UniTs - University of Trieste

Faculty of Scientific and Data Intensive Computing

Department of mathematics informatics and geosciences

Probabilistic Machine Learning

Lecturer:
Prof. Luca Bortolussi

Authors:

Andrea Spinelli
Christian Faccio

March 24, 2025

github.com/Spina02 andreaspinelli2002@gmail.com

Abstract

As a student of Scientific and Data Intensive Computing, I've created these notes while attending the **Probabilistic Machine Learning** course.

This course will cover the "*probabilistic side*" of machine learning. In particular, we will focus on the following topics:

- Basics of probability and probabilistic inference
- Pronabilistic formulation of learning (Empirical Risk Minimization and PAC Learning)
- Graphical Models
- Inference with graphical models: belief propagation
- Hidden Markov Models for sequential data
- Bayesian Linear Regression and Classification, Laplace approximation, Model Selection
- Kernel Regression and Kernel functions, Gaussian Processes for regression (hints)
- Monte Carlo sampling
- Expectation Maximization and Variational Inference
- Bayesian Neural Networks
- Generative Modelling: Variational Autoencoders and Diffusion Processes

While these notes were primarily created for my personal study, they may serve as a valuable resource for fellow students and professionals interested in probabilistic machine learning.

Contents

1 Introduction

1.1 Machine Learning

Machine learning is a field of computer science about **learning models**.

Models

Definition: ***Model***

- A ***Model*** is a hypothesis that certain features of a system of interest are well replicated in another, simpler syetem.
- ***Mathematical Model*** is a model where the simpler system consists of a set of mathematical relations between objects (equations, inequalities, etc).
- A ***Stochastic Model*** is a mathematical model where the objects are probability distributions.

All modelling usually starts by defining a family of models indexed by some parameters, which are tweaked to reflect how well the feature of interest are replicated.
Machine learning deals with algorithms for automatic selecrion of a model from observations of the system.

Machine Learning

Definition: ***Machine learning***

Machine learning *explores the study and construction of algorithms that can learn from and make predictions on data.* *Wikipedia*

There are three main types of machine learning:

Generative and Discriminative Learning

- **Generative Learning** im at describing the full probability distribution of inputs x or input/output pairs (x,y).
$$p(x,y) = p(x)p(y|x)$$
- **Discriminative Learning** aims at describing the conditional probability of output given the input, or a statistics/function of such probability
$$p(y|x) \quad or \quad y = f(x)$$

[to fix:]
- **Supervised Learning**: The algorithm learns from labeled data by mapping inputs to outputs.
- **Unsupervised Learning**: The algorithm identifies patterns or structures in unlabeled data.
- **Data Generatiion**: The algorithm generates new data points.

Inference and Estimation

Two central concepts for probabilistic machine learning are:

- **Inference**: Compute marginals and contitionals probability distributions applying the laws of probability.
- **Estimation**: Given data and a family of models, find the best parameters/models that explains the data.

In the Bayesian world: estimation $\approx$ inference.

Probability

Probability is a mathematical theory that deals with **uncertainty**
When a certain problems has to face practical difficulties due to it's complexity, we can use probability to model the ***aleatorical uncertainty***, which is the uncertainty due to the randomness of the system.
More often, we have a limited knowledge of the system, and we can use probability to model the ***epistemic uncertainty***, which is the uncertainty due to the lack of knowledge.

Tip: *Everything is a probability distribution*

In machine learning **everything is a probability distribution**, even if not explicitly stated.

1.2 Probability basics

1.2.1 Random Variables

Random Variables are functions mapping outcomes of an experiment to real numbers. They serve as abstract representations of the outcomes in randomized experiments. Note that what we observe are the *realizations* (values resulting from an observed outcome) of these random variables.

Example: *Random Variable*

Consider the following example:

$$\{Head, Tail\}, \qquad \{0,1\}, \qquad \left\{\tfrac{1}{2},\tfrac{1}{2}\right\}.$$

Only the second is the random variable itself; the third is its probability distribution, while the first is the sample space of potential outcomes.

We consider a ***Sample Space*** Ω, which is the set of all possible outcomes of a random experiment.
A random variable X is a function:

$$X : \Omega \to E, \qquad \text{where } E \subseteq \mathbb{R} \quad (\text{or } E \subseteq \mathbb{N})$$

with the probability measure

$$P(X \in S) = P(\{\omega \in \Omega \mid X(\omega) \in S\}), \quad S \subseteq E.$$

A model for our random outcome is the probability distribution of X. In particular, if the sample space is finite or countable the **probability mass function (pmf)** is given by:

$$p(x) := P(X = x).$$

If the sample space is infinite, we use the **probability density function (pdf)** where

$$P(a \leq X \leq b) = \int_a^b p(x)dx \quad \text{and} \quad \int_{\mathbb{R}} p(x)dx = 1.$$

1.2.2 Notable Probability Distributions

Below are some of the most common probability distributions.

Discrete Distributions

Distribution	pmf	Mean	Variance
Binomial $\text{Bin}(n,p)$	$\binom{n}{x} p^x(1-p)^{n-x}$	np	$np(1-p)$
Bernoulli $\text{Bern}(p)$	$p \quad (x=1), \quad 1-p \quad (x=0)$	p	$p(1-p)$
Discrete Uniform $\mathscr{U}(a,b)$	$\frac{1}{b-a+1}$	$\frac{a+b}{2}$	$\frac{(b-a+1)^2-1}{12}$
Geometric $\text{Geom}(p)$	$(1-p)^{x-1}p$	$\frac{1}{p}$	$\frac{1-p}{p^2}$
Poisson $\text{Pois}(\lambda)$	$\frac{\lambda^x e^{-\lambda}}{x!}$	λ	λ

Continuous Distributions

Distribution	pdf	Mean	Variance
Continuous Uniform $\mathscr{U}(a,b)$	$\begin{cases} \frac{1}{b-a} & x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Exponential $\text{Exp}(\lambda)$	$\lambda e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gaussian $\mathscr{N}(\mu,\sigma^2)$	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$	μ	σ^2
Beta $\text{Beta}(\alpha,\beta)$	$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
Gamma $\text{Gamma}(\alpha,\beta)$	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$
Dirichlet $Dir(\alpha)$	$\frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$	$\tilde{\alpha}_i$	$\frac{\tilde{\alpha}_i(1-\tilde{\alpha}_i)}{\alpha_0+1}$
Student's t $St(\nu)$	$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$	0	$\begin{cases} \frac{\nu}{\nu-2} & \nu > 2 \\ \infty & 1 < \nu \leq 2 \end{cases}$

Notes:

- For discrete distributions, $n \in \{0,1,2,\ldots\}$, $p \in [0,1]$, and x runs over the support.
- For continuous distributions, parameters such as λ, μ, σ, α, and β belong to $\mathbb{R}$ (with appropriate restrictions) and $x \in \mathbb{R}$.
- In the Dirichlet distribution, $\tilde{\alpha}_i = \frac{\alpha_i}{\sum_{h=1}^{K} \alpha_h}$ and $\alpha_0 = \sum_{i=1}^{K} \alpha_i$.

- For Student’s t-distribution, $\nu > 1$.

2 Empirical Risk Minimization and PAC Learning

In this chapter, we will introduce the concept of **Empirical Risk Minimization** (ERM) in which to frame learning problems, the notion of inductive bias, and the main results of algorithmic learnability, encapsulated in the definition of **Probably Approximately Correct** (PAC) Learning and of complexity of a set of hypothesis, namely VC-dimension and Rademacher complexity.

2.1 Empirical Risk Minimization

We begin by considering a supervised learning setting in which the **input space** X is a subset of $\mathbb{R}^n$, and the **output space** Y can be real-valued (e.g., $Y = \mathbb{R}$), binary (e.g., $Y = \{0,1\}$), or a finite set of classes (e.g., $Y = \{0,1,\dots,K\}$). In this probabilistic framework, each input-output pair (x,y) is drawn from a joint probability distribution

$$p(x,y) \in \mathrm{Dist}(X \times Y),$$

often referred to as the ***data generating distribution***.
By definition, this distribution factors into the marginal $p(x)$ and the conditional $p(y \mid x)$, so that

$$p(x,y) = p(x)\, p(y \mid x).$$

Because $p(x)$ and $p(y \mid x)$ describe how inputs and outputs are related, it is helpful to write them explicitly. The marginal distribution of x is

$$p(x) = \int p(x,y)\, dy,$$

while the conditional distribution of y given x is

$$p(y \mid x) = \frac{p(x,y)}{p(x)}.$$

A typical dataset D in supervised learning consists of N input-output pairs drawn independently from $p(x,y)$. We denote this as

$$D \sim p^N(x,y),$$

which means

$$D = \{(x_i,y_i) \mid i = 1,\dots,N\},$$

where each (x_i,y_i) is sampled according to the joint distribution $p(x,y)$.
In many cases, we assume that $p(y \mid x)$ depends on some unknown function of x. Formally, one might write

$$p(y \mid x) = p\big(y \mid f(x)\big),$$

where f is the function we aim to learn. The central objective in supervised learning—through methods such as empirical risk minimization—is to find or approximate this function f by using the observed data D.

