

Estimation and Probabilistic Learning

Farzad Farnoud

University of Virginia
Aug. 2025

Contents (Summary)

0	Review of Probability	6
1	Probability, Inference, and Learning	23
2	Frequentist Parameter Estimation	31
3	Bayesian Parameter Estimation	42
4	Multivariate Random Variables	53
5	Linear Regression	56
6	Linear Classification	72
7	Expectation-Maximization *	82
8	Basics of Graphical Models	91
9	Independence in Graphical Models	98
10	Parameter Estimation in Graphical Models	105
11	Inference in Graphical Models	112
12	Inference in Hidden Markov Models	123
13	Factor Graphs and Sum/Max-product Algorithms **	131
14	Markov Chains	141
15	Sampling Methods	149
16	Variational Inference *	161
17	Appendix	172

Contents

0	Review of Probability	6
0.1	What is probability?	6
0.1.1	Definitions:	6
0.1.2	Axioms:	6
0.1.3	Interpretations of probability	7
0.2	Sets and their sizes	8
0.3	Random variables and distributions	8
0.3.1	Discrete distributions	9
0.3.2	Continuous distributions	9
0.3.3	Cumulative distribution functions	10
0.3.4	Expected value	11
0.3.5	Common distributions	13
0.4	Joint probability distributions	13
0.4.1	Expectation, correlation, and covariance	14
0.4.2	Independence	16
0.4.3	Conditional probability and conditional distributions	17
0.4.4	Bayes' rule	18
0.5	Inequalities and limits	19
0.5.1	Inequalities	19
0.5.2	Limits	19
0.6	Random vectors	21
0.6.1	Properties of expectation and covariance	21
1	Probability, Inference, and Learning	23
1.1	Introduction	23
1.2	Relationships and joint probability distributions	23
1.3	Inference and decision making	24
1.4	Machine Learning and Probability	26
1.4.1	Empirical Risk Minimization (ERM)	27
1.4.2	Density estimation	27
1.5	Information theory and machine learning	28
1.5.1	Quantifying uncertainty	28
1.5.2	Relative entropy	29
1.5.3	Conditional entropy and mutual entropy*	30
2	Frequentist Parameter Estimation	31
2.1	Overview	31
2.2	Maximum likelihood estimation	31
2.2.1	Maximum likelihood and the closest distribution	34
2.3	Properties of Estimators	34
2.3.1	Bias	35
2.3.2	Mean squared error and variance	37

2.3.3	Consistency	39
2.4	The Cramer-Rao lower bound*	40
2.5	Asymptotic normality of the MLE	41
3	Bayesian Parameter Estimation	42
3.1	From Prior to Posterior	42
3.2	Bayesian Point Estimates	46
3.3	Posterior Predictive Distribution	47
3.4	Gaussian Prior and Likelihood	48
3.5	Conjugate Priors	50
3.6	The Exponential Family (EF)	51
4	Multivariate Random Variables	53
4.1	Gaussian Random Vectors (Multivariate Normal Distribution)	53
4.1.1	Maximum likelihood estimation	53
4.1.2	Bayesian estimation	54
5	Linear Regression	56
5.1	Introduction	56
5.2	Least-squares	57
5.3	Probabilistic Models for Regression	61
5.3.1	General model	61
5.3.2	Gaussian model	61
5.4	Regularized Linear Regression	64
5.5	Error analysis and model selection	66
5.5.1	Bias-variance trade-off for quadratic error	66
5.5.2	Model Selection	67
5.6	Stochastic Gradient Descent	70
6	Linear Classification	72
6.1	Overview of probabilistic models	72
6.2	Generative Probabilistic Models	73
6.2.1	Gaussian Class-Conditionals	73
6.2.2	Linear Discriminant Analysis	73
6.2.3	Quadratic Discriminant Analysis	75
6.2.4	Maximum Likelihood Solution to LDA	75
6.2.5	Generative Model for Discrete Features **	76
6.2.6	Class-conditionals from the exponential family	77
6.3	Discriminative Models and Logistic Regression	77
6.4	Risk minimization and loss functions for classification	78
6.4.1	Zero-one loss	78
6.4.2	Logistic regression	79
6.4.3	Hinge loss (SVM)	79
7	Expectation-Maximization *	82
7.1	Overview	82
7.2	Clustering with EM	85
7.3	EM with general missing data **	87
7.4	The MM Algorithm **	89
7.4.1	Rank Aggregation from Pairwise Comparisons via MM	89
8	Basics of Graphical Models	91
8.1	Introduction	91
8.2	Bayesian Networks	91
8.2.1	Markov Model	94

8.2.2	Why graphical models?	94
8.3	Markov Random Fields	95
8.3.1	Energy-based models	96
8.4	Moralization: Converting BNs to MRFs	97
9	Independence in Graphical Models	98
9.1	Independence for sets of random variables	98
9.2	Independence in Bayesian Networks	99
9.2.1	Simple Bayesian networks	99
9.2.2	d-separation	100
9.2.3	Markov Blanket in Bayesian Networks	102
9.3	Independence in MRFs	102
10	Parameter Estimation in Graphical Models	105
10.1	MLE for Parameters of Bayesian Networks	105
10.2	Bayesian Parameter Estimation for Bayesian Networks	106
10.3	Parameter Estimation in MRFs	111
11	Inference in Graphical Models	112
11.1	Introduction	112
11.2	The Elimination Algorithm	113
11.3	The Sum-Product Algorithm	114
11.4	The Max-Product Algorithm	117
11.5	Sum-product Example	118
12	Inference in Hidden Markov Models	123
13	Factor Graphs and Sum/Max-product Algorithms **	131
14	Markov Chains	141
14.1	Introduction	141
14.2	State distribution as a function of time	143
14.3	Long-term Behavior of Markov Chains	144
14.3.1	How often does the Markov Chain visit each state?	145
14.4	Balance Properties and Finding the Stationary Distribution	146
14.4.1	Detailed Balance	146
14.4.2	Time-Reversibility **	146
14.4.3	Global Balance **	147
15	Sampling Methods	149
15.1	Introduction	149
15.2	Basic Sampling Techniques	150
15.2.1	Deterministic Integration	150
15.2.2	Rejection Sampling	150
15.2.3	Importance Sampling	152
15.3	Metropolis Monte Carlo	153
15.4	Gibbs Sampling	157
15.5	Hamiltonian Monte Carlo **	158
16	Variational Inference *	161
16.1	Background on Calculus of Variations	162
16.2	Mean-field variational inference	165
16.3	Examples	166
16.3.1	CAVI on a MRF for image denoising	166
16.3.2	Bayesian estimation of a univariate Gaussian [3]	167

16.4 Factorized variational approximations are compact	169
17 Appendix	172
17.1 Review of Linear Algebra	172
17.2 Vector and matrix differentiation	174

I would like to thank the students who have taken the course at UVA for their feedback, and my GTAs, especially Hao Lou, for their contributions to this text.

Farzad Farnoud

Chapter 0

Review of Probability

In this chapter, we will review some concepts from probability theory and linear algebra that will be useful in the rest of the course.

This review is not comprehensive. You can refer to the [course webpage](#) for more resources.

0.1 What is probability?

Intuitively, probability is a way of systematically studying events whose outcomes are uncertain. It enables us to quantify information and uncertainty (e.g., the probability of rolling a 6 is $1/6$ or the probability of rain on grounds at 10 am tomorrow is 20%). It can be used to describe relationships and provides ways to transfer our knowledge about one random quantity to another.

From a mathematical point of view, probability deals with sets, and functions that assign real values to those sets, in a way that certain axioms are satisfied. In this sense, probability is similar to geometry, number theory, etc. It can be used to model the real world, but it can also be studied as an abstract subject.

0.1.1 Definitions:

Assuming an experiment with different possible outcomes, consider the following definitions.¹

- Ω : the sample space, the set of all possibilities (*outcomes*)
- $E \subseteq \Omega$: an event, i.e., a set of outcomes
- \Pr : A function from subsets of Ω to \mathbb{R} . $\Pr(E)$ is the probability of the event E .

0.1.2 Axioms:

- $\Pr(E) \geq 0$ for all $E \subseteq \Omega$.
- $\Pr(\Omega) = 1$
- $\Pr(E_1 \cup E_2) = \Pr(E_1) + \Pr(E_2)$ if $E_1 \cap E_2 = \emptyset$.

Based on these axioms, many theorems and other results can be proven. For $A, B \subseteq \Omega$:

- If $A \subseteq B$, then $\Pr(A) \leq \Pr(B)$.
- $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$.

More definitions for basic concepts:

¹These definitions and the following axioms are simplified. We cannot always assign probability to all subsets of Ω . Also, for the third axiom, for any **countable** sequence of mutually exclusive events E_1, E_2, \dots , we require that $\Pr(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \Pr(E_i)$.

- Two events A and B are *independent*, denoted $A \perp B$, if $\Pr(A \cap B) = \Pr(A) \Pr(B)$.
- If $\Pr(B) \neq 0$, the *conditional probability* of A given B is defined as

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)}.$$

- Random variables, distributions, expected value, ...

What these theorems and definitions ‘mean’ depends on what we think probability means.

0.1.3 Interpretations of probability

Probability is the most important concept in modern science, especially as nobody has the slightest notion what it means.

Bertrand Russell

How do we assign probability to events? What does it mean, for example, to say that $\Pr(E) = 1/3$?

- Classical interpretation: If there are K possible outcomes, and we have no reason for some outcomes to be more likely than others, the probability of each outcome is $1/K$.
 - Probability of rolling a 3 is $1/6$.
 - Probability of heads is $1/2$ when tossing a fair coin.
- Frequentist interpretation: Assume that there is a “random” experiment that can be repeated many times. If we repeat it N times and N is very large, then the number of times that the event E occurs is approximately $N \Pr(E)$. In other words, the “frequency” of E occurring is $\Pr(E)$.
 - Probability of heads for a given coin is $\Pr(H) = 1/3$. So if we toss it 3000 times, we should see heads around 1000 times.
 - Probability **distribution** of the number N of children (≤ 18) of a randomly chosen American household:

	$\Pr(N = 0)$	$\Pr(N = 1)$	$\Pr(N = 2)$	$\Pr(N \geq 3)$
1970	0.442	0.182	0.174	0.203
2008	0.541	0.195	0.169	0.095

- Bayesian interpretation: probability indicates the degree of belief in a way that is consistent with the axioms. This allows us to consider events that are, strictly-speaking, not random.
 - $\Pr(\text{Heads}) = 1/2$ (both Bayesian and frequentist)
 - $\Pr(\text{Stock market will hit a certain threshold this year})$
 - $\Pr(\text{Nuclear war this century})$
 - $\Pr(\text{A certain person is guilty of a given crime})$

The classical interpretation is sometimes criticized as being circular. We call a coin fair if $\Pr(H) = \Pr(T) = 1/2$ and we say $\Pr(H) = 1/2$ if the coin is fair. Nevertheless, the definition is relied upon in practice, e.g., in games of chance. The frequentist definition can be criticized for being vague. What do “large” and “approximately” mean? How large is large enough? And how close should two values be for us to call them approximately equal? The Bayesian interpretation is criticized for being subjective and for assigning probabilities to experiments that happen only once (so any given event either happens or does not happen).

Criticism of interpretations of probability does not create any mathematical problems. Mathematically, we only need to assign probabilities in a way that the axioms are satisfied. Different interpretations however lead to different approaches to problems, potentially leading to different real-world decisions.

0.2 Sets and their sizes

Finding the probability of an event is easiest when all outcomes are equally likely. In such cases, if we can measure the size of the set A of desirable outcomes, dividing that by the size of the sample space, will yield the probability,

$$\Pr(A) = \frac{|A|}{|\Omega|},$$

where $|A|$ denotes the size of the set A .

Definition 0.1. A set A is **finite** if there is a natural number n such that the number of elements in A is less than n . Otherwise, it is **infinite**. If the elements of A can be counted, i.e., there is a one-to-one function from A to natural numbers, then A is **countable**. Otherwise, it is **uncountable**. A countable set may be finite (e.g., $\{1, 5, 6\}$) or infinite (e.g., integers, prime numbers, rational numbers).

If A is finite, we define its size (aka, cardinality) as the number of elements. This requires us to be able to count:

- **Sum rule:** If an action can be performed in m ways and another action can be performed in n ways, and further if we can choose which action to perform, in total we have $m + n$ options.
- **Product rule:** If the first action can be performed in m ways and the second action can be performed in n ways, and further if we must perform both actions in order, in total we have $m \times n$ options.
- **Permutations:** The number of ways we can arrange n objects is $n! = 1 \times 2 \times \cdots \times n$.
- **Combinations:** The number of ways we can choose k objects from a set of n objects is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Exercise 0.2. †² Prove that $\binom{n}{x}x = n\binom{n-1}{x-1}$. △

Exercise 0.3. How many 8-bit bytes are there? How many of these have exactly 3 ones? If we pick a random byte, what is the probability that it has exactly 3 ones (binomial distribution)? What is the probability that it has 6 or more consecutive ones? △

Exercise 0.4. How many binary sequences of length n that end with one are there with exactly k ones? △

If the sample space has an infinite, even uncountable, number of outcomes, we may still be able to think of the outcomes as equally likely. For example, if we pick a random number between 0 and 1, we may assume all outcomes are equally likely. In such cases, the size of the set can be measured via length, area, volume, etc.

Exercise 0.5. A random number in the interval $[0, 1]$ is chosen. What is the probability that it is more than $1/2$ but less than $2/3$? What is the probability that it is equal to $1/2$? What is the probability that it is rational (optional)? △

Exercise 0.6. A random point is chosen in a square of unit side. What is the probability that it is inside the circle of diameter one inscribed in the square? What is the probability that it is on the circle? △

0.3 Random variables and distributions

A **random variable (RV)** is a function that assigns real values to outcomes in Ω . In most cases, there is a very natural mapping. For example, let X denote the number showing on a dice. Now X is a random variable, mapping each outcome of the form “the dice shows i ” to the real number i . For this reason, the

²This symbol indicates that the exercise, section, etc., is optional.

fact that random variables are really functions is often overlooked. Information about the probabilities of different outcomes is given by the **distribution** of the random variable.

A random variable is **discrete** if there are a countable number of possibilities (could be infinite but countable, like natural numbers). They can also be **continuous** (uncountable number of outcomes, defined over the real line or some subset of some Euclidean space).

For example, a random variable that is 1 if heads shows when a given coin is flipped and is 0 otherwise is discrete and finite; the number of phone calls made in a given hour is discrete and infinite; the arrival time of a plane from midnight is continuous.

0.3.1 Discrete distributions

The distribution of a discrete random variable X is given by its **probability mass function** (pmf) denoted by $p_X(x)$, where

$$p_X(x) = \Pr(X = x).$$

Clearly, $p_X(x) \geq 0$ for all x and

$$\sum_x p_X(x) = 1. \quad (0.1)$$

If clear from the context, we drop the X in the subscript.

Example 0.7 (Poisson Distribution). An RV X has the Poisson distribution with parameter λ if

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x \in \{0, 1, \dots\}.$$

The number of times an event, e.g., phone calls or car accidents, occurs in a given interval of time is often assumed to have a Poisson distribution (with good reason). \triangle

Exercise 0.8. A red die and a blue die are rolled. Let X denote the number showing on the red die and Y denote the sum of the two dice. Find the pmf of X and the pmf of Y . \triangle

Exercise 0.9. Two cards are drawn at random from a standard deck of 52 cards and let Z denote the number of Aces drawn. Find the pmf of Z . \triangle

0.3.2 Continuous distributions

The distribution of a continuous random variable X is given by its **probability distribution function** (pdf) $p_X(x)$, also sometimes denoted $f_X(x)$. Roughly speaking,

$$\Pr\left(x - \frac{dt}{2} \leq X \leq x + \frac{dt}{2}\right) = p_X(x)dt.$$

For two real numbers a, b ,

$$\Pr(a \leq X \leq b) = \int_a^b p_X(x)dx.$$

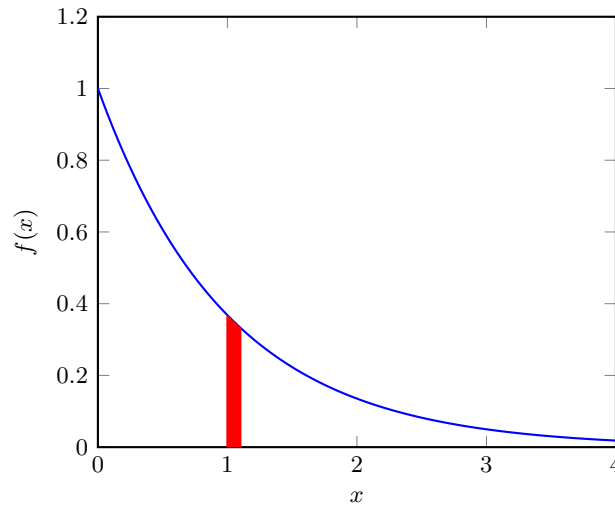
For any pdf, we have $p_X(x) \geq 0$ and

$$\int_{-\infty}^{\infty} p_X(x)dx = 1.$$

Exercise 0.10 (Exponential distribution). An exponential random variable X with parameter λ has distribution

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0.$$

For $\lambda = 1$, the probability that X is between 1 and 1.1 is around $e^{-1} \times 0.1 = 0.37 \times 0.1 = 0.037$. In the figure below, the area colored red represents this probability.



△

0.3.3 Cumulative distribution functions

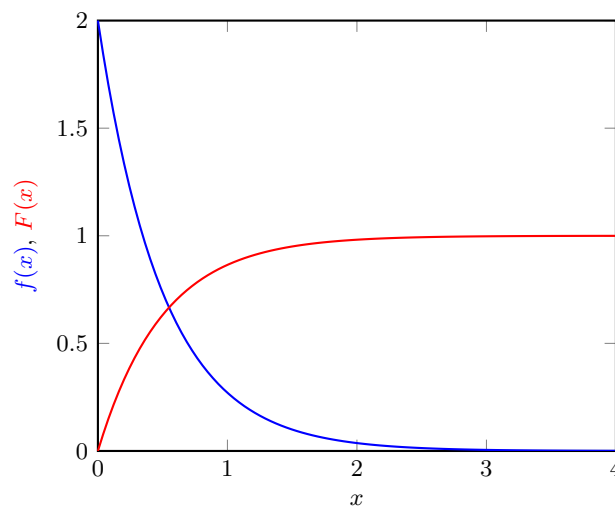
Cumulative distribution functions (CDFs) are defined for both discrete and continuous RVs as $F_X(x) = p_X(X \leq x)$ and can be found via summation or integration:

$$F_X(x) = \sum_{k \leq x} p_X(k)$$

$$F_X(x) = \int_{-\infty}^x p_X(t) dt$$

Example 0.11. The CDF of the exponential RV in Example 0.10 with $\lambda = 2$ is given by

$$F_X(x) = \int_{-\infty}^x \lambda e^{-\lambda t} dt = 1 - e^{-\lambda x}$$



△

0.3.4 Expected value

The **expected value** or the **mean** $\mathbb{E}[X]$ of a random variable X with distribution $p(x)$ is given by

$$\begin{aligned}\mathbb{E}[X] &= \sum_x xp(x), \\ \mathbb{E}[X] &= \int_{-\infty}^{\infty} xp(x)dx.\end{aligned}$$

One way to think about the expected value is as the average of a large number of experiments. For example, if a game pays out $\$X$ each time you play with probability distribution $p(x)$, if you play the game many times, on average you will win $\$ \mathbb{E}[X]$ per game. That is if you play n times, each time winning $\$x_n$, and n is large, then

$$\frac{1}{n}(x_1 + x_2 + \dots + x_n) \simeq \mathbb{E}[X].$$

Exercise 0.12. Find the expected value of the discrete and continuous RVs in the examples above. △

Exercise 0.13. Find $\mathbb{E}[1]$. △

0.3.4.1 Expectation of functions of random variables

For an RV X and a function $f(x)$ it follows from the definition that

$$\begin{aligned}\mathbb{E}[f(X)] &= \sum_x f(x)p(x), \\ \mathbb{E}[f(X)] &= \int_{-\infty}^{\infty} f(x)p(x)dx.\end{aligned}\tag{0.2}$$

Exercise 0.14. A random variable X has distribution

$$p_X(-1) = 0.1, \quad p_X(0) = 0.2, \quad p_X(1) = 0.3, \quad p_X(2) = 0.4.$$

Find $\mathbb{E}X$. Let $Y = X^2$. Find $\mathbb{E}Y$, both by finding the distribution of Y and by using (0.2). △

0.3.4.2 Linearity of expectation

For a RV X , functions $f(x)$ and $g(x)$, and real numbers a and b ,

$$\mathbb{E}[af(X) + bg(X)] = a\mathbb{E}[f(X)] + b\mathbb{E}[g(X)],$$

which can be proven easily from the definition of expectation.

