Sistemas Inteligentes

Probabilistic Graphical Models (Stanford Notes)

José Eduardo Ochoa Luna
Dr. Ciencias - Universidade de São Paulo

Maestría C.C. Universidad Católica San Pablo
Sistemas Inteligentes

22 de noviembre 2018

Introduction

- Probabilistic graphical modeling is a branch of machine learning that studies how to use probability distributions to describe the world and to make useful predictions about it

Introduction

- Probabilistic graphical modeling is a branch of machine learning that studies how to use probability distributions to describe the world and to make useful predictions about it
- It is scientific field that bridges two very different branches of mathematics: probability and graph theory

Introduction

- Probabilistic graphical modeling is a branch of machine learning that studies how to use probability distributions to describe the world and to make useful predictions about it
- It is scientific field that bridges two very different branches of mathematics: probability and graph theory
- Probabilistic modeling also has intriguing connections to philosophy, particularly the question of causality

Introduction II

- Probabilistic modeling is widely used throughout machine learning and in many real-world applications

Introduction II

- Probabilistic modeling is widely used throughout machine learning and in many real-world applications
- This combination of elegant theory and powerful applications makes graphical models one of the most fascinating topics in modern AI and Comp. Science

Introduction II

- Probabilistic modeling is widely used throughout machine learning and in many real-world applications
- This combination of elegant theory and powerful applications makes graphical models one of the most fascinating topics in modern AI and Comp. Science
- The 2011 Turing award (Nobel prize of computer science) was awarded to Judea Pearl for founding the field of probabilistic graphical modeling

Probabilistic modeling

- Often, modeling the real word it involves a significant amount of uncertainty (e.g. the price of a house has a certain chance of going up if a new subway station opens within a certain distance)

Probabilistic modeling

- Often, modeling the real word it involves a significant amount of uncertainty (e.g. the price of a house has a certain chance of going up if a new subway station opens within a certain distance)
- It is very natural to deal with this uncertainty by modeling the world in the form of a probability distribution

$$p(x, y)$$

Probabilistic modeling II

given such a model, we could ask questions such as

- What is the probability that house prices will rise over the next five years?

Probabilistic modeling II

given such a model, we could ask questions such as

- What is the probability that house prices will rise over the next five years?
- given that the house costs $ 100,000, what is the most probability that it has three bedrooms?

Probability Aspect

The probability aspect of modeling is very important, because:

- We cannot perfectly predict the future. We often don't have enough knowledge about the world, and often the world itself is stochastic

Probability Aspect

The probability aspect of modeling is very important, because:

- We cannot perfectly predict the future. We often don't have enough knowledge about the world, and often the world itself is stochastic
- We need to assess the confidence of our predictions; often predicting a single value is not enough, we need the system to output its beliefs about what's going on in the world

Probability Aspect

The probability aspect of modeling is very important, because:

- We cannot perfectly predict the future. We often don't have enough knowledge about the world, and often the world itself is stochastic
- We need to assess the confidence of our predictions; often predicting a single value is not enough, we need the system to output its beliefs about what's going on in the world
- Thus, we will study principled ways of reasoning about uncertainty and use ideas from both probability and graph theory to derive efficient ML algorithms for this task.

Difficulties of probability modeling

Spam Classification

- Suppose we have a model $p(y, x_1, \ldots, x_n)$ of word occurrences in spam and non-spam mail

Difficulties of probability modeling

Spam Classification

- Suppose we have a model $p(y, x_1, \ldots, x_n)$ of word occurrences in spam and non-spam mail
- Each binary variable x_i encodes whether the i-th English word is present in the email

Difficulties of probability modeling

Spam Classification

- Suppose we have a model $p(y, x_1, \ldots, x_n)$ of word occurrences in spam and non-spam mail
- Each binary variable x_i encodes whether the i-th English word is present in the email
- The binary variable y indicates whether the email is spam

Difficulties of probability modeling

Spam Classification

- Suppose we have a model $p(y, x_1, \ldots, x_n)$ of word occurrences in spam and non-spam mail
- Each binary variable x_i encodes whether the i-th English word is present in the email
- The binary variable y indicates whether the email is spam
- In order to classify a new email: $P(y = 1|x_1, \ldots, x_n)$

Difficulties of probability modeling II

What is the size of the function p_θ that we just defined?

- Our model defines a probability in $[0, 1]$ for each combination of inputs $y, x_1, \ldots, x_n$

Difficulties of probability modeling II

What is the size of the function p_θ that we just defined?

- Our model defines a probability in $[0, 1]$ for each combination of inputs $y, x_1, \ldots, x_n$
- Specifying all these probabilities will require us to write down a staggering 2^{n+1} different values, one for each assignment to our $n + 1$ binary variables

Difficulties of probability modeling II

What is the size of the function p_θ that we just defined?

- Our model defines a probability in $[0, 1]$ for each combination of inputs $y, x_1, \ldots, x_n$
- Specifying all these probabilities will require us to write down a staggering 2^{n+1} different values, one for each assignment to our $n + 1$ binary variables
- Since n is the size of the English vocabulary, this is clearly impractical from both a computational and from statistical point of view

Difficulties of probability modeling II

What is the size of the function p_θ that we just defined?

- Our model defines a probability in $[0, 1]$ for each combination of inputs $y, x_1, \ldots, x_n$
- Specifying all these probabilities will require us to write down a staggering 2^{n+1} different values, one for each assignment to our $n + 1$ binary variables
- Since n is the size of the English vocabulary, this is clearly impractical from both a computational and from statistical point of view
- Probabilities are inherently exponentially-sized objects: the only way in which we can manipulate them is by making simplifying assumptions about their structure

Simplifying Assumption

- The main simplifying assumption is that of conditional independence among the variables

Simplifying Assumption

- The main simplifying assumption is that of conditional independence among the variables
- For instance, suppose that English words are all conditionally independent given Y

Simplifying Assumption

- The main simplifying assumption is that of conditional independence among the variables
- For instance, suppose that English words are all conditionally independent given Y
- I.e., the probabilities of seeing two words are independent given that a message is spam

Simplifying Assumption

- The main simplifying assumption is that of conditional independence among the variables
- For instance, suppose that English words are all conditionally independent given Y
- I.e., the probabilities of seeing two words are independent given that a message is spam
- oversimplification: "pills" and "buy" are clearly correlated

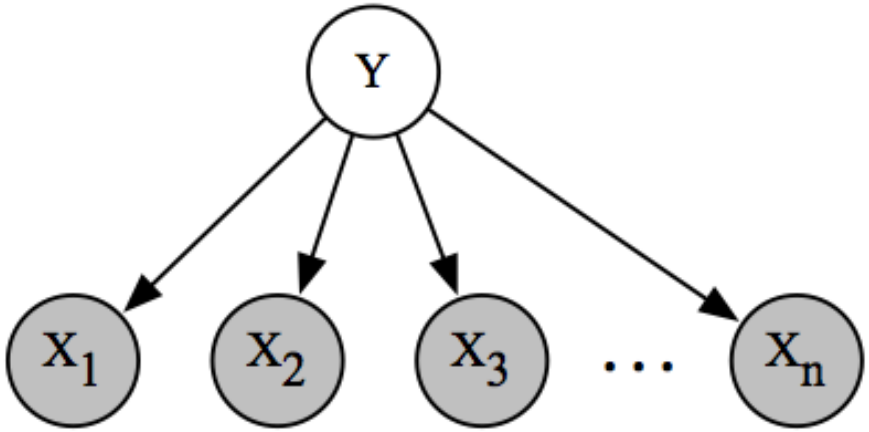
Naive Bayes assumption

Given the assumption of independence, we can write the model probability as a product of factors:

$$P(y, x_1, \ldots, x_n) = p(y) \prod_{i=1}^{n} p(x_i|y)$$

- Each factor $p(x_i|y)$ can be described by a small number of parameters (4)
- The entire distribution is parametrized by $O(n)$ parameters, which we can tractably estimate from data and make predictions

Describing Probabilities with Graphs



The independence assumption can be conveniently represented in the form of a graph
It is easy to understand:
an email was generated by first choosing at random whether the email is spam or not (indicated by y), and then by sampling words one at a time

Graphical Model Themes

- Representation (how to specify a model)

Graphical Model Themes

- Representation (how to specify a model)
- Inference (how to ask the model questions)

Graphical Model Themes

- Representation (how to specify a model)
- Inference (how to ask the model questions)
- Learning (how to fit a model to real-world data)

Representation

- How do we express a probability distribution that models some real-world phenomenon?

Representation

- How do we express a probability distribution that models some real-world phenomenon?
- A naive model for classifying spam messages with n possible words requires us in general to specify $O(2^n)$ parameters

Representation

- How do we express a probability distribution that models some real-world phenomenon?
- A naive model for classifying spam messages with n possible words requires us in general to specify $O(2^n)$ parameters
- To construct tractable models we will make heavy use of graph theory (e.g. connectivity, tree-width)

Inference

Given a probabilistic model, how do we obtain answers to relevant questions about the world?
Two types of questions:

- Marginal inference: what is the probability of a given variable in our model after we sum everything else out?

$$p(x_1) = \sum_{x_2} \sum_{x_3} \cdots \sum_{x_n} p(x_1, x_2, \ldots, x_n)$$

Inference

Given a probabilistic model, how do we obtain answers to relevant questions about the world?
Two types of questions:

- Marginal inference: what is the probability of a given variable in our model after we sum everything else out?

$$p(x_1) = \sum_{x_2} \sum_{x_3} \cdots \sum_{x_n} p(x_1, x_2, \ldots, x_n)$$

- Maximum a posteriori (MAP) inference asks for the most likely assignment of variables. For example, we may try to determine the most likely spam message, solving the problem

$$\max_{x_1,\ldots,x_n} p(x_1, \ldots, x_n, y = 1)$$

Inference

Given a probabilistic model, how do we obtain answers to relevant questions about the world?
Two types of questions:

- Marginal inference: what is the probability of a given variable in our model after we sum everything else out?

$$p(x_1) = \sum_{x_2} \sum_{x_3} \cdots \sum_{x_n} p(x_1, x_2, \ldots, x_n)$$

- Maximum a posteriori (MAP) inference asks for the most likely assignment of variables. For example, we may try to determine the most likely spam message, solving the problem

$$\max_{x_1,\ldots,x_n} p(x_1, \ldots, x_n, y = 1)$$

- Often queries will involve evidence (MAP example), in which case we will fix the assignment of a subset of variables

Inference II

- Inference is a very challenging task

Inference II

- Inference is a very challenging task
- It will be NP-hard to answer any of these questions

Inference II

- Inference is a very challenging task
- It will be NP-hard to answer any of these questions
- Whether inference is tractable will depend on the structure of the graph

Inference II

- Inference is a very challenging task
- It will be NP-hard to answer any of these questions
- Whether inference is tractable will depend on the structure of the graph
- In case intractability, we will still be able to obtain answers via approximate inference

Learning

- This task refers to fitting a model to a dataset, which could be for example a large number of labeled examples of spam

Learning

- This task refers to fitting a model to a dataset, which could be for example a large number of labeled examples of spam
- Learning and inference are linked, since inference will turn out a key subroutine within learning algorithms

Learning

- This task refers to fitting a model to a dataset, which could be for example a large number of labeled examples of spam
- Learning and inference are linked, since inference will turn out a key subroutine within learning algorithms
- Two main tasks: learning of parameters and structure

Elements of Probability

Elements of probability

Sample space Ω

The set of all the outcomes of a random experiment. Here, each outcome $\omega \in \Omega$ can be thought of as a complete description of the state of the real world at the end of the experiment.