2.2 Risk and Empirical Risk

$h \in \mathcal{H} \quad x, y \sim p(x,y)$

loss function $l(x,y,h) \in \mathbb{R}_{\geq}$,

- 0-1 loss: $l(x,y,h) = \mathbb{I}(h(x) \neq y)$, eith $y \in \{0,1\}$.
- squared loss: $l(x,y,h) = (h(x)-y)^2$, with $y \in \mathbb{R}$.

We have a probabilistic process, so we have some inputs that are more likely than others. If a model makes a mistake on a more likely input, it should be penalized more.

Definition: *Risk*

The ***risk*** (or ***generalization error***) is defined as:

$$R(h) = E_{x,y \sim p(x,y)}[l(x,y,h)]$$

Risk minimization principle:

The goal is to find the hypothesis h that minimizes the risk.

$$\text{find } h^* \in \mathcal{H} \text{ such that } h^* = \arg\min_{h \in \mathcal{H}} R(h)$$

Definition: *Empirical Risk*

The ***empirical risk*** (or ***training error***) is defined as:

$$\hat{R} = \frac{1}{N} \sum_{i=1}^{N} l(x_i, y_i, h)$$

Empirical risk minimization principle:

The goal is to find the hypothesis h that minimizes the empirical risk.

$$\text{find } h_D^* = \arg\min_{h \in \mathcal{H}} \hat{R}(h)$$

2.2.1 Bias Variance Trade-off

In this section, we want to analyze the generalization error and decom- pose it according to the sources of error that we are going to commit.

In what follows, we will use the squared loss (hence we will focus on regression problems). Considering $h \in \mathcal{H}$, an explicit expression of the generalization error committed when choosing hypothesis h is:

$$R(h) = E_p[l(x,y,h)] = \int\int (h(x)-y)^2 p(x,y)\,dx\,dy$$

Theorem 1. *The minimizer of the generalization error R is:*

$$\boxed{g(x) = E[y|x] = \int y p(y|x)\,dy}$$

so that $g = \arg\min_h R(h)$, *if* $g \in \mathcal{H}$

We can rewrite the risk as:

$$R(h) = \underbrace{\int (h(x)-g(x))^2 p(x)\,dx}_{=0 \quad iff \quad h(x)=g(x)} + \overbrace{\iint (g(x)-y)^2 p(x,y)\,dx\,dy}^{\text{independent of } h\text{: intrinsic noise}}$$

$$\begin{aligned}E_D[R(h_D^*)] &= \underbrace{\int (E_D[h_D^*(x)-g(x)])^2 p(x)dx}_{bias^2} \\ &+ \underbrace{\int E_D[(h_D^*(x)-E_D[h_D^*(x)])^2 p(x)dx]}_{variance} \\ &+ \underbrace{\iint (g(x)-y)^2 p(x,y)dxdy}_{noise}\end{aligned}$$

2.3 ERM and Maximum Likelihood

Given a dataset $D = \{(x_i, y_i)\}_{i=1,\dots,m}$ s.t. $D \sim p^m, p = p(x,y)$
We factorize the data generating distributions as: $p(x,y) = p(x)p(y|x)$ and we make an hypothesis on $p(y|x)$, trying to express this conditional probability in a parametric form:

$$p(y|x) = p(y|x,\theta)$$

[check recording for missing part]
We consider the log Likelihood:

$$L(\theta;D) = \sum_{i=1}^{m} \log p(y_i|x_i,\theta)$$

Then we apply the maximum likelihood principle:

$$\begin{aligned}\arg\min_\theta -L(\theta;D) &= \arg\min_\theta -\frac{1}{m}\sum_{i=1}^{m}\log p(y_i|x_i,\theta) \\ &= \arg\min_\theta E_{p(x,y)}[-\log p(y|x,\theta)]\end{aligned}$$

since the avarage is an empirical approximation of the expectation.

Observation: ***Empirical Risk***

$-\frac{1}{m}\sum_{i=1}^{m}\log p(y_i|x_i,\theta)$is known as **empirical risk**

2.4 KL divergence

Consider a probability distribution $p(x)$, then $-\log p(x)$ is a measure of **self-information**. Indeed, if $p(x) = 1$ then $-\log p(x) = 0$ (no self-information), describing substantially out (lack of) surprise in observing the event. If instead $p(x) = 0$ then $-\log p(x) = \infty$. In general, the more rare the event is, i.e. the lower is $p(x)$, the more self-information it carries, i.e. the larger is $-\log p(x)$.

Definition: ***Entropy***

In an information-theoretic sense, the **entropy** is a measure of the inforamtion that is carries by a random phenomenon, expressed as the expected amount of self-information that is conveyed by a realization of the random phenomenon.

It is formally defined as:

$$H(p) = E_p[-\log p(x)] = -\int p(x) \log p(x) dx$$

for the continuous case, and:

$$H(p) = E_p[-\log p(x)] = -\sum p(x) \log p(x)$$

for the discrete case.

For the discrete case, the maximum entropy is achieved for te uniform distrivution and is equal to $\log n$, where n is the number of possible outcomes. In the continuous case, for a fixed variance, the distribution that maximizes the entropy is the Gaussian. The entropy is always 0 if we have a deterministic distribution.

Definition: ***Kullback-Leibler divergence***

The **Kullback-Leibler divergence** (or **relative entropy**) between two probability distributions $p(x)$ and $q(x)$ is a measure of how one distribution diverges from a second, expected probability distribution.
It is formally defined as:

$$D_{KL}(p||q) = E_p\left[\log \frac{p(x)}{q(x)}\right] = \int p(x) \log \frac{p(x)}{q(x)} dx$$

for the continuous case, and:

$$D_{KL}(p||q) = E_p\left[\log \frac{p(x)}{q(x)}\right] = \sum p(x) \log \frac{p(x)}{q(x)}$$

for the discrete case.

Intuitively, we are taking a sort of expected difference between p and q, expressed in terms of a log odds ratio. It tells us how different two distributions are: the larger KL the more different are p and q.
Properties of KL:

- $KL[q||p]$ is a convex function of q and p and $KL[q||p] \geq 0$
- KL is non-symmetric, i.e. $KL[q||p] \neq KL[p||q]$
- $KL[q||p] = -H[q] - \mathbb{E}_q[\log p]$, where the first term is the entropy and the second term is known as cross-entropy between p and q.

Suppose moreover, that p is fixed but unknown, $q = q_0$ can vary: what we usually do is trying to find the best q_θ that approximates p.
The **mutual information** between x and y is defined as:

$$I(x,y) = KL[p(x,y)||p(x)p(y)] = \int\int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy$$

$KL[p(x,y)||p(x)p(y)] = 0$ iff x and y are independent.
Moreover, the more dependent they are, the more different is $p(x,y)$ from the product of the marginals, the more information x carries about y and viceversa.
In other words, the higher the mutual information is, the more knowing y will tell us about x, the less residual uncertainty on x we will have.

Consider a dataset: $\underline{x} : x_1, \dots, x_N$:

Definition: ***Empirical distribution***

The **empirical distribution** of a dataset $\underline{x}$ is defined as:

$$\hat{p}(\underline{x}) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i)$$

where $\delta(x)$ is the Dirac delta function.

It is an approximation of the input data generating function$p(x)$. Practically, the more observations we have, the more the empirical distribution will look like $p(x)$.
Given a distribution q, we can compute:

$$KL[p_{emp}||q] = \mathbb{E}_{p_{emp}} \left[\log \frac{p_{emp}(x)}{q(x)} \right] = \frac{1}{N} \sum_{i=1}^{N} \log \frac{p(x_i)}{q(x_i)}$$

If $q = q_0$, this is $-\frac{1}{N} L(\Theta)$ plus a constant. Hence maximizing $L(\Theta)$ is essentially equivalent to minimizing the KL between p_{emp} and q_0. This means that we can always rephrase maximum likelihood in terms of cross-entropy.

2.5 PAC Learning

Our goal is to measure how much we can learn as a function of the model complexity. This results in the **PAC** (Probably Approximately Correct) **learning** framework, which encodes the notion of model complexity and gives also bounds on the error that we commit. Here we consider it in the context of (binary) classification, i.e. $y \in 0, 1$, using the 0-1 loss.

Definition: ***PAC Learning***

A realizable hypothesis set $\mathscr{H}$ is ***PAC learnable*** iff $\forall \varepsilon, \delta \in (0,1), \forall p(x,y), \exists m_{\varepsilon,\delta} \in \mathscr{N} s.t. \forall m \geq m_{\varepsilon,\delta}, \exists D\ p^m, |D| = m$ then $p_D(R(h_D^*) \leq \varepsilon) \geq 1 - \delta$

This means that, fixing two parameters $\varepsilon, \delta \in (0,1)$, governing our precision, and a data generating distribution $p(x,y)$, we can find a number of samples $m_{\varepsilon,\delta}$ such that, with probability at least $1-\delta$, the empirical risk of the best hypothesis h_D^* is less than ε. Note that the probability here is over the dataset D, meaning that our learning will succeed for a fraction $1-\delta$ of sampled datasets.
In a more general setting,

Definition:

Given an hypothesis set $\mathscr{H}$ (not necessarily realizable) and an algorithm A, $\mathscr{H} is$**agnostic PAC-learnable**$iff \forall \varepsilon, \delta \in (0,1), \forall p(x,y), \exists m_{\varepsilon,\delta} \in \mathscr{N}\ p^m, |D| = m \geq m_{\varepsilon,\delta} \Rightarrow$ $p_D(R(h_D^*) \leq R(h^*) + \varepsilon) \geq 1 - \delta$, being h_D^A the result of applying A to $\mathscr{H}$ and D.