Example 0.15. $\mathbb{E}[(X - a)^2] = \mathbb{E}[X^2 - 2aX + a^2] = \mathbb{E}[X^2] - 2a\mathbb{E}X + a^2$. △

Consider a collection of random variables X_1, X_2, \dots, X_n . By the linearity of expectation

$$\mathbb{E}\left[\sum_{i=1}^n X_i\right] = \sum_{i=1}^n \mathbb{E}X_i.\tag{0.3}$$

If all variables are identically distributed, then

$$\mathbb{E} \left[\sum_{i=1}^n X_i \right] = n \mathbb{E} X_1. \quad (0.4)$$

Example 0.16. In a class of n students, what is the expected number of pairs of students who have the same birthday? To find this, for two students i and j , let X_{ij} be equal to 1 if they share a birthday and 0 otherwise and let $X = \sum_{i=1}^{n-1} \sum_{j=i+1}^n X_{ij}$. Now,

$$\mathbb{E} X = \binom{n}{2} \mathbb{E} X_{12} = \binom{n}{2} \Pr(X_{12} = 1) = \binom{n}{2} \frac{1}{365} \simeq \frac{n^2}{730}. \quad (0.5)$$

In particular, having $n = \sqrt{730} \simeq 27$ students in a class is enough to have on average one pair with the same birthday. With $n = 60$ and $n = 85$ students, there should be around 5 and 10 such pairs, respectively. \triangle

0.3.4.3 Variance

Suppose someone offers you a game in which your expected winning is \$100. Will you accept? Which game would you play?

- You always win exactly \$100.
- You win \$0 with probability $1/2$ and \$200 with probability $1/2$.
- You win \$1200 with probability $1/2$ and lose \$1000 with probability $1/2$.

All three have the same mean. So what's different between them?

The mean helps us represent a distribution with one value, which describes the average behavior of the RV. But as this example shows, the behavior around the mean is also important. Denoting the mean of X by μ_X , the variability around the mean is captured to a degree by the variance $\text{Var}[X]$,

$$\text{Var}[X] = \mathbb{E}[(X - \mu_X)^2].$$

The variance gives a sense of *how far X is from its mean μ_X , on average*. The **standard deviation**, σ_X , is defined as

$$\sigma_X = \sqrt{\text{Var}[X]},$$

and the variance is usually denoted as σ_X^2 .

Exercise 0.17. Prove that

$$\text{Var}[X] = \mathbb{E} X^2 - (\mathbb{E} X)^2.$$

\triangle

Exercise 0.18. Find the mean and variance of each of the following RVs [1]:

- $X + c$
- aX
- $aX + c$
- $\frac{X - \mu_X}{\sigma_X}$ (called the **standardized version** of X)

\triangle

0.3.5 Common distributions

We denote X having distribution ‘Dist’ by $X \sim \text{Dist}(a, b, \dots)$, where a, b, \dots , are the parameters of the distribution.

0.3.5.1 Discrete distributions

- $X \sim \text{Ber}(p)$: $\Pr(X = 1) = p$, $\Pr(X = 0) = 1 - p$, $\mathbb{E}[X] = p$, $\text{Var}[X] = p(1 - p)$.
- $X \sim \text{Bin}(n, p)$: ³ $p(x) = \binom{n}{x} p^x (1 - p)^{n-x}$, $0 \leq x \leq n$, $\mathbb{E}[X] = np$, $\text{Var}[X] = np(1 - p)$.
- $X \sim \text{Geo}(p)$: $p(x) = (1 - p)^{x-1} p$, $x \geq 1$, $\mathbb{E}[X] = 1/p$, $\text{Var}[X] = (1/p)^2 - (1/p)$.
- $X \sim \text{NegBin}(k, p)$: $p(x) = \binom{x-1}{k-1} (1 - p)^{x-k} p^k$, $x \geq k$, $\mathbb{E}[X] = k/p$, $\text{Var}[X] = k[(1/p)^2 - (1/p)]$.
- $X \sim \text{Poi}(\lambda)$: $p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$, $x \geq 0$, $\mathbb{E}[X] = \lambda$, $\text{Var}[X] = \lambda$.
- $X \sim \text{Uni}[a, b]$: $p(x) = \frac{1}{b-a+1}$, $x \in \mathbb{Z}, a \leq x \leq b$, $\mathbb{E}[X] = \frac{a+b}{2}$, $\text{Var}[X] = \frac{(b-a+1)^2 - 1}{12}$.

Exercise 0.19. Prove that the mean of $\text{Bin}(n, p)$ is as given using Exercise 0.2. △

0.3.5.2 Continuous distributions

- $X \sim \text{Uni}(a, b)$: $p(x) = \frac{1}{b-a}$, $x \in (a, b)$, $\mathbb{E}[X] = \frac{a+b}{2}$, $\text{Var}[X] = \frac{(b-a)^2}{12}$.
- $X \sim \mathcal{N}(\mu, \sigma^2)$: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$, $x \in \mathbb{R}$, $\mathbb{E}[X] = \mu$, $\text{Var}[X] = \sigma^2$.
- $X \sim \text{Exp}(\lambda)$: $p(x) = \lambda e^{-\lambda x}$, $x \geq 0$, $\mathbb{E}[X] = 1/\lambda$, $\text{Var}[X] = 1/\lambda^2$.

Sometimes, we drop the normalization constant, that is, the constant by which we divide to ensure that the distribution integrates to 1. This could be because the constant is not important (e.g., in Bayesian inference) or because it is hard to determine. In such cases, we use \propto to show proportionality rather than equality. We should be careful which of the entities appearing is the *variable*. For example, viewed as a function of x , we have $f(x) = \frac{\lambda^x e^{-\lambda}}{x!} \propto \frac{\lambda^x}{x!}$ and as a function of λ , we have $g(\lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \propto \lambda^x e^{-\lambda}$.

- $X \sim \text{Beta}(\alpha, \beta)$: $p(x) \propto x^{\alpha-1} (1-x)^{\beta-1}$, $0 \leq x \leq 1$, $\mathbb{E}[X] = \frac{\alpha}{\alpha+\beta}$, $\text{Var}[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.
- $X \sim \text{Gamma}(\alpha, \beta)$: $p(x) \propto x^{\alpha-1} e^{-\beta x}$, $x > 0$, $\mathbb{E}[X] = \frac{\alpha}{\beta}$, $\text{Var}[X] = \frac{\alpha}{\beta^2}$.

Example 0.20. For the distributions given in this section, try changing what the variable is and what the parameters are and check whether another distribution from the list can be obtained with appropriate normalization. For example, $\text{Bin}(n, p)$ viewed as a distribution in p turns into $\text{Beta}(x+1, n-x+1)$. △

0.4 Joint probability distributions

Joint probability distributions allow us to encode information about relationships between quantities, from independence to strong correlation.

For random variables X and Y , the CDF and the pmf/pdf give their joint distribution, depending on their

³Note that sometimes p is used both as a parameter and as the distribution. The meaning should be clear from the context.

type,

$$\begin{aligned}
 F_{X,Y}(x,y) &= \Pr(X \leq x, Y \leq y), && \text{CDF for continuous and discrete} \\
 p_{X,Y}(x,y) &= \Pr(X = x, Y = y), && \text{pmf for discrete} \\
 p_{X,Y}(x,y)dxdy &\simeq \Pr\left(x - \frac{dx}{2} \leq X \leq x + \frac{dx}{2}, y - \frac{dy}{2} \leq Y \leq y + \frac{dy}{2}\right), && \text{pdf for continuous}
 \end{aligned}$$

We can find the distribution for each random variable (in this context these are called the **marginals**) by integration/summation,

$$p_X(x) = \sum_y p_{X,Y}(x,y), \quad p_X(x) = \int_{-\infty}^{\infty} p_{X,Y}(x,y)dy.$$

0.4.1 Expectation, correlation, and covariance

Given two or more RVs, we may be interested in finding the expected value of a function of these RVs, e.g., $\mathbb{E}[XY]$. In such case, similar to (0.2), we have

$$\mathbb{E}[f(X,Y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)p(x,y)dxdy, \quad (0.6)$$

and similarly for discrete variables.

The **correlation** between X and Y is $\mathbb{E}[XY] = \int \int xyp(x,y)dxdy$. The **covariance** $\text{Cov}(X,Y)$ and the **correlation coefficient** $\rho_{X,Y}$ are defined as

$$\begin{aligned}
 \text{Cov}(X,Y) &= \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] \\
 \rho_{X,Y} &= \frac{\text{Cov}(X,Y)}{\sigma_X \sigma_Y}.
 \end{aligned}$$

It can be shown that $-1 \leq \rho_{X,Y} \leq 1$. If $\rho = 0$, then the random variables are **uncorrelated**.

What does the correlation coefficient mean? Let X and Y be random variables, for example, weight and height of a person chosen at random. Suppose that we want to predict the value of Y given X but we are restricted to linear functions of X . Then, in a certain sense,⁴ the best predictor \hat{Y} of Y is

$$\hat{Y} = \mathbb{E}Y + \rho \frac{\sigma_Y}{\sigma_X}(X - \mathbb{E}X),$$

with the “error” being

$$\sigma_Y^2(1 - \rho^2).$$

In particular, if X and Y are standardized, $\hat{Y} = \rho X$ with error $1 - \rho^2$.

Exercise 0.21. If $|\rho|$ is close to 1, the RVs are said to be **strongly correlated**. Why? △

Exercise 0.22. Show that $\text{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}X \mathbb{E}Y$. △

Example 0.23. The bivariate jointly Gaussian distribution for X,Y with means μ_X and μ_Y , variances σ_X and σ_Y , and correlation coefficient ρ is given as

$$p(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)} \left[\frac{(x-\mu_X)^2}{\sigma_X^2} + \frac{(y-\mu_Y)^2}{\sigma_Y^2} - \frac{2\rho(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} \right]}.$$

Examples of this pdf are given in Figure 1. △

⁴Minimizing the Mean Square Error

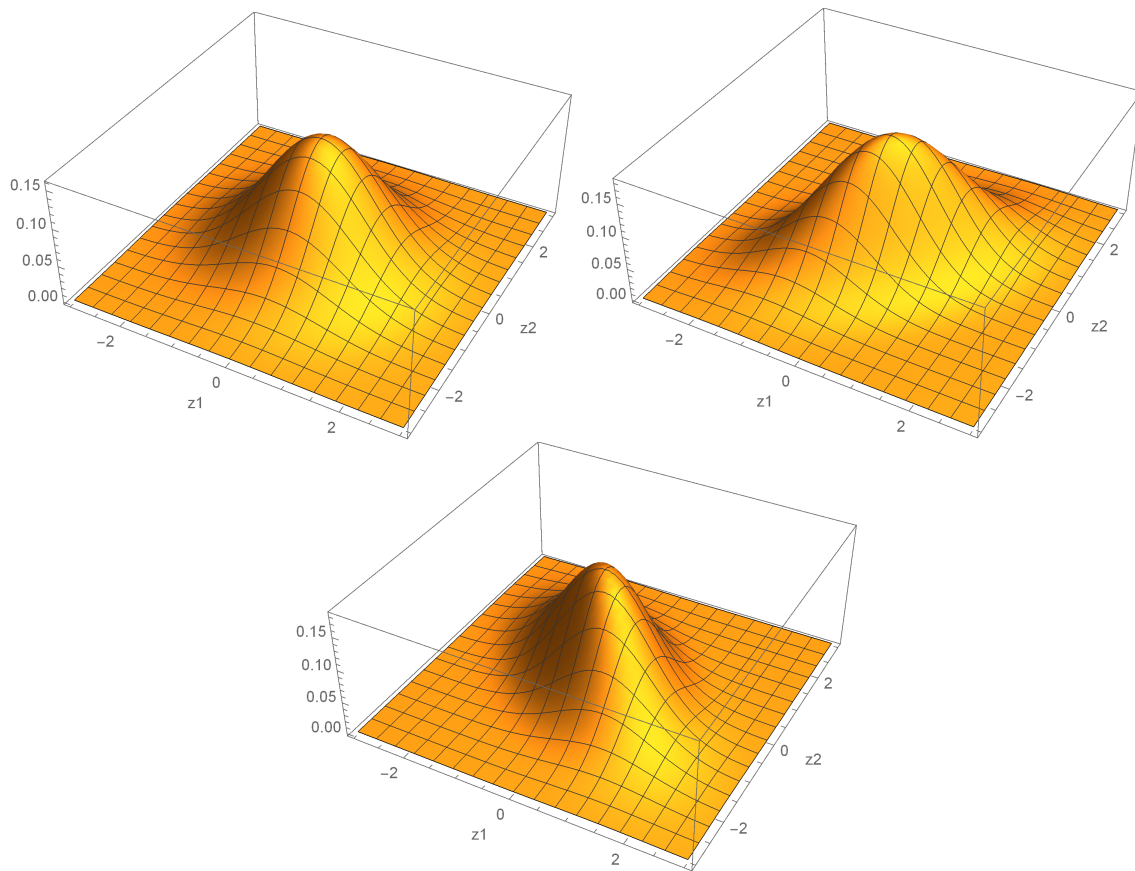


Figure 1: Bivariate Normal pdfs with $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, with $\rho = 0$ (uncorrelated), $\rho = .5$ (positively correlated), and $\rho = -.5$ (negatively correlated), respectively.

Exercise 0.24. For random variables X, Y, Z and constants a, b, c, d, e , prove that

- $\text{Var}(X) = \text{Cov}(X, X)$
- $\text{Cov}(X + Y, Z) = \text{Cov}(X, Z) + \text{Cov}(Y, Z)$
- $\text{Cov}(aX, Y) = a \text{Cov}(X, Y)$
- $\text{Cov}(X, b) = 0$
- $\text{Cov}(aX + bY + c, dZ + e) = ad \text{Cov}(X, Z) + bd \text{Cov}(Y, Z)$

△

Exercise 0.25. Find the expected values and variances of X and Y from Exercise 0.8. Find $\text{Cov}(X, Y)$. △

0.4.2 Independence

Recall that two events A and B are independent iff (if and only if) $\Pr(A \cap B) = \Pr(A) \Pr(B)$. Two random variables X and Y are independent if $\{X \in S_1\}$ and $\{Y \in S_2\}$ are independent for all sets S_1 and S_2 . This implies that

$$p(x, y) = p(x)p(y). \quad (0.7)$$

For two independent random variables, we have

$$\mathbb{E}[XY] = \mathbb{E}[X] \mathbb{E}[Y] \quad (0.8)$$

and $\text{Cov}(X, Y) = 0$.

Exercise 0.26. Prove (0.8) using (0.7). △

Exercise 0.27. For two independent RVs X and Y , find $\text{Var}[X + Y]$ and $\mathbb{E}[(X - Y)^2 + 3XY + 5]$ in terms of means and variances of X and Y . △

A collection X_1, \dots, X_n of random variables that are independent from each other but have the same distribution are called **independent and identically distributed (iid)**. We have

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i). \quad (0.9)$$

Exercise 0.28. For iid RVs X_1, \dots, X_n , let $S_n = \sum_{i=1}^n X_i$. Show that

$$\text{Var}(S_n) = \sum_{i=1}^n \text{Var}(X_i). \quad (0.10)$$

△

Exercise 0.29. For iid RVs X_1, \dots, X_n , suppose $\mathbb{E}[X_i] = \mu$ and $\text{Var}[X_i] = \sigma^2$, and let \bar{X} be their average. Show that

$$\mathbb{E}[\bar{X}] = \mu, \quad \text{Var}[\bar{X}] = \frac{\sigma^2}{n}. \quad (0.11)$$

△

0.4.3 Conditional probability and conditional distributions

For two discrete variables X and Y , the conditional probability distribution of Y given X is given by

$$p_{Y|X}(y|x) = \Pr(Y = y|X = x) = \frac{\Pr(Y = y, X = x)}{\Pr(X = x)} = \frac{p_{X,Y}(x, y)}{p_X(x)}.$$

For continuous RVs, we also have $p_{Y|X}(y|x) = \frac{p_{X,Y}(x,y)}{p_X(x)}$. In this case, however, we interpret the conditional density as

$$p_{Y|X}(y|x) \simeq \frac{\Pr(y - \epsilon/2 \leq Y \leq y + \epsilon/2 | x - \epsilon/2 \leq X \leq x + \epsilon/2)}{\epsilon},$$

for small positive ϵ . This essentially says to find $p_{Y|X}(y|x)$, we first assume that X is in a narrow strip around x and then find the density for Y given this assumption.

Law of total probability. Let A_1, A_2, \dots, A_n be a partition of the sample space. That is, $\cup_{i=1}^n A_i = \Omega$ and for all $i \neq j$, we have $A_i \cap A_j = \emptyset$. For an event B_i , we have

$$\Pr(B) = \sum_{i=1}^n \Pr(B \cap A_i) = \sum_{i=1}^n \Pr(B|A_i) \Pr(A_i).$$

In particular, if X can take on $\{1, 2, \dots, n\}$, then for another RV Y ,

$$p_Y(y) = \sum_{x=1}^n p_{Y|X}(y|x) p_X(x).$$

Chain rule of probability. For events A_1, \dots, A_n , we have

$$\Pr(A_1 \cap A_2 \cap \dots \cap A_n) = \Pr(A_1) \Pr(A_2|A_1) \Pr(A_3|A_1, A_2) \dots \Pr(A_n|A_1, \dots, A_{n-1}),$$

which can be easily proven by induction. A similar rule holds for random variables X_1, \dots, X_n :

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

Conditional expectations are defined based on conditional distributions, e.g.,

$$\mathbb{E}[X|Y = y] = \sum_x x p_{X|Y}(x|y).$$

Exercise 0.30. Suppose the joint pmf is given as

$p_{X,Y}(x, y)$	$x = 0$	$x = 1$
$y = 0$	0.25	0
$y = 1$	0.5	0.25

Find $p(y|x)$, $p(x|y)$, $\mathbb{E}[Y|X = 0]$, $\mathbb{E}[Y|X = 1]$, $\mathbb{E}[X|Y = 0]$, $\mathbb{E}[X|Y = 1]$. △

Exercise 0.31. A point is chosen uniformly at random in a triangle with vertices on $(0, 0)$, $(1, 0)$, $(1, 1)$. Let X and Y determine the x and y coordinates of the chosen point. Find $p(x|y)$, $p(y|x)$, $\mathbb{E}[X|Y = y]$, $\mathbb{E}[Y|X = x]$. △

0.4.3.1 Law of iterated expectations.

Consider a random variable X and a function $g(x)$. We can now obtain $g(X)$ by replacing the deterministic value for x with a random one. Note that $g(X)$ is a random variable. For example, if $X \sim \text{Uni}(-1, 1)$ and $g(x) = |x|$, then $g(X)$ is a random variable with distribution $\text{Uni}(0, 1)$.

Now let $g(x) = \mathbb{E}[Y|X = x]$. This is, of course, a well-defined function. We define $\mathbb{E}[Y|X] = g(X)$, which is as discussed a random variable. Now that we have a random variable, we can compute its expectation, i.e., $\mathbb{E}[\mathbb{E}[Y|X]]$.

Exercise 0.32. A die is rolled, showing X . A coin is then flipped X times resulting in Y heads. Find $\mathbb{E}[Y]$, $\mathbb{E}[Y|X = x]$, the pmf of $\mathbb{E}[Y|X]$, and $\mathbb{E}[\mathbb{E}[Y|X]]$. \triangle

It can be shown that

$$\mathbb{E}[\mathbb{E}[Y|X]] = \mathbb{E}[Y], \quad \mathbb{E}[\mathbb{E}[Y|X, Z]|Z] = \mathbb{E}[Y|Z]. \quad (0.12)$$

0.4.4 Bayes' rule

In Exercise 0.32, the conditional distribution $p(y|x)$ is readily available as

$$p(y|x) = \binom{x}{y} 2^{-x}.$$

But what if we are interested in $p(x|y)$? Since $p(x|y) = \frac{p(x,y)}{p(y)}$ and $p(x, y) = p(y|x)p(x)$, we have

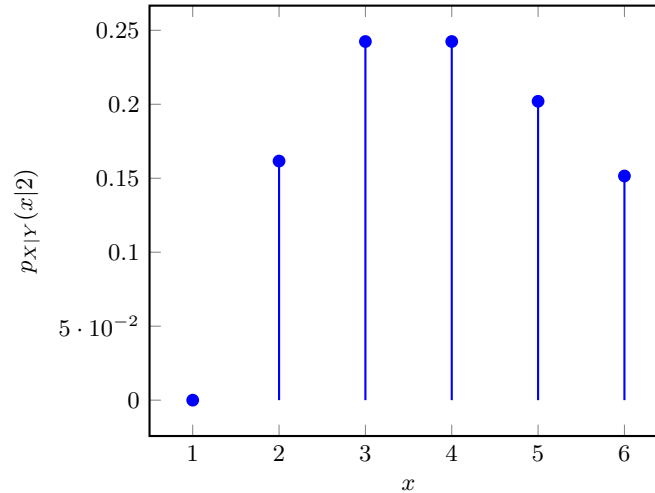
$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')},$$

which is called the **Bayes rule**.