Set of events (or event space) F

A set whose elements $A \in F$ (called events) are subsets of Ω (i.e., $A \subseteq \Omega$ is a collection of possible outcomes of an experiment)

Axioms of Probability

A function $P : F \to \mathbb{R}$ that satisfies the following properties:

$$P(A) \geq 0. \text{ for all } A \in F$$

If $A_1, A_2, \ldots$ are disjoint events (i.e, $A_i \cap A_j = \emptyset$ whenever $i \neq j$) then $P(\cup_i A_i) = \sum_i P(A_i)$
$P(\Omega) = 1$
These three properties are called the **Axioms of Probability**

Example

Consider the event of tossing a six-sided die. The sample space is $\Omega = \{1, 2, 3, 4, 5, 6\}$. The simplest event space is the trivial event space $F = \{\emptyset, \Omega\}$

For this event space, the unique probability measure satisfying the requirements is given by $P(\emptyset) = 0, P(\Omega) = 1$

Properties

If $A \subseteq B \Rightarrow P(A) \leq P(B)$
$P(A \cap B) \leq \min(P(A), P(B))$
Union Bound $P(A \cup B) \leq P(A) + P(B)$

$$P(\Omega - A) = 1 - P(A)$$

Law ot Total Probability if $A_1, \ldots, A_k$ are a set of disjoint events such that $\bigcup_{i=1}^{k} A_i = \Omega$ then $\sum_{i=1}^{k} P(A_k) = 1$

Conditional Probability

Let B be an event with non-zero probability

The conditional probability of any event A given B is defined as $P(A|B) = \frac{P(A \cap B)}{P(B)}$

In other words, $P(A|B)$ is the probability measure of the event A after observing the occurrence of event B

Chain Rule

Let $S_1, \ldots S_k$ be events, $P(S_i) > 0$. Then $P(S_1 \cap S_2 \cap \cdots \cap S_k) = P(S_1)P(S_2|S_1)P(S_3|S_2 \cap S_1) \cdots p(S_k|S_1 \cap S_2 \cap \cdots S_{k-1})$
Note that for $k = 2$ events, this is just the definition of conditional probability

$$P(S_1 \cap S_2) = P(S_1)P(S_2|S_1)$$

In general, it is derived by applying the definition of conditional independence multiple times

Independence

Two events are called independent if and only if $P(A \cap B) = P(A)P(B)$ (or equivalently $P(A|B) = P(A)$)
Independence is equivalent to saying that observing B does not have any effect on the probability of A

Real World Applications

Applications

- Images:, Generation, In-Painting, Denoising

Applications

- Images:, Generation, In-Painting, Denoising
- Language: Generation, Translation

Applications

- Images:, Generation, In-Painting, Denoising
- Language: Generation, Translation
- Audio: Super-Resolution, Speech Synthesis, Speech Recognition

Applications

- Images:, Generation, In-Painting, Denoising
- Language: Generation, Translation
- Audio: Super-Resolution, Speech Synthesis, Speech Recognition
- Science: Computational Biology, Ecology, Economics

Applications

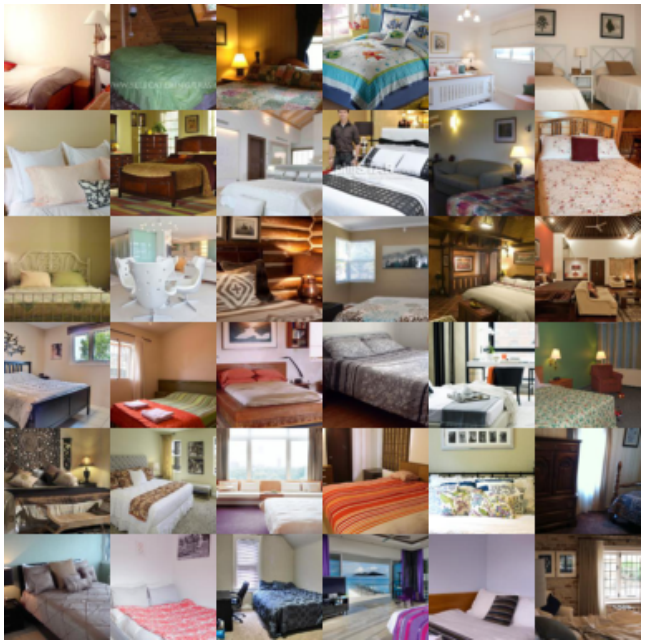
- Images:, Generation, In-Painting, Denoising
- Language: Generation, Translation
- Audio: Super-Resolution, Speech Synthesis, Speech Recognition
- Science: Computational Biology, Ecology, Economics
- Health Care and Medicine: Diagnosis

Image Models

Suppose we are able to learn a probability distribution $p(x)$ over images (a matrix of pixels) that assigns high probability to images that look realistic, and low probability to everything else
Given this model, there are a number of task that can be solved:

Sampling

Suppose we are somehow able to learn a probability distribution that assigns high probability to images that look like bedrooms (based on some training data):



Sampling

If we sample $x \approx p(x)$, we are generating new (realistic) images:
Generated Data:

Sampling

If we train the model on human faces, we can generate new ones

In Painting

Suppose we have our probability distribution $p(x)$, and a patch of an existing image (e.g. a piece of a photograph). If we sample from $p(Image|patch)$, we will generate different possible ways of completing the image:

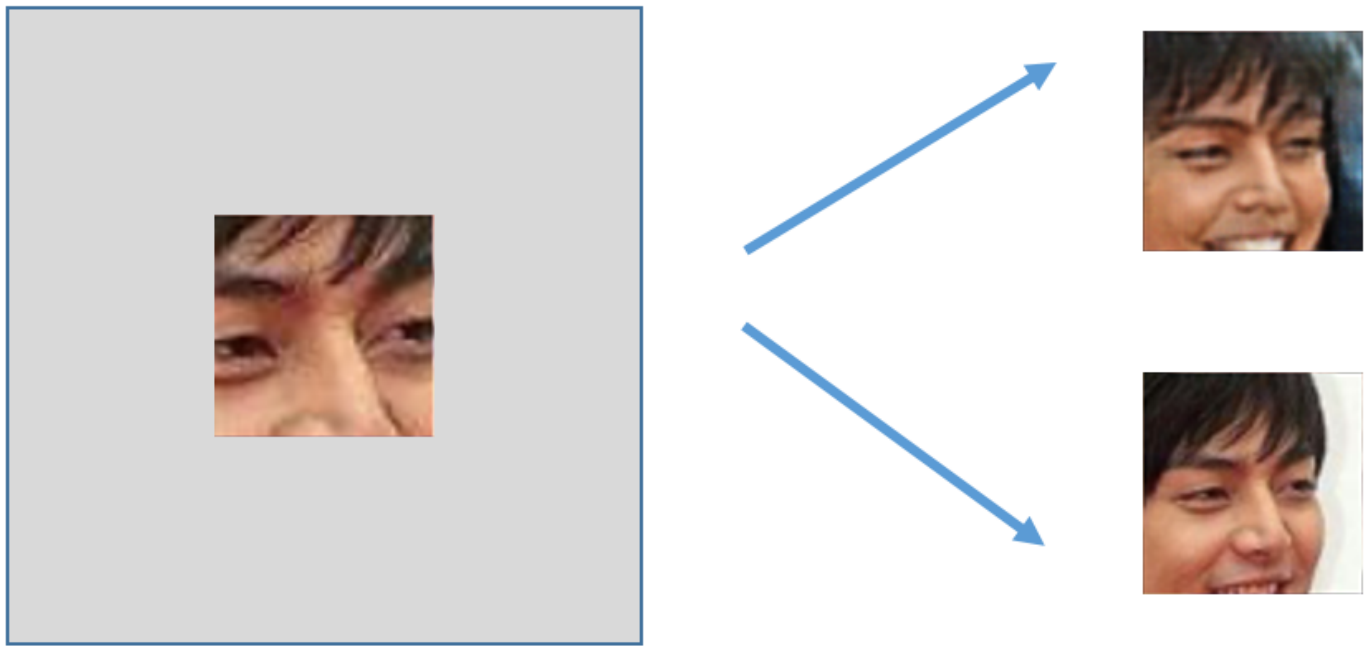


Image Denoising

Given an image corrupted by noise (e.g. an old photograph), we can attempt to restore it based on our probabilistic model of what images look like

Language Models

Suppose we can construct a probability distribution $p(x)$ over sequences of words or characters x that assigns high probability to (English) sentences.

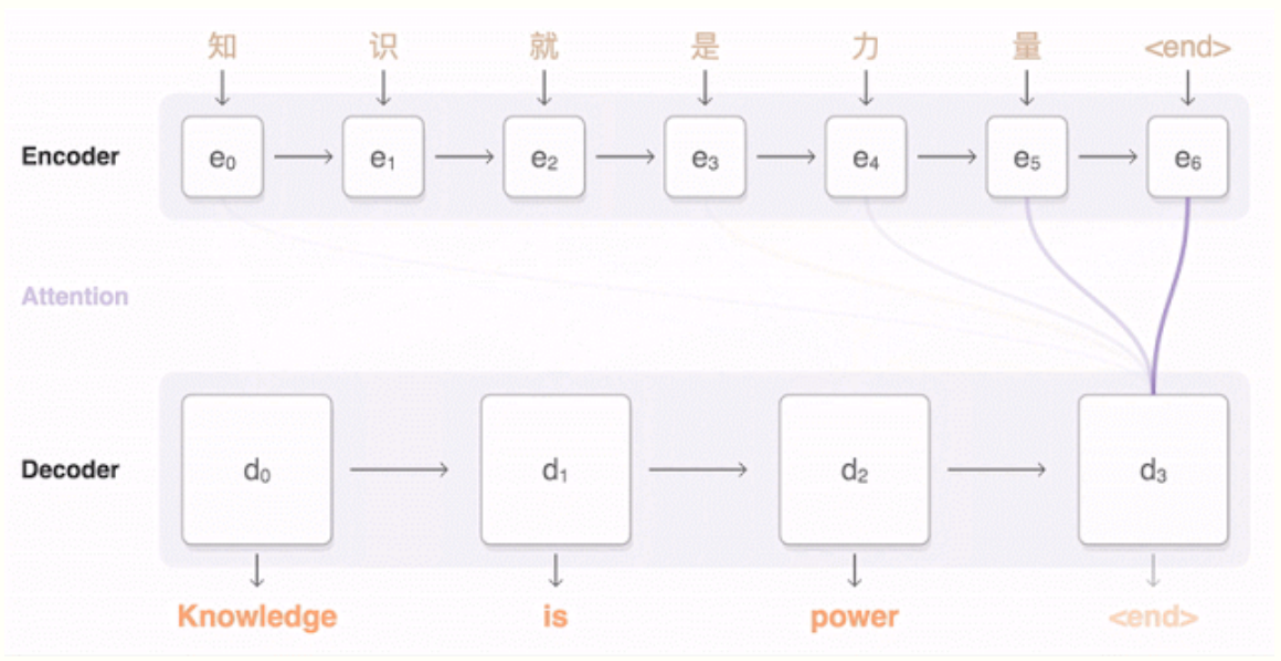
Generation

Suppose we use Wikipedia as our training data, and learn a model $p(x)$ based on it. We can then sample from the model, generating new Wikipedia-like articles like the following one:

Naturalism and decision for the majority of Arab countries? capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham?s sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25—21]] to note, the Kingdom of Costa Rica, ...