This means that, fixing two parameters $\varepsilon, \delta \in (0,1)$, governing our precision, and a data generating distribution $p(x,y)$, we can find a number of samples $m_{\varepsilon,\delta}$ such that, with probability at least $1-\delta$, the empirical risk of the best hypothesis h_D^* is less than the risk of the best hypothesis h^* plus ε.

Finite hypothesis sets

An hypothesis set is said to be **finite** is $\mathscr{H}$ is s.t. $|\mathscr{H}| < \infty$.

Using combinatorial arguments, we can prove that finite hypothesis sets are agnostic PAC-learnable with:

$$m_{\varepsilon,\delta} \leq \lceil \frac{2\log(\frac{2\mathscr{H}}{\delta})}{\varepsilon^2} \rceil$$

hence with polynomial dependency on ε and δ. In this framework, $\log(|\mathscr{H}|)$ is a measure of the complexity of the set $\mathscr{H}$.

Warning:

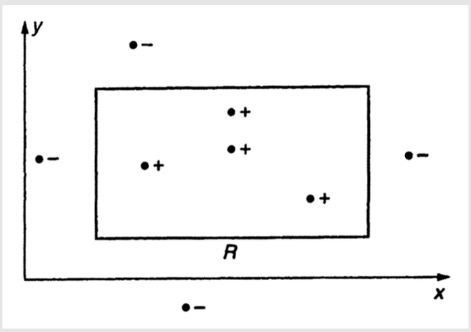
If $\mathscr{H}$ is described by d parameters of type double when represented in a computer (64 bits), it holds that $|\mathscr{H}| \leq 2^{d64}$, so we have a finite set of hypothesis, hence we can provide a bound on every implementable set of hypothesis functions.
In this case,

$$m_{\varepsilon,\delta} \leq \frac{128d + 2\lg(\frac{2}{\delta})}{\varepsilon^2}$$

i.e. we have linear dependency on the number of parameters.

Example: ***Pac learning example***

We consider an example introduced by Kearns and Vazirani in the book "An Introduction to Computational Learning Theory". [**kearns1994introduction**].
The goal is to learn a target axis-aligned rectangle R lying in $\mathbb{R}^2$ and $z \in -1, 1$ distributed according to a distribution $p(\mathbf{x}, z)$. The hypothesis set $\mathscr{H}$ is the set of all axis-aligned rectangles lying in $\mathbb{R}^2$. We assume there is a true rectangle of this king such that all positive points are inside it, and all negative points are outside.



We consider out hypothesis h to be the axis-aligned rectangle *R'* with the smallest area that includes all the positive examples and none of the negative ones.
What is the minimum number $m_{\varepsilon,\delta}$ of training examples so that, wiht probability at least $1-\delta$, h has an error at most ε with respect to the true rectangle and the distribution $p(\mathbf{x}, z)$?
Solution
We have to find an $m_{\varepsilon,\delta}$ such that $\forall m > m_{\varepsilon,\delta} P(error(h_m) > \varepsilon) \leq \delta$. First of all, let's notice that the error $error(h_m)$ is equal to the area of the target rectangle R minus the area of the internal rectangle $R' = h_m$ that we found.
Then, given an arbitrary error bound ε, we build within R 4 rectangles having each one an area of $\varepsilon/4$, on the sides of R.

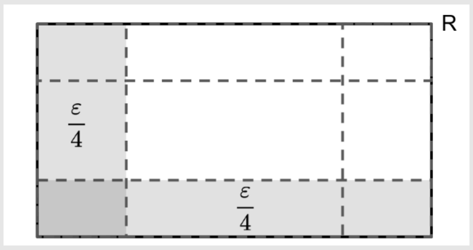


Figure 2.1: The 4 rectangles of area $\varepsilon/4$

Let's consider the m observations drawn from $p(\mathbf{x}, y)$. If each of the four rectangles defined above contains at least one point, we have $error(h_m) \leq \varepsilon$, because the area difference between R and R' would be fully covered by the 4 rectangles.

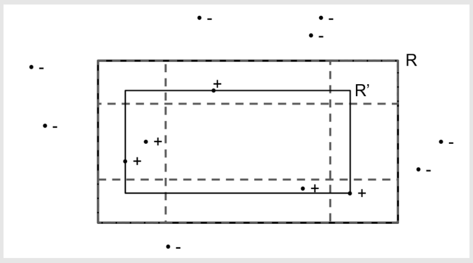


Figure 2.2: Configuration showing event

If we call this event B, we have:

$$B \rightarrow error(h_m) \leq \varepsilon$$

equivalently, by modus tollens:

$$error(h_m) > \varepsilon \rightarrow \neg B$$

This implies:

$$P(\neg B) \geq P(error(h_m) > \varepsilon)$$

$P(\neg B)$ is the probability that at least one of the 4 rectangles doesn't contain any of the m points. This probability is $(1-\varepsilon/4)^m$ for a single rectangle, hence we have $P(\neg B) \leq 4(1-\varepsilon/4)^m$. Thys the following chain of inequalities holds:

$$4(1-\varepsilon/4)^m \geq P(\neg B) \geq P(error(h_m) > \varepsilon)$$

Now, let $m_{\varepsilon,\delta}$ be such that:

$$\delta \geq 4(1-\varepsilon/4)^m_{\varepsilon,\delta} \geq P(\neg B) \geq P(error(h_m) > \varepsilon)$$

Finding $m_{\varepsilon,\delta}$ that satisfy this inequality would prove that $\forall m > m_{\varepsilon,\delta} P(error(h_m) > \varepsilon) \leq \delta$. We then require:

$$4(1-\varepsilon/4)^m_{\varepsilon,\delta} \leq \delta$$

and we use the inequality $(1-k) \leq e^{-k}$ to obtain:

$$m_{\varepsilon,\delta} \geq (4/\varepsilon) \cdot \ln(4/\delta)$$

In summary, provided a sample of at least $(4/\varepsilon) \cdot \ln(4/\delta)$ examples in order to choose an hypothesis rectangle R', we can assert that with probability at least $1 - delta, R'$ will misclassify a new point (drawn according to the same distribution from which the sample was chosen) with probability at most ε.
This proves that our hypothesis set $\mathscr{H}$ is PAC-learnable.

2.6 VC Dimension (Vapnik-Chervonenkis)

Consider a class of hypotheses functions $\mathscr{H} = h : X \to 0,1$ and a subset $C = c_1, \dots, c_m \subseteq X$ of input points.
Define $\mathscr{H}_c = (h(c_1), \dots, h(c_m))|h \in \mathscr{H}$, the set of all tuples of Booleans obtained by applying all possible hypothesis functions $h \in \mathscr{H}$ to all points in C. We say that $\mathscr{H}$ **shatters** the set C iff $|\mathscr{H}_C| = 2^m$.
Practically, this means that for any label assignment to points in C, we have a function in our hypothesis set which is able to match such an assignment. Namely, we can exactly describe every possible dataset with inputs in C.

Definition: ***VC Dimension***

The **VC dimension of** $\mathscr{H}$ is defined as:

$$VC(\mathscr{H}) = \max\{m \in \mathbb{N} | \mathscr{H} \text{ shatters some } C \subseteq X, |C| = m\}$$

Remark: In calculating the VC dimension, it is enough that we find one set of m points that can be shattered, it is not necessary that we are able to shatter any m points.

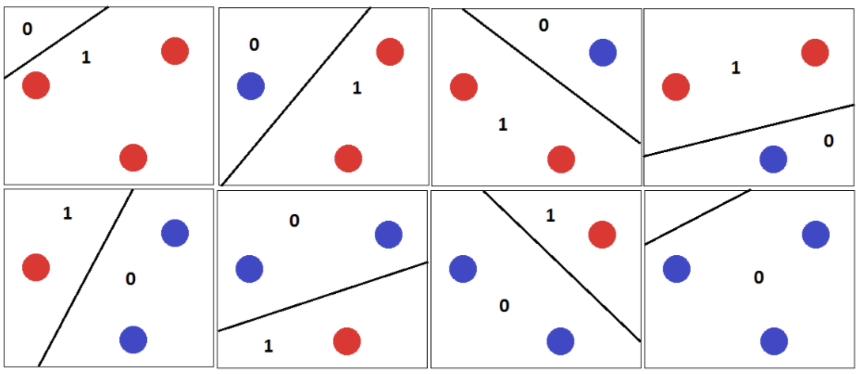


Figure 2.3: Proof that VCdim($\mathscr{H}_l \geq 3$)

2.6.1 VC dimension and PAC learning

In what follows, we will explore the reasons why VC dimension in crucial for PAC learnability.
Theorem 2. $If \mathscr{H} shatters C, |C| \geq 2m, then we cannot learn \mathscr{H} with m samples$
Hence, there will be an assignment of m samples to classes in which we are going to commit a large error.

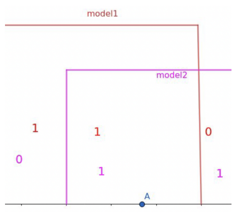


Figure 2.4: Visual interpretation of the theorem: it is impossible to train a model of type $\mathscr{H}_a+$ with only a point A with known classification, because the points different from A could have any classification.

