*** * * Artificial Intelligence and Machine Learning * * ***

Probabilistic Graphical Models (PGMs)

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Summer term 2025

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

Agenda for this Unit

Section:

Introduction

Refresher on important Concepts in Statistics
Important Rules for Probabilities
Introduction to graphical Models

Important Concepts

❶ What is a random variable $\mathcal{X}$?

A random number whose value is subject to variations due to chance.

❷ What is a distribution $p(\mathcal{X} = x)$?

Describes the probability density that the random variable $\mathcal{X}$ will be equal to a certain value x.

❸ What is a joint distribution?

Given M random variables, the **joint distribution** specifies the probability for all pairs of outcomes.

Important Concepts (Ctd.)

❹ What is a marginal distribution?

The **marginal distribution** of a subset of random variables describes the probability distribution of the variables in the subset.

❺ What is a conditional distribution?

A **conditional distribution** describes the probability of an outcome given the occurrence of a particular event.

Important Rules for Probabilities

Conditional probability:

$$p(\mathcal{X}|\mathcal{Y}) = \frac{p(\mathcal{X} \cap \mathcal{Y})}{p(\mathcal{Y})} \quad \Longleftrightarrow \quad p(\mathcal{X} \cap \mathcal{Y}) = p(\mathcal{X}|\mathcal{Y})p(\mathcal{Y}) \tag{1}$$

BAYES' rule:

$$p(\mathcal{X}|\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{X})p(\mathcal{X})}{p(\mathcal{Y})} \tag{2}$$

Important Rules for Probabilities (Ctd.)

Sum rule for probabilities: Given a joint probability distribution $p(\mathcal{X} \cap \mathcal{Y})$, we can easily compute the marginal $p(\mathcal{X})$ by summing out the variable $\mathcal{Y}$ which we are not interested in **(marginalization)**:

$$p(\mathcal{X}) = \sum_{y \in \text{dom}(\mathcal{Y})} p(\mathcal{X} \cap \mathcal{Y} = y) \tag{3}$$

For continuous variables we have to replace the sum by an integral!

Important Rules for Probabilities (Ctd.)

Chain rule (product rule) for probabilities:

$$p(\mathcal{X}_1 \cap \mathcal{X}_2 \cap \mathcal{X}_3) = p(\mathcal{X}_1|\mathcal{X}_2 \cap \mathcal{X}_3)p(\mathcal{X}_2 \cap \mathcal{X}_3)$$
$$= p(\mathcal{X}_1|\mathcal{X}_2 \cap \mathcal{X}_3)p(\mathcal{X}_2|\mathcal{X}_3)p(\mathcal{X}_3) \quad (4)$$

In general the chain rule can be applied to an arbitrary number of random variables! Also note that we can choose a different order of the random variables.

Notation: In the following we will write $p(\mathcal{X}, \mathcal{Y})$ instead of $p(\mathcal{X} \cap \mathcal{Y})$

What are graphical Models?

- Probabilities play a central role in pattern recognition and machine learning
- We can represent probability distributions in **graphical form**
- Such representations are called **probabilistic graphical models (PGMs)**

Advantages:

- Simple way to **visualize the structure** of a probabilistic model
- Insights into the properties of the model, including **conditional independence properties**
- Inference can be expressed in terms of **graphical manipulations**

What are graphical Models? (Ctd.)

- A graph comprises **nodes** *(also called **vertices**)* connected by **links** *(also known as **edges** or **arcs**)*
- Each node represents a **random variable**
- Links express **probabilistic relationships** between the variables
- The graph describes the way in which the **joint distribution can be decomposed into a product of factors**
- A directed graphical model *(i. e. links have arrows)* is called a **BAYESian network** or **BN** for short
- An undirected graphical model is called **MARKOV random field (MRF)**

Section:

BAYESian Networks (BNs)

Motivation for the Use of Directed Graphical Models
Representation of large Probability Distributions
Independencies encoded in a BAYESian Network
d-Separation
Naïve BAYES as a BAYESian Network

Motivation

- Consider an arbitrary joint distribution $p(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ of three random variables
- **Please note:**
  - We do not make any assumptions about the random variables
  - They can be discrete or continuous
  - Also, we do not specify the type of distribution (Multinomial, GAUSSian, etc.)
- According to the chain rule (4), this joint distributions factorizes into

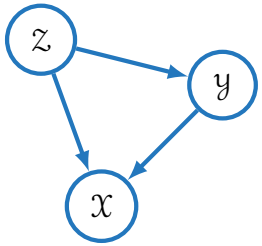
$$p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = p(\mathcal{X}|\mathcal{Y}, \mathcal{Z})p(\mathcal{Y}|\mathcal{Z})p(\mathcal{Z}) \tag{5}$$

- We now translate (5) into a directed graphical model, i. e. a BAYESian network

A simple BAYESian Network

Procedure:

1. Draw a node for each random variable
2. Associate each node with the corresponding conditional distribution on the right-hand side of equation (5)
3. For each conditional distribution we add directed links from the variables in the conditioning set



$$p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = p(\mathcal{X}|\mathcal{Y}, \mathcal{Z})p(\mathcal{Y}|\mathcal{Z})p(\mathcal{Z})$$

Some Remarks

- If there is a link going from node $\mathcal{A}$ to node $\mathcal{B}$, we call $\mathcal{A}$ a **parent** and $\mathcal{B}$ a **child**
- **Please note:** We do not make any formal distinction between a node and the corresponding random variable
- The BAYESian network above is **fully connected**
  (each pair of nodes is connected by a link; we ignore the directionality here)

Please note: A BAYESian network has to be a **directed acyclic graph (DAG)**, i. e. no directed cycles are allowed! *(Note that the network above does not contain a directed cycle)*

Let's add more Variables

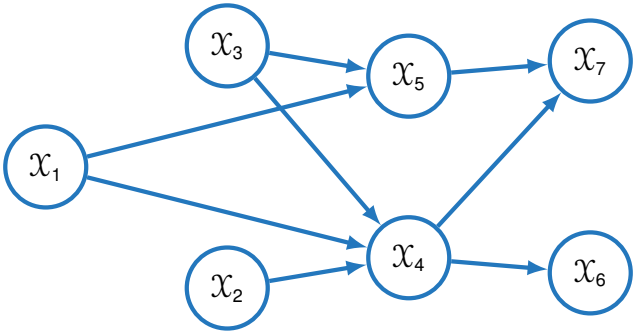
- Let us now consider a joint distribution over M variables $p(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M)$
- Repeatedly applying the chain rule yields:

$$p(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M) = p(\mathcal{X}_1|\mathcal{X}_2, \ldots, \mathcal{X}_M) \ldots p(\mathcal{X}_{M-1}|\mathcal{X}_M)p(\mathcal{X}_M) \tag{6}$$

- Analogously to above we could construct a BAYESian network which represents this joint distribution
- This would again lead to a **fully connected network**

It is the **absence of links** which conveys interesting information!

Let's add more Variables (Ctd.)



$$p(X_1, X_2, X_3, X_4, X_5, X_6, X_7) = ? \tag{7}$$

Let's add more Variables (Ctd.)