Translation

Suppose we have learned probabilistic models for both English and Chinese. We can use the model to generate an English sentence conditioned on the corresponding Chinese one (translation):

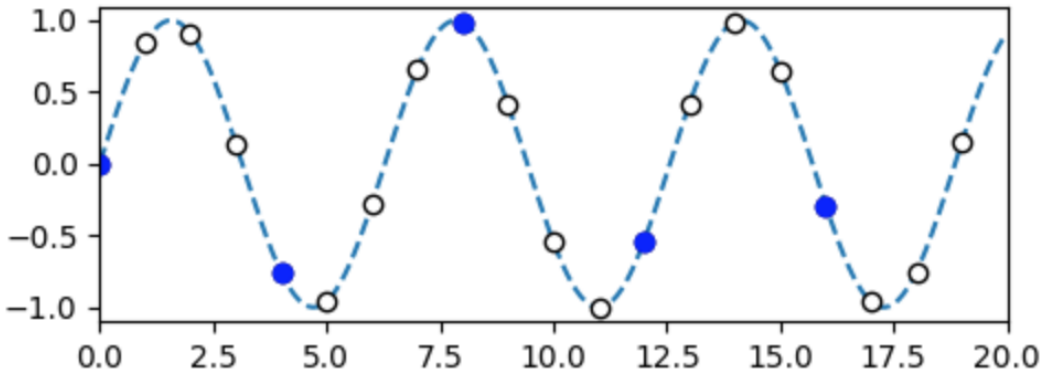


Audio Models

Suppose we can construct a probability distribution $p(x)$ over audio signals that assigns high probability to ones that sounds like human speech

Upsampling or super-resolution

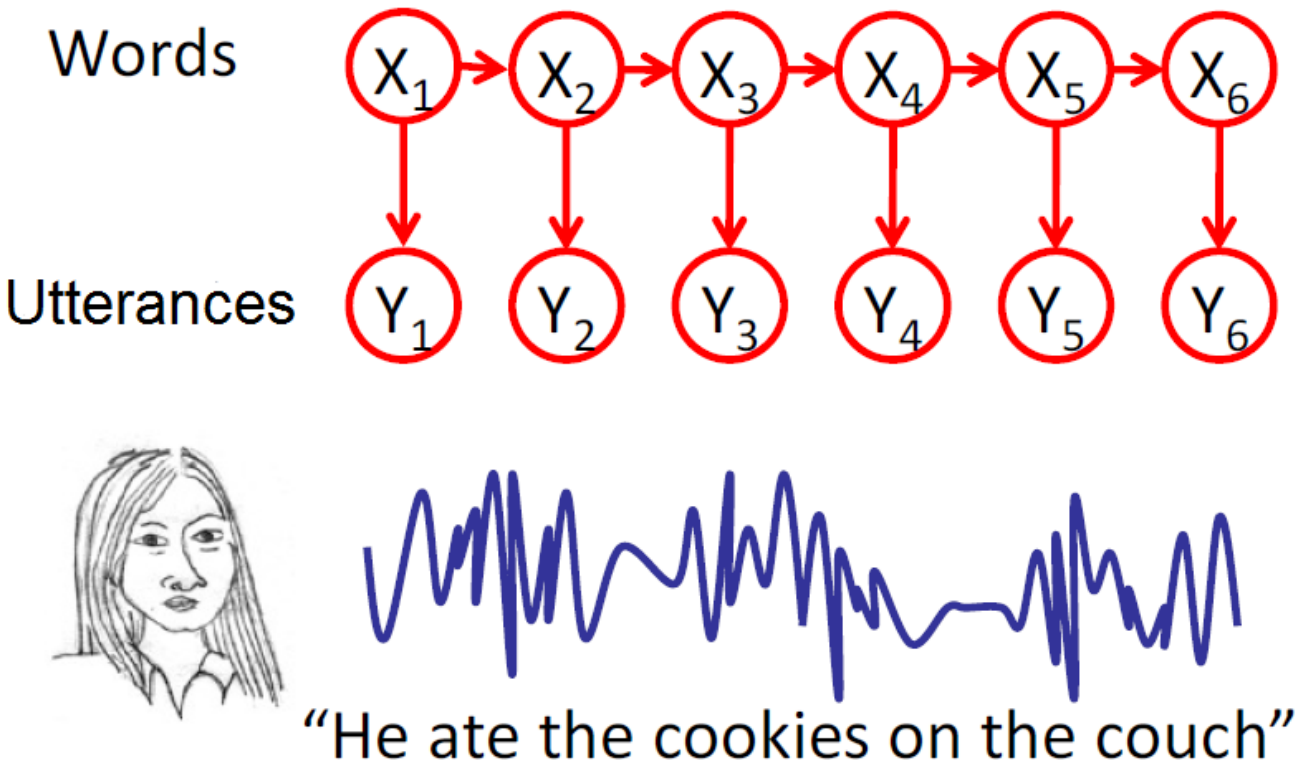
Given a low resolution version of an audio signal, we can attempt to increase its resolution



Given observed audio signals (blue) and some underlying model of the audio, we aim to reconstruct a higher-fidelity version of the original signal (dotted line) by predicting intermediate signals (white). $P(\mathbf{I}|\mathbf{O})$

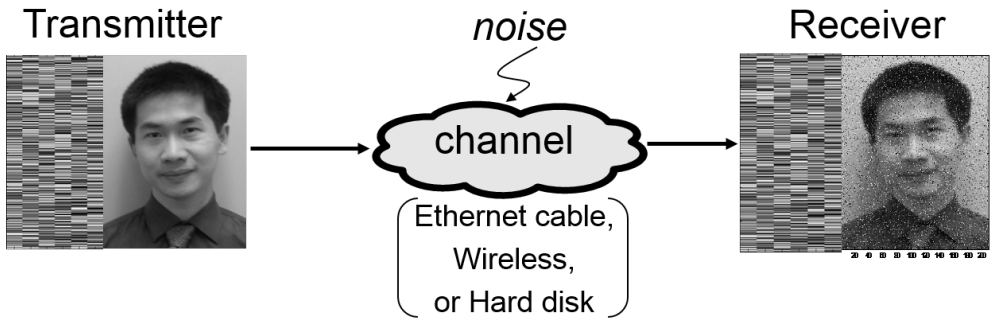
Speech recognition

Given a (joint) model of speech signals and language (text), we can attempt to infer spoken words from audio signals



Error Correcting codes

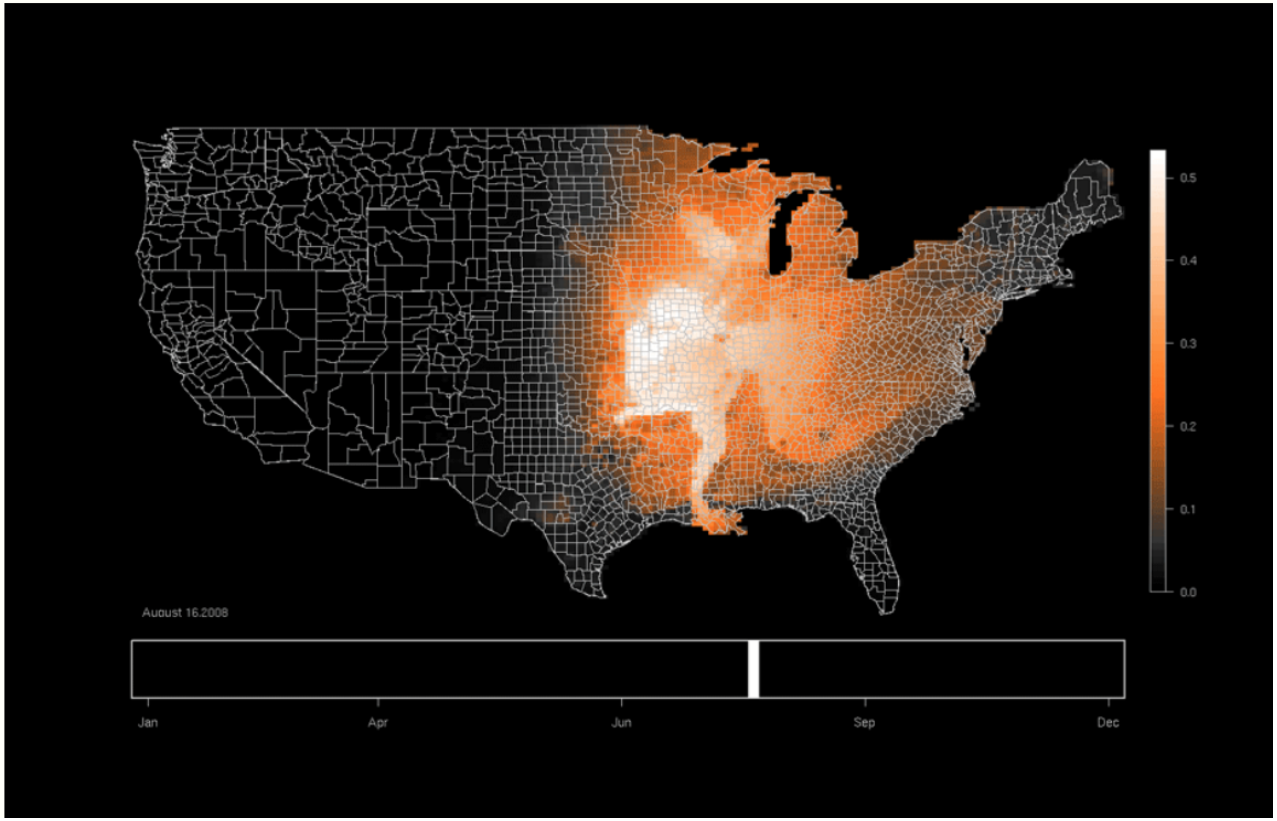
Probabilistic models are often used to model communication channels (e.g., Ethernet or Wifi), i.e., the fact that if you send a message over a channel, you might get something different on the other end due to noise. Error correcting codes and techniques based on graphical models are used to detect and correct communication errors.