If the $VCdim(\mathscr{H}) = \infty$, then $\mathscr{H}$ is not (agnostic) PAC learnable, indeed:

Theorem 3. $\mathscr{H} is (agnostic) PAC learnable iff VCdim(\mathscr{H}) < \infty$
In this case: $\exists c_1, c_2$ s.t.

$$c_1 \frac{VCdim(\mathscr{H}) + \log(\frac{1}{\delta})}{\varepsilon^2} \leq m_{\varepsilon,\delta} \leq c_2 \frac{VCdim(\mathscr{H}) + \log(\frac{1}{\delta})}{\varepsilon^2}$$

Hence VC dimension gives us control on what we can or cannot learn.

2.6.2 Rademacher Complexity

Consider the data generating distribution $p(x,y)$, a dataset $D\ p^m$ and an hypothesis class $\mathscr{H} = h : X \to -1, 1$.

Definition: ***Rademacher Distribution***

A distribution $\sigma = (\sigma_1, \dots, \sigma_m) s.t. \sigma_i \in -1, 1 \forall i and p(\sigma_i = 1) = 0.5$ is called **Rademacher Distribution**

Definition: ***Rademacher Complexity***

The **Rademacher Complexity** of $\mathscr{H}$ is defined as:

$$R_m(\mathscr{H}) = \mathbb{E}_\sigma \left[\sup_{h \in \mathscr{H}} \frac{1}{m} \sum_{i=1}^m \sigma_i h(x_i) \right]$$

Remark: this is a property both of the function h and of the dataset D.

Observation:

$\sum_{i=1}^m \sigma_i h(x_i) = \sigma \cdot h(\underline{x})$ is the scalar product of the Rademacher distribution $\sigma$with the function h evaluated on our dataset, so that $\frac{1}{m} \sigma \cdot h(\underline{x}) \in [-1, 1]$, essentially is a measure of correlation of h with random noise σ.

Hence, for a specific choice of noise σ, we are going to look at the dataset and choose the best h that correlates with this noise; then we take the expectation w.r.t. σ.

Definition: ***Rademacher Complexity***

Taking into account the data generating mechanism p, the **data-independent Rademacher complexity** is defined as:

$$\mathscr{R}_m(\mathscr{H}) = \mathbb{E}_{D\ p^m}[\hat{\mathscr{R}}_D(\mathscr{H})]$$

Fix $\mathscr{H} and p(x,y)$, then $\forall \sigma > 0$ with probability at least $1 - \delta, \forall D\ p^m, |D| = m, \forall h \in \mathscr{H}$ we have:

$$R(h) \leq \hat{(R_D)}(h) + \underbrace{\mathscr{R}_m(\mathscr{H}) + \sqrt{\frac{\log(\frac{1}{\delta})}{2m}}}_{\varepsilon_1}$$

$$R(h) \leq \hat{(R_D)}(h) + \hat{\mathscr{R}}_D(\mathscr{H}) + 3\sqrt{\frac{\log(\frac{2}{\delta})}{sm}}\ \varepsilon_2$$

Remark: computing the Rademacher complexity can be challenging, as it requires the solution of an optimization problem of any possible value of the Rademacher distribution.

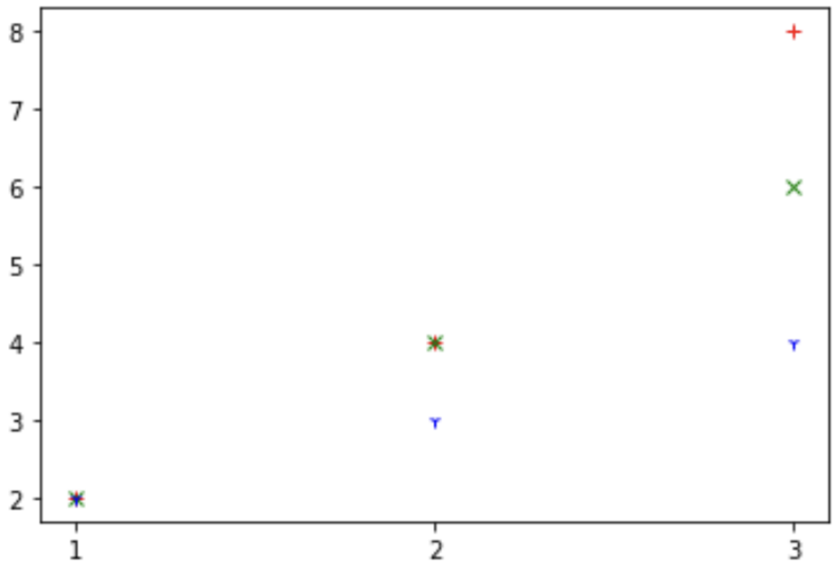


Figure 2.5: Plot of the growth functions of $\mathscr{H}_a$(blue), $\mathscr{H}_{a+}$(green), $\mathscr{H}_l$(red) for $1 \leq m \leq 3$

2.6.3 Rademacher complexity and VC dimension

Definition: ***Growth Function***

The **growth function** $\prod_{\mathscr{H}}$ is defined as:

$$\prod_{\mathscr{H}} : \mathbb{N} \to \mathbb{N}, \prod_{\mathscr{H}}(m) = \max_{x_1,\dots,x_m \in X} |\mathscr{H}_D|, |D| = m$$

with $\mathscr{H}_D = h(x_1), \dots, h(x_m) | h \in \mathscr{H}, D = x_1, \dots, x_m$

Hence the growth function describes how the complexity of what we can explain with out hypothesis set $\mathscr{H}$ increases with the cardinality of the data points that we have.
We can moreover define the $VCdim(\mathscr{H})$ in terms of the growth function:

$$VCdim(\mathscr{H}) = \max\{m \in \mathbb{N} | \prod_{\mathscr{H}}(m) = 2^m\}$$

Intuitively, this comes from the fact that if a set C is shattered by $\mathscr{H}$, then $\mathscr{H}_D = 2^m$.
Theorem 4 (Sauer Lemma)**.**

$$\prod_{\mathscr{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d \leq O(m^d)$$

with $d = VCdim(\mathscr{H})$
Moreover it also holds that:

$$\mathscr{R}_m(\mathscr{H}) \leq \sqrt{\frac{2\log(\frac{2}{\delta})}{m}} + \sqrt{\frac{2\log(\frac{2}{\delta})}{m}} \leq O(\sqrt{\frac{d}{m}})$$

We can say that $\mathscr{R}_m(\mathscr{H})$ and $VCdim(\mathscr{H})$ are essentially equivalent, in the sense that when the VC dimension is ∞ then the upper bound on the Redemarcher complexity is independent of m and greater than 1 (if you substitute it into the PAC bound using Redemacher complexity, you get an error which remains always large, no matter how large is m). Otherwise, when the VC dimension is $< \infty$, the bound tends to go to 0 as m grows to infinity.
Summarizing, we have ways to measure the complexity of our hypotheses space: if this complexity is finite (in the sense of VC dimensionality), then what we have as a result is that we can constrain

the error and provide bounds that tell us that this error is going towards 0, as we increase the data points (i.e. we will eventually learn). Instead if the VC dimension is infinite, no matter how many data points we have, we are not going to be able to learn, because the complexity of our model is too high, hence it can always overfit the data.
Hence, in order to be able to actually learn, we need to put some constraints in our hypothesis set, and this formalizes our inductive bias.

3 Probabilistic Graphical Models

3.1 Probabilitic Inference

Suppose having a set of random variates (either discrete or continuous) $x = (x_1,\dots,x_n), z = (z_1,\dots,z_m)$ with joint probability distribution $p(x,z)$.
To reason about such a model, we would perform:

- **Marginalization**: $p(x) = \sum_z p(x,z) = \int p(x,z)\, dz\, dx_{-i}$
- **Conditioning**: $p(x|z) = \frac{p(x,z)}{p(z)}$

These operations can be combined to perform **inference** on the model, i.e. to compute the probability of some variables given the values of others.

$$p(z_j|x_i = \bar{x}_i) = \frac{p(\bar{x}_i), z_j}{p(\bar{x}_i)} = \frac{\sum_{\bar{z}_j} p(\bar{x}_i, \bar{z}_j)}{p(\bar{x}_i)} = \frac{\int p(x,z)\, dx_{-i}\, dz_j}{\int P(x,z)\, dx_{-i}\, dz}$$

What is the cost of computation of performing these operations?
Well, focusing on discrete values for z such that $z_i \in 0,1$ and considering the marginalization over it:

$$p(x) = \sum_{z \in Z} p(x,z)$$

Here $z \in Z, with |Z| = 2^m$, resulting in an asympotically exponential cost in the number of variables.
Let's set up a strategy to perform inference and marginalizaton in a more efficient way.