- We write the joint distribution in terms of the product of a set of conditional probabilities *(one for each node)*
- Each conditional probability will be conditioned **only on the parents** of the corresponding node
- Therefore, the joint distribution $p(\mathcal{X}_1, \ldots, \mathcal{X}_7)$ factorizes according to

$$p(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4, \mathcal{X}_5, \mathcal{X}_6, \mathcal{X}_7) =$$

$$p(\mathcal{X}_1)p(\mathcal{X}_2)p(\mathcal{X}_3)p(\mathcal{X}_4|\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)p(\mathcal{X}_5|\mathcal{X}_1, \mathcal{X}_3)p(\mathcal{X}_6|\mathcal{X}_4)p(\mathcal{X}_7|\mathcal{X}_4, \mathcal{X}_5) \quad (8)$$

Factorization of the Joint Probability Distribution in General

Factorization of the joint distribution in a BAYESian network:

$$p(\mathcal{X}_1, \ldots, \mathcal{X}_M) = \prod_{m=1}^{M} p\big(\mathcal{X}_m \mid \mathrm{pa}(\mathcal{X}_m)\big) \tag{9}$$

The joint distribution $p(\mathcal{X}_1, \ldots, \mathcal{X}_M)$ defined by the graph is given by the product, over all nodes of the graph, of a conditional distribution for each node $\mathcal{X}_m$ conditioned on the variables corresponding to the parents of that node in the graph $\mathrm{pa}(\mathcal{X}_m)$

(Very important!)

Number of Parameters in General

Consider a discrete random variable $\mathcal{X}$ which can take ℓ possible states

Question: How many parameters are needed to specify the joint distribution?

Answer: $\ell - 1$ **(why?)**

Now consider M random variables each of which can take ℓ different states

Question: How many parameters does the joint distribution have now?

Answer: $M^\ell - 1 \Rightarrow$ **exponentially many!**

Number of Parameters in BAYESian Networks

- **BAYESian networks usually need much fewer parameters!**
- Consider the following graph *(no links at all)*:

- The joint distribution factorizes according to $p(\mathcal{X}_1)p(\mathcal{X}_2)\ldots p(\mathcal{X}_M)$
- In this extreme case we have $M(\ell - 1) \ll M^\ell - 1$ parameters

The number of parameters now grows linearly with the number of variables!

Number of Parameters in BAYESian Networks (Ctd.)

- Admittedly, the previous graph is not very practical
- Consider indested a linear chain

- The joint distribution now factorizes according to $p(\mathcal{X}_1)p(\mathcal{X}_2|\mathcal{X}_1)\ldots p(\mathcal{X}_M|\mathcal{X}_{M-1})$

How many parameters do we have now? Answer: $\ell - 1 + (M-1)\ell(\ell-1)$
*(which is **quadratic** in ℓ and linear in M)*

Example: Number of Parameters

Let us consider a simple network: Grade $\mathcal{G}$ is influenced by intelligence $\mathcal{I}$

(this model comprises three free parameters)

	$\mathcal{I} = \text{high}$	$\mathcal{I} = \text{low}$
$p(\mathcal{I})$	0.85	0.15

$p(\mathcal{G} \mid \mathcal{I})$	$\mathcal{I} = \text{high}$	$\mathcal{I} = \text{low}$
$\mathcal{G} = \text{a}$	0.90	0.50
$\mathcal{G} = \text{b}$	0.10	0.50

$$p(\mathcal{G} = \text{b}, \mathcal{I} = \text{high}) = p(\mathcal{G} = \text{b} | \mathcal{I} = \text{high}) p(\mathcal{I} = \text{high}) = 0.85 \cdot 0.1 = 0.085$$

Conditional Independence

- Important concept: **conditional independence**
- Consider three random variables $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ for which the conditional distribution of $\mathcal{A}$ given $\mathcal{B}$ and $\mathcal{C}$ is such that it does not depend on the value of $\mathcal{B}$, i. e.

$$p(\mathcal{A}|\mathcal{B}, \mathcal{C}) = p(\mathcal{A}|\mathcal{C}) \tag{10}$$

- Therefore, we can write: $p(\mathcal{A}, \mathcal{B}|\mathcal{C}) \overset{(4)}{=} p(\mathcal{A}|\mathcal{B}, \mathcal{C})p(\mathcal{B}|\mathcal{C}) = p(\mathcal{A}|\mathcal{C})p(\mathcal{B}|\mathcal{C})$

$\mathcal{A}$ and $\mathcal{B}$ are said to be **statistically independent** given $\mathcal{C}$

Conditional Independence (Ctd.)

- We use the symbol $\perp\!\!\!\perp$ to denote independence of random variables
- For equation (10) we would write

$$\left(\mathcal{A} \perp\!\!\!\perp \mathcal{B}\right) \mid \mathcal{C} \tag{11}$$

Conditional independence properties play an important role in using probabilistic models for pattern recognition by simplifying both the structure of a model and the computations needed to perform inference and learning under that model

Local MARKOV Assumption

How to read off the independencies from a BAYESian network?

Local MARKOV assumption:

From equation (9) we already know that a variable $\mathcal{X}_m$ is independent of its non-descendants $\text{nd}(\mathcal{X}_m)$ given its parents $\text{pa}(\mathcal{X}_m)$:

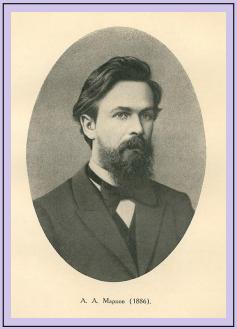
$$\mathcal{X}_m \perp\!\!\!\perp \text{nd}(\mathcal{X}_m) \mid \text{pa}(\mathcal{X}_m) \qquad \forall\, m = 1, 2, \ldots, M \tag{12}$$

Portrait: ANDREY ANDREYEVICH MARKOV

ANDREY ANDREYEVICH MARKOV (14 June 1856 – 20 July 1922) was a Russian mathematician best known for his work on **stochastic processes**. A primary subject of his research later became known as the **MARKOV chain**. He was also a strong, close to master-level, chess player.

MARKOV and his younger brother VLADIMIR ANDREEVICH MARKOV (1871 – 1897) proved the MARKOV BROTHERS' INEQUALITY. His son, another ANDREY ANDREYEVICH MARKOV (1903 – 1979), was also a notable mathematician, making contributions to constructive mathematics and recursive function theory.

(Wikipedia)



Representation Theorem

- We write $I_{\text{LM}}(\mathcal{G})$ for the set of the conditional independencies implied by the local MARKOV assumption
- **Question:** Is $\mathcal{G}$ an **I-map** (independence map) of p?

$$I_{\text{LM}}(\mathcal{G}) \overset{?}{\subseteq} I(p)$$

Representation theorem:

$$I_{\text{LM}}(\mathcal{G}) \subseteq I(p) \iff p(\mathcal{X}_1, \ldots, \mathcal{X}_M) = \prod_{m=1}^{M} p\big(\mathcal{X}_m \mid \text{pa}(\mathcal{X}_m)\big) \qquad (13)$$

Independencies in real Problems

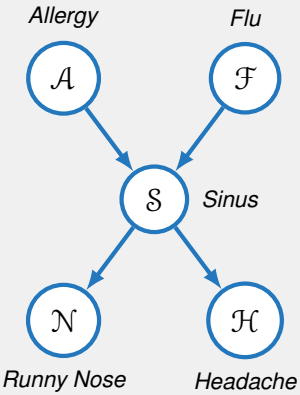
Real world

The true distribution p contains independency assertions $I(p)$

Model

The graph $\mathcal{G}$ encodes local independency assumptions $I_{\text{LM}}(\mathcal{G})$

Example: Local MARKOV Assumption



Let us consider an example:

Let the five binary variables *(yes, no)*

$\mathcal{A}$ (allergy), $\mathcal{F}$ (flu), $\mathcal{S}$ (sinus infection),
$\mathcal{N}$ (runny nose), $\mathcal{H}$ (headache),

and the BAYESian network depicted on the left-hand side be given

Example: Local MARKOV Assumption (Ctd.)