Example 0.33. In Exercise 0.32, we can use the Bayes rule to find $p(x|y)$,

$$p(x|y) = \frac{\binom{x}{y} 2^{-x} (1/6)}{\sum_{x'=y}^6 \binom{x'}{y} 2^{-x'} (1/6)} = \frac{\binom{x}{y} 2^{-x}}{\sum_{x'=y}^6 \binom{x'}{y} 2^{-x'}}$$

We may ask for example, what is the likeliest value for X if $Y = 2$. Below, $p_{X|Y}(x|2)$, i.e., the conditional distribution of X given $Y = 2$. We can see that the likeliest values for X are 3, 4.



\triangle

Bayes' rule is used in *evidential reasoning*, examples of which we will see in the next chapter. In this setting, the goal is to find the probabilities of different causes based on the evidence.

Bayesian inference takes its name from Bayes rule. In this setting, it is often the case that we know the

distribution of data given the parameters. But what we actually have is data and need to find the distribution of the parameters. The Bayes rule allows us to find this conditional distribution, a topic we will discuss in detail later.

0.5 Inequalities and limits

0.5.1 Inequalities

0.5.1.1 Markov inequality

Suppose the average length of a blue whale is 22m and we do not know anything else about the distribution of the lengths of blue whales. Can we say anything about the probability that the length of a randomly chosen blue whale is ≥ 30 m? For example, is it possible that this probability is 0.8 or larger? No, since in that case, the average would be $\geq 0.8 \times 30\text{m} = 24\text{m}$. So only knowing the mean enables us to say something about the extremes of the probability distribution.

This observation is formalized via the **Markov inequality**. For a *non-negative* random variable X , we have

$$\Pr(X \geq a) \leq \frac{\mathbb{E} X}{a}.$$

Exercise 0.34. Prove the Markov inequality. △

A special case of this occurs when X counts something, i.e., it only takes non-negative integer values. Then,

$$\Pr(X \geq 1) = \Pr(X > 0) \leq \mathbb{E} X, \quad \Pr(X = 0) \geq 1 - \mathbb{E} X.$$

In particular, if the mean $\mathbb{E} X$ is small, then there is a large probability that $X = 0$.

Exercise 0.35 (†). Provide a bound on the probability that in a random binary sequence of length n , there exists a run (consecutive occurrences) of 1s of length at least $2 \log_2 n$? (The result will tell you that this is unlikely for large n .) △

0.5.1.2 Chebyshev inequality

If in addition to the mean, we also have the variance, we can use the Chebyshev bound. For a random variable X with mean μ and variance σ^2 ,

$$\Pr\left(\left|\frac{X - \mu}{\sigma}\right| \geq a\right) \leq \frac{1}{a^2}.$$

Exercise 0.36. Prove the Chebyshev bound using the Markov bound. △

Example 0.37. The Chebyshev bound tells us that being k standard deviations away from the mean has probability at most $1/k^2$.

k	2	3	4	5	6	7	8	9	10
Probability of deviating more than $k \times \text{std}$ is \leq	25%	11.1%	6.25 %	4%	2.78%	2.04%	1.56%	1.23%	1%

In particular, being 10 standard deviations away from the mean has probability at most 1%. △

0.5.2 Limits

Limits in probability provide a way to understand what happens when the number of experiments grows or many random effects accumulate. Limit theorems are beneficial given that we often deal with large volumes

of data. The following limit theorems will be helpful to us later in the course.

0.5.2.1 Law of large numbers

Let X_1, \dots, X_n be random variables with mean μ and variance $\leq \sigma^2$ and suppose that for each i and j , X_i and X_j are uncorrelated (in particular, independent). Also, let $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$. Then, for any $\epsilon > 0$,

$$\Pr(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n\epsilon^2}. \quad (0.13)$$

As n becomes large the right side becomes smaller and smaller. So for large n the probability of \bar{X}_n being too far from the mean is very small. This is referred to as the **Law of Large Numbers** (LLN). In other words, if we take n independent samples from a random variable X , then the average of those samples will be close to the mean $\mathbb{E}X$,

$$\frac{1}{n}(x_1 + x_2 + \dots + x_n) \simeq \mathbb{E}[X],$$

which is what we used to motivate expected value.

Exercise 0.38. Use the Chebyshev inequality to prove LLN when random variables are independent and all have the same variance σ^2 . \triangle

Example 0.39. Suppose $X_i \sim \text{Poi}(2)$, $1 \leq i \leq 500$, and let \bar{X}_n be the average of the first n X_i s. Figure 2 shows the plot for \bar{X}_n for a realization of X_i s obtained via computer simulation. It is observed that for large values of n , \bar{X}_n is close to 2, the mean of the Poisson distribution. \triangle

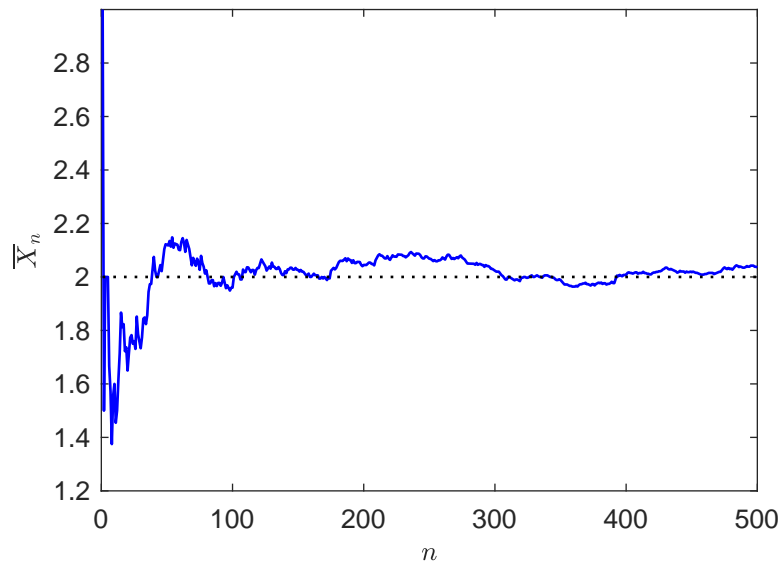


Figure 2: \bar{X}_n based on $X_i \sim \text{Poi}(2)$ as a function of n .

0.5.2.2 Central limit theorem

Let X_1, X_2, \dots be iid random variables with mean μ and variance σ^2 and let $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$. As $n \rightarrow \infty$. The **Central Limit Theorem (CLT)** states that

$$\text{distribution of } \sqrt{n}(\bar{X}_n - \mu) \rightarrow \mathcal{N}(0, \sigma^2). \quad (0.14)$$

That is, the distribution of $\sqrt{n}(\bar{X}_n - \mu)$ approaches the distribution of a normal random variable with mean 0 and variance σ^2 .

Loosely speaking, the CLT also means $S_n = \sum_{i=1}^n X_i$ has distribution $\mathcal{N}(n\mu, n\sigma^2)$.

Example 0.40. Let $X_i \sim \text{Uni}(0, 1)$, $1 \leq i \leq n = 10$. We produce 50,000 samples of \bar{X}_n (and S_n), and plot the normalized histograms for $\sqrt{n}(\bar{X}_n - \mu)$ and the pdf of $\mathcal{N}(0, \sigma^2)$ and the normalized histogram for S_n and the pdf of $\mathcal{N}(n\mu, n\sigma^2)$ in Figure 3. \triangle

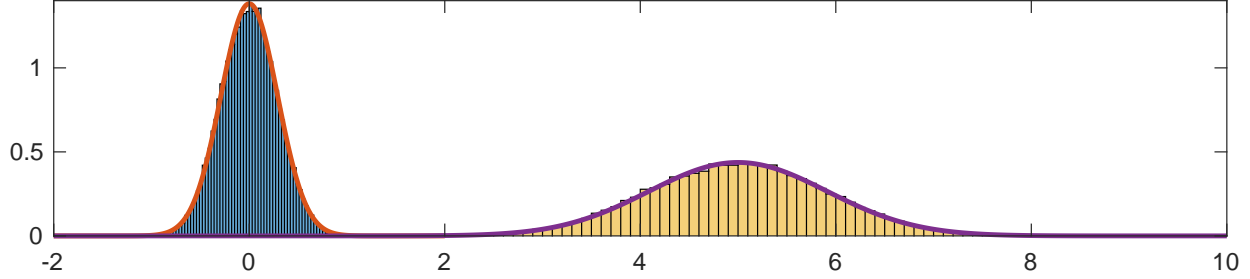


Figure 3: The normalized histograms for $\sqrt{n}(\bar{X}_n - \mu)$ and the pdf of $\mathcal{N}(0, \sigma^2)$ (on the left) and the normalized histogram for S_n and the pdf of $\mathcal{N}(n\mu, n\sigma^2)$ (on the right) for uniform X_i with $\mu = 1/2$ and $\sigma^2 = 1/12$ and with $n = 10$.

0.6 Random vectors

A **random vector** is a vector of random variables.⁵ Consider the random vectors \mathbf{X} and \mathbf{Y}

$$\mathbf{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_m \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}. \quad (0.15)$$

The **expected value** of \mathbf{X} is

$$\mathbb{E} \mathbf{X} = \begin{pmatrix} \mathbb{E} X_1 \\ \vdots \\ \mathbb{E} X_m \end{pmatrix}. \quad (0.16)$$

The **correlation matrix** of \mathbf{X} and \mathbf{Y} is the $m \times n$ matrix $\mathbb{E}[\mathbf{X}\mathbf{Y}^T]$, whose i, j th element is $\mathbb{E}[X_i Y_j]$. The **cross-covariance matrix** $\text{Cov}(\mathbf{X}, \mathbf{Y})$ of \mathbf{X} and \mathbf{Y} is the matrix $\mathbb{E}[(\mathbf{X} - \mathbb{E} \mathbf{X})(\mathbf{Y} - \mathbb{E} \mathbf{Y})^T]$, whose i, j th element is $\text{Cov}(X_i, Y_j)$. The covariance of a vector \mathbf{X} is $\text{Cov}(\mathbf{X}) = \text{Cov}(\mathbf{X}, \mathbf{X})$. The **conditional expectation** $\mathbb{E}[\mathbf{X}|\mathbf{Y}]$ of \mathbf{X} given \mathbf{Y} is a vector whose i th element is $\mathbb{E}[X_i|\mathbf{Y}]$.

If the elements of \mathbf{X} are uncorrelated, then $\text{Cov}(X_i, X_j) = 0$ for $i \neq j$ and the covariance matrix becomes diagonal. If, in addition, $\text{Cov}(X_i, X_i) = \text{Var}(X_i) = \sigma^2$, i.e., all elements of \mathbf{X} have the same variance σ^2 , then $\text{Cov}(\mathbf{X}) = \sigma^2 I$.

0.6.1 Properties of expectation and covariance

For deterministic matrices \mathbf{A}, \mathbf{B} , deterministic vectors \mathbf{a}, \mathbf{b} , and random vectors $\mathbf{X}, \mathbf{Y}, \mathbf{W}, \mathbf{Z}$, we have [1]

1. $\mathbb{E}[\mathbf{A}\mathbf{X} + \mathbf{a}] = \mathbf{A} \mathbb{E} \mathbf{X} + \mathbf{a}$
2. $\text{Cov}(\mathbf{X}, \mathbf{Y}) = \mathbb{E}[\mathbf{X}(\mathbf{Y} - \mathbb{E} \mathbf{Y})^T] = \mathbb{E}[(\mathbf{X} - \mathbb{E} \mathbf{X})(\mathbf{Y} - \mathbb{E} \mathbf{Y})^T] = \mathbb{E}[\mathbf{X}\mathbf{Y}^T] - \mathbb{E} \mathbf{X} \mathbb{E} \mathbf{Y}^T$
3. $\mathbb{E}[(\mathbf{A}\mathbf{X})(\mathbf{B}\mathbf{Y})^T] = \mathbf{A} \mathbb{E}[\mathbf{X}\mathbf{Y}^T] \mathbf{B}^T$

⁵We use lowercase bold letters to denote deterministic vectors, uppercase bold letters to denote random vectors, and uppercase sans serif letters, such as \mathbf{A} , to denote matrices.

4. $\text{Cov}(\mathbf{A}\mathbf{X} + \mathbf{a}, \mathbf{B}\mathbf{Y} + \mathbf{b}) = \mathbf{A} \text{Cov}(\mathbf{X}, \mathbf{Y}) \mathbf{B}^T$
5. $\text{Cov}(\mathbf{A}\mathbf{X} + \mathbf{a}) = \mathbf{A} \text{Cov}(\mathbf{X}) \mathbf{A}^T$
6. $\text{Cov}(\mathbf{W} + \mathbf{X}, \mathbf{Y} + \mathbf{Z}) = \text{Cov}(\mathbf{W}, \mathbf{Y}) + \text{Cov}(\mathbf{W}, \mathbf{Z}) + \text{Cov}(\mathbf{X}, \mathbf{Y}) + \text{Cov}(\mathbf{X}, \mathbf{Z})$

Example 0.41. For a random vector \mathbf{X} and constants a, \mathbf{b} , from property 5, we have $\text{Cov}(a\mathbf{X} + \mathbf{b}) = a^2 \text{Cov}(\mathbf{X})$. We also prove this using the other properties. The relevant properties are given in each step.

$$\text{Cov}(a\mathbf{X} + \mathbf{b}) = \text{Cov}(a\mathbf{X} + \mathbf{b}, a\mathbf{X} + \mathbf{b}) \quad (0.17)$$

$$\stackrel{6}{=} \text{Cov}(a\mathbf{X}, a\mathbf{X}) + \text{Cov}(a\mathbf{X}, \mathbf{b}) + \text{Cov}(\mathbf{b}, a\mathbf{X}) + \text{Cov}(\mathbf{b}, \mathbf{b}) \quad (0.18)$$

$$\stackrel{2}{=} a^2 \text{Cov}(\mathbf{X}, \mathbf{X}) + 0 + 0 + 0 \quad (0.19)$$

△

References

- [1] Bruce Hajek. *Random Processes for Engineers*. Illinois, 2014. URL: <http://hajek.ece.illinois.edu/Papers/randomprocJuly14.pdf> (visited on 01/30/2017).

Chapter 1

Probability, Inference, and Learning

1.1 Introduction

In this chapter, we will study the role of probability in inference, codifying relationships, and machine learning. When considering these problems, we deal with uncertainty, and that's where probability comes in. In other words, we are interested in probability because it allows us to model uncertainty (or equivalently, belief and knowledge). Sources of uncertainty, for example in machine learning, include:

- Noise: aggregate contribution of factors that we do not (wish to) consider (models focus on the most important quantities).
- Finite sample size: finite size of data makes it impossible to determine relationships (i.e., probability distributions) as some configuration may never happen or happen few times in finite data.

1.2 Relationships and joint probability distributions

Is there any relationship between the arrival times of two people working at a business (opening at 9:00 am), both living in the same area? If so, how can we represent this relationship? How can we make prediction about one being late given the other is late (e.g., if we need at least one person be present)?

In the same way that we can encode our information about a random quantity as a distribution, we can encode information about random quantities, as well as their relationships, as joint distributions.

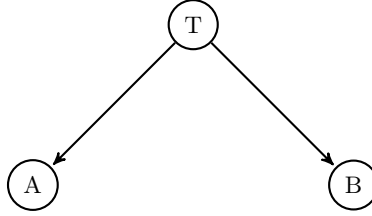
In our example, there's obviously a relationship, that is, the arrival times are not independent. For example, both are affected by traffic. Let

T_0 : normal traffic
 T_1 : heavy traffic
 A_0 : Alice is on time
 A_1 : Alice is late
 B_0, B_1 for Bob

and assume

$$\begin{aligned}\Pr(T_0) &= 0.65, \\ \Pr(A_0|T_0) &= 0.9, \\ \Pr(B_0|T_0) &= 0.82, \\ \Pr(A_0|T_1) &= 0.5, \\ \Pr(B_0|T_1) &= 0.15.\end{aligned}$$

Finally, conditioned on the traffic situation, Alice and Bob's arrival times are independent. This information completely determines all probabilities. As we will see in much greater depth later, the fact that the Alice and Bob's arrival times are only related through traffic can be shown *graphically* as



Causal reasoning:

$$\begin{aligned}\Pr(A_0) &= \Pr(T_0) \Pr(A_0|T_0) + \Pr(T_1) \Pr(A_0|T_1) = (0.65 \times 0.9) + (0.35 \times 0.5) = 0.76 \\ \Pr(B_0) &= \Pr(T_0) \Pr(B_0|T_0) + \Pr(T_1) \Pr(B_0|T_1) = (0.65 \times 0.82) + (0.35 \times 0.15) = 0.5855\end{aligned}$$

Evidential reasoning (inverse probabilities, uses Bayes rule):

$$\begin{aligned}\Pr(T_0|A_0) &= \Pr(A_0|T_0) \Pr(T_0) / \Pr(A_0) = 0.65 \times 0.9 / 0.76 = 0.7697 \\ \Pr(T_0|B_0) &= \Pr(B_0|T_0) \Pr(T_0) / \Pr(B_0) = 0.65 \times 0.82 / 0.5855 = 0.9103\end{aligned}$$

The common cause makes the events A_i and B_i dependent. Recall that two events E_1 and E_2 are independent, denoted $E_1 \perp\!\!\!\perp E_2$ if $\Pr(E_1 E_2) = \Pr(E_1) \Pr(E_2)$, or, if $\Pr(E_2) \neq 0$, $\Pr(E_1|E_2) = \Pr(E_1)$. We have

$$\begin{aligned}\Pr(A_0|B_0) &= \Pr(A_0 B_0) / \Pr(B_0) \\ \Pr(A_0 B_0) &= (0.65 \times 0.82 \times 0.9) + (0.35 \times 0.15 \times 0.5) = 0.506 \\ \Pr(A_0|B_0) &= 0.506 / 0.586 = 0.863 \neq \Pr(A_0) \\ \Pr(B_0|A_0) &= 0.506 / 0.76 = 0.6658 \neq \Pr(B_0)\end{aligned}$$

So $A_0 \not\perp\!\!\!\perp B_0$.

However, they are *conditionally independent*, by assumption

$$\Pr(A_0 B_0 | T_0) = \Pr(A_0 | T_0) \Pr(B_0 | T_0),$$

which is denoted as $A_0 \perp\!\!\!\perp B_0 | T_0$.

What is the source of uncertainty in this problem? Since we have assumed the distribution is known, finite sample size is not an issue. The source is noise. For example, if we had information about other factors affecting Bob, e.g., how reliable his car is, if he needs to drop off his kids, etc., we could reduce the amount of noise and make better predictions.

1.3 Inference and decision making

Let us consider a problem about **inferring** unknown values and making decisions and use probability to solve it, using both frequentist and Bayesian views. Suppose that the probability that someone with a given allele of a gene will develop a certain disease is θ . We are interested in determining θ . In particular, we may be interested in comparing this with the fraction of people in the general population with that disease, say 0.01. Different interpretations lead to different approaches to problems. But to decide, both frequentists and Bayesians need data.

Data (\mathcal{D}): Among a sample of 100 people with this allele, 2 had the disease.

- A Frequentist thinks of θ as unknown non-random parameter. She starts by asking “What is the probability of the observation as a function of θ ?” We can view each of the 100 people chosen to be an

independent Bernoulli trial with probability θ . So the distribution is Binomial and the probability of the observation as a function of θ is

$$L(\theta) = \binom{100}{2} \theta^2 (1 - \theta)^{98}.$$

Probability of the observation as a function of the parameter is called the *likelihood function*. So what value for θ makes the most sense? Since the observation has actually happened, we would expect it to have a high probability so we find θ that maximizes the likelihood. This method is called *maximum likelihood estimation*, and we'll discuss it in much more detail later. In this case, we estimate θ to be

$$\hat{\theta} = \arg \max_{\theta} L(\theta) = \frac{2}{100},$$

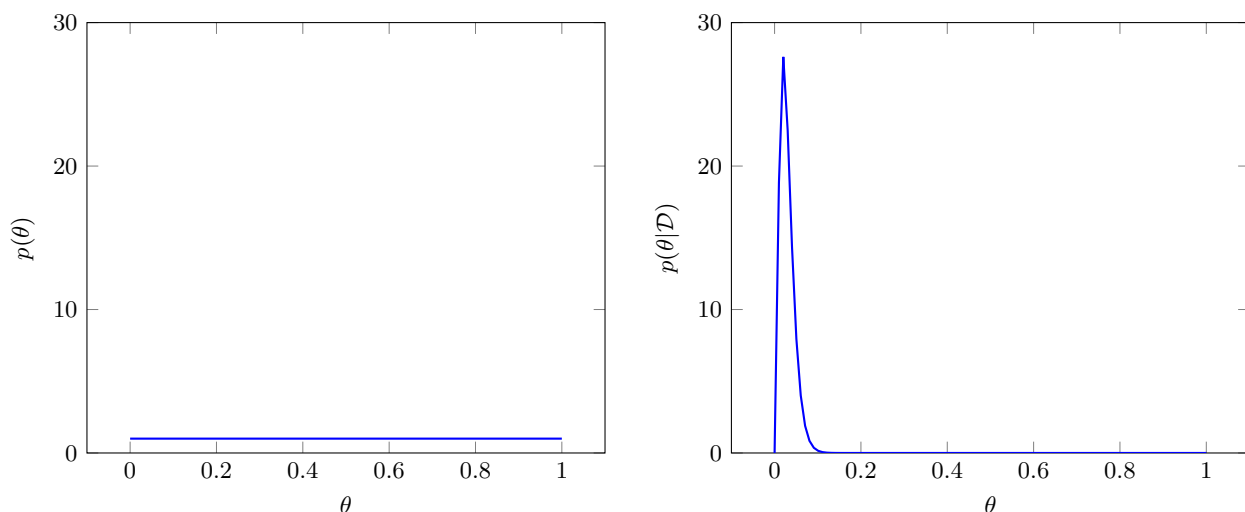
which is a reasonable estimate. But how close is the estimate to the true value? For frequentists, this is a tricky question to answer probabilistically since the true value and the estimate are both deterministic at this point. With some clever reasoning (some would say mental gymnastics), frequentists come up with *confidence intervals* and *confidence levels* to quantify the accuracy of estimators.