Iterative message passing decoding

Ecology

Graphical models are used to study phenomena that evolve over space and time, capturing spatial and temporal dependencies. For example, they can be used to study bird migrations



Medical Diagnosis

PGM can assist doctors in diagnosing diseases and predicting adverse outcomes

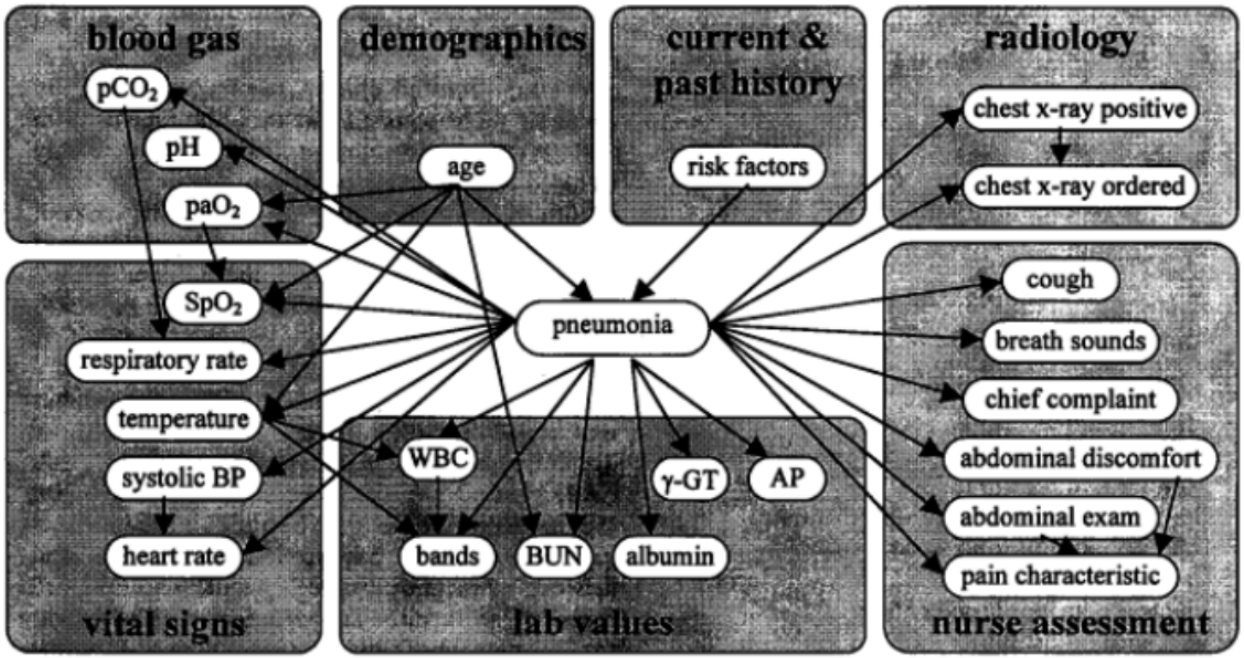


Figure 2: Structure of the Bayesian network. All variables are available in the HELP system during a patient's encounter in the emergency room with the exception of the chest x-ray information ("chest x-ray positive").

In 1998 the LDS Hospital in Salt Lake City, Utah developed a Bayesian network for diagnosing pneumonia. Their model was able to distinguish patients with pneumonia

Other Applications

- Fault diagnosis

Other Applications

- Fault diagnosis
- Natural language processing

Other Applications

- Fault diagnosis
- Natural language processing
- Traffic analysis

Other Applications

- Fault diagnosis
- Natural language processing
- Traffic analysis
- Social networks models

Other Applications

- Fault diagnosis
- Natural language processing
- Traffic analysis
- Social networks models
- Robot localization and mapping

Other Applications

- Fault diagnosis
- Natural language processing
- Traffic analysis
- Social networks models
- Robot localization and mapping
- Computer vision

Bayesian Networks

Probabilistic modeling with Bayesian Networks

- Directed graphical models (a.k.a Bayesian networks) are a family of probability distributions that admit a compact parametrization that can be naturally described using a directed graph

Probabilistic modeling with Bayesian Networks

- Directed graphical models (a.k.a Bayesian networks) are a family of probability distributions that admit a compact parametrization that can be naturally described using a directed graph
- By the chain rule, we can write any probability p as:

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2|x_1)\ldots p(x_n|x_{n-1}, \ldots, x_2, x_1)$$

Probabilistic modeling with Bayesian Networks

- Directed graphical models (a.k.a Bayesian networks) are a family of probability distributions that admit a compact parametrization that can be naturally described using a directed graph
- By the chain rule, we can write any probability p as:

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2|x_1)\ldots p(x_n|x_{n-1}, \ldots, x_2, x_1)$$

- A compact BN is a distribution in which each factor on the right hand side depends only on a small number of ancestor variables x_{A_i}:

$$p(x_i|x_{i-1}\ldots x_1) = p(x_i|x_{A_i})$$

Probabilistic modeling with Bayesian Networks

- For example, in a model with five variables, we may choose to approximate the factor $p(x_5|x_4, x_3, x_2, x_1)$ with $p(x_5|x_4, x_3)$, thus $x_{A_i} = \{x_4, x_3\}$

Probabilistic modeling with Bayesian Networks

- For example, in a model with five variables, we may choose to approximate the factor $p(x_5|x_4, x_3, x_2, x_1)$ with $p(x_5|x_4, x_3)$, thus $x_{A_i} = \{x_4, x_3\}$
- When the variables are discrete, we may think of the factors $p(x_i|x_{A_i})$ as probability tables (CPD), in which rows correspond to assignments to x_{A_i} and columns correspond to values of x_i

Probabilistic modeling with Bayesian Networks

- For example, in a model with five variables, we may choose to approximate the factor $p(x_5|x_4, x_3, x_2, x_1)$ with $p(x_5|x_4, x_3)$, thus $x_{A_i} = \{x_4, x_3\}$
- When the variables are discrete, we may think of the factors $p(x_i|x_{A_i})$ as probability tables (CPD), in which rows correspond to assignments to x_{A_i} and columns correspond to values of x_i
- the entries contain the actual probabilities $p(x_i|x_{A_i})$

Probabilistic modeling with Bayesian Networks

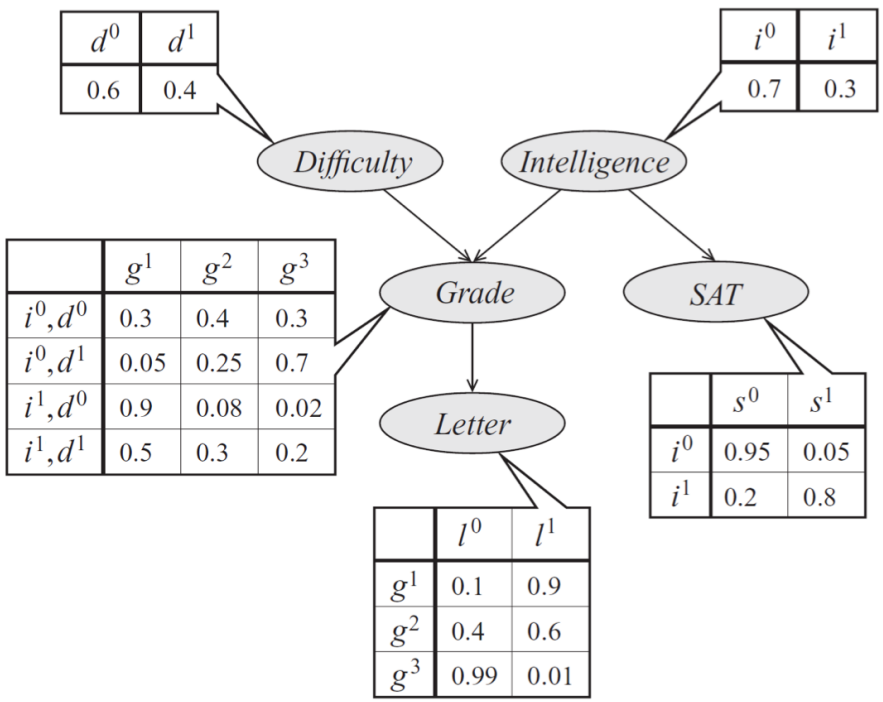
- For example, in a model with five variables, we may choose to approximate the factor $p(x_5|x_4, x_3, x_2, x_1)$ with $p(x_5|x_4, x_3)$, thus $x_{A_i} = \{x_4, x_3\}$
- When the variables are discrete, we may think of the factors $p(x_i|x_{A_i})$ as probability tables (CPD), in which rows correspond to assignments to x_{A_i} and columns correspond to values of x_i
- the entries contain the actual probabilities $p(x_i|x_{A_i})$
- If each variable takes d values and has at most k ancestors, then the entire table will contain at most $O(d^{k+1})$ entries

Probabilistic modeling with Bayesian Networks

- For example, in a model with five variables, we may choose to approximate the factor $p(x_5|x_4, x_3, x_2, x_1)$ with $p(x_5|x_4, x_3)$, thus $x_{A_i} = \{x_4, x_3\}$
- When the variables are discrete, we may think of the factors $p(x_i|x_{A_i})$ as probability tables (CPD), in which rows correspond to assignments to x_{A_i} and columns correspond to values of x_i
- the entries contain the actual probabilities $p(x_i|x_{A_i})$
- If each variable takes d values and has at most k ancestors, then the entire table will contain at most $O(d^{k+1})$ entries
- Since we have one table per variable, the entire probability distribution can be compactly described with only $O(nd^k)$ parameteres

Graphical Representation

Distributions of this form can be naturally expressed as directed acyclic graphs, in which vertices correspond to variables x_i and edges indicate dependency relationships. In particular we set the parents of each node to x_i to its ancestors x_{A_i}



Graphical Representation

The joint probability distribution over 5 variables naturally factorizes:

$$p(l, g, i, d, s) = p(l|g)p(g|i, d)p(i)p(d)p(s|i)$$

- The graphical representation is a DAG that visually specifies how random variables depend on each other. The graph indicates that the letter depends on the grade, which in turn depends on the student's intelligence and the difficulty of the exam

Graphical Representation

The joint probability distribution over 5 variables naturally factorizes:

$$p(l, g, i, d, s) = p(l|g)p(g|i, d)p(i)p(d)p(s|i)$$

- The graphical representation is a DAG that visually specifies how random variables depend on each other. The graph indicates that the letter depends on the grade, which in turn depends on the student's intelligence and the difficulty of the exam
- another interpretation: to determine the quality of the reference letter, we may first sample an intelligence level and an exam difficulty

Graphical Representation

The joint probability distribution over 5 variables naturally factorizes:

$$p(l, g, i, d, s) = p(l|g)p(g|i, d)p(i)p(d)p(s|i)$$

- The graphical representation is a DAG that visually specifies how random variables depend on each other. The graph indicates that the letter depends on the grade, which in turn depends on the student's intelligence and the difficulty of the exam
- another interpretation: to determine the quality of the reference letter, we may first sample an intelligence level and an exam difficulty
- a student's grade is sampled given these parameters

Graphical Representation

The joint probability distribution over 5 variables naturally factorizes:

$$p(l, g, i, d, s) = p(l|g)p(g|i, d)p(i)p(d)p(s|i)$$

- The graphical representation is a DAG that visually specifies how random variables depend on each other. The graph indicates that the letter depends on the grade, which in turn depends on the student's intelligence and the difficulty of the exam
- another interpretation: to determine the quality of the reference letter, we may first sample an intelligence level and an exam difficulty
- a student's grade is sampled given these parameters
- finally, the recommendation letter is generated based on that grade

Formal Definition

A BN is a directed graph $G = (V, E)$, together with
A random variable x_i for each node $i \in V$
One conditional probability distribution (CPD) $p(x_i|x_{A_i})$ per node, specifying the probability of x_i conditioned on its parents' values
A BN defines a probability distribution p. Conversely, we say that a probability p *factorizes* over a DAG G if it can be decomposed into a product of factors, as specified by G.