Node $\mathcal{F}$:

$$\mathsf{pa}(\mathcal{F}) = \emptyset$$
$$\mathsf{nd}(\mathcal{F}) = \{\mathcal{A}\}$$

Independencies:

$$\mathcal{A} \perp\!\!\!\perp \mathcal{F}$$

Node $\mathcal{N}$:

$$\mathsf{pa}(\mathcal{N}) = \mathcal{S}$$
$$\mathsf{nd}(\mathcal{N}) = \{\mathcal{A}, \mathcal{F}, \mathcal{H}\}$$

Independencies:

$$\mathcal{N} \perp\!\!\!\perp \{\mathcal{A}, \mathcal{F}, \mathcal{H}\} \mid \mathcal{S}$$

Node $\mathcal{S}$:

$$\mathsf{pa}(\mathcal{S}) = \{\mathcal{A}, \mathcal{F}\}$$
$$\mathsf{nd}(\mathcal{S}) = \emptyset$$

Independencies:

$$\emptyset$$

Example: Local MARKOV Assumption (Ctd.)

- According to the **chain rule**, the joint probability distribution is given by:

$$p(\mathcal{A}, \mathcal{F}, \mathcal{S}, \mathcal{H}, \mathcal{N}) = p(\mathcal{F}) \cdot p(\mathcal{A}|\mathcal{F}) \cdot p(\mathcal{S}|\mathcal{F}, \mathcal{A}) \cdot p(\mathcal{H}|\mathcal{S}, \mathcal{F}, \mathcal{A}) \cdot p(\mathcal{N}|\mathcal{S}, \mathcal{F}, \mathcal{A}, \mathcal{H})$$

- By applying the **local MARKOV assumption** we get:

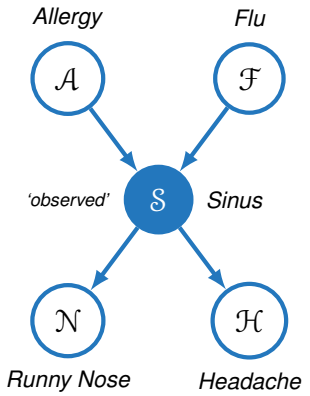
$$p(\mathcal{A}, \mathcal{F}, \mathcal{S}, \mathcal{H}, \mathcal{N}) = p(\mathcal{A}) \cdot p(\mathcal{F}) \cdot p(\mathcal{S}|\mathcal{A}, \mathcal{F}) \cdot p(\mathcal{N}|\mathcal{S}) \cdot p(\mathcal{H}|\mathcal{S})$$

$\Longrightarrow$ **Much less parameters due to the local MARKOV assumption!**

Explaining Away / BERKSON's Paradox

- From above we know $\mathcal{A} \perp\!\!\!\perp \mathcal{F}$
- Let us assume we observe $\mathcal{S}$
- Two causes ($\mathcal{A}$ and $\mathcal{F}$) *compete* to explain the observed data ($\mathcal{S}$)
- It follows: $\neg(\mathcal{A} \perp\!\!\!\perp \mathcal{F} \mid \mathcal{S})$, although $\mathcal{A} \perp\!\!\!\perp \mathcal{F}$

$\mathcal{A}$ and $\mathcal{F}$ become dependent when we observe $\mathcal{S}$!



Independencies encoded in a BAYESian Network

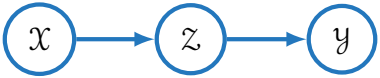
- A graph encodes more dependencies than are implied by the **local MARKOV assumption**
- Consider the following network:

  - Using the local MARKOV assumption we get $\mathcal{D} \perp\!\!\!\perp \{\mathcal{A}, \mathcal{B}\} \mid \mathcal{C}$
  - But we also have $\mathcal{D} \perp\!\!\!\perp \mathcal{A} \mid \mathcal{C}$ and $\mathcal{D} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{C}$ *(this is not covered by the local* MARKOV *assumption)*

This leads us to the concept of **d-separation (dependency separation)**

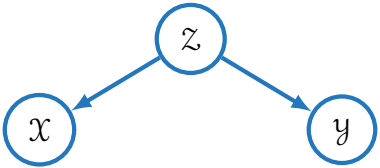
Four Example Graphs

Indirect causal effect:

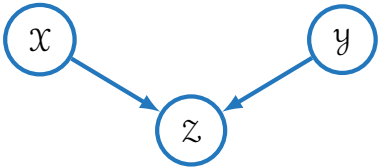


Indirect evidential effect:

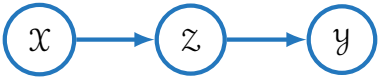
Common cause:



Common effect / v-structure:



Indirect causal Effect – No Variables observed



- The joint distribution factorizes into

$$p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = p(\mathcal{X})p(\mathcal{Z}|\mathcal{X})p(\mathcal{Y}|\mathcal{Z}) \tag{14}$$

- **Question:** Are $\mathcal{X}$ and $\mathcal{Y}$ independent?

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

Indirect causal Effect – No Variables observed (Ctd.)

$$p(\mathcal{X}, \mathcal{Y}) \overset{(3)}{=} \sum_{\mathcal{Z}} p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) \overset{(14)}{=} \sum_{\mathcal{Z}} p(\mathcal{X}) p(\mathcal{Z}|\mathcal{X}) p(\mathcal{Y}|\mathcal{Z})$$

$$= p(\mathcal{X}) \sum_{\mathcal{Z}} p(\mathcal{Z}|\mathcal{X}) p(\mathcal{Y}|\mathcal{Z}) = p(\mathcal{X}) \sum_{\mathcal{Z}} p(\mathcal{Y}, \mathcal{Z}|\mathcal{X})$$

$$= p(\mathcal{X}) p(\mathcal{Y}|\mathcal{X})$$

$$\neq p(\mathcal{X}) p(\mathcal{Y})$$

Answer: If $\mathcal{Z}$ is **not observed**, then $\mathcal{X}$ and $\mathcal{Y}$ are **dependent**: $\neg(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \emptyset)$

Indirect causal Effect – Variabes observed

- Now suppose that we observe the value of $\mathcal{Z}$
- **Question:** Are $\mathcal{X}$ and $\mathcal{Y}$ independent given $\mathcal{Z}$?

$$p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) \stackrel{?}{=} p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})$$

Indirect causal Effect – Variabes observed (Ctd.)

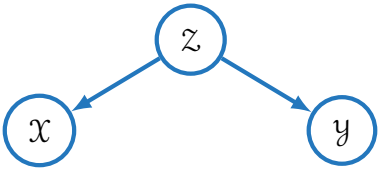
$$p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) \overset{(1)}{=} \frac{p(\mathcal{X}, \mathcal{Y}, \mathcal{Z})}{p(\mathcal{Z})} \overset{(14)}{=} \frac{p(\mathcal{X})p(\mathcal{Z}|\mathcal{X})p(\mathcal{Y}|\mathcal{Z})}{p(\mathcal{Z})}$$

$$= \frac{p(\mathcal{X})p(\mathcal{Z}|\mathcal{X})}{p(\mathcal{Z})} p(\mathcal{Y}|\mathcal{Z})$$

$$\overset{(2)}{=} p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})$$

Answer: If $\mathcal{Z}$ is **observed**, then $\mathcal{X}$ and $\mathcal{Y}$ are **independent**: $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z}$

Common Cause – No Variables observed



- Factorization of the joint probability:

$$p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})p(\mathcal{Z}) \tag{15}$$

- **Question:** Is $\mathcal{X}$ indpendent from $\mathcal{Y}$?