- A Bayesian thinks of θ as random and assigns to it a distribution, called the *prior*, before seeing the data. Thinking of θ as random is imaginative (some would say questionable) since there is no repeatable experiment and there is a single value that is true. One way to justify randomness of θ is to think of our universe being drawn from a set of possible universes. Regardless, the Bayesian view is used widely in practice.

Our Bayesian statistician then looks at the data and updates her distribution for θ , thus obtaining the *posterior* distribution. Assume that before seeing the data, we believe that the distribution for θ is uniform, i.e., $p(\theta) \sim \text{Uni}[0, 1] = \text{Beta}(1, 1)$. This means that while we do not know what θ is, we believe it is equally likely to be any value between 0 and 1. When we see the data, we can update this belief,

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \quad (\text{Bayes' rule})$$

It turns out $p(\theta|\mathcal{D}) \sim \text{Beta}(3, 99)$.



In contrast to the frequentist view, the Bayesian view is consistent and flexible. For example, we can show that

$$p(\theta > 0.01|\mathcal{D}) = 0.92.$$

What is the source of uncertainty in this problem? It is the finite sample size. If we know the status of a very large number of people with the allele, we would know the distribution/ the value of θ .

1.4 Machine Learning and Probability

Let us consider the generic form of supervised machine learning problems, which have the following components:

- **Data:** $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$, $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$. \mathcal{X} is called the feature space, and \mathcal{Y} is called the label space. As an example, each x_i could be a vector providing information about a house, e.g., (location, lot size, square footage, number of bedrooms, \dots), and y_i can be the sale price of the house.
- **Assumption:** (x_i, y_i) are iid samples of random variables X and Y . The joint distribution (X, Y) is (partially) unknown.
- **Goal:** Find the “best” function f to predict y corresponding to a given x . In other words, the function f produces an estimate $\hat{y} = f(x)$ of y given data x . Continuing our example, y would be the true but unknown price of the house with features x , and $f(x)$ would be a prediction (similar to what Zillow does).
- **Evaluation:** How do we define “best”? For a given data point (x, y) , evaluate the success of f using a loss function $L(y, f(x))$, e.g., $L(y, f(x)) = |y - f(x)|$. Ideally, we would like to minimize the expected loss over all possible outcomes weighted by their probabilities, so we define

$$\mathcal{L}(f) = \mathbb{E}[L(Y, f(X))], \quad (1.1)$$

also known as the **population risk**, where the expectation is over the distribution $p(x, y)$ of (X, Y) . Our goal then becomes finding

$$f^{**} = \arg \min_f \mathcal{L}(f) = \arg \min_f \mathbb{E}[L(Y, f(X))]. \quad (1.2)$$

- **Learning Algorithm:** The algorithm that finds f^{**} , or tries to.

The expectation in (1.2) is computed using the joint distribution $p(x, y)$. Here is where we face our main machine learning challenge: ***What we have is the data set \mathcal{D} consisting of samples from $p(x, y)$, but what we need to find f^{**} is the joint distribution $p(x, y)$.*** We can address this mismatch in two ways, either through the Empirical Risk Minimization framework discussed in §1.4.1, or through estimating the unknown distribution $p(x, y)$ using \mathcal{D} as discussed in §1.4.2.

Before proceeding further, let us consider two common problems in supervised learning:

- **Regression:** \mathcal{Y} consists of **scalars or vectors of reals**. For example, predicting stock price based on financial information, or determining the score someone will assign a movie based on previous scores. A common loss function is the **quadratic** or **squared error** loss function:

$$L(y, f(x)) = (y - f(x))^2. \quad (1.3)$$

It can be shown that for this loss, *if the distribution is known*,

$$f^{**}(x) = \mathbb{E}[Y|X = x]. \quad (1.4)$$

- **Classification:** \mathcal{Y} consists of **classes or categories**. For example, speech recognition, hand writing recognition, the presence or absence of a disease. A common loss function is the **0-1 loss**:

$$L(y, f(x)) = \begin{cases} 1, & \text{if } y \neq f(x). \\ 0, & \text{if } y = f(x). \end{cases} \quad (1.5)$$

In this case, *if the distribution is known*, then the best classifier is

$$f^{**}(x) = \arg \max_{y \in \mathcal{Y}} p(y|x). \quad (1.6)$$

We emphasize again that to solve the problem optimally as in (1.4) and (1.6), we need to know the joint distribution of x and y or the conditional distribution of y given x .

1.4.1 Empirical Risk Minimization (ERM)

Since we usually do not know the distribution but have access to data $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$, we cannot directly minimize the expected loss as in (1.2). Instead we can minimize the **empirical risk**, i.e., the loss on observed data points,

$$f^{**} = \arg \min_f \mathbb{E}[L(Y, f(X))] \quad \rightarrow \quad f_N^{**} = \arg \min_f \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)). \quad (1.7)$$

So instead of the best possible solution based on the distribution, f^{**} , we should try to find f_N^{**} based on N data points. But finding f_N^{**} is still problematic, as it only provides a way for us to determine the value of $f(x)$ for $x \in \{x_1, \dots, x_N\}$. In other words, it is not able to extrapolate or generalize.

A common solution, which is also helpful from a practical point of view, is to restrict the choices for f to a set \mathcal{H} , called the **hypothesis set**. This leads to the ERM formulation of the learning problem

$$f^* = \arg \min_{f \in \mathcal{H}} \mathbb{E}[L(Y, f(X))] \quad \rightarrow \quad f_N^* = \arg \min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)). \quad (1.8)$$

For example, we may choose \mathcal{H} to be the set of linear or sigmoid functions. By restricting predictors to the hypothesis set \mathcal{H} , we have introduced our prior knowledge, or *bias* towards the learning task.

1.4.2 Density estimation

As mentioned, distribution estimation, aka density estimation, is another way to use data for prediction. Here we discuss only *parametric density estimation*, where we can (or choose to) represent the joint distribution of (X, Y) using a probabilistic model with some unknown parameters, for example, a graphical model with known structure and unknown parameters. There are also nonparametric ways of estimating distributions.

Let us consider maximum likelihood, which is one method for parameter estimation. Suppose the distribution has a set of unknown parameters θ and we represent the distribution as p_θ . So what should we choose as the value of θ ? If an outcome has a small probability, the chance it appears in our dataset \mathcal{D} is small. So those outcomes observed in \mathcal{D} must have large probability. Hence, we must choose θ such that the probability assigned to \mathcal{D} is large, that is,

$$\begin{aligned} \hat{\theta} &= \arg \max_{\theta} p_{\theta}(\mathcal{D}) \\ &= \arg \max_{\theta} \prod_{i=1}^N p_{\theta}(x_i, y_i) \\ &= \arg \max_{\theta} \sum_{i=1}^N \log p_{\theta}(x_i, y_i), \end{aligned}$$

where in the last step, we use the monotonously of the log function to convert the product to a simpler-to-deal-with summation. We'll cover this in more detail later. For now, let us assume we can find $\hat{\theta}$, and in turn, $p_{\hat{\theta}}(x, y)$ as our estimate of the joint distribution $p(x, y)$.

With $p_{\hat{\theta}}(x, y)$ in hand, we can solve (1.2) as

$$\hat{f}_N = \arg \min_f \mathbb{E}_{\hat{\theta}}[L(Y, f(X))],$$

where $\mathbb{E}_{\hat{\theta}}$ is expectation computed using the estimated distribution $p_{\hat{\theta}}$. As we have seen in (1.4), for quadratic

and 0-1 losses, we respectively have

$$\begin{aligned}\hat{f}_N(x) &= \mathbb{E}_{\hat{\theta}}[Y|X = x], \\ \hat{f}_N(x) &= \arg \max_{y \in \mathcal{Y}} p_{\hat{\theta}}(y|x).\end{aligned}$$

1.5 Information theory and machine learning

Information theory deals with quantifying information and the rules that govern its transmission, storage, and transformation from one form to another. It has applications in communications, data storage, machine learning, and biology. In machine learning it can be used to help better understand relationships between knowns and unknowns, design loss functions, and establish fundamental limits on how well we can do with a certain amount of data (regardless of the type of algorithm and computational resources).

1.5.1 Quantifying uncertainty

Let X be a Bernoulli random variable that is equal to 1 if it is raining in Seattle and 0 otherwise. Similarly, let Y indicate whether it is raining in Phoenix. How much information do X and Y provide us? Alternatively, before they are revealed, how uncertain are we about X and about Y ? Can we measure the information content of a random variable, or equivalently, our uncertainty about them.

Let's look at specific outcomes for each variable:

- $X = 1$: It's raining in Seattle. This is a statement with a fair amount of information as rain in Seattle is almost 50/50.
- $Y = 0$: It's not raining in Phoenix. This statement doesn't provide a lot of information as this outcome is expected and has a high probability.
- $Y = 1$: It's raining in Phoenix. This provides a lot of information as this outcome is unlikely and surprising.

So as a function of probability, the amount of information of a given statement decreases as the probability increases. If the probability of an outcome is p , what is a good function describing the amount of information we gain from learning that the outcome has occurred? It turns out a good choice is $I(p) = \log \frac{1}{p}$, which is called the *self-information* function and shown in Figure 1.1 when the base of the log is 2. Then the information content of the statement ' $X = x_i$ ' is

$$I(p(x_i)) = \log \frac{1}{p(x_i)}.$$

And the amount of information *on average* for a random variable X that takes values in the set $\mathcal{X} = \{x_1, \dots, x_m\}$ is

$$H(X) = \mathbb{E} \left[\log \frac{1}{p(X)} \right] = \sum_{i=1}^m p(x_i) \log \frac{1}{p(x_i)},$$

where for continuous RVs, the sum must be replaced with an integral. This is called the *entropy*. If the log is base 2, then the unit is a *bit*.

If there are m different possible outcomes, then the maximum value that entropy can take is $\log m$. So

$$0 \leq H(X) \leq \log m.$$

An important special case is the binary entropy function $H_b(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}$ for experiments

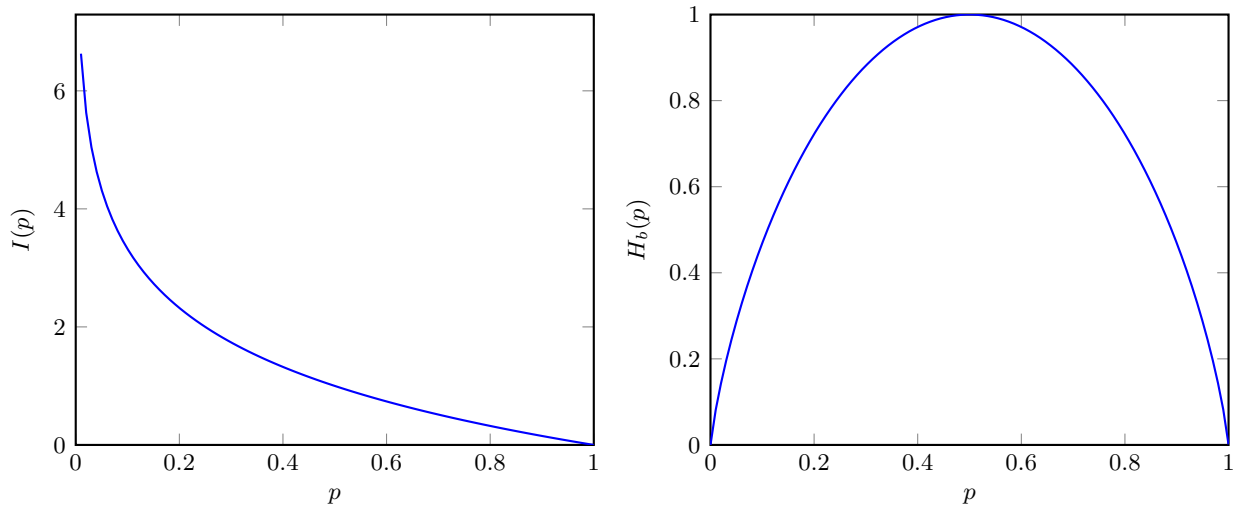


Figure 1.1: Self-information (left) for an event with probability p and binary entropy (right) for a Bernoulli RV with probability of success equal to p .

with two outcomes with probabilities p and $1 - p$. For example,

$$H(\text{Fair coin}) = H_b\left(\frac{1}{2}\right) = 1,$$

$$H(6 \text{ on a die}) = H_b\left(\frac{1}{6}\right) = 0.65,$$

$$H(\text{Rainy day in Seattle}) = H_b\left(\frac{150}{365}\right) = 0.977,$$

$$H(\text{Rainy day in Phoenix}) = H_b\left(\frac{33}{365}\right) = 0.43784,$$

$$H(\text{Rainy day in the Sahara}) = H_b\left(\frac{1}{365}\right) = 0.027267.$$

The plot for binary entropy is given in Figure 1.1. The maximum entropy is 1 bit. This makes sense since we can represent the outcome with 1 bit. Random variables with equal chances of 0 and 1 have the highest entropy (and maximum uncertainty). Those with predictable outcomes have lower entropies.

Entropy was introduced by Shannon in his article “A mathematical theory of communication” in 1948. It is also the minimum amount of “bandwidth” you need to transmit the outcome of the experiment. He also popularized the term *bit* (Binary digit).

“My greatest concern was what to call it. I thought of calling it ‘information,’ but the word was overly used, so I decided to call it ‘uncertainty.’ When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, ‘You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, no one really knows what entropy really is, so in a debate you will always have the advantage.’” – Claude Shannon, Scientific American (1971), volume 225, page 180.

1.5.2 Relative entropy

Let X be a random variable with set of possible values denoted as \mathcal{X} and its distribution as $p(x)$. Let q be another distribution also over \mathcal{X} . For example, let X be a random Latin letter with p given by the English letter frequencies and q by the French letter frequencies. For example, we have $p(E) = 12.6\%$ and

$q(E) = 15.1\%$.

The *relative entropy*, or the *Kullback–Leibler divergence*, between two distributions p and q is defined as

$$D_{KL}(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \quad (1.9)$$

The divergence can be viewed as the difference between the entropy of X when self-information is computed based on an approximate distribution and when it is based on the “true” distribution since

$$\begin{aligned} D_{KL}(p||q) &= \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{q(x)} - \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} \\ &= \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{q(x)} - H(X). \end{aligned}$$

Relative entropy provides a measure of difference between two distributions. It is always non-negative and equals 0 if and only if $q = p$. In machine learning, it is used to measure how good our estimated distribution q is to the true distribution p . It is not symmetric, so $D_{KL}(p||q)$ is not necessarily equal to $D_{KL}(q||p)$.

A related quantity is cross-entropy, which is also used as a loss function,

$$H(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{q(x)}.$$

So $D_{KL}(p||q) = H(p||q) - H(X)$.

1.5.3 Conditional entropy and mutual entropy*

We can also measure the information in multiple random variables using entropy. The information in both X and Y is denoted $H(X, Y)$ and is defined as

$$H(X, Y) = \mathbb{E} \left[\log \frac{1}{p(X, Y)} \right] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{1}{p(x, y)}.$$

If we know Y , how much information is left in X ? This is denoted $H(X|Y)$. If, for example $X = Y + 2$, then $H(X|Y) = 0$ since if we know Y , we also know X . Conditional entropy is defined as

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) H(X|Y = y) = E \left[\log \frac{1}{p(X|Y)} \right] = H(X, Y) - H(Y)$$

Mutual information, $I(X; Y)$, represents the amount of information that one random variable has about the other, and is defined as

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

While this quick overview is sufficient for our purposes in this course, if you are interested, you can check out the slides for this [Short Lecture on Information Theory](#), or the course [Mathematics of Information](#).

Chapter 2

Frequentist Parameter Estimation

2.1 Overview

Parameter estimation can be used to infer unknowns about the real world (e.g, the frequency of a given disease among individuals with a certain genetic mutation) and to estimate the distribution of the data in machine learning problems.

There are two main frameworks for parameter estimation:

- Frequentist methods: In the frequentists' perspective, the true parameter value θ^* is unknown and fixed. The estimate $\hat{\theta}$ is a function of the data, which provides a single “best” estimate of θ^* . Frequentists have different methods for estimation including *maximum likelihood*, which we will discuss in detail, and the *moment method*, which finds the parameters by solving equations obtained by equating empirical moments and theoretical moments.
- Bayesian methods: Parameters are considered to be random and are treated as such. The Bayesian method provides a unified approach consisting of the following steps:
 1. Start with the prior distribution for the parameter
 2. Collect data
 3. Obtain posterior distribution by updating the prior distribution using data and Bayes' theorem

2.2 Maximum likelihood estimation

Suppose data x is collected. We model this data as a realization of a random variable X with distribution p_X , which has an unknown parameter θ^* . The probability of observing x , assuming θ , is $p_X(x; \theta)$. To estimate θ^* , **Maximum likelihood estimation (MLE)** chooses the parameter that assigns the highest probability to the data:

$$\hat{\theta}_{\text{mle}} = \arg \max_{\theta} p_X(x; \theta).$$

The expression $p(x; \theta)$, viewed as a function of θ , is called the **likelihood**; hence the name maximum likelihood estimation. As shorthand, we use $L(\theta) = p_X(x; \theta)$ and $\ell(\theta) = \ln L(\theta)$, where $\ell(\theta)$ is the **log-likelihood**. Clearly, the value of θ that maximizes $L(\theta)$ is the same as the one that maximizes $\ell(\theta)$:

$$\hat{\theta}_{\text{mle}} = \arg \max_{\theta} \ell(\theta) = \arg \max_{\theta} \ln p_X(x; \theta)$$

Example 2.1. In this example, we attempt to show the intuition behind maximum likelihood. Suppose that a given road has heavy traffic or light traffic. We denote the probability of light traffic by θ^* . To estimate

data, we count the number of times X that the road has light traffic in a period of 100 days. After collecting this data, we observe that $X = 65$. We have

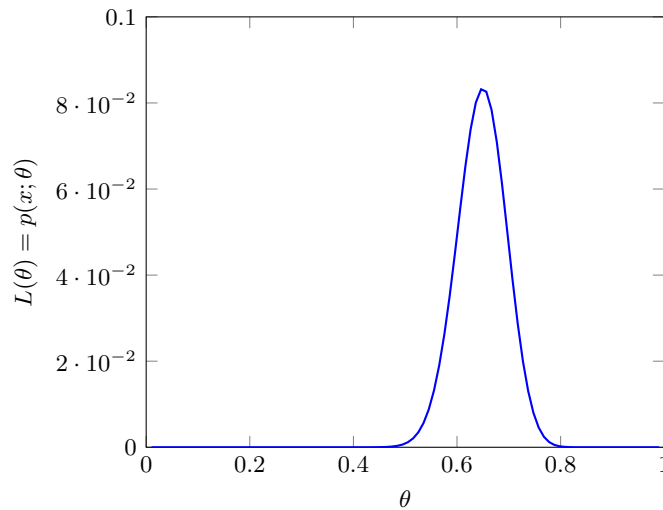
$$p_X(65; \theta) = \binom{100}{65} \theta^{65} (1 - \theta)^{35}$$

Let's try a few different choices for θ , e.g., $\theta \in \{0.2, 0.4, 0.6, 0.8\}$, and see which one makes more sense:

$$\begin{aligned} p(65, \theta = 0.2) &= 1.6 \times 10^{-22}, \\ p(65, \theta = 0.4) &= 0.00000026, \\ p(65, \theta = 0.6) &= 0.0491, \\ p(65, \theta = 0.8) &= 0.00019, \end{aligned}$$

If $\theta = 0.2$, the probability of 65 days with light traffic is extremely small. So observing $x = 65$ would be very unlikely, which in turn would make $\theta = 0.2$ an unreasonable guess. Among the presented choices, $\theta = 0.6$ appears the most reasonable. This reasoning suggests the following: *The value of the parameter that assigns a higher probability to the observation is a better choice.*

Since we are not limited to a specific set of choices, we can find the parameter that **maximizes** the probability of the observation. In the figure below, $L(\theta) = p(x; \theta)$ is plotted as a function of θ . This is the likelihood function.