Formal Definition

A probability represented by a BN will be valid:

- It will be non-negative and

Formal Definition

A probability represented by a BN will be valid:

- It will be non-negative and
- one can show using induction argument that the sum over all variable assignments will be one

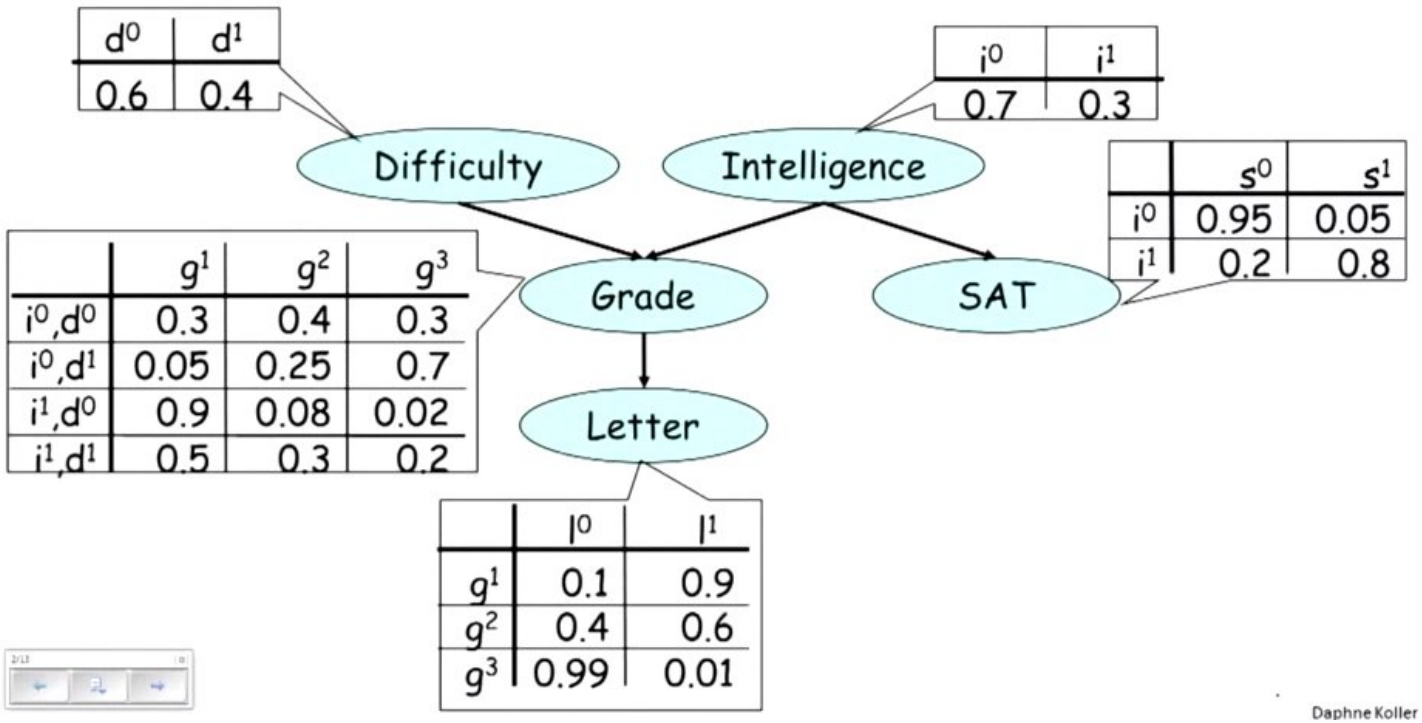
Formal Definition

A probability represented by a BN will be valid:

- It will be non-negative and
- one can show using induction argument that the sum over all variable assignments will be one
- Conversely, we can also show by counter-example that when G contains cycles, its associated probability may not sum to one