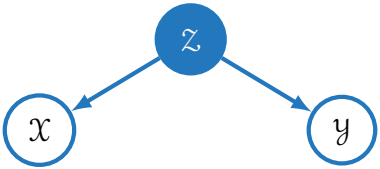
$$p(\mathcal{X}, \mathcal{Y}) \stackrel{?}{=} p(\mathcal{X})p(\mathcal{Y})$$

Common Cause – No Variables observed (Ctd.)

$$
\begin{aligned}
p(\mathcal{X}, \mathcal{Y}) &\overset{(3)}{=} \sum_{\mathcal{Z}} p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) \\
&\overset{(15)}{=} \sum_{\mathcal{Z}} p(\mathcal{X}|\mathcal{Z}) p(\mathcal{Y}|\mathcal{Z}) p(\mathcal{Z}) \\
&\neq p(\mathcal{X}) p(\mathcal{Y})
\end{aligned}
$$

Answer: If $\mathcal{Z}$ is **not observed**, then $\mathcal{X}$ and $\mathcal{Y}$ are **dependent**: $\neg(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \emptyset)$

Common Cause – Variables observed



- Again, we observe the value of $\mathcal{Z}$
- **Question:** Are $\mathcal{X}$ and $\mathcal{Y}$ independent given $\mathcal{Z}$?

$$p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) \stackrel{?}{=} p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})$$

Common Cause – Variables observed (Ctd.)

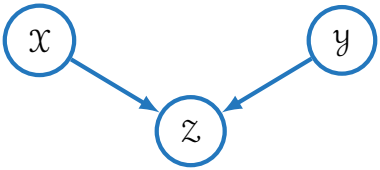
$$p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) \overset{(1)}{=} \frac{p(\mathcal{X}, \mathcal{Y}, \mathcal{Z})}{p(\mathcal{Z})}$$

$$\overset{(15)}{=} \frac{p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})p(\mathcal{Z})}{p(\mathcal{Z})}$$

$$= p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})$$

Answer: If $\mathcal{Z}$ is **observed**, then $\mathcal{X}$ and $\mathcal{Y}$ are **independent**: $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z}$

Common Effect – No Variables observed



- Factorization of the joint probability:

$$p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = p(\mathcal{X})p(\mathcal{Y})p(\mathcal{Z}|\mathcal{X}, \mathcal{Y}) \tag{16}$$

- **Question:** Is $\mathcal{X}$ indpendent from $\mathcal{Y}$?

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

Common Effect – No Variables observed (Ctd.)

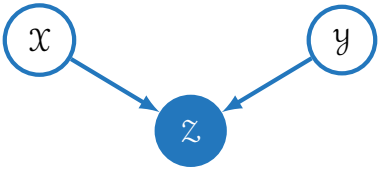
$$p(\mathcal{X}, \mathcal{Y}) \overset{(3)}{=} \sum_{\mathcal{Z}} p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) \overset{(16)}{=} \sum_{\mathcal{Z}} p(\mathcal{X}) p(\mathcal{Y}) p(\mathcal{Z}|\mathcal{X}, \mathcal{Y})$$

$$= p(\mathcal{X}) p(\mathcal{Y})$$

Answer: If $\mathcal{Z}$ is **not observed**, then $\mathcal{X}$ and $\mathcal{Y}$ are **independent**: $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \emptyset$

Note that this pattern behaves differently than the previous patterns!

Common Effect – Variables observed



- Let's see how this pattern behaves when observing the value of $\mathcal{Z}$
- **Question:** Are $\mathcal{X}$ and $\mathcal{Y}$ independent given $\mathcal{Z}$?

$$p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) \stackrel{?}{=} p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})$$

Common Effect – Variables observed (Ctd.)

$$p(\mathcal{X}, \mathcal{Y}|\mathcal{Z}) \overset{(1)}{=} \frac{p(\mathcal{X}, \mathcal{Y}, \mathcal{Z})}{p(\mathcal{Z})} \overset{(16)}{=} \frac{p(\mathcal{X})p(\mathcal{Y})p(\mathcal{Z}|\mathcal{X}, \mathcal{Y})}{p(\mathcal{Z})} \neq p(\mathcal{X}|\mathcal{Z})p(\mathcal{Y}|\mathcal{Z})$$

Answer: If $\mathcal{Z}$ is **observed**, then $\mathcal{X}$ and $\mathcal{Y}$ are **dependent**: $\neg(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z})$

Note that this pattern behaves again differently than the previous patterns!

$\mathcal{X}$ and $\mathcal{Y}$ also become dependent if a descendant of $\mathcal{Z}$ is observed (but not $\mathcal{Z}$ itself)

Summary

Let $\mathcal{X}$ and $\mathcal{Y}$ be two random variables which are connected via a third node $\mathcal{Z}$

The arrows at node $\mathcal{Z}$ meet either **head-to-tail** or **tail-to-tail**: $\mathcal{X}$ and $\mathcal{Y}$ are dependent if $\mathcal{Z}$ is not observed, else they are independent.
Examples: Indirect causal effect, indirect evidential effect, common cause

The arrows at node $\mathcal{Z}$ meet **head-to-head**: $\mathcal{X}$ and $\mathcal{Y}$ are independent if $\mathcal{Z}$ is not observed, else they are dependent.
Example: Common effect / v-structure

d-Separation [PEARL.1988]

- Consider a general directed graph in which ***A***, ***B***, and ***C*** are arbitrary **nonintersecting** sets of nodes *(whose union may be smaller than the complete set of nodes in the graph)*
- We wish to determine whether a particular conditional independence statement

$$\boldsymbol{A} \perp\!\!\!\perp \boldsymbol{B} \mid \boldsymbol{C}$$

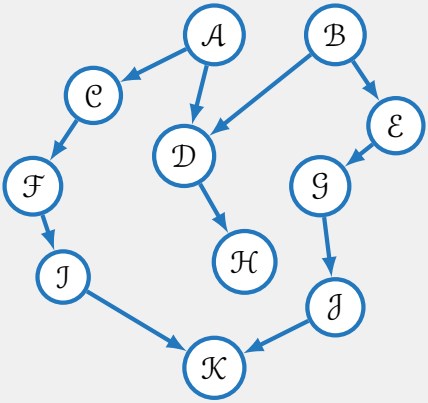
  is implied by a given directed acyclic graph
- To do so, we consider **all possible paths** from any node in ***A*** to any node in ***B***

d-Separation [PEARL.1988] (Ctd.)

- Any such path is said to be **blocked** if it includes a node such that either
  1. the arrows on the path meet either *head-to-tail* or *tail-to-tail* at the node, and the node is in the set $\boldsymbol{C}$, or
  2. the arrows meet *head-to-head* at the node, and neither the node, nor any of its descendants, is in the set $\boldsymbol{C}$
- If all paths are blocked, then $\boldsymbol{A}$ is said to be **d-separated** from $\boldsymbol{B}$ by $\boldsymbol{C}$
- The joint distribution over all of the variables in the graph will satisfy

$$\boldsymbol{A} \perp\!\!\!\perp \boldsymbol{B} \mid \boldsymbol{C}$$

Example: d-Separation

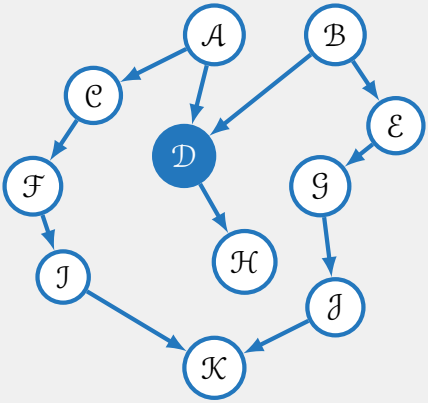


- **Question:** $\mathcal{F} \perp\!\!\!\perp \mathcal{G}$?
- Have a look at all consecutive triplets:

$\mathcal{F} - \mathcal{I} - \mathcal{K}$	**Active**
$\mathcal{I} - \mathcal{K} - \mathcal{J}$	**Inactive** (v-structure)
$\mathcal{K} - \mathcal{J} - \mathcal{G}$	**Active**

- This trail is not active
- Do the same with the other path
  *(it is also inactive – **why?**)*
- **Answer:** We have $\mathcal{F} \perp\!\!\!\perp \mathcal{G}$

Example: d-Separation (Ctd.)