Factorisation

Consider $p(x_1,\dots,x_n)$. Applying the laws of probability we can write

$$\begin{aligned} p(x_1,\dots,x_n) &= p(x_1)p(x_2|x_1)p(x_3|x_1,x_2)\dots p(x_n|x_1,\dots,x_{n-1}) \\ &= \prod_{i=1}^n p(x_i|x_1,\dots,x_{i-1}) \\ &= p(x_n|x_1,\dots,x_{n-1})p(x_{n-1}|x_1,\dots,x_{n-2})\dots p(x_1) \end{aligned}$$

What is then the cost in terms of computer memory to store these three representation of out distribution? Suppose $x_i \in 1,\dots,k$, we have the following costs:

- $p(x_1)$ costs $k-1 = O(k)$ floating point numbers (single or double precision)
- $p(x_2|x_1)$ costs $k(k-1) = O(k^2)$ float numbers (we have different distributions for x_2, each costing $k-1$ floats)
- In general $p(x_n|x_1,\dots,x_{n-1})$ costs $O(k^n)$ float numbers
- Factorization costs $O(k^n)$ float numbers, unfeasible

Probabilistic Graphical Models (PGM) are models of a probability distribution that describe/impose a certain factorization, hence allowing us to store and make inference effectively with joint distributions. There are three main kind of PGM:

- Bayesian Networks
- Markov Random Fields

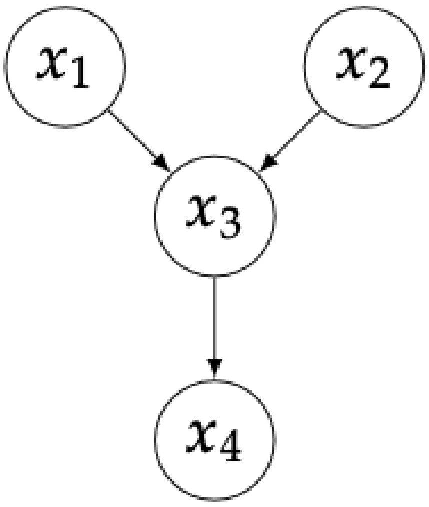
- Factor Graphs

3.2 Bayesian Networks

Definition: ***Bayesian Network***

A Bayesian Network is a directed acyclic graph (DAG) $G = (V, E)$ where:
- $V = \{x_1, \ldots, x_n\}$ is a set of random variables
- $E \subseteq V \times V$ is a set of directed edges

such that each node x_i is associated with a conditional probability distribution $p(x_i | pa(x_i))$ where $pa(x_i)$ is the set of parents of x_i in G.



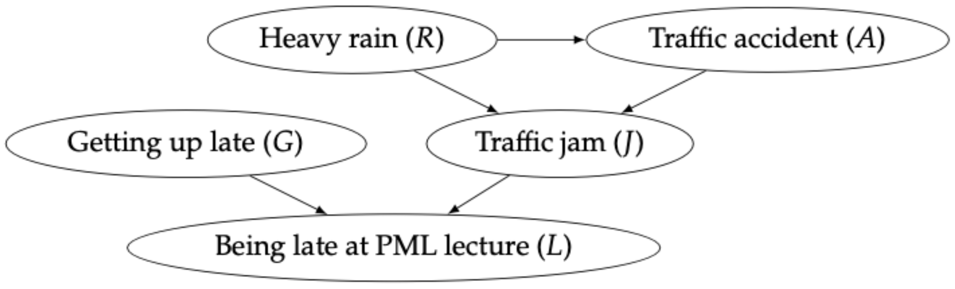


Figure 3.1: Example of a Bayesian Network

This BN corresponds to the following factorization:

$$p(L, G, J, R, A) = p(L|G, J)p(J|R, A)p(A|R)p(R)p(G)$$

Tip:

Directed edges does mean causality. BN **are not** a causal model.

Notational Conventions

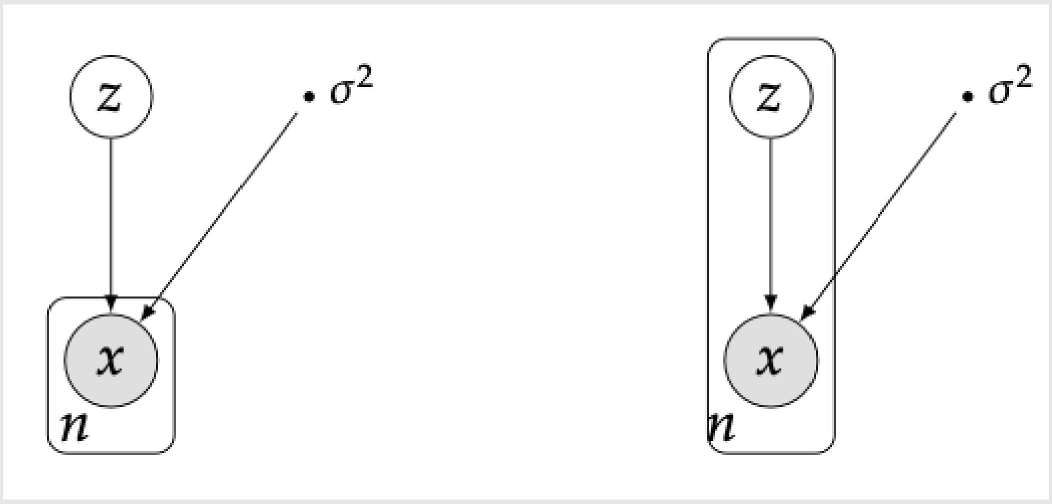
- Observed nodes (Random variables of which we know the value) are **coloured** or shadowed
- Plated nodes are a shorthand for a collection of nodes.
- Deterministic quantities represented as small solid circles. These are fixed parameters of the model (represented graphically as $\dot{\alpha}$)

Example: ***Mixture of Gaussians***

As an example suppose to have a RV z that follows a discrete distribution and another RV x that is normally distributed with mean $\mu(z)$, depending on the value of z, and variance σ^2. This model is known as a **mixture of gaussians**.
We can write the joint distribution as:

$$p(x,z) = p(z)p(x|z) = p(z)\mathcal{N}(x|\mu(z),\sigma^2)$$

with $p(x|z) = \mathcal{N}(x|\mu_z,\sigma^2)$
Typically, in this scenario we have at our disposal n observations of the variable x, $\bar{x} = x_1,\dots,x_n$, while the corresponding variable z is unobserved (hence z is a **latent variable**). We typically want to compute $p(z|\bar{x})$.
We can have two scenarios here: in the first $\bar{x}$ are sampled from a single realization of the variable z, while in the second one each x_i may have been drawn with a different z_i, hence we are interested in computing $p(z_i!x_i)$:



3.2.1 Sampling and reasoning in BN

Recalling the initial example

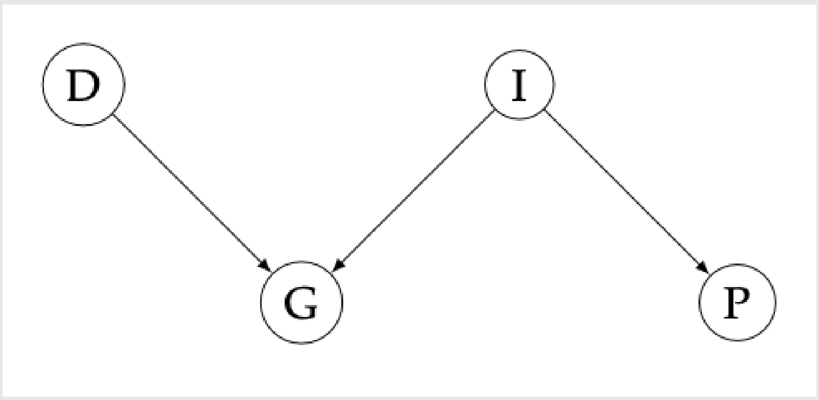
$$p(x_1,x_2,x_3,x_4) = p(x_4|x_3)p(x_3|x_2,x_1)p(x_2)p(x_1)$$

$$\begin{tikzcd} x_1 \arrow[dr] & & x_2 \arrow[dl] \\ & x_3 \arrow[d] & \\ & x_4 & \end{tikzcd}$$

Ancestral sampling is a sampling technique that allows us to sample from a given distribution, if we know its factorization.
One can start to sample from the top of the network (our **ancestrals**), in the example x_1 and x_2, then move to their children and sample, in the example, x_3 from $p(x_3|x_2,x_1)$ using the values for x_1 and x_2 that have already been sampled, and so on. It is a simple and effective way to sample probability distribution, provided it is easy to sample from wach marginal and conditional.

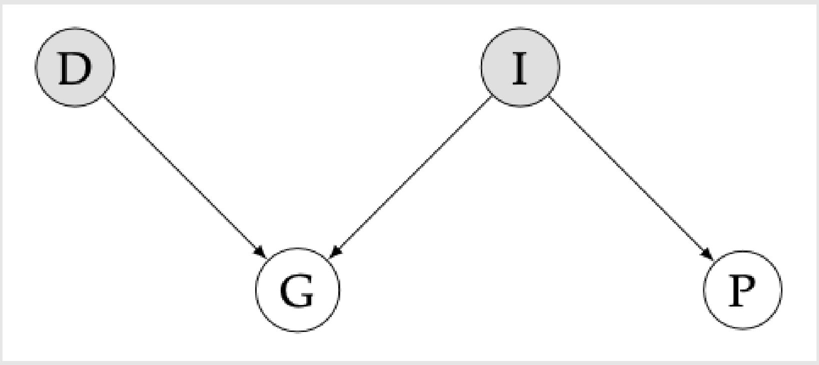
Example: ***Reasoning with BN***

We consider here a model of students' grade in an exam.