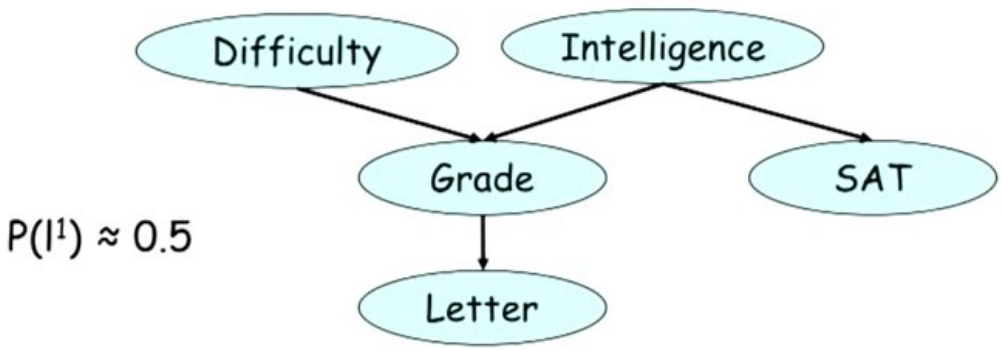
Reasoning Patterns

The Student Network

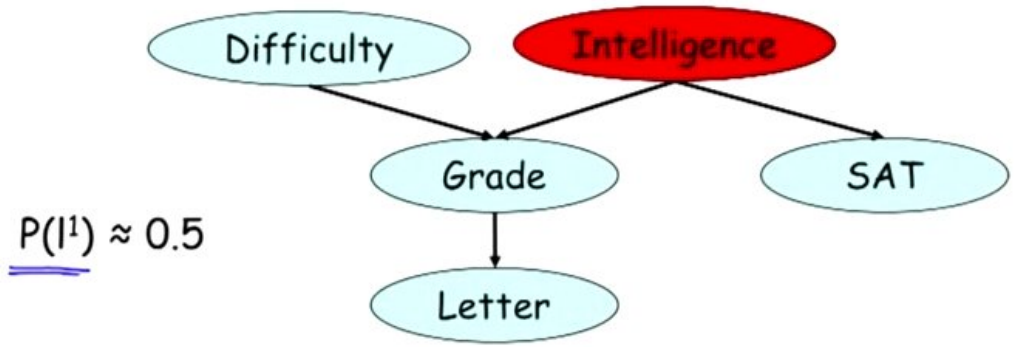


Daphne Koller

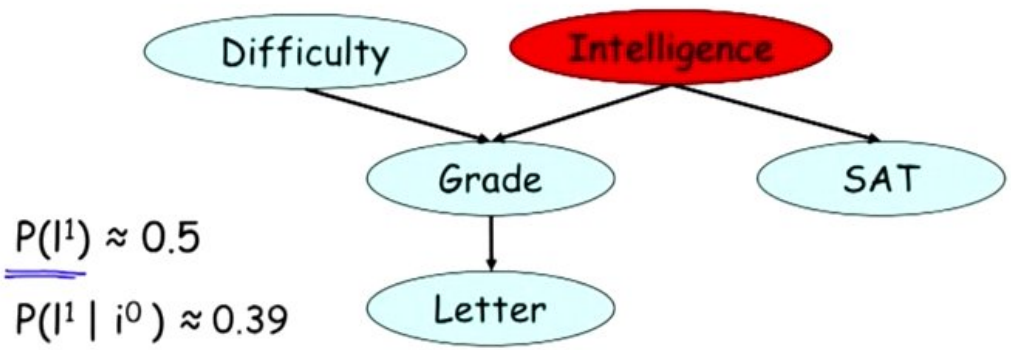
Causal Reasoning



Causal Reasoning



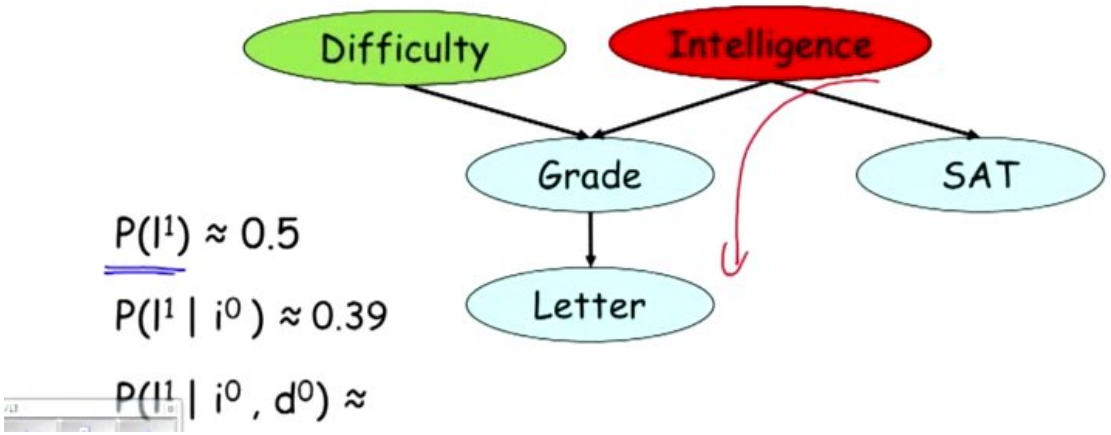
Causal Reasoning



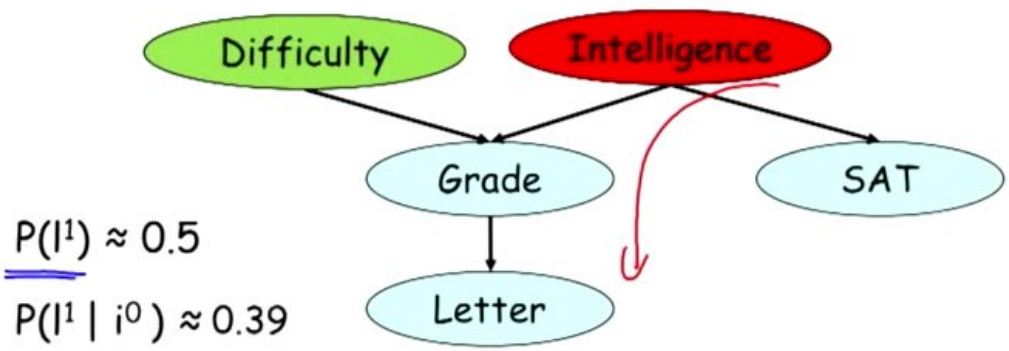
$P(l^1) \approx 0.5$

$P(l^1 \mid i^0) \approx 0.39$

Causal Reasoning



Causal Reasoning



$P(l^1) \approx 0.5$

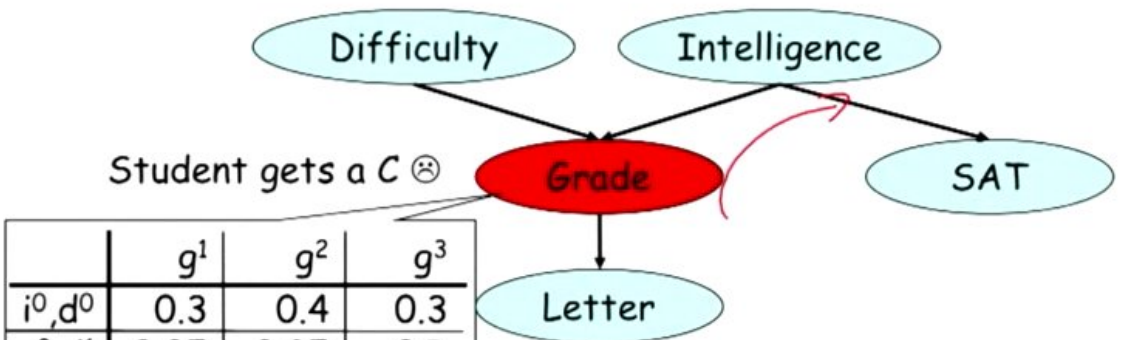
$P(l^1 \mid i^0) \approx 0.39$

$P(l^1 \mid i^0, d^0) \approx 0.51$

Evidential Reasoning

$P(d^1) = 0.4$ $P(i^1) = 0.3$

$P(d^1 \mid g^3) \approx$ $P(i^1 \mid g^3) \approx$

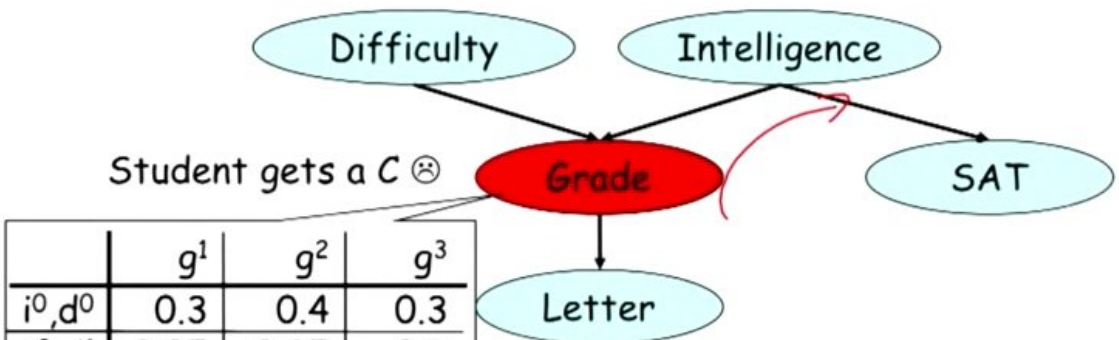


	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

Evidential Reasoning

→ $P(d^1) = 0.4$
$P(d^1 \mid g^3) \approx 0.63$

→ $P(i^1) = 0.3$
$P(i^1 \mid g^3) \approx 0.08$

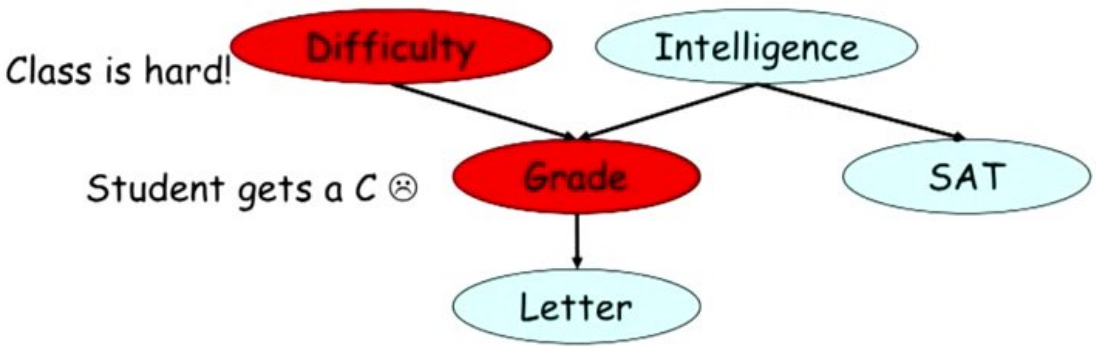


	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

Intercausal Reasoning

$P(d^1) = 0.4$ $P(i^1) = 0.3$

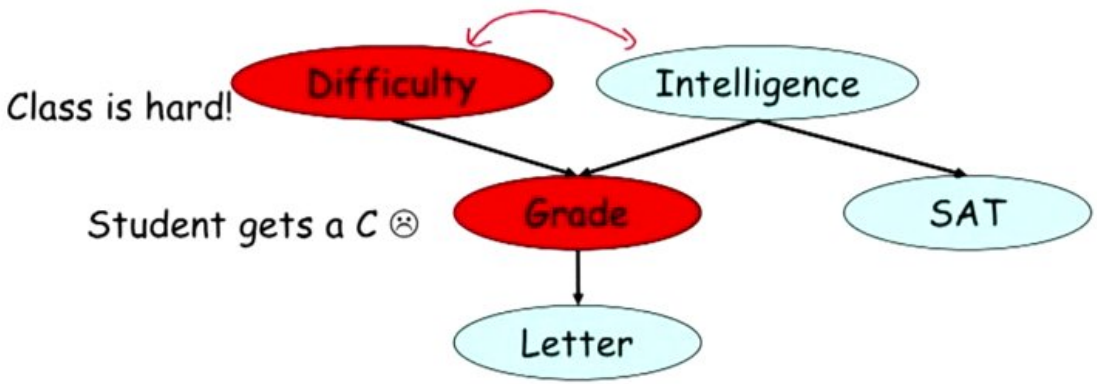
$P(d^1 \mid g^3) \approx 0.63$ $P(i^1 \mid g^3) \approx 0.08$



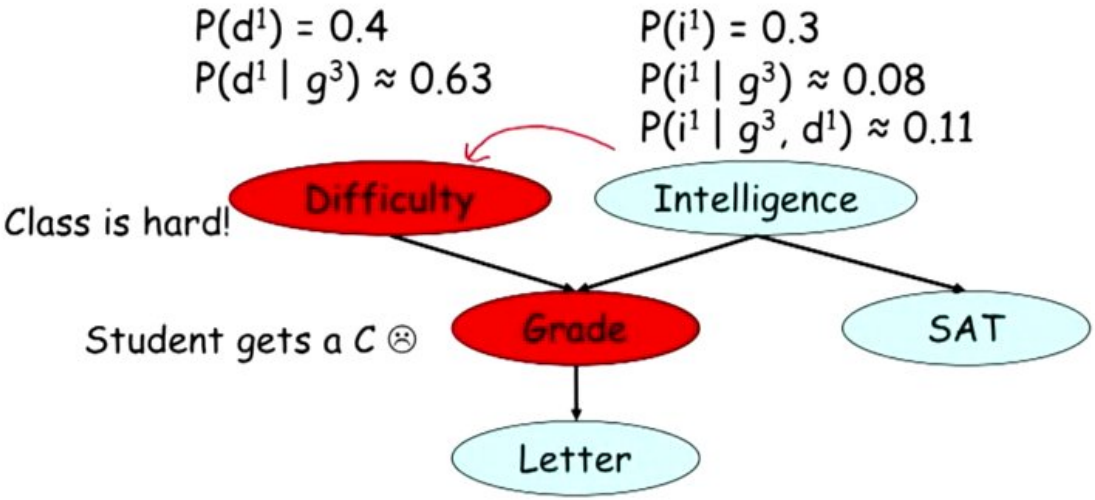
Intercausal Reasoning

$P(d^1) = 0.4$ $P(i^1) = 0.3$

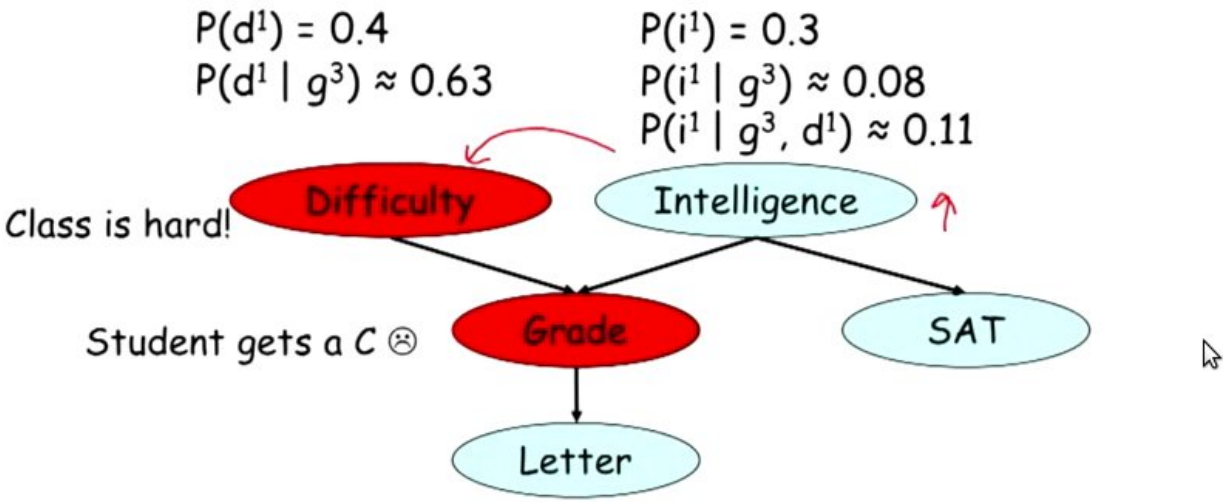
$P(d^1 \mid g^3) \approx 0.63$ $P(i^1 \mid g^3) \approx 0.08$