- **Question:** $\mathcal{F} \perp\!\!\!\perp \mathcal{G} \mid \mathcal{D}$?
- Have a look at all consecutive triplets:

$\mathcal{F} - \mathcal{C} - \mathcal{A}$	**Active**
$\mathcal{C} - \mathcal{A} - \mathcal{D}$	**Active**
$\mathcal{A} - \mathcal{D} - \mathcal{B}$	**Active** *(v-structure)*
$\mathcal{D} - \mathcal{B} - \mathcal{E}$	**Active**
$\mathcal{B} - \mathcal{E} - \mathcal{G}$	**Active**

- This trail is active!
- We have $\neg(\mathcal{F} \perp\!\!\!\perp \mathcal{G} \mid \mathcal{D})$

Soundness of d-Separation

We define $I(\mathcal{G})$ to be the set of **all** conditional independencies (including d-separation)

Soundness:

$$p \text{ factorizes according to } \mathcal{G} \Longrightarrow I(\mathcal{G}) \subseteq I(p) \tag{17}$$

- Hence, d-separation captures only **true independencies**
- We have $I(\mathcal{G}) \subseteq I(p)$ and not only $I_{\text{LM}}(\mathcal{G}) \subseteq I(p)$

Completeness of d-Separation

Completeness:

- One can also show:

$$p \text{ factorizes according to } \mathcal{G} \Longrightarrow I(p) \subseteq I(\mathcal{G}) \tag{18}$$

  for 'almost all' distributions p

- In this case p is called **faithful** *(a faithful distribution does **not declare extra independence assumptions** that **cannot be read off** from $\mathcal{G}$)*
- $\mathcal{G}$ is called a perfect map, **P-map** $\Longleftrightarrow I(\mathcal{G}) = I(p)$

Summary: D-Maps, I-Maps, and perfect Maps

A graph $\mathcal{G}$ is said to be an **I-map** (independence map) of a specific distribution p, if every conditional independence statement implied by $\mathcal{G}$ is satisfied by p

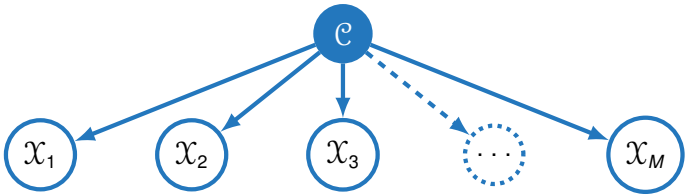
$\mathcal{G}$ is said to be a **D-map** (dependency map) of a distribution p, if every conditional independence statement satisfied by p is reflected in $\mathcal{G}$

$\mathcal{G}$ is called **P-map** (perfect map) if it is both, a D-map and an I-map

Graphical Model for Naïve BAYES

- The naïve BAYES model can be expressed as a graphical model ('common cause')
- **Assumption:**

$$\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \mid \mathcal{C} \qquad \forall\, \boldsymbol{X}, \boldsymbol{Y} \text{ subsets of } \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M\}$$



$$p(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M, \mathcal{C}) = p(\mathcal{C}) \prod_{m=1}^{M} p(\mathcal{X}_m|\mathcal{C})$$

Section:

Inference and Sampling in Graphical Models

Complexity of Inference in BAYESian Networks
Exact Inference in BAYESian Networks
Approximate Inference in BAYESian Networks

Inference in BAYESian Networks

- We want to use a BAYESian network to answer queries, i. e. we have to compute the probability of certain events
- This is called **inference**

In general, inference in BAYESian networks is hopeless.
Inference in BAYESian networks (even approximate inference) is NP-hard!

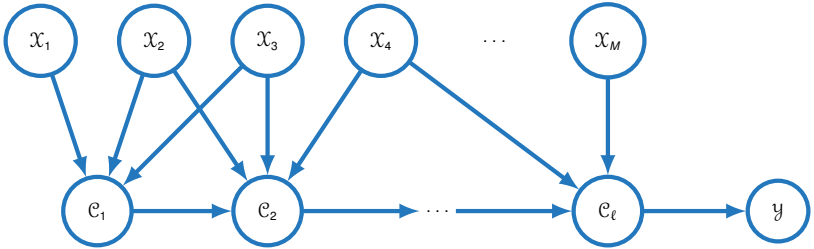
- In practice we can exploit the **structure of the network** to be more efficient
- There are some effective **approximation algorithms**

Complexity of Inference

- Consider a reduction to **3-SAT (3-satisfiability)**
  - We have groups (also called 'clauses') of three random variables
  - The random variables are connected via logical ORs $\vee$
  - The clauses are connected via logical ANDs $\wedge$
  - This problem is known to be **NP-hard**
- Suppose we have M boolean variables: **Does a satisfying assignment exist?**

$$\underbrace{\left(\neg x_1 \vee x_2 \vee x_3\right)}_{=:c_1} \wedge \underbrace{\left(\neg x_2 \vee x_3 \vee \neg x_4\right)}_{=:c_2} \wedge \ldots$$

Complexity of Inference (Ctd.)

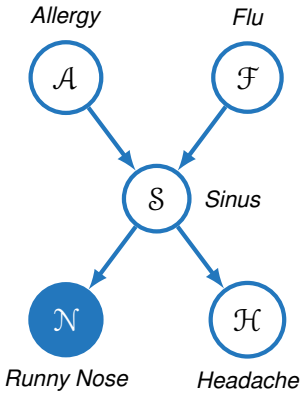


- There are 2^M possible assignments of the variables X_m $(1 \leqslant m \leqslant M)$
- $p(Y = 1) =$ # satisfying assignments$/2^M$
- **This problem is in #P**

Exact Inference

- Suppose we have the **conditional probability query**: *'What is the probability of having an allergy given a runny nose?'*, i. e. $p(\mathcal{A} = \mathrm{t} | \mathcal{N} = \mathrm{t})$
- We rewrite the expression using (1):

$$p(\mathcal{A} = \mathrm{t} | \mathcal{N} = \mathrm{t}) = \frac{p(\mathcal{A} = \mathrm{t}, \mathcal{N} = \mathrm{t})}{p(\mathcal{N} = \mathrm{t})}$$



Exact Inference (Ctd.)

- We know:

$$p(\mathcal{A}, \mathcal{F}, \mathcal{S}, \mathcal{H}, \mathcal{N}) = p(\mathcal{A}) \cdot p(\mathcal{F}) \cdot p(\mathcal{S}|\mathcal{A}, \mathcal{F}) \cdot p(\mathcal{N}|\mathcal{S}) \cdot p(\mathcal{H}|\mathcal{S})$$

- We compute $p(\mathcal{A} = \mathrm{t}, \mathcal{N} = \mathrm{t})$ by **summing out** all variables we are not interest in:

$$p(\mathcal{A} = t, \mathcal{N} = t) \overset{(3)}{=} \sum_{\mathcal{F}} \sum_{\mathcal{S}} \sum_{\mathcal{H}} p(\mathcal{A} = \mathrm{t}) \cdot p(\mathcal{F}) \cdot p(\mathcal{S}|\mathcal{A} = \mathrm{t}, \mathcal{F}) \cdot p(\mathcal{N} = \mathrm{t}|\mathcal{S}) \cdot p(\mathcal{H}|\mathcal{S})$$

$$= p(\mathcal{A} = \mathrm{t}) \sum_{\mathcal{S}} p(\mathcal{N} = \mathrm{t}|\mathcal{S}) \sum_{\mathcal{F}} p(\mathcal{F}) \cdot p(\mathcal{S}|\mathcal{A} = \mathrm{t}, \mathcal{F}) \sum_{\mathcal{H}} p(\mathcal{H}|\mathcal{S})$$