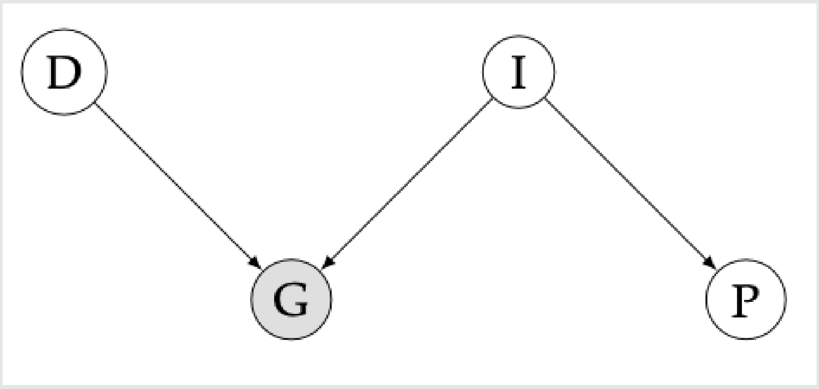


The di culty of the exam (D) and the intelligence of the student (I) are independent, but the grade you get when you take an exam (G) is dependent on both (studying of course helps too). Having passed a hard exam (P) likely depends on the student's abilities but not on the di culty and the grade of the current exam. We have three di erent forms of reasoning here:

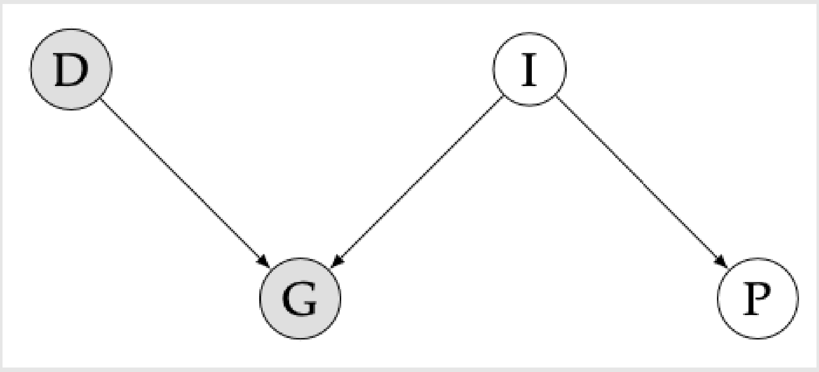
- **Causal Reasoning**: We observe something (the di culty of the exams, how intelligent we are) and we infer the grade and if we have passed a hard exam. The reasoning goes from top to bottom.



- **Evidential Reasoning**: If we instead observe the grade, we might want to get information regarding the di culty of the exams and how intelligent the student is. The reasoning goes from bottom to top.



- **Intercausal Reasoning**: If we observe the grade and the di□culty of the exams, we might want to infer the intelligence of the student. The reasoning goes from the middle to the top.



Bayesian networks of course can get very large and probabilistic inference becomes a complicated and important task to perform in the real world.
Remark: note that dependency in a BN can model both causality and correlation, and the two are not necessarily distinguishable within the model.

3.2.2 Conditional Independence in BN

Definition: ***Conditional Independence***

Consider the RVs a, b, c. We say that a is **conditional independent** of b given c (written $a \perp\!\!\!\perp b|c$) if:

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

or equivalently:

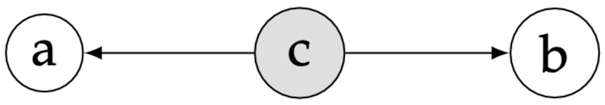
$$p(a|b,c) = p(a|c)$$

We have three scenarios to consider in Bayesian Networks to understand conditional independence.

Tail to Tail

Consider the RVs a, b, c. We say that a and b are tail to tail with c if:

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



We know that this Bayesian network implies the following factorization:

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

Also

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

Which means that $a \perp\!\!\!\perp b \not\perp\!\!\!\perp b|c$.

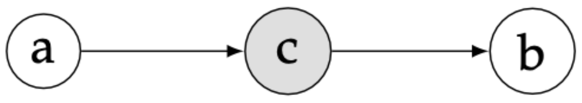
Head to Tail

Consider the RVs a, b, c. We say that a and b are head to tail with c if:

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

or

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



Again, we can use this factorization and study independence of our variables:

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

Also

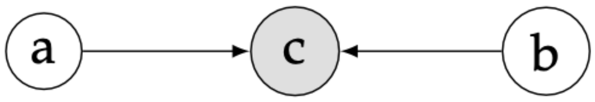
$$p(a,b|c) = \frac{p(b|c)p(c|a)p(a)}{p(c)} = p(b|c)p(a|c)$$

Which means, again, that $a \perp\!\!\!\perp b \not\perp\!\!\!\perp b|c$.

Head to Head

Consider the RVs a,b,c. We say that a and b are head to head with c if:

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



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

Also

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

Which means, this time, that $a \perp\!\!\!\perp b and a \not\perp\!\!\!\perp b|c$.
Notice that also $a \perp\!\!\!\perp b|d, \forall d$ descendant of c.

Formalization

Formalizing the notion of conditional independence on Bayesian Network that we just saw,

Definition:

Given RVs a,b,c a path from a to b is **blocked** by c if:

- c is observed and the path is head-to-tail or tail-to-tail in c.
- c is not observed, nor any descendant of c, and the path is head-to-head in c.

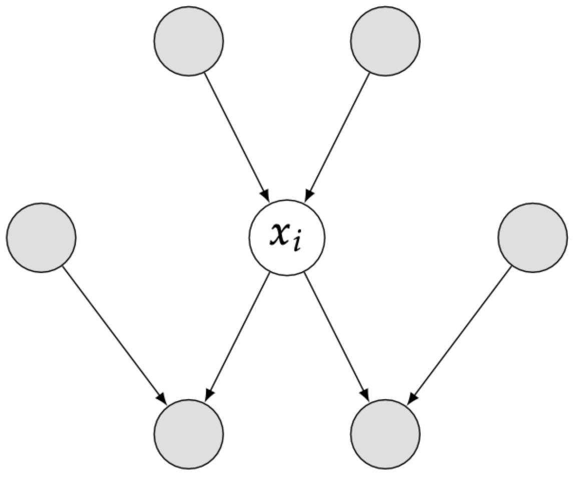
Proposition: Let A,B,C subsets, if all paths from a node in A to a node in B are blocked by a node in C then $A \perp\!\!\!\perp B|C$.

Markov Blanket

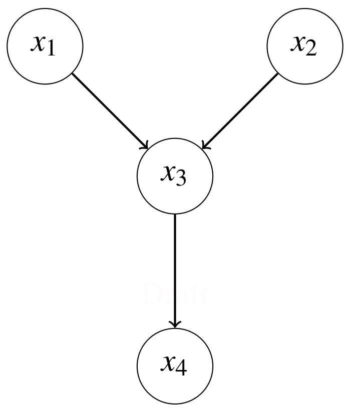
Consider a node x_i on the network and condition on everything else, i.e. on $x_{-i} = \{x_1,\dots,x_{i-1},x_{i+1},\dots,x_n\}$. Which nodes will remain in the conditioning set? If we make the computation

$$p(x_i|x_{-i}) = \frac{p(x_1,\dots,x_n)}{p(x_1,\dots,x_{i-1},x_{i+1},\dots,x_n)} = \frac{\prod_j p(x_j|pa_j)}{\sum_{x_i}\prod_j p(x_j|pa_j)}$$

In the denominator, each term in which x_i does not appear either as x_j or in $+pa_j$ will get out of the summation and cancel with the respective term in the numerator. So the only nodes that remain are thos belonging to pa_i, or those for which $x_i \in pa_j$ is known as the **Markov Blanket** of x_i (it contains parents, children and co-parents of x_i). Each node conditioned on its Markov blanket is independent of the rest of the network.



3.2.3 Naive Bayes

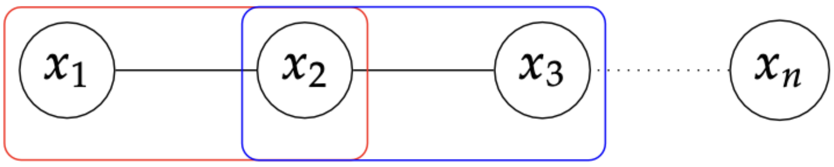


4 Exact Inference in Probabilistic Graphical Models

In the context of inference, out task is, given a joint distribution $p(X) = p(x_1,\dots,x_n)$ to compute marginals or conditionals of the distribution.
We formalize this by considering two disjoint subsets of r.v.: $y \subseteq x, z \subseteq x$ s.t. $y \cap z = \emptyset$ and $y \cup z \subseteq x$.
Given that we observe y, we want to compute the conditional

$$\begin{aligned} p(z|y=\bar{y}) &= \sum_x p(z,x'|y=\bar{y}) discreter.v. \\ &= p(z|y=\bar{y} = \int_x p(z,x'|y=\bar{y})) continuousr.v. \end{aligned}$$

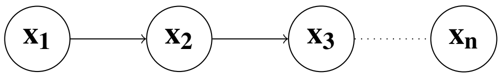
being x' s.t. $x' \cup y \cup z = x$.
Remark: this summation can be of the order of $O(\mathscr{K}^m)(|x'| = m)$, hence obtaining this marginalization is computationally challenging!
Our goal is to carry out this computation efficiently, by leverage the factorization implied by the PGM.
As an example, consider a Markov Random Field where variables are connected in a chain (colored rectangles represent the cliques):



The factorization implied by this PGM is:

$$p(x) = \frac{1}{Z} \cdot \psi_{1,2}(x_1,x_2) \cdot \psi_{2,3}(x_2,x_3) \dots \psi_{x_{n-1},x_n}(x_{n-1},x_n)$$

Consider now the equivalent model in case of a Bayesian Network:



The factorization in this case reads as

$$p(x) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_2) \dots p(x_n|x_{n-1})$$

Observation: *Markov processes*

Chain structures like the previous are common because they repre- sent temporal series (more specifically, they are Markov processes).