Intercausal Reasoning

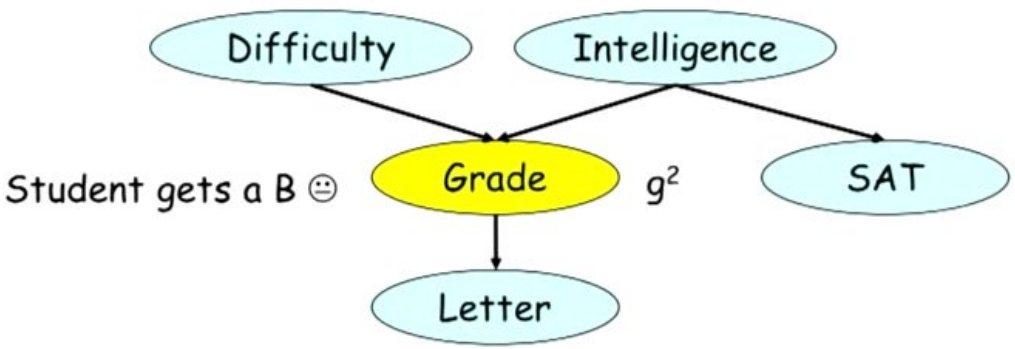


Intercausal Reasoning

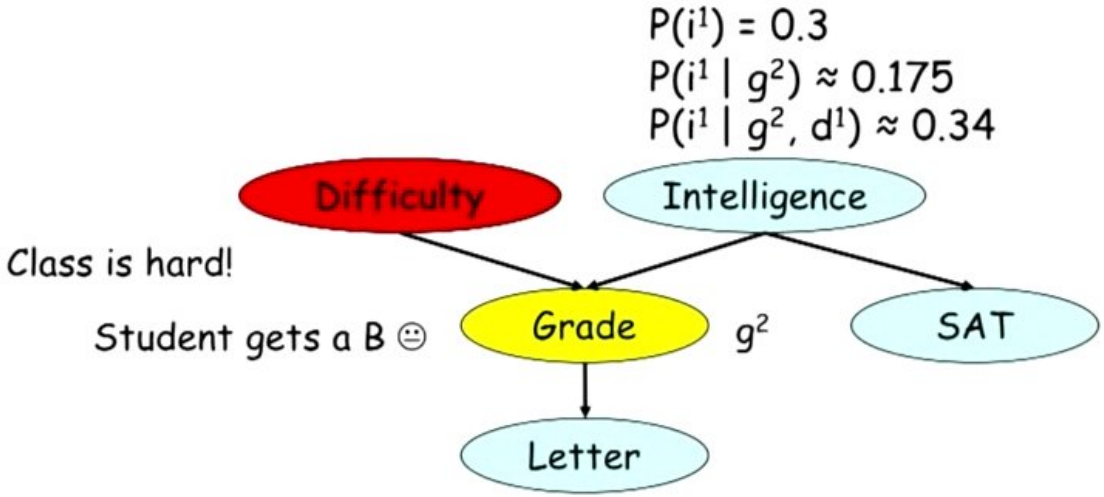


Intercausal Reasoning II

$P(i^1) = 0.3$
$P(i^1 \mid g^2) \approx 0.175$

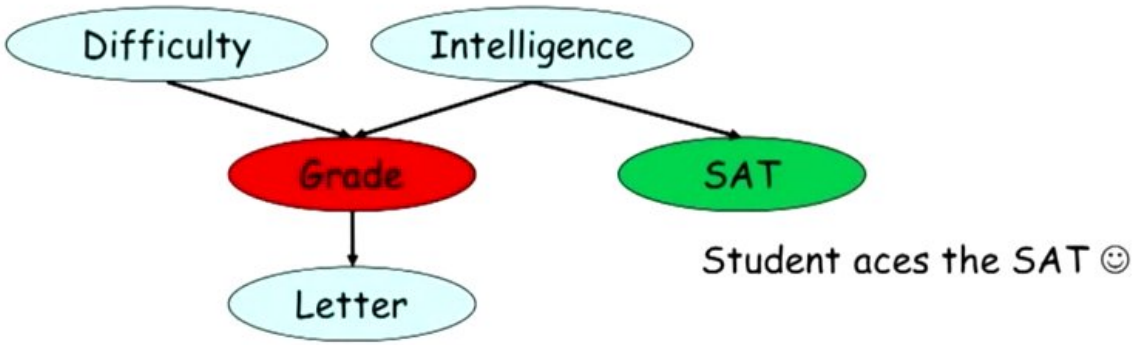


Intercausal Reasoning II



Student Aces the SAT

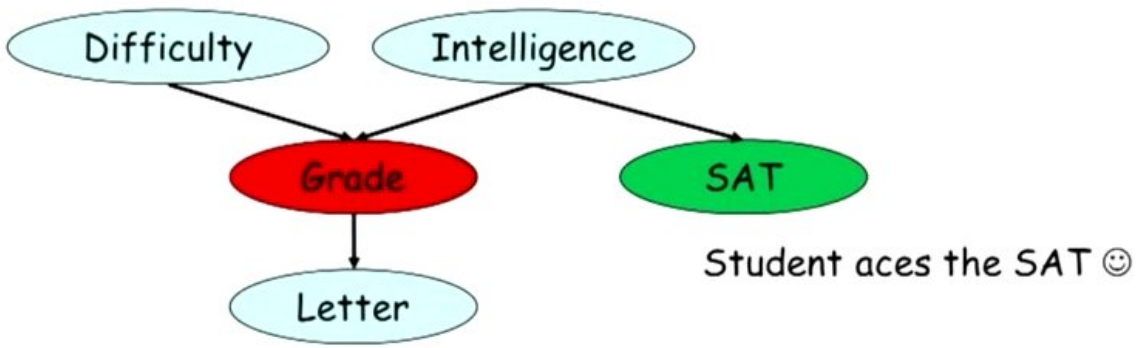
- What happens to the posterior probability that the class is hard?



Student Aces the SAT

$P(d^1) = 0.4$ $P(i^1) = 0.3$

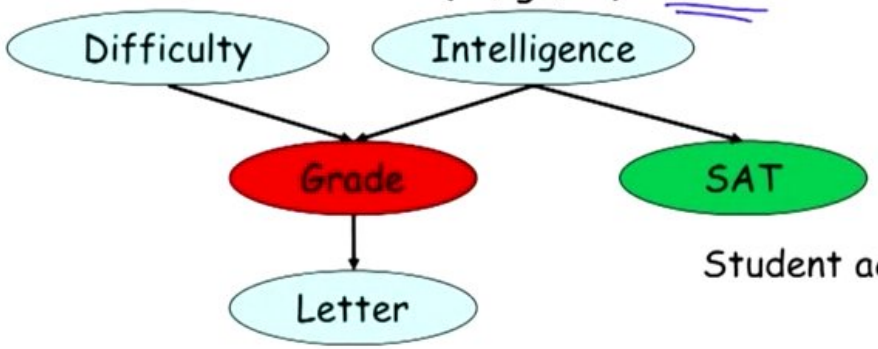
$P(d^1 \mid g^3) \approx 0.63$ $P(i^1 \mid g^3) \approx 0.08$



Student Aces the SAT

$P(d^1) = 0.4$
$P(d^1 \mid g^3) \approx 0.63$

$P(i^1) = 0.3$
$P(i^1 \mid g^3) \approx 0.08$
$P(i^1 \mid g^3, s^1) \approx 0.58$



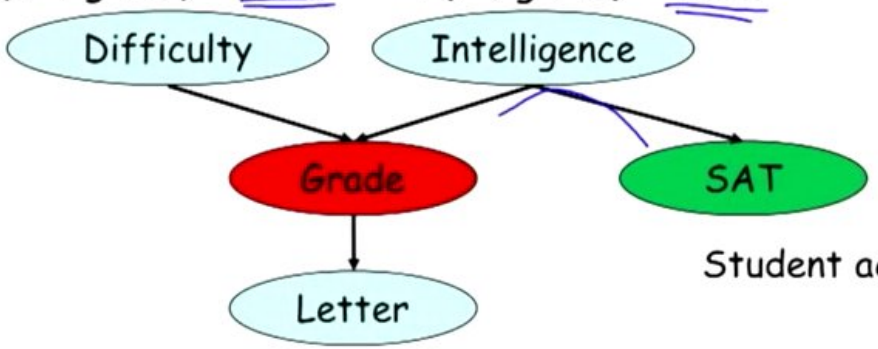
Student aces the SAT ☺

Student gets a C ☹

Student Aces the SAT

$P(d^1) = 0.4$
$P(d^1 \mid g^3) \approx 0.63$
$P(d^1 \mid g^3, s^1) \approx 0.76$

$P(i^1) = 0.3$
$P(i^1 \mid g^3) \approx 0.08$
$P(i^1 \mid g^3, s^1) \approx 0.58$



Student aces the SAT ☺

Student gets a C ☹

The Dependencies of a Bayes Net

- BNs represent probabilities that can be formed via product of smaller, local conditional probability distributions

The Dependencies of a Bayes Net

- BNs represent probabilities that can be formed via product of smaller, local conditional probability distributions
- By expressing a probability in this form, we are introducing into our model assumptions that certain variables are independent

The Dependencies of a Bayes Net

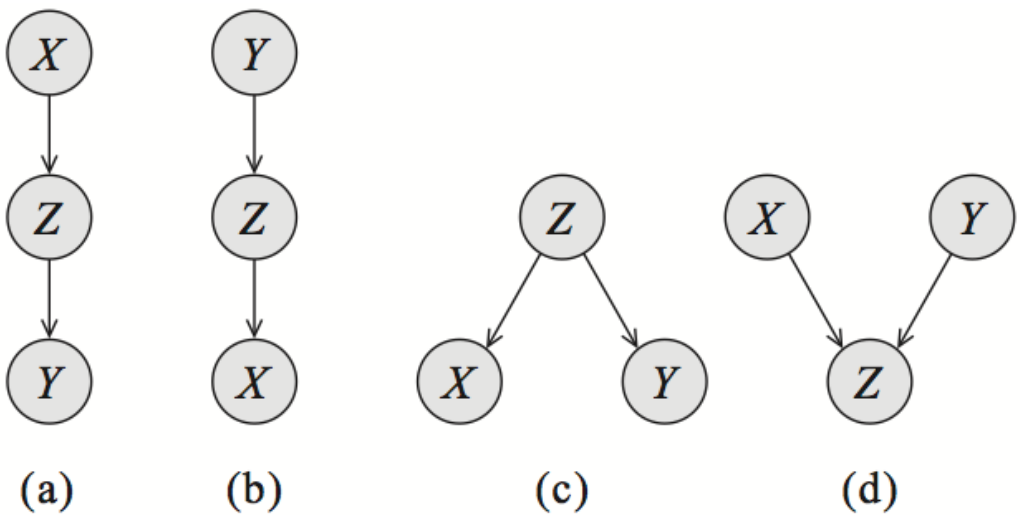
- BNs represent probabilities that can be formed via product of smaller, local conditional probability distributions
- By expressing a probability in this form, we are introducing into our model assumptions that certain variables are independent
- Which independence assumptions are we exactly making by using a mode BN with a given structure described by G?

The Dependencies of a Bayes Net

- BNs represent probabilities that can be formed via product of smaller, local conditional probability distributions
- By expressing a probability in this form, we are introducing into our model assumptions that certain variables are independent
- Which independence assumptions are we exactly making by using a mode BN with a given structure described by G?
- $I(p)$ denotes the set of all conditional independencies that hold for a joint distribution p

Independencies described by directed graphs

It turns out that a BN p describes many independencies in $I(p)$ which can be recovered from the graph by looking three types of structures.
For example, a BN with three nodes



dependencies: cascade (a,b), common parent (c), and v-structure (d)

Common parent

- If G is of the form $A \leftarrow B \rightarrow C$, and B is observed, then $A \perp C|B$

Common parent

- If G is of the form $A \leftarrow B \rightarrow C$, and B is observed, then $A \perp C|B$
- If B is unobserved, then $A \not\perp C$

Common parent

- If G is of the form $A \leftarrow B \rightarrow C$, and B is observed, then $A \perp C|B$
- If B is unobserved, then $A \not\perp C$
- Intuitively, this stems from the fact that B contains all the information that determines the outcomes of A and C; once it is observed, there is nothing else that affects these variables' outcomes

Cascade

- If G equals $A \rightarrow B \rightarrow C$, and B is observed, then, $A \perp C|B$

Cascade

- If G equals $A \rightarrow B \rightarrow C$, and B is observed, then, $A \perp C|B$
- If B is unobserved, then $A \not\perp C$

Cascade

- If G equals $A \rightarrow B \rightarrow C$, and B is observed, then, $A \perp C|B$
- If B is unobserved, then $A \not\perp C$
- The intuition is that B holds all the information that determines the outcome of C; it does not matter what value A takes

V-structure

- It is also known as *explaining away*

V-structure

- It is also known as *explaining away*
- If G is $A \rightarrow C \leftarrow B$ then knowing C couples A and B.