- Do the same for $p(\mathcal{N} = \mathrm{t})$ and compute $p(\mathcal{A} = \mathrm{t}|\mathcal{N} = \mathrm{t})$

Variable Elimination

Have: $p(\mathcal{A}, \mathcal{F}, \mathcal{S}, \mathcal{H}, \mathcal{N}) = p(\mathcal{A}) \cdot p(\mathcal{F}) \cdot p(\mathcal{S}|\mathcal{A}, \mathcal{F}) \cdot p(\mathcal{N}|\mathcal{S}) \cdot p(\mathcal{H}|\mathcal{S})$

Want: $p(\mathcal{H})$

Assume: Elimination order: $\mathcal{A}, \mathcal{F}, \mathcal{N}, \mathcal{S}$

Eliminate $\mathcal{A}$: $\psi_{\mathcal{A}}(\mathcal{F}, \mathcal{S}) = \sum_{a \in \mathcal{A}} p(a) \cdot p(\mathcal{S}|a, \mathcal{F})$ $\Rightarrow \psi_{\mathcal{A}}(\mathcal{F}, \mathcal{S}) \cdot p(\mathcal{F}) \cdot p(\mathcal{N}|\mathcal{S}) \cdot p(\mathcal{H}|\mathcal{S})$

Eliminate $\mathcal{F}$: $\psi_{\mathcal{F}}(\mathcal{S}) = \sum_{f \in \mathcal{F}} \psi_{\mathcal{A}}(f, \mathcal{S}) \cdot p(f)$ $\Rightarrow \psi_{\mathcal{F}}(\mathcal{S}) \cdot p(\mathcal{N}|\mathcal{S}) \cdot p(\mathcal{H}|\mathcal{S})$

Eliminate $\mathcal{N}$: $\psi_{\mathcal{N}}(\mathcal{S}) = \sum_{n \in \mathcal{N}} p(n|\mathcal{S})$ $\Rightarrow \psi_{\mathcal{F}}(\mathcal{S}) \cdot \psi_{\mathcal{N}}(\mathcal{S}) \cdot p(\mathcal{H}|\mathcal{S})$

Eliminate $\mathcal{S}$: $\psi_{\mathcal{S}}(\mathcal{H}) = \sum_{s \in \mathcal{S}} \psi_{\mathcal{F}}(s) \cdot \psi_{\mathcal{N}}(s) \cdot p(\mathcal{H}|s)$ $\Rightarrow \boxed{\psi_{\mathcal{S}}(\mathcal{H})}$

Input: BAYESian network $\mathcal{G}$, query $p(\boldsymbol{X}|\boldsymbol{O})$

1 Instantiate the evidence variables in $\boldsymbol{O}$

2 Choose an ordering of the variables to be eliminated $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M\}$

3 Initialize the factors

$$\boldsymbol{\Psi} := \{\psi_1, \psi_2, \ldots, \psi_M : \psi_m := p(\mathcal{X}_m \mid \text{pa}(\mathcal{X}_m))\}$$

4 **foreach** $m \in \{1, 2, \ldots, M\}$ **do**

5 **if** $\mathcal{X}_m \notin \{\boldsymbol{X}, \boldsymbol{O}\}$ **then**

6 Remove the factors $\psi_1, \psi_2, \ldots, \psi_L$ from $\boldsymbol{\Psi}$ that include $\mathcal{X}_m$

7 Generate a new factor $\widetilde{\psi}$ by eliminating $\mathcal{X}_m$ from these factors: $\widetilde{\psi} := \sum_{\mathcal{X}_m} \prod_{\ell=1}^{L} \psi_\ell$

8 Add $\widetilde{\psi}$ to the set of factors $\boldsymbol{\Psi}$

9 **end**

10 **end**

11 Normalize probabilities

12 **return** *answer to query* $p(\boldsymbol{X}|\boldsymbol{O})$

Approximate Inference

- We have seen already that exact inference is in NP-hard
- In the following we will introduce some methods for **approximate inference**
- Some common methods include:
  - **Forward sampling** (without evidence, i. e. observed variables)
  - **Rejection sampling** (with evidence, i. e. observed variables)
  - **Gibbs sampling** (MCMC – Markov Chain Monte Carlo)
  - **Likelihood weighting**
- In this lecture we shall cover forward sampling, rejection sampling, and Gibbs sampling

Forward Sampling (without Evidence)

Input: Bayesian network $\mathcal{G}$, number of nodes M, number of samples N

Initialize set of samples: $\boldsymbol{S} := \emptyset$
for $n \in \{1, 2, \ldots, N\}$ **do**
    for $m \in \{1, 2, \ldots, M\}$ **do**
        For $\mathcal{X}_m$ sample value $s_m^{(n)}$ according to $p(\mathcal{X}_m \mid \text{pa}(\mathcal{X}_m))$
    end
    Append $\boldsymbol{s}^n$ to the list of samples $\boldsymbol{S}$
end
return *set of samples* $\boldsymbol{S} := \{\boldsymbol{s}^1, \boldsymbol{s}^2, \ldots, \boldsymbol{s}^N\}$

Forward Sampling: Answering Queries

- Suppose we have collected several samples $\boldsymbol{S} := \left\{ \boldsymbol{s}^1, \boldsymbol{s}^2, \ldots, \boldsymbol{s}^N \right\}$
- **Question:** How can we do inference with these samples?
- **Answer:** Count the number of samples for which $\mathcal{X}_m = x_i$ holds true and divide by the total number of samples

$$p(\mathcal{X}_m = x_i) \approx \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left\{ s_m^{(n)} = x_i \right\}$$

How about evidence, i. e. queries of the form $p(\mathcal{X}_m | \mathcal{X}_k)$?

Rejection Sampling (Forward Sampling with Evidence)

When we have evidence (i. e. some random variables are observed), then the **sample has to be consistent with the evidence!**

- **A simple approach:** Use forward sampling and ignore the evidence
- If the sample is not consistent: **Reject the sample** ($\Rightarrow$ rejection sampling)

Problem: What if the evidence has low probability? $\Rightarrow$ **Most samples will be rejected!** Rejection sampling can be slow...

GIBBS Sampling

- We shall now consider the **GIBBS sampling** method
- It belongs to the family of **MARKOV Chain Monte Carlo (MCMC)** methods
- The samples are **dependent** and form a **MARKOV chain**

Sampling process:

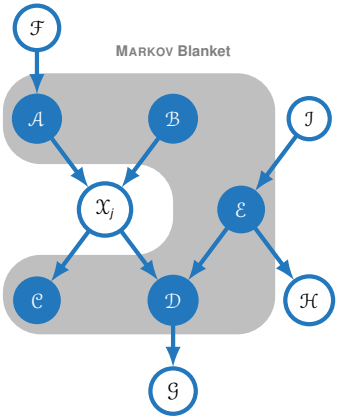
- Fix the values of evidence / observed variables $\boldsymbol{O}$
- Initialize the first sample $\boldsymbol{s}^0$ randomly
- Generate the next sample $\boldsymbol{s}^{n+1}$ based on the current one $\boldsymbol{s}^n$

Ordered Gibbs Sampler

- **Main idea:** Generate the next sample $\boldsymbol{s}^{n+1}$ based on the current one $\boldsymbol{s}^{n}$
- Sample variables **in order**:

$$
\begin{aligned}
\mathcal{X}_1 &: \quad s_1^{(n+1)} \sim p\left(s_1 \mid s_2^{(n)}, s_3^{(n)}, \ldots, s_m^{(n)}, \boldsymbol{O}\right) \\
\mathcal{X}_2 &: \quad s_2^{(n+1)} \sim p\left(s_2 \mid s_1^{(n+1)}, s_3^{(n)}, \ldots, s_m^{(n)}, \boldsymbol{O}\right) \\
&\vdots \\
\mathcal{X}_m &: \quad s_m^{(n+1)} \sim p\left(s_m \mid s_1^{(n+1)}, s_2^{(n+1)}, \ldots, s_{m-1}^{(n+1)}, \boldsymbol{O}\right)
\end{aligned}
$$

Markov Blanket



- We have to sample the value for $\mathcal{X}_m$ given all of the other variables in the network
- The **Markov blanket** simplifies that
- The Markov blanket of a node consists of its parents, co-parents, and children

A node is independent of all other nodes in the network given its Markov blanket

Section:

Hidden MARKOV Models (HMMs)

What is a hidden MARKOV Model?