Suppose that, given the model above, we want to compute the marginal

$$p(x_k) = \sum_{x_1,\dots,x_{k-1},x_{k+1},x_n} p(x_1,\dots,x_n)$$

with $1 < k < n$. In principle this has complexity $O(\mathscr{K}^{n-1})$.
However, plugging the factorization implied by the Markov Random Field, we get(using the distributive laws of sum and product):

$$\begin{aligned} p(x_k) &= \textstyle\sum_{x\_k} \frac{1}{Z} \psi_{1,2}(x_1,x_2) \dots \psi_{n-1,n}(x_{n-1},x_n) \\ &= \tfrac{1}{Z} \textstyle\sum_{x_{k-1}} \psi_{k-1,k}(x_{k-1},x_k) \dots \left[\sum_{x_2} \psi_{2,3}(x_2,x_3) \left[\sum_{x_1} \psi(x_1,x_2)\right]\right] + \\ &+ \textstyle\sum_{x_{k+1}} \psi_{k,k+1}(x_k,x_{k+1}) \dots \left[\sum_{x_{n-1}} \psi_{n-2,n-1}(x_{n-2},x_{n-1}) \left[\sum_{x_n} \psi_{n-1,n}(x_{n-1,}x_n)\right]\right] \end{aligned}$$

Note that complexity is now $O(n \cdot k)$, i.e. linear in n.
We can define a dynamic programming algorithm, called **message-passing algorithm**, in which the information from the graph is summarized by local edge information.
We break the chain in two parts, the past and the future w.r.t. x_k, and we define the following:

- forward message: $\mu_\alpha(x_k) = \sum_{x_{k-1}} \psi_{k-1,k}(x_{k-1},x_k) \cdot \mu_\alpha(x_{k-1})$, with base case $\mu_\alpha(x_1) = 1$
- backward message: $\mu_\beta(x_k) = \sum_{x_{k+1}} \psi_{k,k-1}(x_k,x_{k+1}) \cdot \mu_\beta(x_{k+1})$, with base case $\mu_\beta(x_n) = 1$

In this algorithm, each node sends a message to its neighbors, and the messages are computed by the above formulas.

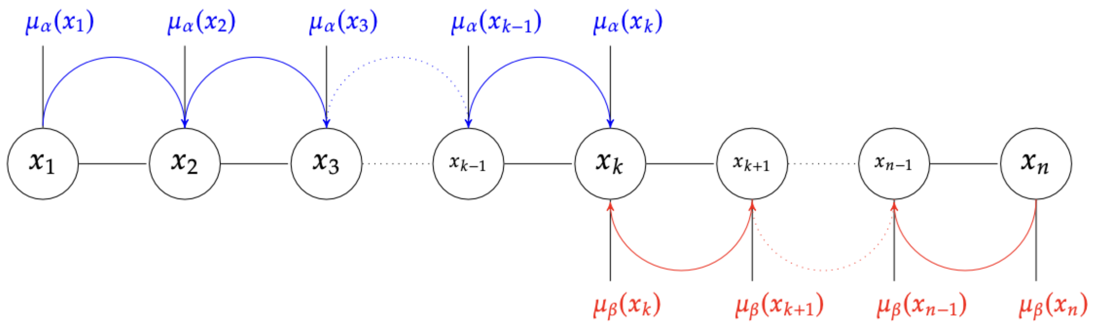


Figure 4.1: Message-passing algorithm

Once we have both $\mu_\alpha(x_k)$ and $\mu_\beta(x_k)$, we can observe that:

$$p(x_k) = \frac{1}{Z_k} \cdot \mu_\alpha(x_k)\mu_\beta(x_k) \propto \mu_\alpha(x_k)\mu_\beta(x_k)$$

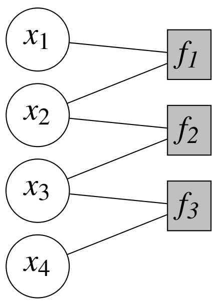
with the normalization constant Z_k computed as $Z_k = \sum_{x_k} \mu_\alpha(x_k)\mu_\beta(x_k)$.
Summarizing, the idea behind this algorithm is to start from the extremes of the chain and pass messages (forward and backward) until the other end is reached. Once there, messages are combined to obtain an (un)normalized marginal distribution.

4.1 Factor Graphs

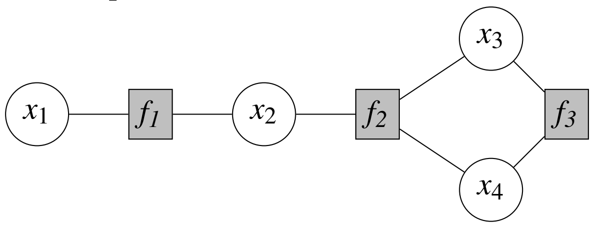
Our goal in what follows is to extend what we have done for chains to more general graph structures, i.e. trees and politrees. In order to do so, we need to introduce the formalism of **Factor Graphs**.
The idea behind them is to expose in a more clear way what are the factors and what are the variables. For this reason, Factor Graphs are bipartite graphs (i.e. nodes are divided in two classes), in which we have:

Variable nodes	**Factor nodes**
x	f

The canonical way of represent bipartite graphs is the following:

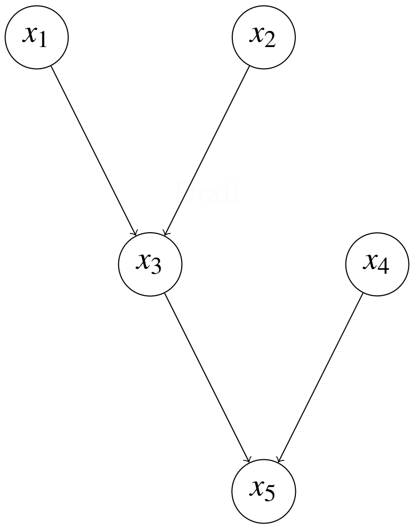


That is, all variable nodes are put on the left and all factor nodes on the right.
A more convenient, equivalent, representation could be:



4.1.1 From Bayesian Networks to Factor Graphs

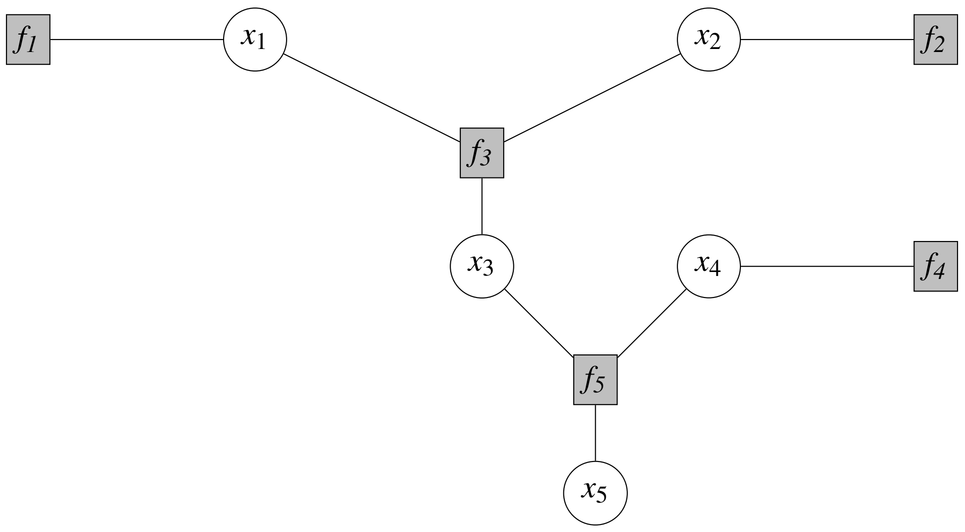
Consider the following Bayesian Network:



Which corresponds to the factorization:

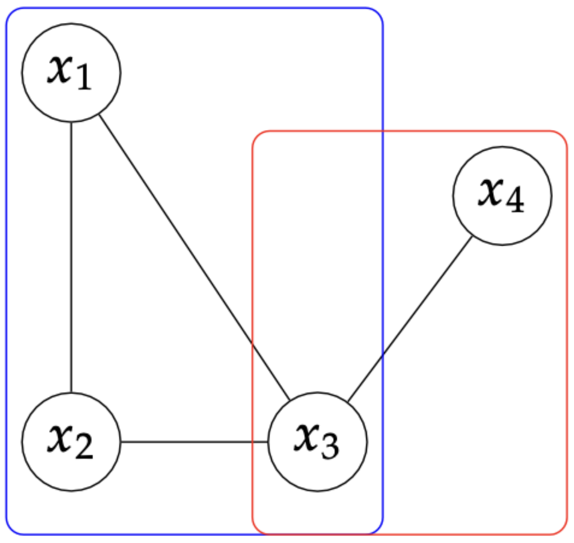
$$\begin{aligned} p(x) &= p(x_1)p(x_2)p(x_3|x_1,x_2)p(x_4)p(x_5|x_3,x_4) \\ &= f_1(x_1)f_2(x_2)f_3(x_1,x_2,x_3)f_4(x_4)f_5(x_3,x_4,x_5) \end{aligned}$$