V-structure

- It is also known as *explaining away*
- If G is $A \rightarrow C \leftarrow B$ then knowing C couples A and B.
- In other words, $A \perp B$ if C is unobserved, but $A \not\perp B | C$ if C is observed

V-structure

- Suppose that C is a Boolean variable that indicates whether our lawn is wet one morning

V-structure

- Suppose that C is a Boolean variable that indicates whether our lawn is wet one morning
- A and B are two explanations for it being wet: either it rained (indicated by A), or the sprinkler turned on (indicated by B).

V-structure

- Suppose that C is a Boolean variable that indicates whether our lawn is wet one morning
- A and B are two explanations for it being wet: either it rained (indicated by A), or the sprinkler turned on (indicated by B).
- If we know that the grass is wet (C is true) and the sprinkler didn?t go on (B is false), then the probability that A is true must be one, because that is the only other possible explanation.

V-structure

- Suppose that C is a Boolean variable that indicates whether our lawn is wet one morning
- A and B are two explanations for it being wet: either it rained (indicated by A), or the sprinkler turned on (indicated by B).
- If we know that the grass is wet (C is true) and the sprinkler didn?t go on (B is false), then the probability that A is true must be one, because that is the only other possible explanation.
- Hence, A and B are not independent given C.

d-separation

- We say that Q, W are d-separated when variables O are observed if they are not connected by an *active path*

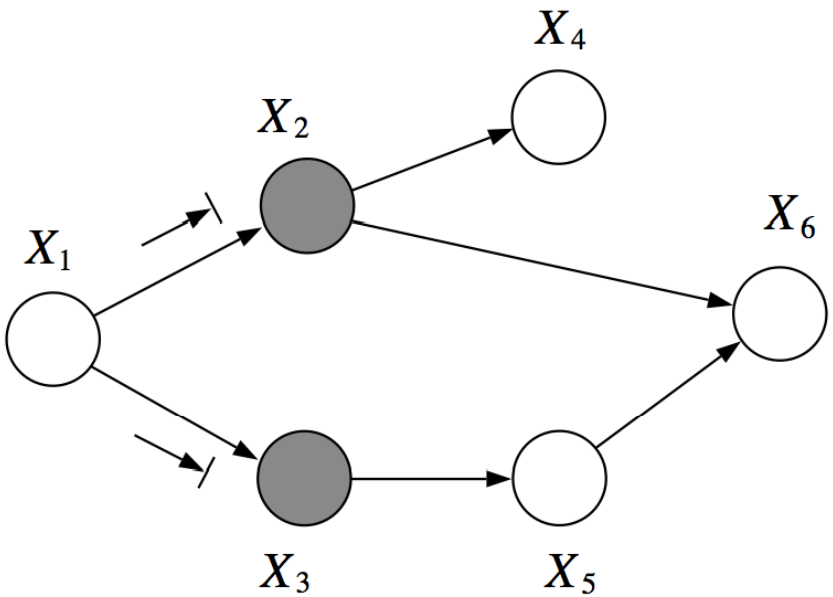
d-separation

- We say that Q, W are d-separated when variables O are observed if they are not connected by an *active path*
- An undirected path in the BN structure G is called *active* given observed variables O if for every consecutive triple of variables X, Y, Z on the path, one of the following holds:

d-separation

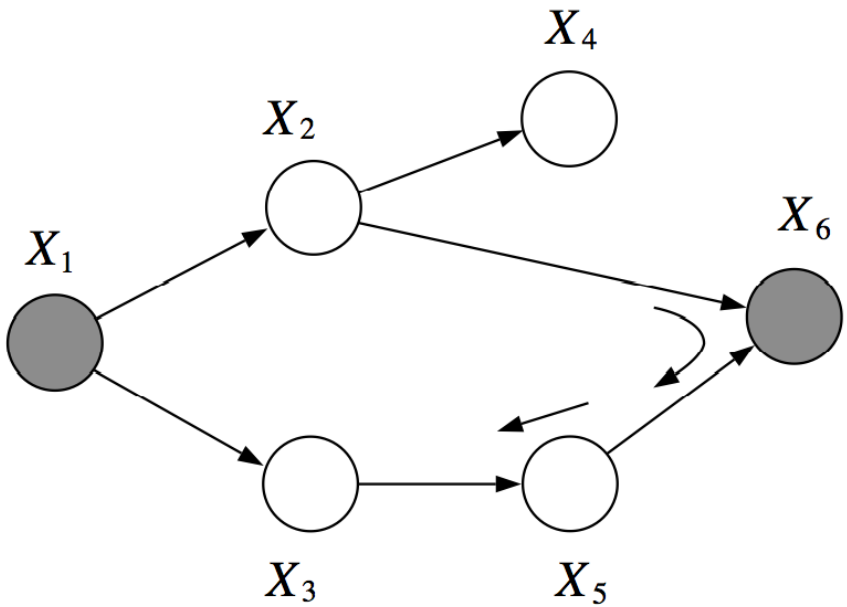
$X \leftarrow Y \leftarrow Z$, and Y is unobserved $Y \notin O$
$X \rightarrow Y \rightarrow Z$, and Y is unobserved $Y \notin O$
$X \leftarrow Y \rightarrow Z$, and Y is unobserved $Y \notin O$
$X \rightarrow Y \leftarrow Z$, and Y or any of its descendants are observed

d-separation Example



X_1 and X_6 are d-separated given X_2, X_3.

d-separation Example



X_2, X_3 are not d-separated given X_1, X_6, because we can find an active path (X_2, X_6, X_5, X_3)

d-separation III

Let $I(G) = \{(X \perp Y|Z) : X, Y$ are d-sep given $Z\}$ be a set of variables that are d-separated in G
Theor: If p factorizes over G, then $I(G) \subseteq I(p)$. We say that G is an I-map (independence map) for p.
In other words, all the independencies encoded in G are sound: variables that are d-separated in G are truly independent in p.
However, the converse is not true: a distribution may factorize over G, yet have independencies that are not captured in G

Bayes Nets I-equivalence

- Are independence maps unique when they exist?

Bayes Nets I-equivalence

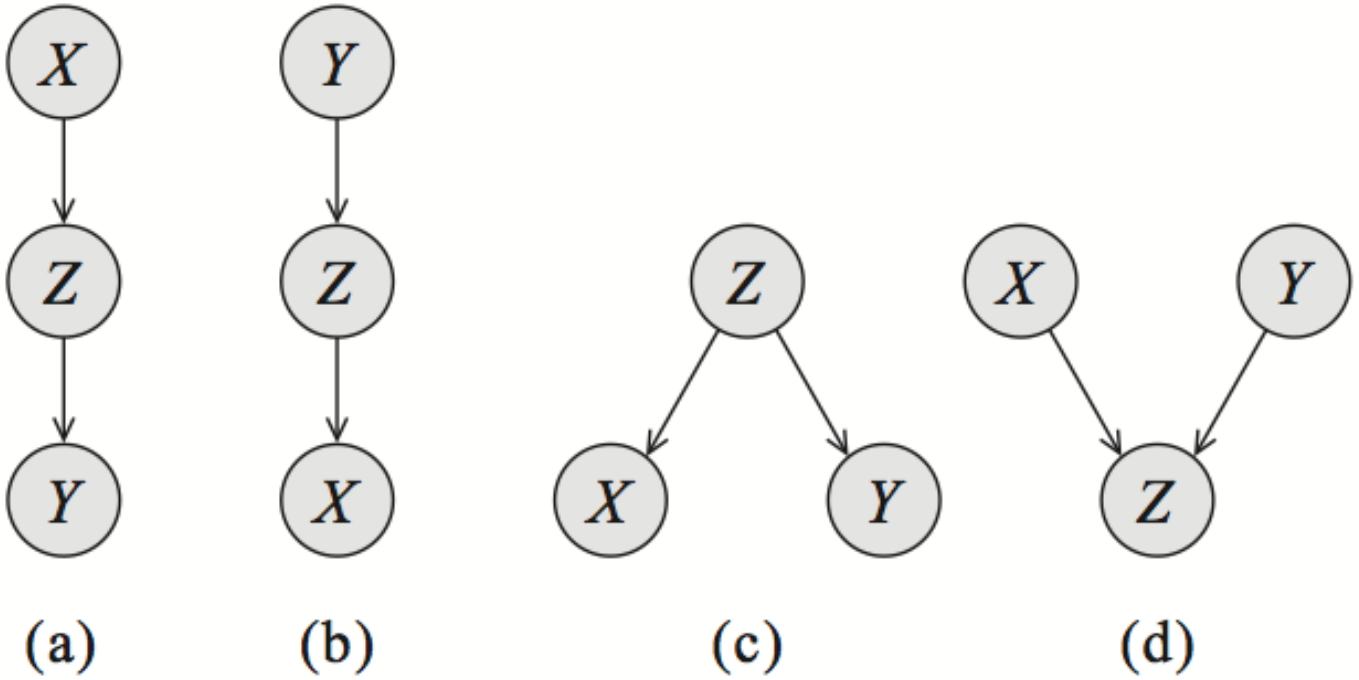
- Are independence maps unique when they exist?
- No, $X \rightarrow y$ and $X \leftarrow Y$ encode the same independencies, yet form differents graphs

Bayes Nets I-equivalence

- Are independence maps unique when they exist?
- No, $X \rightarrow y$ and $X \leftarrow Y$ encode the same independencies, yet form differents graphs
- We say that two BNs G_1, G_2 are I-equivalent if they encode the same dependencies $I(G_1) = I(G_2)$

Bayes Nets I-equivalence

We say that each of the graphs below have the same skeleton, meaning that if we drop the directionality of the arrows, we obtain the same undirected graph in each case.



Bayes Nets I-equivalence

- The cascade-type structures (a,b) are clearly symmetric and the directionality of arrows does not matter

Bayes Nets I-equivalence

- The cascade-type structures (a,b) are clearly symmetric and the directionality of arrows does not matter
- In fact, (a,b,c) encode exactly the same dependencies

Bayes Nets I-equivalence

- The cascade-type structures (a,b) are clearly symmetric and the directionality of arrows does not matter
- In fact, (a,b,c) encode exactly the same dependencies
- We can change the directions of the arrows as long as we don't turn them into a V-structure (d)

Bayes Nets I-equivalence

- The cascade-type structures (a,b) are clearly symmetric and the directionality of arrows does not matter
- In fact, (a,b,c) encode exactly the same dependencies
- We can change the directions of the arrows as long as we don't turn them into a V-structure (d)
- When we do have a V-structure, we cannot change any arrows: d) is the only that describes the dependency $X \not\perp Y|Z$

Bayes Nets I-equivalence

Theor: If G, G' have the same skeleton and the same v-structures, then $I(G) = I(G')$

Two graphs are I-equivalente if the d-separation between variables is the same

We can flip the directionality of any edge, unless it forms a v-structure and the d-connectivity of the graph will be unchanged

Tarea

Jupyter notebook Semana8