We can see that $\theta = 0.65$ maximizes the likelihood and hence is the maximum-likelihood estimate. We can also show this analytically. First, the likelihood is given as

$$L(\theta) = p(x; \theta) = \binom{100}{65} \theta^{65} (1 - \theta)^{35}.$$

We usually use the log-likelihood as the function to optimize:

$$\ell(\theta) = \log L(\theta) = \log \left(\binom{100}{65} \theta^{65} (1 - \theta)^{35} \right) \doteq 65 \log \theta + 35 \log(1 - \theta), \quad (2.1)$$

where \doteq denotes equality but with ignoring additive terms that are constant in θ (and thus do not alter the value of θ that maximize the log-likelihood). We differentiate $\ell(\theta)$ to find the value of θ that maximizes $\ell(\theta)$.

$$\frac{d\ell(\theta)}{d\theta} = \frac{65}{\theta} - \frac{35}{1 - \theta} = 0 \implies 65 - 65\theta = 35\theta \implies \hat{\theta}_{\text{mle}} = \frac{65}{100}. \quad (2.2)$$

Note that this result is intuitive as it agrees with our observation that 65% of the days had light traffic. \triangle

A note on notation: In general, our data is a vector, which we denote by bold symbols such as \mathbf{x} . The corresponding random variable is \mathbf{X} .

Example 2.2 (Parameters of the normal distribution). A device for measuring an unknown quantity μ^* (e.g., the mass of an electron) is used n times producing values $\mathbf{Y} = (Y_1, \dots, Y_n)$. Each measurement is independent and for each i we have $Y_i = \mu^* + Z_i$, where Z_i is the measurement noise satisfying $Z_i \sim \mathcal{N}(0, (\sigma^*)^2)$. Note that this implies $Y_i \sim \mathcal{N}(\mu^*, (\sigma^*)^2)$.

Suppose we have collected data $\mathbf{y} = (y_1, \dots, y_n)$. We consider the problem in two cases: μ^* is unknown but σ^* is known; and both μ^* and σ^* are unknown.

- Known σ^* , unknown μ^* : We have

$$p_{Y_i}(y_i; \mu) = \frac{1}{\sigma^* \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{y_i - \mu}{\sigma^*}\right)^2\right)$$

$$L(\mu) = p_{\mathbf{Y}}(\mathbf{y}; \mu) = \prod_{i=1}^n p_{Y_i}(y_i; \mu)$$

$$\ell(\mu) = \sum_{i=1}^n \ln p_{Y_i}(y_i; \mu) = \sum_{i=1}^n \left(-\ln(\sigma^* \sqrt{2\pi}) - \frac{1}{2} \left(\frac{y_i - \mu}{\sigma^*}\right)^2 \right) \doteq -\frac{1}{2} \sum_{i=1}^n \left(\frac{y_i - \mu}{\sigma^*}\right)^2$$

and so

$$\frac{d\ell}{d\mu} = \sum_{i=1}^n \frac{y_i - \mu}{\sigma^*} = 0 \implies \hat{\mu}_{\text{mle}} = \frac{1}{n} \sum_{i=1}^n y_i = \bar{y}.$$

- Unknown σ^*, μ^* : We have

$$\ell(\mu, \sigma) = \sum_{i=1}^n \left(-\ln(\sigma \sqrt{2\pi}) - \frac{1}{2} \left(\frac{y_i - \mu}{\sigma}\right)^2 \right) \doteq -n \ln \sigma - \frac{1}{2} \sum_{i=1}^n \left(\frac{y_i - \mu}{\sigma}\right)^2$$

and so

$$\frac{\partial \ell}{\partial \mu} = \sum_{i=1}^n \frac{y_i - \mu}{\sigma} = 0,$$

$$\frac{\partial \ell}{\partial \sigma} = -\frac{n}{\sigma} + \sum_{i=1}^n \frac{(y_i - \mu)^2}{\sigma^3} = 0.$$

Solving this system of equations for μ and σ yields

$$\hat{\mu}_{\text{mle}} = \frac{1}{n} \sum_{i=1}^n y_i = \bar{y},$$

$$\hat{\sigma}_{\text{mle}}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2.$$

\triangle

2.2.1 Maximum likelihood and the closest distribution

We have described maximum likelihood as aiming to find a distribution that gives a high probability to the observed data. An alternative view relates it to the empirical distribution of the data, denoted $p_{\mathbf{x}}$. Given $\mathbf{x} = \{x_1, \dots, x_n\}$, let $\#_x$ denote the number of times x appears in \mathbf{x} . The empirical distribution is given as

$$p_{\mathbf{x}}(x) = \frac{\#_x}{n} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(\mathbf{x} = \mathbf{x}_i)$$

where $\mathbb{1}(\cdot)$ equals 1 if the enclosed condition is true and 0 otherwise.

Now consider a parameterized family of distributions p_{θ} . It makes sense to choose θ such that p_{θ} is close to $p_{\mathbf{x}}$. In other words, we would like θ to be chosen such that p_{θ} describes the observed data well. A standard way of measuring the “closeness” of p_{θ} to the empirical distribution $p_{\mathbf{x}}$ is relative entropy, $D_{KL}(p_{\mathbf{x}}||p_{\theta})$.

It turns out the closest distribution is in fact given by maximum likelihood, i.e.,

$$\hat{\theta}_{\text{mle}} = \arg \min_{\theta} D_{KL}(p_{\mathbf{x}}||p_{\theta}). \quad (2.3)$$

This fact provide further evidence for the soundness of MLE strategy. Note in particular that if there exists θ such that $p_{\theta} = p_{\mathbf{x}}$, it will be chosen by MLE. This is because relative entropy is always non-negative and equals to 0 if and only if the two distributions are the same. So choosing $p_{\theta} = p_{\mathbf{x}}$, if possible, provides the smallest value for the relative entropy, i.e., 0.

Exercise 2.3. Prove (2.3). △

Exercise 2.4. (†) Note that relative entropy is not symmetric. Instead of $D_{KL}(p_{\mathbf{x}}||p_{\theta})$, we could minimize $D_{KL}(p_{\theta}||p_{\mathbf{x}})$. What are the differences between the two formulations and which one is more suitable for parameter estimation? △

2.3 Properties of Estimators

Maximum likelihood is just one way of estimating parameters. We can choose any function of the data as the estimate. For instance, in Example 2.2, we could choose the middle (median) value among y_1, \dots, y_n as the estimate for μ^* . Given the fact that there are many estimators, how do we evaluate them and select one?

Clearly, we would like the estimate to be close to the true value. But stating this condition in a rigorous probabilistic way is a bit challenging in the frequentist framework. We are specifically interested in the error:

$$\hat{\theta}(\mathbf{x}) - \theta^*,$$

where $\hat{\theta}(\mathbf{x})$ is the estimate based on data \mathbf{x} and θ^* is the true value¹. Evaluating $\hat{\theta}(\mathbf{x})$ is difficult because, obviously, the true value is unknown.

So instead of finding the specific error, we may try to find the probability that the true value θ^* is within say 10% of the estimate $\hat{\theta}$. But after the estimate is produced based on a given data set, the estimate is a deterministic value. For instance, in Example 2.1, the MLE is given as $\hat{\theta}_{\text{mle}} = 0.65$. So questions such as “What is the probability that the difference between θ^* and $\hat{\theta}(\mathbf{x})$ is larger than 0.05?” are not meaningful because, while θ^* is unknown, both θ^* and $\hat{\theta}(\mathbf{x})$ are deterministic after data is collected and the estimation task is performed.

The solution to these difficulties is to study the properties of the estimator not based on a specific realization \mathbf{x} of the data but in general, over all possible data sets that could be produced and all the resulting estimated values. We can think of the thought experiment in which many, many, data sets are collected and the

¹Note the slight abuse of notation: sometimes θ is used as the generic parameter, e.g., as the argument of the likelihood function, and sometimes as the true value of the parameter. The distinction should be clear from the context

estimation task is performed based on each. The estimate itself is a random variable because each time we perform the estimation task, new data samples are obtained and these are random, following a certain distribution. In other words, instead of considering a single estimate $\hat{\theta}(\mathbf{x})$ for a specific realization \mathbf{x} , we study the estimator $\hat{\theta}(\mathbf{X})$, i.e., a random variable. Then it makes sense to ask “What is the probability that the difference between θ^* and $\hat{\theta}(\mathbf{X})$ is larger than 0.05?” since $\hat{\theta}(\mathbf{X})$ is a random variable with some distribution. It may be difficult to find the distribution of $\hat{\theta}(\mathbf{x})$ and it may depend on the unknown parameter θ^* but at least the question is meaningful. In this section, we will see some of the evaluation criteria based on this view.

A note on notation: Typically, we use θ as the generic parameter, with θ^* denoting its true value, according to which \mathbf{X} is distributed. For a given data \mathbf{x} , the estimate is shown by $\hat{\theta}(\mathbf{x})$ or $\hat{\theta}$. So, $\hat{\theta}$ denotes both the estimator, i.e., a function that produces the estimate given the data, and the estimate; the intent should be clear from the context. Finally, we may use $\hat{\Theta} = \hat{\theta}(\mathbf{X})$ to denote the estimate as a random variable.

2.3.1 Bias

Bias is the expected estimation error,

$$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}(\mathbf{X}) - \theta^*] = \mathbb{E}[\hat{\theta}(\mathbf{X})] - \theta^* \quad (2.4)$$

As discussed, the expected value is taken over the randomness in \mathbf{X} . Bias of the estimator tells us whether in general the estimator over- or under-estimates the true value. If bias is equal to 0, then the estimator is called **unbiased**.

Example 2.5 (Example 2.1 continued). Previously, we obtained the maximum likelihood estimate for the probability θ of having light traffic. Let us find its bias. Again we collect data over 100 days and let X denote the number of days when there is light traffic. We know that $\hat{\theta}_{\text{mle}}$, as a function of data, is given by

$$\hat{\theta}_{\text{mle}}(X) = \frac{X}{100},$$

Note that instead of using a specific value for the number of days with light traffic, such as 65, we use a random variable X representing this quantity. Dropping the dependence on X for simplicity, the expected value of $\hat{\theta}_{\text{mle}}$ is given by

$$\mathbb{E}[\hat{\theta}_{\text{mle}}(\mathbf{X})] = \frac{\mathbb{E}[X]}{100}.$$

Assuming θ^* to be the true value, the number X of days when there is light traffic follows $\text{Bin}(100, \theta^*)$, and so $\mathbb{E}[X] = 100\theta^*$. It follows that

$$\mathbb{E}[\hat{\theta}_{\text{mle}}(\mathbf{X})] = \frac{100\theta^*}{100} = \theta^*.$$

Hence, the maximum likelihood estimate is an unbiased estimator. \triangle

Example 2.6. Given iid data $\mathbf{y} = (y_1, \dots, y_n)$, $n \geq 3$, with mean θ^* , let us find the bias of each of the following estimators,

$$\begin{aligned} \hat{\theta}_1(\mathbf{y}) &= \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \\ \hat{\theta}_2(\mathbf{y}) &= y_1, \\ \hat{\theta}_3(\mathbf{y}) &= \frac{2y_2 + y_3}{3}. \end{aligned}$$

Let Y_i be the random variable corresponding to observation y_i and $\bar{Y} = \sum_{i=1}^n Y_i$. We have

$$\begin{aligned}\mathbb{E} \hat{\theta}_1(\mathbf{Y}) &= \mathbb{E} \bar{Y} = \frac{1}{n} \sum_{i=1}^n \mathbb{E} Y_i = \frac{1}{n} \sum_{i=1}^n \theta^* = \theta^*, \\ \mathbb{E} \hat{\theta}_2(\mathbf{Y}) &= \mathbb{E} Y_1 = \theta^*, \\ \mathbb{E} \hat{\theta}_3(\mathbf{Y}) &= \mathbb{E} \left[\frac{2Y_2 + Y_3}{3} \right] = \frac{2 \mathbb{E} Y_2 + \mathbb{E} Y_3}{3} = \theta^*.\end{aligned}$$

So all of these estimators are unbiased. △

Example 2.7. Given n samples $\mathbf{y} = (y_1, \dots, y_n)$ from a distribution with mean μ^* and variance $(\sigma^*)^2$, are the estimators

$$\hat{\mu} = \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

for the mean and variance, respectively, unbiased?

For $\hat{\mu}$, we have

$$\mathbb{E}[\hat{\mu}(\mathbf{Y})] = \mathbb{E}[\bar{Y}] = \mathbb{E} \left[\frac{1}{n} \sum_{i=1}^n Y_i \right] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[Y_i] = \frac{1}{n} n \mathbb{E}[Y_1] = \mu^*$$

and so the estimator for the mean is unbiased. We can show (how?) that

$$\mathbb{E}[\hat{\sigma}^2(\mathbf{Y})] = \frac{n-1}{n} (\sigma^*)^2$$

and the bias of estimating $(\sigma^*)^2$ is

$$\mathbb{E}[\hat{\sigma}^2(\mathbf{Y})] - (\sigma^*)^2 = -\frac{1}{n} (\sigma^*)^2.$$

Based on this, we can create an unbiased estimator for the variance as

$$\hat{\sigma}_u^2(\mathbf{y}) = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2.$$

△

Example 2.8. [1, Example 2.8.2] An urn has m^* balls, numbered $1, 2, \dots, m^*$. Suppose however that m^* is unknown to us. We pick one random ball from the urn and the number on the ball is y . We estimate m^* using maximum likelihood. First, let Y be the random variable corresponding to observation y , with distribution $p_Y(y; m^*)$. We have

$$p_Y(y; m) = \begin{cases} \frac{1}{m} & y \leq m, \\ 0 & y > m. \end{cases}$$

and thus

$$L(m) = \begin{cases} \frac{1}{m} & m \geq y, \\ 0 & m < y. \end{cases}$$

Hence, $L(m)$ is maximized by choosing $m(y) = y$ and so $\hat{m}_{\text{mle}} = y$. To find the bias of \hat{m}_{mle} ,

$$\begin{aligned}\mathbb{E}[\hat{m}_{\text{mle}}(Y)] &= \mathbb{E}[Y] = \sum_{i=1}^{m^*} i \cdot \frac{1}{m^*} = \frac{m^* + 1}{2}, \\ \text{Bias}(\hat{m}_{\text{mle}}) &= \frac{m^* + 1}{2} - m^* = -\frac{m^* - 1}{2},\end{aligned}$$

which means that the ML estimator tends to underestimates m^* by almost a factor of 2. △

Example 2.9 (Linear unbiased estimator). Can we design an unbiased estimator for Example 2.8? There are many options, but for simplicity we may choose an estimator that is linear in the data, in particular, one of the form

$$\hat{m}_L(y) = ay + b.$$

We find a and b such that \hat{m}_L is unbiased. We have

$$\mathbb{E}[\hat{m}_L(Y)] = a \mathbb{E}Y + b = a \frac{m^* + 1}{2} + b.$$

Setting this equal to m^* (equality should hold for any m^*) yields $a = 2$ and $b = -1$, i.e.,

$$\hat{m}_L(y) = 2y - 1.$$

△

Example 2.10 (Survival of Humanity (!)). The human species will eventually die out. We use two methods to estimate the total number of humans m who will ever live. Let humans be enumerated by birth order as $h_1, h_2, \dots, h_y, \dots, h_m$, where h_1 represents Adam, h_2 represents Eve, h_y represents you, and h_m represents the last human to live. Assuming that your birth order y is random, the problem is similar to estimating the number of balls in an urn in Example 2.8.

Assuming that 100 billion humans have been born so far, we have $\hat{m}_{\text{mle}} = 100$ billion and $\hat{m}_L = 200$ billion. The ML estimate predicts that the end is here. Further, assuming that there will be 140 million births each year, the unbiased estimator predicts the end of humanity to occur in around 700 years. △

2.3.2 Mean squared error and variance

Example 2.11. Consider an unbiased estimator $\hat{\theta}$ and define $\hat{\theta}' = \hat{\theta} + W$, where W is a zero-mean random variable with a large variance. Now, $\hat{\theta}'$ is unbiased, similar to $\hat{\theta}$, but it is not a good estimator (regardless of how good $\hat{\theta}$ is). So clearly, being unbiased alone is not sufficient to ensure that an estimator is “good.” △

For an estimator $\hat{\theta}$, where the random variable describing data is denoted by \mathbf{X} , the mean squared error (MSE) is defined as

$$\text{MSE}(\hat{\theta}) = \mathbb{E} \left[\left(\hat{\theta}(\mathbf{X}) - \theta^* \right)^2 \right].$$

The smaller the MSE, the more accurate the estimator.

Let $\hat{\Theta} = \hat{\theta}(\mathbf{X})$. Note that

$$\begin{aligned} \text{MSE}(\hat{\theta}) &= \mathbb{E} \left[\left(\hat{\Theta} - \theta^* \right)^2 \right] \\ &= \mathbb{E} \left[\left((\hat{\Theta} - \mathbb{E} \hat{\Theta}) + (\mathbb{E} \hat{\Theta} - \theta^*) \right)^2 \right] \\ &= \mathbb{E} \left[(\hat{\Theta} - \mathbb{E} \hat{\Theta})^2 \right] + (\mathbb{E} \hat{\Theta} - \theta^*)^2 + 2 \mathbb{E} \left[(\hat{\Theta} - \mathbb{E} \hat{\Theta}) \right] (\mathbb{E} \hat{\Theta} - \theta^*) \\ &= \mathbb{E} \left[(\hat{\Theta} - \mathbb{E} \hat{\Theta})^2 \right] + (\mathbb{E} \hat{\Theta} - \theta^*)^2, \end{aligned}$$

where, the third equality uses the fact that $\mathbb{E} \hat{\Theta} - \theta^*$ is a deterministic constant and the fourth equality the fact that $\mathbb{E} \left[(\hat{\Theta} - \mathbb{E} \hat{\Theta}) \right] = 0$. Hence,

$$\text{MSE}(\hat{\theta}) = \text{Var}(\hat{\theta}) + (\text{Bias}(\hat{\theta}))^2.$$

For unbiased estimators, the variance is an important quantity since it is equal to the MSE.

Example 2.12 (Example 2.1 re-revisit). We saw in Example 2.5 that the maximum likelihood estimate for the probability of traffic θ^* is unbiased. Now, let us find its variance. Again, we write $\hat{\theta}_{\text{mle}}(\mathbf{X}) = \frac{X}{100}$ and

$$\text{Var}(\hat{\theta}_{\text{mle}}) = \frac{\text{Var}(X)}{100^2} = \frac{\theta^*(1 - \theta^*)}{100},$$

where X is the number of days without traffic, which follows $\text{Bin}(100, \theta^*)$ with variance $100\theta^*(1 - \theta^*)$. As we can see, the variance (hence, MSE) increases as the true value of θ^* approaches $1/2$, i.e., every data point contains more uncertainty. Furthermore, we can extend this result to the more general case where we collect data for n days. By the same argument, we get

$$\text{MSE}(\hat{\theta}_{\text{mle}}) = \text{Var}(\hat{\theta}_{\text{mle}}) = \frac{\theta^*(1 - \theta^*)}{n}.$$

△

Example 2.13. Consider data $\mathbf{y} = (y_1, \dots, y_n)$, where the corresponding random variables Y_i are iid with distribution $\mathcal{N}(\mu, \sigma^2)$. The ML estimator for the mean μ is $\hat{\theta}_{\text{mle}}(\mathbf{y}) = \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ is unbiased. We have

$$\text{MSE}(\hat{\theta}_{\text{mle}}) = \text{Var}(\bar{Y}) = \frac{\sigma^2}{n}.$$

△

Note that as n increases, the MSE decreases and the estimate becomes more accurate, as would be expected. This property is studied next.

