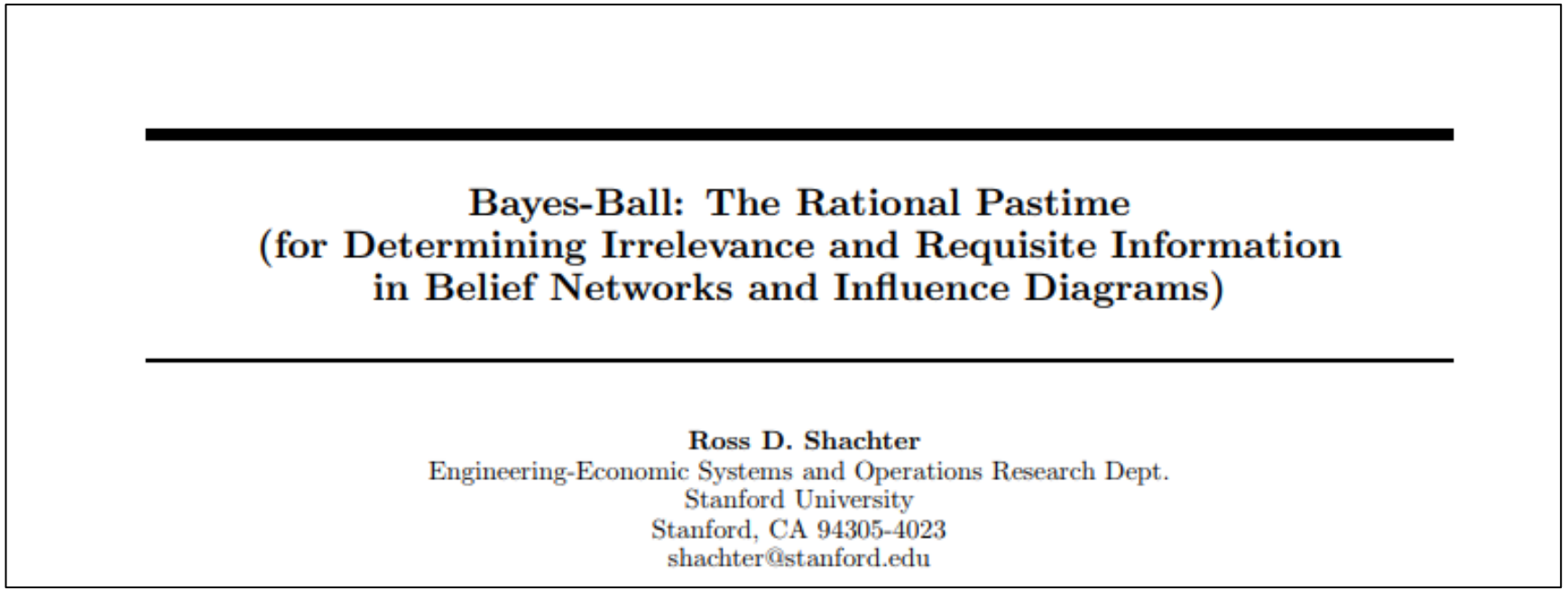
Bayes-Ball

Automating d-separation testing

Baseball: "The National Pastime"



Bayes-ball: "The Rational Pastime"



Bayes-Ball: The Rational Pastime
(for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams)

Ross D. Shachter
Engineering-Economic Systems and Operations Research Dept.
Stanford University
Stanford, CA 94305-4023
shachter@stanford.edu

Ross D. Shachter. 1998. Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98)

Bayes Ball Algorithm: Intuition

- This is a "reachability" algorithm:

  1. Shade the nodes in set C. ("given" nodes)
  2. Place a ball at each of the nodes in set A.
  3. Let the balls "bounce around" the graph according to the d-separation rules:

  IF none of the balls reach B THEN

  $$A \perp B \mid C,$$

  ELSE

  $$A \not\perp B \mid C,$$

- Can be implemented as a breadth-first search.

Single-Check Algorithm Sketch

- Given Graph G and sets of nodes X, Y and Z
- Output True if $X \perp Y \mid Z$, False otherwise.
- 2 Phases:
  - **Phase 1:** "Find all the unblocked v-structures".
    - Traverse the graph from the leaves to the roots, marking all nodes that are in Z or have descendants in Z.
  - **Phase 2:** "Traverse all the trails starting from X"
    - Apply breadth first search, stopping a specific traversal when we hit a blocked node.
    - If Y is reached during this traversal, return False.
- Else, return True.

Previous (2018) Coding Exercise Problem 1 (10%)

Phase 1: Finding unblocked v-structures

```
// Store all ancestors of Z
// so we can check unblocked V structures easily
Given G, X (source), Y (target), Z (observations)
L = Z // Nodes to be visited
A = ∅ // Z and its ancestors

while L ≠ ∅
        C = L.pop() //get some node from L
        if C ∉ A then
                L = L.push(C's parents)
        A = A.push(C)
```

Phase 2 is for you to figure out!