Fundamentally, the idea is to have a factor node (connected to the proper variable nodes) for each term of the factorization implied by the Bayesian Network.
Formally: $\forall p(x_j|pa_j) \to f_j \curvearrowright x_j \cup pa_j$.
Hence the Factor Graph equivalent to the Bayesian Network above is:



4.1.2 From Markov Random Fields to Factor Graphs

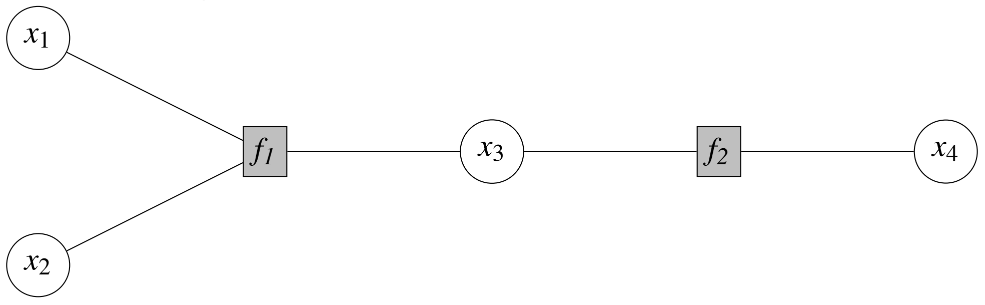
Consider the following Markov Random Field (rectangles correspond to the cliques):



Which implies the factorization:

$$p(x) = \frac{1}{Z}\psi_1(x_1,x_2,x_3)\psi_2(x_3,x_4) = \frac{1}{Z}f_1(x_1,x_2,x_3)f_2(x_3,x_4)$$

This leads to the following Factor Graph:

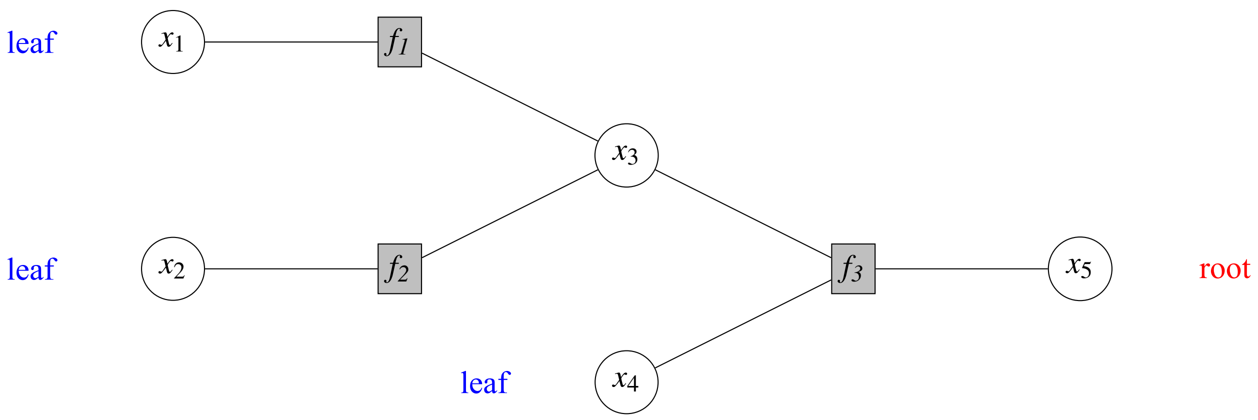


Hence, the conversion from Markov Random Fields to Factor Graphs works as: $\forall c \in \mathscr{C} \to f_c(x_c) \sim \psi_c(x_c)$ (equality does not hold because of the normalization constant).

So, whether we are starting from a Bayesian Network or from a Markov Random Field, we can convert our PGM into a Factor Graph and perform message passing on it, without loss of generality.

4.2 Sum Product Algotithm

Our goal is now to do inference in Factor Graphs, generalizing to more general graph structures what we did with chains previously. Consider the following Factor Graph:

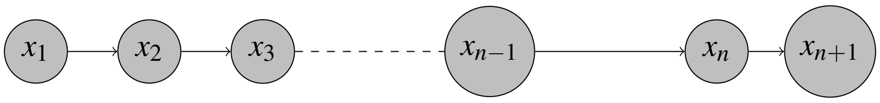


The joint probabbility distribution implied by thie Factor Graph is:

$$p(x) = f_1(x_1,x_3)f_2(x_2,x_3)f_3(x_3,x_4,x_5)$$

5 Hidden Markov Models

In what follows our goal is to model **time series data**. This kind of data are observed from a process evolving in time, typically at different time steps $x_1, x_2, \dots, x_N$ (i.e. assuming a discrete model of time). Since sequential data often arise through measurement of time series, there is correlation between observations at different time steps.
These data can come from very different domains: financial data, weather forecast data, speech data, epidemiological data.
Markov Chains are natural models for sequential data, the following is a Markov Chain of order 1:



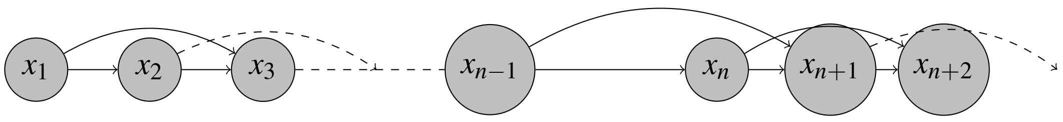
Since all he points (up to a certain step N) are observed, the factorization by the model is:

$$p(x_1, x_2, \dots, x_N) = p(x_1)p(x_2|x_1)p(x_3|x_2)\dots p(x_N|x_{N-1})$$

It holds that future observations are independent of all but the most recent observation:

$$x_{n+1} \perp x_1, x_2, \dots, x_n | x_n$$

A **time homogeneus** process is a process whose transiiton probability does not change in time, i.e. such that $p(x_1|x_{n-q}) = p(x_2|x_1)$.
These chains are not always the best model for describing sequential observations, indeed often there is a deeper dependency on the past, and first-order Markov chains suffer from too short memory.
In these cases, we can mode to **Markov models of order k**, where the dependency of x_n is on the previous k steps. The following is a second order Markov Chain:



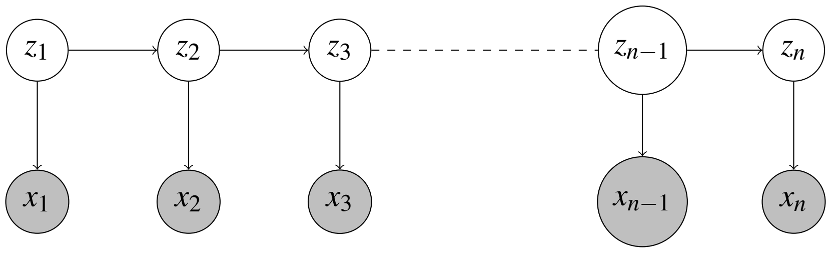
In this case, the factorization of the model is:

$$p(x_1, x_2, \dots, x_N) = p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)\dots p(x_{n+1}|x_n, x_{n-1})\dots$$

And it holds that:

$$x_{n+2} \perp x_{n-1} | x_n, x_{n+1}$$

Remark: if x_i are discrete, we talk about *Markov chains*; if x_i are continuous and $p(x_n|\dots)$ are Gaussian, we talk about *autoregressive models* (of order k).
If we want to build a model for sequential data that is not limited to the Markov assumption of any order, we can rely on **state space models**, which introduce **latent variables**. In terms of graphical model, we have that latent variables from a Markov chain, and each of them corresponds to an observation:



If we consider two observations x_i,x_j, then $x_i \not\perp x_j | \underline{x}$ (with $\underline{x} = x_1, \dots, x_n$) since there is not any obserbed node in the path from x_i to x_j.
In order to describe the space models like the one above, we need:

- **transition probabilities** $p(z_n|z_{n-1})$, which describe the evolution of the latent variables. They can be arranged in a matrix A s.t. $A_{ij} = p(z_n = j|z_{n-1} = i)$;
- **initial distribution** $p(z_1)$, specified by a vector π s.t. $\pi_i = p(z_1 = i)$,
- **emission probabilities** $p(x_n|z_n)$, parametrized by ψ. The emission probability for discrete x is a categorical distribution, for continuous x is typically a Gaussian or a mixture of Gaussians.

Definition: ***Hidden Markov Model***

The **Hidden Markov Model** is a specific instance of the state space model described before, in which the latent variables z_i are discrete. If instead the variables z_i are continuous and the transition probabilities $p(z_i|z_{i-1})$ are Gaussian, we talk about **Linear Dynamical System**.

Consider the parameters $\Theta = (A, \pi, \psi)$ and the variables

$$\underline{x}, \underline{z} = (x_1, \dots, x_n, z_1, \dots, z_n)$$

then

$$p(\underline{x}, \underline{z}|\Theta) = p(z_1|\pi) \left[\prod_{n=2}^{N} p(z_n|z_{n-1}, A) \right] \left[\prod_{n=1}^{N} p(x_n|z_n, \psi) \right]$$

Example:

We can graphically represent the states of $\underline{z}$ and the transition matrix as follows (note that this is not a PGM):