- We shall now discuss the use of BAYESian networks for **sequence classification**
- Consider e. g. the task of **part-of-speech tagging**

Part-of-speech tagging (POS tagging) is the task of assigning **part-of-speech tags** (e. g. NN – noun, VB – verb, etc.) to a set of given words, e. g.

The	fans	watch	the	race
DET	*NN*	*VB*	*DET*	*NN*

Task: Predict the most probable sequence of tags given the observed words

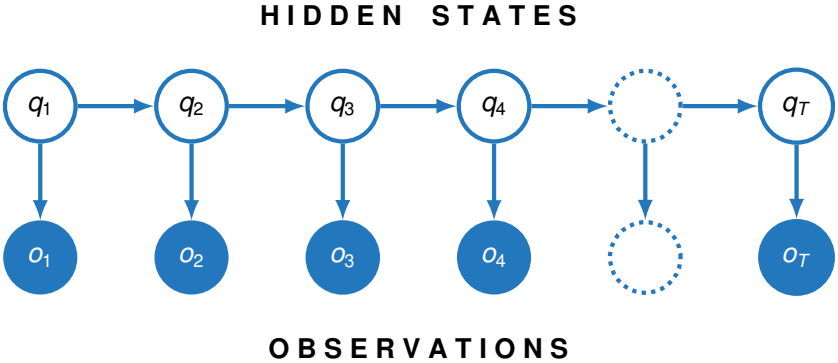
What is a hidden MARKOV Model? (Ctd.)

- We call the sequence of words **observed**
- The sequence of tags is **hidden** as we do not observe the tags in the real world

Can you imagine why this task is not as easy as it might seem?

In the following we will introduce **Hidden MARKOV Models (HMMs)** – a special kind of BAYESian network which allows us to compute the most probable sequence of hidden states given the observations

What does an HMM look like?



A running Example

To illustrate the concepts we will use the following scenario:

- Imagine you are a climatologist in the year 2799 studying the history of global warming
- Unfortunately, you cannot find any records for Mannheim for the summer of 2021
- But you do have Peter's diary who kept track of how much ice cream he ate every day that summer
- Our goal is to **use these observations to infer the temperature every day**
- We assume that there are only two kinds of days: **Cold (C)** and **hot (H)**

Formal Definition of HMMs

An HMM is specified by the following components:

$Q = q_1, q_2, \ldots, q_N$	A set of N possible **(hidden) states**
$\boldsymbol{A} = a_{11}, a_{12}, \ldots, a_{N1}, \ldots, a_{NN}$	An $N \times N$ **transition probability matrix**, each a_{ij} representing the probability of moving from state i to state j, such that $\sum_{j=1}^{N} a_{ij} = 1$ for all states $i = 1, 2, \ldots, N$
$O = o_1, o_2, \ldots, o_T$	A sequence of **observations**, each one drawn from a vocabulary $V = v_1, \ldots, v_K$

Formal Definition of HMMs (Ctd.)

An HMM is specified by the following components:

$\boldsymbol{B} = b_i(o_t)$	A sequence of **emission probabilities**, each expressing the probability of an observation o_t being generated from state i
q_0, q_F	A special **start state** q_0 and **end state** q_F (not associated with observations)

Running Example: Definition of the Components

- Let the set of states $Q := \{\text{Hot}, \text{Cold}\}$
- We use the start state <start>, but we do not use a dedicated end state, instead we allow both states, Hot and Cold to be final states
- Transition and emission probabilities are given in the tables below:

Transition probabilities *A*

$p(q_i \mid q_{i-1})$	Hot	Cold
<start>	0.80	0.20
Hot	0.70	0.30
Cold	0.40	0.60

Emission probabilities *B*

$p(o_i \mid q_i)$	1	2	3
Hot	0.20	0.40	0.40
Cold	0.50	0.40	0.10

Fundamental Problems

HMMs are characterized by three fundamental problems:

1. **Computation of the likelihood**
   Compute the likelihood of an observation sequence $\boldsymbol{o}$
2. **Decoding** *(we will focus on decoding in the lecture)*
   Given a sequence of observations $\boldsymbol{o} = o_1, o_2, \ldots, o_T$, find the most probable sequence of (hidden) states $\boldsymbol{q} = q_1, q_2, \ldots, q_T$
3. **Learning**
   Learn transition and emission probabilities from data

Decoding in Hidden Markov Models

- The most probable hidden state sequence $\widetilde{\boldsymbol{q}}$ given the observations $\boldsymbol{o}$ is:

$$\widetilde{\boldsymbol{q}} := \arg\max_{\boldsymbol{q}} p(\boldsymbol{q}|\boldsymbol{o}) \tag{19}$$

- This quantity is hard to compute. Let's apply **BAYES' rule** (2):

$$\widetilde{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} \frac{p(\boldsymbol{o}|\boldsymbol{q})p(\boldsymbol{q})}{p(\boldsymbol{o})} \propto \arg\max_{\boldsymbol{q}} p(\boldsymbol{o}|\boldsymbol{q})p(\boldsymbol{q}) \tag{20}$$

- Equation (20) is still not easy to compute, which is why we introduce **two simplifying assumptions**:

The Markov Assumption

Assumption 1 (Markov assumption): The probability of a specific state is dependent only on the previous state, i. e.

$$p(q_i|q_1, \ldots, q_{i-1}) = p(q_i|q_{i-1}) \tag{21}$$

('The future is independent of the past given the present')

Therefore:

$$p(\boldsymbol{q}) \overset{(4)}{=} p(q_T|q_1, \ldots, q_{T-1}) \cdots p(q_1) \overset{(21)}{=} \prod_{i=1}^{T} p(q_i|q_{i-1}) \tag{22}$$

The Output Independence Assumption

Assumption 2 (Output independence assumption): The probability of an observation o_i is dependent only on the state q_i that produced the observation, and not on any other states or observations, i. e.

$$p(o_i|q_1, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = p(o_i|q_i) \tag{23}$$

Therefore:

$$p(\boldsymbol{o}|\boldsymbol{q}) = \prod_{i=1}^{T} p(o_i|q_i) \tag{24}$$

Putting everything together...

- Plugging equations (22) and (24) into (20) we obtain:

$$\widetilde{\boldsymbol{q}} = \arg\max_{\boldsymbol{q}} p(\boldsymbol{o}|\boldsymbol{q})p(\boldsymbol{q}) = \arg\max_{\boldsymbol{q}} \prod_{i=1}^{T} p(o_i|q_i)p(q_i|q_{i-1}) \tag{25}$$

- This equation contains two types of probabilities:
  - **Transition probabilities:**

$$p(q_i|q_{i-1})$$

  - **Emission probabilities:**

$$p(o_i|q_i)$$

Efficient Computation

- To find $\widetilde{\boldsymbol{q}}$ we could enumerate all possible sequences of hidden states $\boldsymbol{q}$, evaluate equation (25), and pick the one which maximizes $p(\boldsymbol{o}|\boldsymbol{q})$
- This brute-force approach has computational complexity $\mathcal{O}(N^T)$ **(why?)**

The complexity is exponential in the length T of the sequence!