Exercise 2.14. For the estimators in Example 2.6, find the MSE, assuming the variance is $(\sigma^*)^2$. △

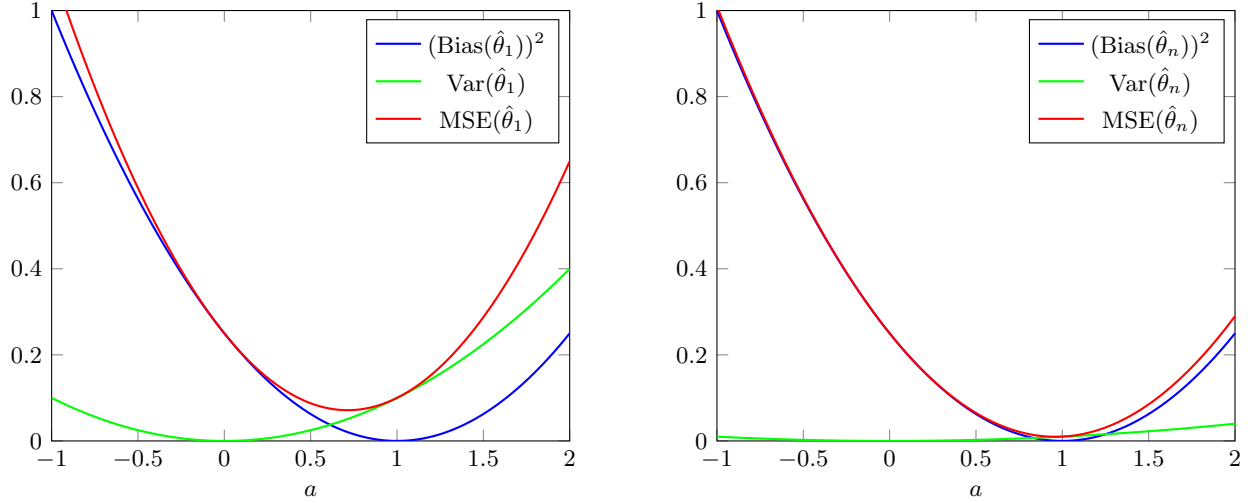
Exercise 2.15 (Bias-variance trade-off). Given iid data $\mathbf{y} = (y_1, \dots, y_n)$, $n \geq 3$, with mean θ^* and variance σ^2 , show that the MSE of

$$\begin{aligned} \hat{\theta}_1 &= ay_1, \\ \hat{\theta}_n &= a\bar{y} = \frac{a}{n} \sum_{i=1}^n y_i, \end{aligned}$$

for some constant $a \in \mathbb{R}$ is given as

$$\begin{aligned} \text{MSE}(\hat{\theta}_1) &= (a - 1)^2(\theta^*)^2 + a^2\sigma^2, \\ \text{MSE}(\hat{\theta}_n) &= (a - 1)^2(\theta^*)^2 + a^2\sigma^2/n. \end{aligned}$$

What is a good value for a ? Does anything other than $a = 1$ make sense? The components of the MSE are given in the plots below for $\hat{\theta}_1$ and $\hat{\theta}_n$ with $n = 10$, for $\theta^* = 0.5, \sigma^2 = 0.1$. A trade-off between the bias and variance is evident. Why is it not feasible to design an estimator by optimizing for a ? What is the difference between estimation based on little data ($\hat{\theta}_1$) and a lot of data ($\hat{\theta}_n, n = 10$)?



△

2.3.3 Consistency

Consider an estimator $\hat{\theta}_n(\mathbf{x})$ based on n samples $\mathbf{x} = (x_1, \dots, x_n)$. Let $\mathbf{X} = (X_1, \dots, X_n)$ be the random variables that describe the n data samples and let $\hat{\Theta}_n = \hat{\theta}_n(\mathbf{X})$ be the random variable that corresponds to the estimate. The estimator $\hat{\theta}_n$ is said to be **consistent** if $\hat{\Theta}_n \rightarrow \theta^*$ as $n \rightarrow \infty$. More precisely, for all $\epsilon > 0$, we need

$$\lim_{n \rightarrow \infty} \Pr(|\hat{\Theta}_n - \theta^*| \geq \epsilon) = 0.$$

In other words, the estimator is accurate if the size of the data is large.

Example 2.16. The ML and linear estimators described in Examples 2.8 and 2.9 are very different for a single data point. But how do they behave if we have a lot of data. First we need to define these for n data samples. Suppose that we take n samples from the urn with replacement, resulting in $\mathbf{y} = (y_1, y_2, \dots, y_n)$. Define

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

To extend the linear estimator to n data points, we can choose

$$\hat{m}_{L,n} = 2\bar{y} - 1.$$

For the ML estimator, we have (why?)

$$\hat{m}_{\text{mle},n} = \max_i y_i.$$

Both of these, although they look very different, are consistent and converge to m^* as $n \rightarrow \infty$.

- As $n \rightarrow \infty$, by LLN, \bar{Y} converges to the mean of the distribution, i.e., $\mathbb{E}[Y_1] = \frac{m^*+1}{2}$. Hence, $\hat{m}_{L,n} \rightarrow 2 \cdot \frac{m^*+1}{2} - 1 = m^*$.
- For the ML estimator, as $n \rightarrow \infty$, at some point, we will pick the ball numbered m^* and so we will eventually have $\hat{m}_{\text{mle}} = m^*$.

Given the two estimators, the bad news is that the estimators disagree significantly for small data. However, as the size of the sample data increases, the two estimators agree. △

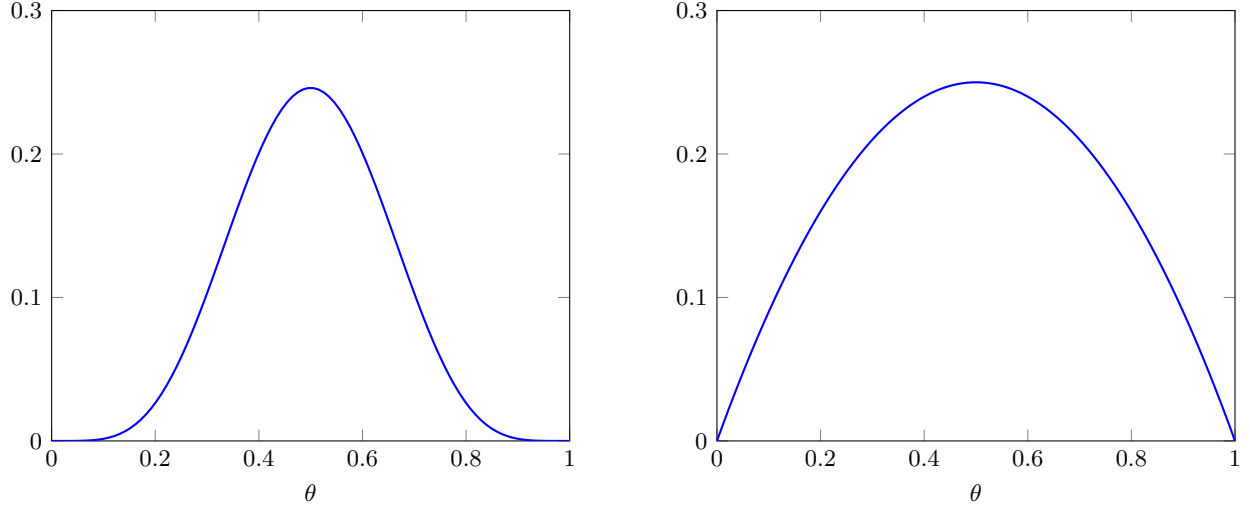


Figure 2.1: The likelihood function on the left demonstrates strong dependence on θ compared to the one on the right.

2.4 The Cramer-Rao lower bound*

For an unbiased estimator, the MSE is equal to the variance, and thus the variance represents the accuracy of the estimator. This leads to the following question: *For a given distribution of data, what is the smallest possible variance of an unbiased estimator?*

The accuracy of estimating a parameter θ depends on how strongly the distribution of the data \mathbf{X} depends on θ . If the dependence is strong, i.e., for values of θ other than the true value θ^* , the probability of the observed data falls sharply, then we may expect to find θ^* with accuracy. On the other hand, if the dependence is weak, then it will be difficult to find θ^* with precision. These two cases are shown in Figure 2.1.

Let the data be encoded as a vector \mathbf{X} , whose distribution is given by p with parameter θ^* . Assuming $\mathbf{X} = \mathbf{x}$, the log-likelihood is $p(\mathbf{x}; \theta)$. The sharpness of the log-likelihood $\ell(\theta)$ at the true value θ^* can be quantified as

$$-\left. \frac{\partial^2 \ell(\theta)}{\partial \theta^2} \right|_{\theta=\theta^*} = -\left. \frac{\partial^2 \ln p(\mathbf{x}; \theta)}{\partial \theta^2} \right|_{\theta=\theta^*}. \quad (2.5)$$

Given the randomness of the data \mathbf{X} , the above quantity is random,

$$-\left. \frac{\partial^2 \ln p(\mathbf{X}; \theta)}{\partial \theta^2} \right|_{\theta=\theta^*}$$

So to average over the data, we define

$$I(\theta^*) = -\mathbb{E} \left[\left. \frac{\partial^2 \ln p(\mathbf{X}; \theta)}{\partial \theta^2} \right|_{\theta=\theta^*} \right] = -\int \left. \frac{\partial^2 \ln p(\mathbf{x}; \theta)}{\partial \theta^2} \right|_{\theta=\theta^*} p(\mathbf{x}; \theta^*) d\mathbf{x},$$

which is called the **Fisher Information**.

The following theorem provides a lower bound on the variance, which is referred to as the Cramer-Rao lower bound (CRLB).

Theorem 2.17 (CRLB). *Given that the log-likelihood $\ell(\theta)$ satisfies certain regularity conditions, the variance of any unbiased estimator $\hat{\theta}$ of θ^* satisfies*

$$\text{Var}(\hat{\theta}) \geq \frac{1}{I(\theta^*)}.$$

If an estimator achieves the CRLB, i.e., $\text{Var}(\hat{\theta}) = 1/I(\theta^*)$, then it is called **efficient**.

As a special case, consider when we have n iid data points, and denote the estimator based on this data as $\hat{\theta}_n$. Denote the Fisher information based on n data points as $I_n(\theta^*)$ and based on one data point as $I_1(\theta^*) = I(\theta^*)$. Since the Fisher information is additive (Why? Hint: definition), we have $I_n(\theta^*) = nI(\theta^*)$. Thus, the variance of an unbiased estimator $\hat{\theta}_n$ based on n independent observations satisfies

$$\text{Var}(\hat{\theta}_n) \geq \frac{1}{nI(\theta^*)}. \quad (2.6)$$

Example 2.18. In Example 2.2, where we estimated the mean μ^* of a Gaussian distribution with known σ^2 based on n iid samples y_1, \dots, y_n , the log-likelihood, ignoring constant terms, was given as

$$\ell(\mu) \doteq -\sum_{i=1}^n \frac{(y_i - \mu)^2}{2\sigma^2}.$$

And,

$$\frac{\partial \ell(\mu)}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^n (y_i - \mu). \quad (2.7)$$

Observe that

$$\frac{\partial^2 \ell(\mu)}{\partial \mu^2} = -\frac{n}{\sigma^2} \implies I(\mu^*) = -\mathbb{E} \left[\frac{\partial^2 \ell(\mu^*)}{\partial \mu^2} \right] = \frac{n}{\sigma^2}.$$

Based on the CRLB, the variance of the estimator satisfies

$$\text{Var}(\hat{\mu}) \geq \frac{\sigma^2}{n}.$$

The variance of the estimator is $\text{Var}(\hat{\mu}) = \frac{\sigma^2}{n}$. Hence, the ML estimator is efficient in this case. \triangle

2.5 Asymptotic normality of the MLE

As shown before, the maximum-likelihood estimator is not necessarily unbiased. However, if we have a large amount of data, under some regularity conditions, the ML estimator $\hat{\Theta}_n$ based on n iid data points satisfies

$$\sqrt{n}(\hat{\Theta}_n - \theta^*) \rightarrow \mathcal{N}(0, I^{-1}(\theta^*)).$$

So for large data, $\hat{\Theta}_n$ is nearly normally distributed with mean θ^* (hence unbiased) and variance $I^{-1}(\theta^*)/n$ (efficient).

While we stated the CRLB and the asymptotic normality of the MLE for scalar parameters, almost identical results also hold for a vector of parameters.

References

- [1] Bruce Hajek. *Random Processes for Engineers*. Illinois, 2014. URL: <http://hajek.ece.illinois.edu/Papers/randomprocJuly14.pdf> (visited on 01/30/2017).

Chapter 3

Bayesian Parameter Estimation

3.1 From Prior to Posterior

In the Bayesian philosophy, unknown parameters are viewed as random. So, our knowledge about the parameter can be encoded as a distribution. The distribution representing our belief before observing the data is called the **prior distribution**. After we observe the data, our belief changes, resulting in the **posterior distribution**.

Specifically, the components of a Bayesian estimation problem are:

- Data \mathbf{x} : The data is a realization of a random variable \mathbf{X} . The distribution of \mathbf{X} depends on a parameter Θ .
- Parameter Θ : The parameter of the distribution of \mathbf{X} , which is unknown, and hence a random variable in the Bayesian framework.
- Joint and marginal distributions p : A joint distribution $p_{\mathbf{X},\Theta}$ and its marginals $p_{\mathbf{X}}$ and p_{Θ} .

The steps of Bayesian estimation of a parameter θ are:

1. Identifying the **prior** distribution, $p_{\Theta}(\theta)$. This is called the prior because it encodes our beliefs about Θ before seeing any data.
2. Collecting **data** \mathbf{x} and forming the likelihood: $p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)$
3. Finding the **posterior** distribution $p_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$ as

$$p_{\Theta|\mathbf{X}}(\theta|\mathbf{x}) = \frac{p_{\Theta}(\theta)p_{\mathbf{X}|\Theta}(\mathbf{x}|\theta)}{p_{\mathbf{X}}(\mathbf{x})}, \quad (3.1)$$

The distribution $p_{\Theta|\mathbf{X}}$ is called the posterior distribution since it encodes our knowledge about the parameter after observing the data. Usually, since the distribution is clear from the argument, we drop the subscripts of p , writing the above equation as

$$p(\theta|\mathbf{x}) = \frac{p(\theta)p(\mathbf{x}|\theta)}{p(\mathbf{x})}, \quad (3.2)$$

Normalizing distributions. Finding the posterior distribution requires computing the integral $p(\mathbf{x}) = \int_{\theta} p(\theta)p(\mathbf{x}|\theta)d\theta$. Since we have to compute an integral anyway, we might as well drop all multiplicative terms that are constant in θ and then normalize the final distribution. In particular, $p(\mathbf{x})$ is one such term. So we often first find a function *proportional* to $p(\theta|\mathbf{x})$ as

$$p(\theta|\mathbf{x}) \propto p(\theta)p(\mathbf{x}|\theta), \quad (3.3)$$

where we can also drop constant terms in θ from $p(\theta)$ and $p(\mathbf{x}|\theta)$. We can then normalize the result by integration. This is often difficult to do. Sometimes, given this function, we can identify the distribution. More generally, we can use computational methods, such as Markov Chain Monte Carlo, or approximation methods, such as variational inference, as we will see later. Finally, in certain cases, we can find what we need without any integration. For example, if our goal is to find the value of θ maximizing $p(\theta|\mathbf{x})$.

Example 3.1. Let Θ denote the unknown parameter of a geometric random variable X , where

$$p_{X|\Theta}(x|\theta) = \theta(1 - \theta)^{x-1}.$$

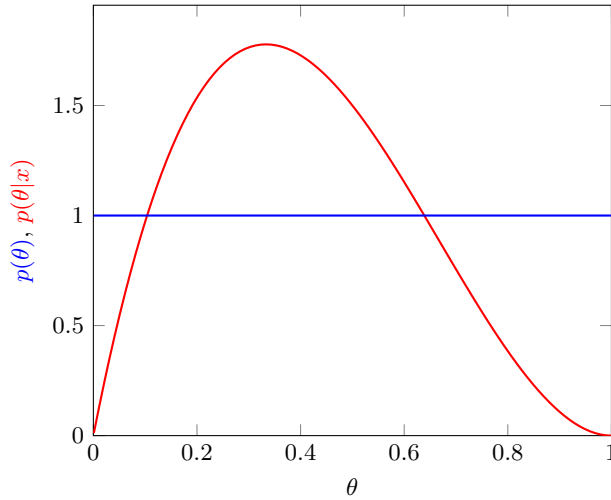
Suppose we observe $X = x$. We would like to estimate Θ based on this observation. If all possible values of Θ are equally likely, we may choose $\Theta \sim \text{Uni}(0, 1)$. We then have

$$p(\theta) = 1 \tag{3.4}$$

$$p(x|\theta) = \theta(1 - \theta)^{x-1} \tag{3.5}$$

$$p(\theta|x) \propto p(\theta)p(x|\theta) \propto \theta(1 - \theta)^{x-1} \tag{3.6}$$

The expression $\theta(1 - \theta)^{x-1}$ as a function of x is the geometric distribution. But as a function of θ , it is proportional to $\text{Beta}(2, x)$. As an example, if $x = 3$, then $\Theta|x \sim \text{Beta}(2, 3)$:



△

Exercise 3.2. The probability of 1 (success) in a Bernoulli experiment (e.g., flipping a coin, a system working or not working, etc) is Θ , which we would like to estimate. Suppose that the experiment is performed once and the outcome x is observed to be $x = 1$. Assuming a uniform prior, find the posterior distribution of Θ , i.e., $p_{\Theta|X}(\theta|1)$. △

Example 3.3. The probability of success in a Bernoulli experiment is Θ , which we would like to estimate. We show success in the i th trial with $y_i = 1$ and failure by $y_i = 0$.

- **Prior distribution:** Assuming that a priori we do not know anything about Θ , it is appropriate to choose $p_{\Theta} \sim \text{Uni}[0, 1]$, i.e., $p(\theta) = 1$ in the interval $[0, 1]$.
- **Likelihood:** We then perform the experiment n times. Suppose that we observe s successes and f failures. Let us denote this observation as $\mathbf{x} = (s, f)$. The likelihood is

$$p(\mathbf{x}|\theta) = \binom{n}{s} \theta^s (1 - \theta)^f \tag{3.7}$$

- The posterior distribution:

$$p(\theta|\mathbf{x}) \propto 1 \cdot \theta^s (1 - \theta)^f = \theta^s (1 - \theta)^f \quad (3.8)$$

We observe that this distribution is of the form of a beta distribution, $\text{Beta}(y; \alpha, \beta) \sim y^{\alpha-1} (1 - y)^{\beta-1}$. Hence,

$$p(\theta|\mathbf{x}) \sim \text{Beta}(s + 1, f + 1). \quad (3.9)$$

△

Note that since we are interested in Θ , we can drop multiplicative terms that are constant with respect to θ , such as $\binom{n}{s}$, in the above example.

Now that we have the posterior distribution, we can answer questions about the parameter, for example, What is the probability that $0.4 < \Theta < 0.6$?

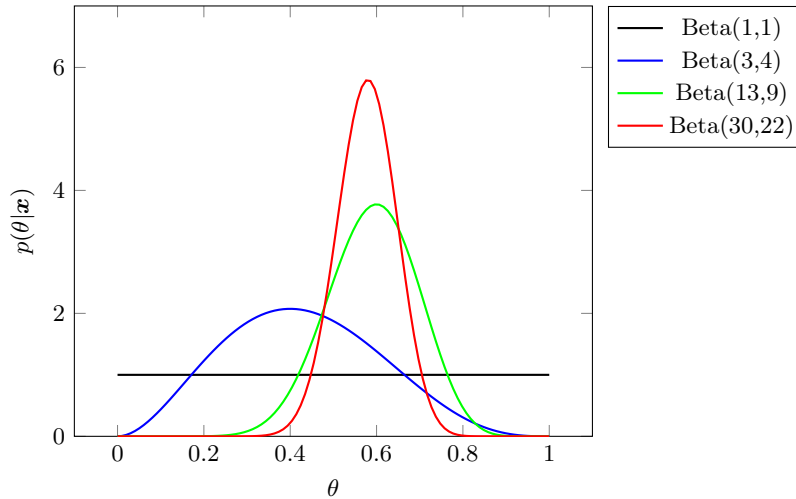
$$\int_{0.4}^{0.6} p(\theta|\mathbf{x}) d\theta \quad (3.10)$$

Example 3.4 (Consecutive Bayesian updating). Continuing the previous example, suppose that we collect more data $\mathbf{x}' = (s', f')$, consisting of s' successes and f' failures. Our prior distribution now is the posterior of the previous example, $p(\theta) \propto \theta^s (1 - \theta)^f$. We have

$$\begin{aligned} p(\mathbf{x}'|\theta) &= \binom{s' + f'}{s'} \theta^{s'} (1 - \theta)^{f'} \\ p(\theta|\mathbf{x}') &\propto \theta^s (1 - \theta)^f \theta^{s'} (1 - \theta)^{f'} \\ &= \theta^{s+s'} (1 - \theta)^{f+f'} \\ \Theta|\mathbf{x}' &\sim \text{Beta}(s + s' + 1, f + f' + 1). \end{aligned} \quad (3.11)$$

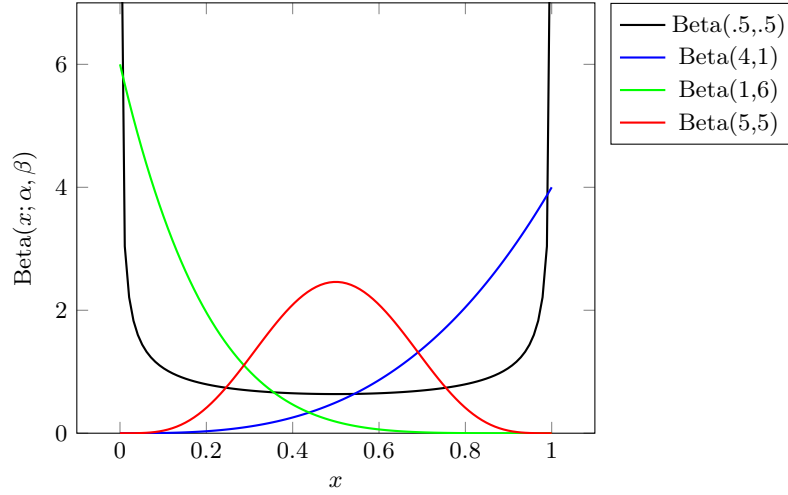
Equivalently, we can update our uniform prior $p(\theta) \propto 1$ with data $(s + s', f + f')$ to obtain $p(\theta|(s + s', f + f')) \sim \text{Beta}(s + s' + 1, f + f' + 1)$. As we can see, the Bayesian approach provides a way to update our belief in a consistent manner.