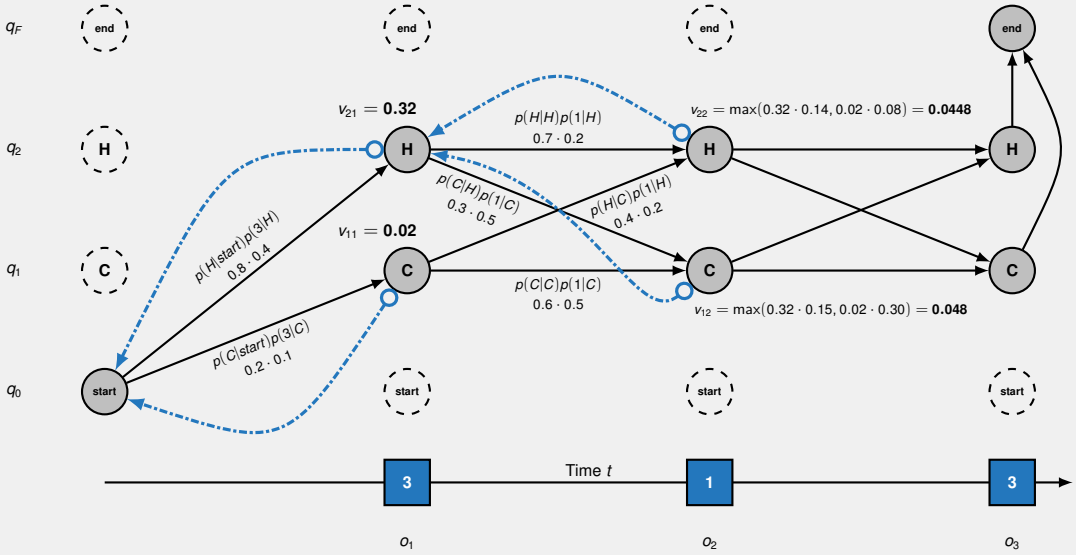
We shall now discuss how to improve on the computational complexity, leading to the **VITERBI algorithm**

The VITERBI Algorithm

- The **VITERBI algorithm** is a **dynamic programming** method which is able to compute $\widetilde{\boldsymbol{q}}$ efficiently
- It is named after ANDREW J. VITERBI
- We will introduce the algorithm by example first and formalize it afterwards

Coming back to our running example:
Suppose we observe the sequence $\boldsymbol{o} := \big(3, 1, 3\big)$. Our goal is to find the most probable sequence of hidden states $\widetilde{\boldsymbol{q}}$

Running Example: VITERBI Algorithm



VITERBI Algorithm: Initialization Phase

Input: $\boldsymbol{o} = o_1, o_2, \ldots, o_T$, state graph of length N

1 Create a path probability matrix $\boldsymbol{V} := \left[v_{ij}\right]_{j=1,\ldots,T}^{i=1,\ldots,N}$

2 Create a matrix of backpointers $\boldsymbol{P} := \left[p_{ij}\right]_{j=1,\ldots,T}^{i=1,\ldots,N+1}$

3 **foreach** *state* $q \in \{1, 2, \ldots, N\}$ **do**

4 $\quad v_{q1} = a_{0q} \cdot b_q(o_1)$

5 $\quad p_{q1} = 0$

6 **end**

Remarks:

- The value v_{qt} represents the probability of being in state q after seeing the first t observations and passing through the most probable state sequence $q_0, q_1, \ldots, q_{t-1}$
- We need the backpointers to retrieve the most probable state sequence

VITERBI Algorithm: Main Phase

foreach *time step* $t \in \{2, 3, \ldots, T\}$ **do**
  foreach *state* $q \in \{1, 2, \ldots, N\}$ **do**
    Compute:

$$v_{qt} = \max_{q'=1,\ldots N} v_{q',t-1} \cdot a_{q',q} \cdot b_q(o_t)$$

$$p_{qt} = \arg\max_{q'=1,\ldots N} v_{q',t-1} \cdot a_{q',q} \cdot b_q(o_t)$$

  end
end

Remarks:

- Compute the path through the VITERBI trellis
- By taking the max we **discontinue paths** which cannot be optimal

VITERBI Algorithm: Termination Phase

1 Compute:

$$v_{N+1,T} = \max_{q=1,\ldots,N} v_{qT} \cdot a_{q,N+1}$$

$$p_{N+1,T} = \arg\max_{q=1,\ldots,N} v_{qT} \cdot a_{q,N+1}$$

---

2 **return** *backtrace path in reverse order*

Remarks:

- The element $v_{N+1,T}$ represents the probability of the **most probable sequence** of hidden states
- The backtrace has to be returned in **reverse order!**

Complexity of the VITERBI Algorithm

- The VITERBI trellis consists of T layers (one for each observation)
- At each node in the VITERBI trellis we only have to consider the N nodes in the **preceeding layer**
- Each layer in the VITERBI trellis comprises N nodes
- The overall complexity of the VITERBI algorithm therefore is

$$\mathcal{O}(TN^2)$$

The VITERBI algorithm is much more efficient than brute-force!

Section:

Wrap-Up

Summary
Recommended Literature
Self-Test Questions
Lecture Outlook

Summary: BAYESian Networks (BNs)

- BAYESian networks are **directed acyclic graphs** which represent exponentially large probability distributions
- **Local MARKOV assumption:** A variable is independent of its non-descendants given its parents (and only its parents)
- **Representation theorem**
- The concept of **d-separation** can be used to find more independencies
- Inference (even approximate) is **NP-hard**
  - Exact inference: Variable elimination
  - Approximate inference: Forward sampling, rejection sampling, GIBBS sampling

Summary: Hidden MARKOV Models (HMMs)

- An HMM is a **sequence classifier** *(as such it takes the context into account)*
- This is useful e. g. in part-of-speech (POS) tagging
- **Two simplifying assumptions:**
  1. **MARKOV assumption:** The probability of a state q_i is dependent only on the previous state q_{i-1}
  2. **Output independence assumption:** The probability of an observation o_i depends only on the hidden state q_i
- Find the **most probable hidden sequence** (*decoding*) by applying the **VITERBI algorithm** *(dynamic programming)*

Recommended Literature

1. [Bishop.2006], chapter 8
2. [Koller.2009], chapter 3
3. [Koller.2009], chapter 9
4. [Jurafsky.2006], chapter 6 (HMMs)

(For free PDF versions, see list in GitHub readme!)

Self-Test Questions

1. What is a marginal / conditional distribution?
2. Write down the sum rule / chain rule for probabilities!
3. What is a BAYESian network?
4. How does a joint probability distribution factorize in a BAYESian network?
5. What is the local MARKOV assumption / the representation theorem?
6. What is d-separation?
7. Explain how inference can be made in BAYESian networks?
8. What are the simplifying assumptions a hidden MARKOV model makes?
9. Explain why the VITERBI algorithm is more efficient than brute force!

What's next...?

	I	Machine Learning Introduction
	II	Optimization Techniques
	III	Bayesian Decision Theory
	IV	Non-parametric Density Estimation
	V	Probabilistic Graphical Models
•	**VI**	Linear Regression
	VII	Logistic Regression
	VIII	Deep Learning
	IX	Evaluation
	X	Decision Trees
	XI	Support Vector Machines
	XII	Clustering
	XIII	Principal Component Analysis
	XIV	Reinforcement Learning
	XV	Advanced Regression

Thank you very much for the attention!

*** * * Artificial Intelligence and Machine Learning * * ***

Topic: Probabilistic Graphical Models (PGMs)

Term: Summer term 2025

Contact:
Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?