The figure below provides an example of the posterior with 0, 5, 20, and 50 samples. It can be observed that the posterior becomes sharper as more data is collected. △



Example 3.5. Beta is a common prior for the probability of Bernoulli experiments. Based on the discussion above, one way to interpret a Beta prior with parameters $\alpha \geq 1, \beta \geq 1$ is to imagine that, starting with the uniform prior, we have already collected $\alpha + \beta - 2$ samples, with $\alpha - 1$ successes. The following plot shows

the Beta distribution with different parameters to give a sense of the range of possible priors. \triangle



Example 3.6. Suppose that Y has a Poisson distribution with parameter Λ . That is, $p_{Y|\Lambda}(y|\lambda) = \text{Poi}(y; \lambda)$. Hence,

$$p(y|\lambda) = \frac{\lambda^y e^{-\lambda}}{y!}, \quad y \in \{0, 1, \dots\}$$

We intend to estimate Λ based on n iid samples $y_1^n = (y_1, \dots, y_n)$ of Y .

We assume that the prior for Λ is given as $p(\lambda) = \text{Gamma}(\lambda; \alpha, \beta) \propto \lambda^{\alpha-1} e^{-\beta\lambda}$. We have

$$p(\lambda) \propto \lambda^{\alpha-1} e^{-\beta\lambda} \quad (3.12)$$

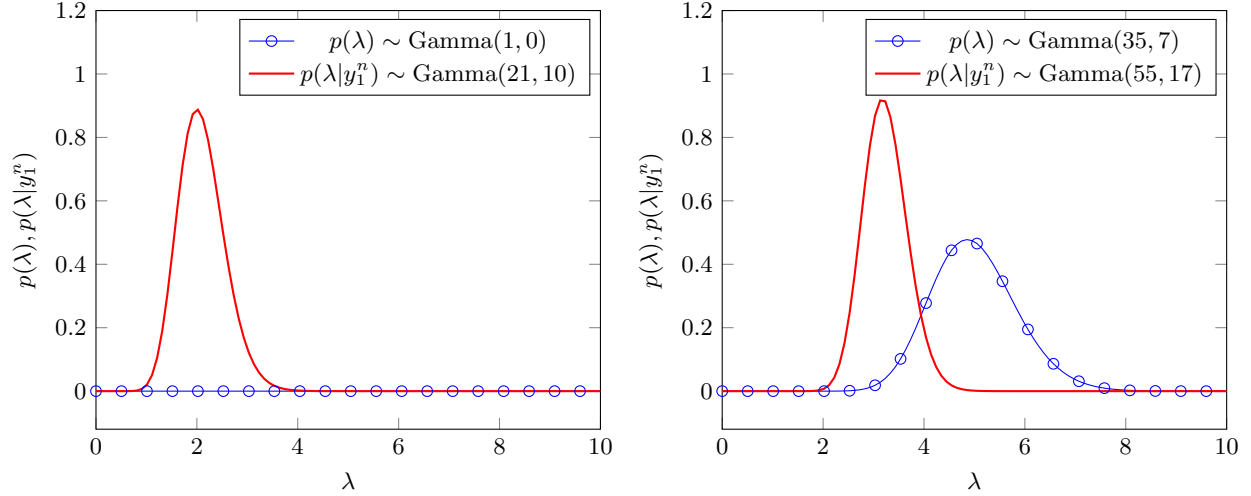
$$p(y_1^n|\lambda) = \prod_{i=1}^n \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} \propto \prod_{i=1}^n \lambda^{y_i} e^{-\lambda} = e^{-n\lambda} \lambda^{n\bar{y}}, \quad (3.13)$$

where $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$. Note that while $p(y_1^n|\lambda)$ is a distribution in y_1^n , we still dropped the $y_i!$ from its expression since our final goal is to find a distribution in λ and for this purpose terms that are independent of λ can be viewed as constant. The posterior is

$$p(\lambda|y_1^n) \propto \lambda^{\alpha-1} e^{-\beta\lambda} e^{-n\lambda} \lambda^{n\bar{y}} = \lambda^{\alpha+n\bar{y}-1} e^{-\lambda(n+\beta)} \propto \text{Gamma}(\lambda; \alpha + n\bar{y}, n + \beta). \quad (3.14)$$

If we choose $\alpha = 1, \beta = 0$, then the Gamma prior is flat, giving all possible values the same prior probability. But this is not a proper distribution. However, as long as the final posterior is a proper distribution, an **improper prior** is deemed acceptable.

Suppose that $n = 10$ and $\bar{y} = 2$. The figure below shows the posterior distribution with different priors. The prior on the left is called a **non-informative prior** because it is flat and the one on the right is an **informative prior** given that it represents a prior belief that certain values have a higher probability.



△

3.2 Bayesian Point Estimates

Having the complete distribution for $p_{\Theta|\mathbf{X}}(\theta|\mathbf{x})$ is useful since it provides the probability for different values for θ . But sometimes we want to estimate Θ with a single value $\hat{\theta} = \hat{\theta}(\mathbf{x})$ as a function of the data, similar to maximum likelihood. It is very important to note that in the Bayesian framework, the data is given and the estimate is known (and not random). The best choice for $\hat{\theta}$ then depends on how we characterize the estimation error:

Average Error	Optimal Estimator
$\mathbb{E}[(\Theta - \hat{\theta})^2 \mathbf{x}]$	$\hat{\theta} = \mathbb{E}[\Theta \mathbf{x}]$ (mean)
$\mathbb{E}[\Theta - \hat{\theta} \mathbf{x}]$	$\hat{\theta} = \mathbf{median}$ of $p(\theta \mathbf{x})$
$\Pr(\Theta \neq \hat{\theta} \mathbf{x}) = \mathbb{E}[I(\Theta \neq \hat{\theta}) \mathbf{x}]$	$\hat{\theta} = \arg \max_{\theta} p(\theta \mathbf{x})$ (mode)

In the table, $I(\text{condition})$ is 1 if the condition is satisfied and is 0 otherwise.

We prove the first case in the table. Let $\bar{\theta} = \mathbb{E}[\Theta|\mathbf{x}]$. We have

$$\mathbb{E}[(\hat{\theta} - \Theta)^2|\mathbf{x}] = \mathbb{E}[(\hat{\theta} - \bar{\theta}) - (\Theta - \bar{\theta})]^2|\mathbf{x}] \quad (3.15)$$

$$= \mathbb{E}[(\hat{\theta} - \bar{\theta})^2 - 2(\hat{\theta} - \bar{\theta})(\Theta - \bar{\theta}) + (\Theta - \bar{\theta})^2|\mathbf{x}] \quad (3.16)$$

$$= (\hat{\theta} - \bar{\theta})^2 - 2(\hat{\theta} - \bar{\theta}) \mathbb{E}[(\Theta - \bar{\theta})|\mathbf{x}] + \mathbb{E}[(\Theta - \bar{\theta})^2|\mathbf{x}] \quad (3.17)$$

$$= (\hat{\theta} - \bar{\theta})^2 + \mathbb{E}[(\Theta - \bar{\theta})^2|\mathbf{x}] \quad (3.18)$$

$$= (\hat{\theta} - \bar{\theta})^2 + \text{Var}(\Theta|\mathbf{x}) \quad (3.19)$$

$$\geq \text{Var}(\Theta|\mathbf{x}), \quad (3.20)$$

and the lower bound on the error is achieved when $\hat{\theta} = \bar{\theta}$.

Example 3.7. Generalizing Example 3.3 by assuming $p(\theta) = \text{Beta}(\theta; \alpha, \beta)$, we obtain $p(\theta|\mathbf{x}) = \text{Beta}(\theta; \alpha +$

$s, \beta + f$) (for Uniform, $\alpha = \beta = 1$). We have

$$\text{Mean} = \frac{s + \alpha}{s + f + \alpha + \beta}, \quad (3.21)$$

$$\text{Median} \simeq \frac{s + \alpha - 1/3}{s + f + \alpha + \beta - 2/3}, \quad (3.22)$$

$$\text{Mode} = \frac{s + \alpha - 1}{s + f + \alpha + \beta - 2}. \quad (3.23)$$

Generally speaking, Bayesian point estimates are between what is suggested only using the prior and what would be obtained using only the likelihood. For example, the mean of the prior is $\frac{\alpha}{\alpha + \beta}$ and the maximum likelihood solution is $\frac{s}{s + f}$. The mean of the posterior, $\frac{s + \alpha}{s + f + \alpha + \beta}$, is between these two. \triangle

3.3 Posterior Predictive Distribution

Given n iid samples, $y_1^n = (y_1, \dots, y_n)$, we are often interested in the distribution of the next (unobserved) value, $p_{Y_{n+1}|Y_1^n}(y_{n+1}|y_1^n)$. This distribution is referred to as *predictive posterior*. We have

$$p(y_{n+1}|y_1^n) = \int p(y_{n+1}, \theta|y_1^n) d\theta \quad (3.24)$$

$$= \int p(\theta|y_1^n) p(y_{n+1}|\theta, y_1^n) d\theta \quad (3.25)$$

$$= \int p(\theta|y_1^n) p(y_{n+1}|\theta) d\theta, \quad (3.26)$$

where we have used the fact that $Y_{n+1} \perp\!\!\!\perp Y_1^n | \Theta$. We have thus written the predictive posterior in terms of two known distributions.

Example 3.8. Continuing Example 3.3, let success in the $n + 1$ st experiment be denoted by $Y_{n+1} = 1$ and failure by $Y_{n+1} = 0$. We have

$$p_{Y_{n+1}|Y_1^n}(1|y_1^n) = \int \theta p(\Theta|y_1^n) = \mathbb{E}[\Theta|y_1^n] = \frac{s + 1}{s + f + 2}, \quad (3.27)$$

where we have used the facts that $p_{Y_{n+1}|\Theta}(1|\theta) = \theta$ and that the mean of $\text{Beta}(s + 1, f + 1)$ is $\frac{s + 1}{s + f + 2}$. \triangle

We may also ask about the expected value of Y_{n+1} given y_1^n , i.e., $\mathbb{E}[Y_{n+1}|y_1^n]$. We can find this by first finding $p(y_{n+1}|y_1^n)$ explicitly. But it is often easier to use the law of iterated expectations, since y_1^n influences Y_{n+1} through Θ . Recall that

$$\mathbb{E}[\mathbb{E}[Y|X]] = \mathbb{E}[Y], \quad \mathbb{E}[\mathbb{E}[Y|X, Z = z]|Z = z] = \mathbb{E}[Y|Z = z]. \quad (3.28)$$

Hence,

$$\mathbb{E}[Y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[Y_{n+1}|\Theta, y_1^n]|y_1^n] = \mathbb{E}[\mathbb{E}[Y_{n+1}|\Theta]|y_1^n], \quad (3.29)$$

where the last step follows from the fact that $Y_{n+1} \perp\!\!\!\perp Y_1^n | \Theta$, implying that $\mathbb{E}[Y_{n+1}|\Theta, y_1^n] = \mathbb{E}[Y_{n+1}|\Theta]$.

Example 3.9. Let's find $\mathbb{E}[Y_{n+1}|y_1^n]$ in Example 3.6. First, observe that $\mathbb{E}[Y_{n+1}|\Lambda] = \Lambda$. We hence need to find $\mathbb{E}[\Lambda|y_1^n]$. We know from before that $\Lambda|y_1^n$ is distributed according to $\text{Gamma}(\alpha + n\bar{y}, \beta + n)$. Therefore, $\mathbb{E}[Y_{n+1}|y_1^n] = \mathbb{E}[\Lambda|y_1^n] = \frac{\alpha + n\bar{y}}{\beta + n}$. \triangle

3.4 Gaussian Prior and Likelihood

Suppose that we want to estimate the mean of a Gaussian distribution with known variance,

$$p(y_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-\theta)^2}{2\sigma^2}} \quad (3.30)$$

given iid data $\{y_1, \dots, y_n\}$.

Improper priors. Assuming that we have no information about this mean, it makes sense to choose the prior

$$p(\theta) \propto 1. \quad (3.31)$$

But since the integral $\int_{-\infty}^{\infty} 1d\theta = \infty$, this does not lead to a valid distribution. Nevertheless, such a choice is acceptable, if the posterior is a valid distribution. Such priors are called *improper priors*. An improper prior does not necessarily have to be uniform.

Example 3.10. Consider the above likelihood and prior and let $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$. We have

$$p(\theta|y_1^n) \propto p(y_1^n|\theta) \cdot 1 \quad (3.32)$$

$$\propto \exp\left(-\frac{\sum_{i=1}^n (y_i - \theta)^2}{2\sigma^2}\right) \quad (3.33)$$

$$\propto \exp\left(-\frac{\sum_{i=1}^n (\theta^2 - 2y_i\theta + y_i^2)}{2\sigma^2}\right) \quad (3.34)$$

$$\propto \exp\left(-\frac{\theta^2 - 2\bar{y}\theta}{2\sigma^2/n}\right) \quad (3.35)$$

$$\propto \exp\left(-\frac{(\theta - \bar{y})^2}{2\sigma^2/n}\right) \quad (3.36)$$

$$\Theta|y_1^n \sim \mathcal{N}(\bar{y}, \sigma^2/n). \quad (3.37)$$

For the expected value of the next sample, we have

$$\mathbb{E}[Y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[Y_{n+1}|\Theta]|y_1^n] = \mathbb{E}[\Theta|y_1^n] = \bar{y}. \quad (3.38)$$

We can see more explicitly as well,

$$\mathbb{E}[Y_{n+1}|y_1^n] = \int y_{n+1} p(y_{n+1}|y_1^n) dy_{n+1} \quad (3.39)$$

$$= \int y_{n+1} \int p(y_{n+1}, \theta|y_1^n) d\theta dy_{n+1} \quad (3.40)$$

$$= \int y_{n+1} \int p(y_{n+1}|\theta) p(\theta|y_1^n) d\theta dy_{n+1} \quad (3.41)$$

$$= \int p(\theta|y_1^n) \int y_{n+1} p(y_{n+1}|\theta) dy_{n+1} d\theta \quad (3.42)$$

$$= \int \theta p(\theta|y_1^n) d\theta \quad (3.43)$$

$$= \mathbb{E}[\Theta|y_1^n] \quad (3.44)$$

$$= \bar{y}. \quad (3.45)$$

△

For the posterior predictive variance, we have

$$\text{Var}(Y_{n+1}|y_1^n) = \sigma^2 + \sigma^2/n. \quad (3.46)$$

It can be shown that $Y_{n+1}|y_1^n$ has a Gaussian distribution, and as we know its conditional mean and variance, we have

$$Y_{n+1}|y_1^n \sim \mathcal{N}(\bar{y}, \sigma^2 + \sigma^2/n) \quad (3.47)$$

From the variance, we can see that there are two sources of uncertainty. One is the inherent randomness in Y , quantified by σ^2 and the other is the result of the uncertainty of our estimate of the mean, quantified by σ^2/n .

Example 3.11 (†). Let us prove that $\text{Var}(Y_{n+1}|y_1^n) = \sigma^2 + \sigma^2/n$:

$$\begin{aligned} \text{Var}(Y_{n+1}|y_1^n) &= \mathbb{E}[(Y_{n+1} - \bar{y})^2|y_1^n] \\ &= \mathbb{E}[\mathbb{E}[(Y_{n+1} - \bar{y})^2|\Theta, y_1^n]|y_1^n] \\ &= \mathbb{E}[\sigma^2 + (\Theta - \bar{y})^2|y_1^n] \\ &= \sigma^2 + \mathbb{E}[(\Theta - \bar{y})^2|y_1^n] \\ &= \sigma^2 + \sigma^2/n \end{aligned}$$

△

We now consider the same problem with a proper Gaussian prior. Note that below as $\tau_0 \rightarrow \infty$, the proper prior below tends to the improper prior $p(\theta) \propto 1$.

Example 3.12. We would like to estimate the mean Θ of normally distributed independent values $y_1^n = (y_1, \dots, y_n)$. Let $\bar{y} = \sum y_i/n$. We assume

$$\Theta \sim \mathcal{N}(\theta_0, \tau_0^2) \quad (3.48)$$

$$Y_i|\theta \sim \mathcal{N}(\theta, \sigma^2) \quad (3.49)$$

where θ_0 and τ_0^2 are the prior mean and variance, respectively, and σ^2 is known. We have

$$p(\theta|y_1^n) \propto p(\theta)p(y_1^n|\theta) \quad (3.50)$$

$$\propto \frac{1}{\sigma\tau_0} \exp\left(-\frac{\sum_{i=1}^n (y_i - \theta)^2}{2\sigma^2} - \frac{(\theta - \theta_0)^2}{2\tau_0^2}\right) \quad (3.51)$$

With some algebra, it can be shown that conditioned on y_1^n , Θ is normally distributed,

$$\Theta|y_1^n \sim \mathcal{N}\left(\frac{\frac{n\bar{y}}{\sigma^2} + \frac{\theta_0}{\tau_0^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}, \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}\right). \quad (3.52)$$

△

Example 3.13. (†) Let us prove (3.52) using (3.51). We start with the following **claim**: If $p_X(x) \propto e^{-f(x)}$, where $f(x) = ax^2 - bx + c$ with $a > 0$, then $X \sim \mathcal{N}(\frac{b}{2a}, \frac{1}{2a})$. Observe that

$$ax^2 - bx + c = \frac{x^2 - bx/a + c/a}{1/a} = \frac{(x - \frac{b}{2a})^2 - (\frac{b}{2a})^2 + \frac{c}{a}}{2(1/(2a))} = \frac{(x - b/(2a))^2}{2(1/(2a))} + C, \quad (3.53)$$

where C is a constant independent from x . Hence,

$$p_X(x) \propto \exp\left(-\frac{(x - b/(2a))^2}{2(1/(2a))}\right), \quad (3.54)$$

proving the claim. Then, (3.52) can be proven by setting

$$a = \frac{n}{2\sigma^2} + \frac{1}{2\tau_0^2}, \quad b = \frac{n\bar{y}}{\sigma^2} + \frac{\theta_0}{\tau_0^2}. \quad (3.55)$$

△

Example 3.14 (Bias-variance trade-off for a Bayesian point estimator). Suppose that the prior for Θ is $\Theta \sim \mathcal{N}(0, \tau_0^2)$. Then, from (3.52), the mean (also the mode and median) Bayesian point estimator for Θ is

$$\hat{\theta}_B = \bar{y} \left(\frac{\tau_0^2}{\tau_0^2 + \sigma^2/n} \right), \quad (3.56)$$

while the maximum-likelihood estimator is $\hat{\theta}_{mle} = \bar{y}$. We can evaluate both estimators in the frequentist framework, finding their MSE.

Note that the frequentist framework requires us to assume a true value θ^* and view $\hat{\theta}_B$ and $\hat{\theta}_{mle}$ as functions of random data Y_1^n . So they are random variables (however, we won't switch to capital letters to represent them here). First, as the MLE is unbiased,

$$\text{MSE}(\hat{\theta}_{mle}) = \text{Var}(\hat{\theta}_{mle}) = \text{Var}(\bar{Y}) = \sigma^2/n, \quad (3.57)$$

and by CRLB, this is the best unbiased estimator.

For the Bayesian estimator, we have

$$\text{Bias}(\hat{\theta}_B) = \mathbb{E}[\hat{\theta}_B] - \theta^* = \theta^* \left(\frac{\tau_0^2}{\tau_0^2 + \sigma^2/n} \right) - \theta^* = -\theta^* \left(\frac{\sigma^2/n}{\tau_0^2 + \sigma^2/n} \right) \quad (3.58)$$

$$\text{Var}(\hat{\theta}_B) = \frac{\sigma^2}{n} \left(\frac{\tau_0^2}{\tau_0^2 + \sigma^2/n} \right)^2 \quad (3.59)$$

$$\text{MSE}(\hat{\theta}_B) = (\theta^*)^2 \left(\frac{\sigma^2/n}{\tau_0^2 + \sigma^2/n} \right)^2 + \frac{\sigma^2}{n} \left(\frac{\tau_0^2}{\tau_0^2 + \sigma^2/n} \right)^2 \quad (3.60)$$

We can see that the bias term is decreasing in τ_0^2 while the variance term is increasing. So, there is a trade-off between the two types of error. Smaller values of τ_0 mean we have a strong prior, thus leading to bias. A strong prior is also less sensitive to data, thus leading to a smaller variance.

In particular, for $\tau_0^2 = (\theta^*)^2$, we have

$$\text{MSE}(\hat{\theta}_B) = \frac{(\theta^*)^2 \sigma^2/n}{(\theta^*)^2 + \sigma^2/n} < \sigma^2/n = \text{MSE}(\hat{\theta}_{mle}). \quad (3.61)$$

So, with the right prior, $\hat{\theta}_B$ has lower MSE than the maximum-likelihood estimator. Of course, this requires knowledge of (θ^*) , which is not available. However, a good prior found based on experience or intuition can provide good results.

△

3.5 Conjugate Priors

Given a likelihood function, the *conjugate prior* is a distribution that leads to a posterior that is from the same family as the prior. Several examples are given below.

- Bernoulli/Beta: ($y = \sum_{i=1}^n y_i$)

$$p(y_i|\theta) = \theta^{y_i} (1 - \theta)^{1-y_i} \quad \text{Ber}(\theta) \quad (3.62)$$

$$p(y_1^n|\theta) = \theta^y (1 - \theta)^{n-y} \quad (3.63)$$

$$p(\theta) \propto \theta^{\alpha-1} (1 - \theta)^{\beta-1} \quad \text{Beta}(\alpha, \beta) \quad (3.64)$$

$$p(\theta|y) \propto \theta^{y+\alpha-1} (1 - \theta)^{n-y+\beta-1} \quad \text{Beta}(y + \alpha, n - y + \beta) \quad (3.65)$$

- Exponential/Gamma: ($y = \sum_{i=1}^n y_i$)

$$p(y_i|\theta) = \theta \exp(-\theta y_i) \quad \text{Exp}(\theta) = \text{Gamma}(1, \theta) \quad (3.66)$$

$$p(y_1^n|\theta) = \theta^n \exp(-\theta y) \quad (3.67)$$

$$p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} \exp(-\beta\theta) \quad \text{Gamma}(\alpha, \beta) \quad (3.68)$$

$$p(\theta|y_1^n) \propto \theta^{n+\alpha-1} \exp(-(y + \beta)\theta) \quad \text{Gamma}(n + \alpha, y + \beta) \quad (3.69)$$

- Gaussian/Gaussian (with known σ^2): ($\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$)

$$p(y_i|\theta) \propto \exp\left(-\frac{(y_i - \theta)^2}{2\sigma^2}\right) \quad \mathcal{N}(\theta, \sigma^2) \quad (3.70)$$

$$p(y_1^n|\theta) \propto \exp\left(-\frac{\sum_{i=1}^n (y_i - \theta)^2}{2\sigma^2}\right) \quad (3.71)$$

$$p(\theta) \propto \exp\left(-\frac{(\theta - \mu_0)^2}{2\tau_0^2}\right) \quad \mathcal{N}(\mu_0, \tau_0^2) \quad (3.72)$$

$$p(\theta|y_1^n) \propto \exp\left(-\frac{(\theta - \mu_1)^2}{2\tau_1^2}\right) \quad \mathcal{N}(\mu_1, \tau_1^2), \quad (3.73)$$

where

$$\mu_1 = \frac{\frac{1}{\tau_0^2} \mu_0 + \frac{1}{\sigma^2/n} \bar{y}}{\frac{1}{\tau_0^2} + \frac{1}{\sigma^2/n}}, \quad (3.74)$$

$$\frac{1}{\tau_1^2} = \frac{1}{\tau_0^2} + \frac{1}{\sigma^2/n}. \quad (3.75)$$

Note that if a prior is conjugate for the likelihood of a single observation, it is also conjugate for the likelihood of many iid observations. One way to see this is to note that updating the distribution using n iid observations is equivalent to updating the distribution n times using single observations consecutively.

Conjugate priors provide a way to fully determine the posterior distribution without the need to integrate to find the missing constants.

3.6 The Exponential Family (EF)

For a random variable Y with parameter Θ , $p(y|\theta)$ is said to be from the exponential family if it has the following form

$$p(y|\theta) = \exp(a(y)^T b(\theta) + f(y) + g(\theta)), \quad (3.76)$$

where a, b, y, θ can be vectors and f, g are scalar functions. $b(\theta)$ is referred to as the *natural parameter*.

The exponential family includes many common distributions such as Gaussian, Beta, Gamma, Binomial, etc. For likelihoods in this family, we can identify the conjugate prior, thus simplifying Bayesian estimation. Furthermore, for these distributions all information in the data can be summarized in the *sufficient statistics* described below.

Maximum Likelihood. Suppose that we have n iid observation, leading to the likelihood function

$$p(y_1^n | \theta) \propto \exp\left(\sum_{i=1}^n a(y_i)^T b(\theta) + ng(\theta)\right), \quad (3.77)$$

Define the *sufficient statistics* for this likelihood as $t(y_1^n) = \sum_{i=1}^n a(y_i)$. We then have

$$p(y_1^n | \theta) \propto \exp(t(y_1^n)^T b(\theta) + ng(\theta)). \quad (3.78)$$

So for finding the maximum likelihood solution, we can summarize all our data as $t(y_1^n)$ and the rest of the information in y_1^n is irrelevant. This is also true for Bayesian estimation. Note that the size of $t(y_1^n)$ is independent of n .

Bayesian Estimation with Conjugate Priors. In this case, we have the general form of the conjugate prior

$$p(y_i | \theta) \propto \exp(a(y_i)^T b(\theta) + g(\theta)) \quad (3.79)$$

$$p(y_1^n | \theta) \propto \exp(t(y_1^n)^T b(\theta) + ng(\theta)) \quad (3.80)$$

$$p(\theta) \propto \exp(\nu^T b(\theta) + mg(\theta)) \quad \text{Dist}(\nu, m) \quad (3.81)$$

$$p(\theta | y_1^n) \propto \exp((\nu + t(y_1^n))^T b(\theta) + (m + n)g(\theta)) \quad \text{Dist}(\nu + t(y_1^n), m + n), \quad (3.82)$$

where *Dist* refers to a specific type distribution.

Pseudo-observations. The parameters in conjugate priors can be interpreted as representing pseudo-observations by comparing the forms of $p(y_1^n | \theta)$ and $p(\theta)$. In particular, ν plays the same role as $t(y_1^n)$ and m represents the number of pseudo-observations.

Example 3.15. The likelihood for a Bernoulli observation is

$$p(y_i | \theta) = \theta^{y_i} (1 - \theta)^{1-y_i} \quad (3.83)$$

$$= \exp(y_i \ln \theta + (1 - y_i) \ln(1 - \theta)) \quad (3.84)$$

$$= \exp\left(y_i \ln \frac{\theta}{1 - \theta} + \ln(1 - \theta)\right). \quad (3.85)$$

We thus let $a(y_i) = y_i$, $b(\theta) = \ln \frac{\theta}{1 - \theta}$, and $g(\theta) = \ln(1 - \theta)$. Furthermore, let $y = t(y_1^n) = \sum_{i=1}^n a(y_i) = \sum_{i=1}^n y_i$. Then,

$$p(y_1^n | \theta) = \exp\left(y \ln \frac{\theta}{1 - \theta} + n \ln(1 - \theta)\right) \quad (3.86)$$

$$p(\theta) = \exp\left(\nu \ln \frac{\theta}{1 - \theta} + m \ln(1 - \theta)\right) \quad (3.87)$$

$$= \theta^\nu (1 - \theta)^{m-\nu}, \quad \text{Beta}(\nu + 1, m - \nu + 1) \quad (3.88)$$

$$p(\theta | y_1^n) = \exp\left((\nu + y) \ln \frac{\theta}{1 - \theta} + (m + n) \ln(1 - \theta)\right), \quad (3.89)$$

$$\text{Beta}(\nu + y + 1, m + n - \nu - y + 1) \quad (3.90)$$

△

Chapter 4

Multivariate Random Variables

In this chapter, we will review some topics related to random vectors, which will be of use in the following chapters.

4.1 Gaussian Random Vectors (Multivariate Normal Distribution)

Recall that a random variable X is Gaussian (normal) with mean μ and variance $\sigma^2 > 0$ if the pdf of X is given by

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{(x - \mu)^2}{2\sigma^2}. \quad (4.1)$$

Definition 4.1. A collection of random variables is **jointly Gaussian** if any linear combination of these variables is Gaussian. A **Gaussian random vector**, also known as a *multivariate normal vector*, is a vector whose elements are jointly Gaussian. A collection of random vectors is jointly Gaussian if the vector obtained by concatenating them is jointly Gaussian.

Example 4.2. If $\begin{pmatrix} X \\ Y \end{pmatrix}$ is a Gaussian vector, then $Z = 2X + 3Y$ is Gaussian. Furthermore,

$$\mathbb{E}[Z] = 2\mathbb{E}[X] + 3\mathbb{E}[Y], \quad (4.2)$$

$$\text{Var}(Z) = \text{Cov}(2X + 3Y, 2X + 3Y) = 4\text{Cov}(X, X) + 12\text{Cov}(X, Y) + 9\text{Cov}(Y, Y) \quad (4.3)$$

$$= 4\text{Var}(X) + 12\text{Cov}(X, Y) + 9\text{Var}(Y), \quad (4.4)$$

which completely characterizes the distribution of Z as $Z \sim \mathcal{N}(\mathbb{E}[Z], \text{Var}(Z))$. \triangle

For a Gaussian random vector \mathbf{X} of dimension d , with mean $\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu}$ and covariance matrix $\mathbf{K} = \text{Cov}(\mathbf{X}) = \mathbb{E}[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T]$, we have

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\mathbf{K}|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{K}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right), \quad (4.5)$$

provided that the covariance matrix is invertible.

The elements of \mathbf{X} are **independent** if and only if the covariance matrix is diagonal.

4.1.1 Maximum likelihood estimation

Consider a d -dimensional random vector $\mathbf{X} = (X_1, \dots, X_d)$ with distribution $\mathcal{N}(\boldsymbol{\theta}^*, \mathbf{K}^*)$ given in (4.5), where $\boldsymbol{\theta}^*, \mathbf{K}^*$ are unknown. Suppose we are interested in the relationship between X_d and X_1, \dots, X_{d-1} . For example, for $\mathbf{X}^T = (X_1, X_2, X_3)$, X_1 and X_2 could indicate the heights of the parents and X_3 could be the

height of the child. We may, for example, be interested in finding $\mathbb{E}[X_d|X_1, \dots, X_{d-1}]$, thus estimating X_d based on X_1, \dots, X_{d-1} . If we find the distribution, in other words, $\boldsymbol{\theta}^*, \mathbf{K}^*$, we can do so. Furthermore, the matrix \mathbf{K}^* can indicate which dimensions are more strongly correlated.

Consider a set of n iid samples $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, where each \mathbf{x}_i is a sample of \mathbf{X} . We denote the elements of \mathbf{x}_i as $\mathbf{x}_i = (x_{i1}, \dots, x_{id})$.

To estimate $\boldsymbol{\theta}^*$ and \mathbf{K}^* , we write

$$\ell(\boldsymbol{\theta}, \mathbf{K}) = \ln p(\mathcal{D}; \boldsymbol{\theta}, \mathbf{K}) = \sum_{i=1}^n \ln p(\mathbf{x}_i; \boldsymbol{\theta}, \mathbf{K}) \quad (4.6)$$

$$\doteq \frac{n}{2} \ln |\mathbf{K}^{-1}| - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\theta})^T \mathbf{K}^{-1} (\mathbf{x}_i - \boldsymbol{\theta}), \quad (4.7)$$

where we have used the fact that $|\mathbf{K}^{-1}| = \frac{1}{|\mathbf{K}|}$.

As seen in the appendix (last chapter), for a symmetric matrix \mathbf{A} , we have $\frac{d}{d\mathbf{v}}(\mathbf{y}^T \mathbf{A} \mathbf{y}) = 2\mathbf{y}^T \mathbf{A} \frac{d\mathbf{y}}{d\mathbf{v}}$. Hence,

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = -\frac{1}{2} \sum_{i=1}^n 2(\mathbf{x}_i - \boldsymbol{\theta})^T \mathbf{K}^{-1} (-\mathbf{I}) = \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\theta})^T \mathbf{K}^{-1}. \quad (4.8)$$

Setting this equal to zero yields

$$\hat{\boldsymbol{\theta}}_{ML} = \bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i. \quad (4.9)$$

Exercise 4.3. Using the facts

$$\frac{\partial}{\partial \mathbf{A}} \mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x} \mathbf{x}^T, \quad \frac{\partial}{\partial \mathbf{A}} \ln |\mathbf{A}| = \mathbf{A}^{-T} \quad (4.10)$$

prove that

$$\hat{\mathbf{K}}_{ML} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (4.11)$$

△

4.1.2 Bayesian estimation

We now solve the same problem using Bayesian estimation, with the following likelihood

$$\mathbf{X}|\boldsymbol{\Theta} \sim \mathcal{N}(\boldsymbol{\Theta}, \mathbf{K}), \quad (4.12)$$

$$p(\mathbf{x}_1^n | \boldsymbol{\theta}) \propto \exp \left(-\frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\theta})^T \mathbf{K}^{-1} (\mathbf{x}_i - \boldsymbol{\theta}) \right), \quad (4.13)$$

where, for simplicity, we assume \mathbf{K} is known and we only need to estimate $\boldsymbol{\Theta}$. As the prior, we choose

$$\boldsymbol{\Theta} \sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{S}_0) \quad (4.14)$$

$$p(\boldsymbol{\theta}) \propto \exp \left(-\frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\mu}_0)^T \mathbf{S}_0^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_0) \right). \quad (4.15)$$

Hence,

$$p(\boldsymbol{\theta} | \mathbf{x}_1^n) \propto \exp \left(-\frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\mu}_0)^T \mathbf{S}_0^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_0) - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\theta})^T \mathbf{K}^{-1} (\mathbf{x}_i - \boldsymbol{\theta}) \right). \quad (4.16)$$

The exponent in the posterior is quadratic in $\boldsymbol{\theta}$, indicating that $\boldsymbol{\Theta}$ has a Gaussian distribution. So $\boldsymbol{\Theta}|\mathbf{x}_1^n \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_n, \mathbf{S}_n)$, for appropriate choices of $\hat{\boldsymbol{\theta}}_n$ and \mathbf{S}_n ,

$$p(\boldsymbol{\theta}|\mathbf{x}_1^n) \propto \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)^T \mathbf{S}_n^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)\right). \quad (4.17)$$

To find $\hat{\boldsymbol{\theta}}_n$ and \mathbf{S}_n , we equate (4.16) and (4.17), ignoring constant multiplicative factors, which leads to

$$(\boldsymbol{\theta} - \boldsymbol{\mu}_0)^T \mathbf{S}_0^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}_0) + \sum_{i=1}^n (\boldsymbol{\theta} - \mathbf{x}_i)^T \mathbf{K}^{-1}(\boldsymbol{\theta} - \mathbf{x}_i) \doteq (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)^T \mathbf{S}_n^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n), \quad (4.18)$$

$$\boldsymbol{\theta}^T \mathbf{S}_0^{-1} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T \mathbf{S}_0^{-1} \boldsymbol{\mu}_0 + n\boldsymbol{\theta}^T \mathbf{K}^{-1} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T \mathbf{K}^{-1} \sum_{i=1}^n \mathbf{x}_i \doteq \boldsymbol{\theta}^T \mathbf{S}_n^{-1} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T \mathbf{S}_n^{-1} \hat{\boldsymbol{\theta}}_n. \quad (4.19)$$

Here, we have used the fact that

$$(\mathbf{a} - \mathbf{b})^T \mathbf{A}(\mathbf{a} - \mathbf{b}) = \mathbf{a}^T \mathbf{A} \mathbf{a} - \mathbf{a}^T \mathbf{A} \mathbf{b} - \mathbf{b}^T \mathbf{A} \mathbf{a} + \mathbf{b}^T \mathbf{A} \mathbf{b} = \mathbf{a}^T \mathbf{A} \mathbf{a} - 2\mathbf{a}^T \mathbf{A} \mathbf{b} + \mathbf{b}^T \mathbf{A} \mathbf{b},$$

for vectors \mathbf{a}, \mathbf{b} and a symmetric matrix \mathbf{A} . Note that $\mathbf{a}^T \mathbf{A} \mathbf{b} = \mathbf{b}^T \mathbf{A} \mathbf{a}$, as both sides are scalars and $\mathbf{a}^T \mathbf{A} \mathbf{b} = (\mathbf{a}^T \mathbf{A} \mathbf{b})^T = \mathbf{b}^T \mathbf{A} \mathbf{a}$.

We now collect the terms of the form $\boldsymbol{\theta}^T \mathbf{A} \boldsymbol{\theta}$,

$$\boldsymbol{\theta}^T (\mathbf{S}_0^{-1} + n\mathbf{K}^{-1}) \boldsymbol{\theta} - 2\boldsymbol{\theta}^T (\mathbf{S}_0^{-1} \boldsymbol{\mu}_0 + \mathbf{K}^{-1} \sum_{i=1}^n \mathbf{x}_i) \doteq \boldsymbol{\theta}^T \mathbf{S}_n^{-1} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T \mathbf{S}_n^{-1} \hat{\boldsymbol{\theta}}_n, \quad (4.20)$$

leading to the following values for the parameters of the posterior distribution $\boldsymbol{\Theta}|\mathbf{x}_1^n \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_n, \mathbf{S}_n)$,

$$\mathbf{S}_n^{-1} = \mathbf{S}_0^{-1} + n\mathbf{K}^{-1}, \quad (4.21)$$

$$\hat{\boldsymbol{\theta}}_n = \mathbf{S}_n (\mathbf{S}_0^{-1} \boldsymbol{\mu}_0 + n\mathbf{K}^{-1} \bar{\mathbf{x}}) \quad (4.22)$$

$$= (\mathbf{S}_0^{-1} + n\mathbf{K}^{-1})^{-1} (\mathbf{S}_0^{-1} \boldsymbol{\mu}_0 + n\mathbf{K}^{-1} \bar{\mathbf{x}}), \quad (4.23)$$

where $\bar{\mathbf{x}}$ is $\sum_{i=1}^n \mathbf{x}_i / n$. The posterior mean, $\hat{\boldsymbol{\theta}}_n$, which we can also view as a point estimate, is the weighted average of the prior mean $\boldsymbol{\mu}_0$ and what is suggested by the data $\bar{\mathbf{x}}$.

Exercise 4.4. Find $\hat{\boldsymbol{\theta}}_n$ and \mathbf{S}_n^{-1} when $\mathbf{S}_0 = s^2 \mathbf{I}$ and $\mathbf{K} = \sigma^2 \mathbf{I}$ and interpret the results. \triangle

Chapter 5

Linear Regression

5.1 Introduction

The goal of *regression* is to predict a real value y as a function of the input variable \mathbf{x} . (The vector \mathbf{x} is referred to as the feature vector, while y is called the target variable.) For example, in a marketing campaign, we may be interested in predicting total sales, given ad budgets in various platforms based on prior experience. Our data is a set of pairs $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, collected from n prior ad campaigns for n previous products,

Product	TV ads	Print ads	Web ads	Sale	
1	$\mathbf{x}_1^T =$	\$20k	\$10k	\$10k	$y_1 = \$500k$
\vdots		\vdots			\vdots
i	$\mathbf{x}_i^T =$	x_{i1}	x_{i2}	x_{i3}	y_i
\vdots		\vdots			\vdots
n	$\mathbf{x}_n^T =$	x_{n1}	x_{n2}	x_{n3}	y_n

If we can predict y for any given value of \mathbf{x} , we can predict the outcome of a marketing campaign or optimize the marketing budget. We can also study what types of ads are more helpful, etc.

In *linear regression* our prediction for y is $\hat{y} = \mathbf{x}^T \boldsymbol{\theta}$, where \mathbf{x} and $\boldsymbol{\theta}$ are elements of \mathbb{R}^d . In our marketing example, our goal becomes to find $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ such that $\hat{y} = \mathbf{x}^T \boldsymbol{\theta} = \boldsymbol{\theta}^T \mathbf{x} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$ is a good predictor for y .

From a probabilistic standpoint, we may consider each (\mathbf{x}_i, y_i) to be an independent realization of the random pair (\mathbf{X}, Y) with some joint distribution $p_{\mathbf{X}, Y}$. We then formulate the linear regression problem as follows: Find

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \mathbb{E}[L(Y, \mathbf{X}^T \boldsymbol{\theta})], \quad (5.1)$$

for a given loss function L . As we typically do not have the joint distribution for \mathbf{X}, Y , we aim to find

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^n L(y_i, \mathbf{x}_i^T \boldsymbol{\theta}). \quad (5.2)$$

The linear form, $\hat{y} = \mathbf{x}^T \boldsymbol{\theta} = \sum_{j=1}^d \theta_j x_j$, may appear restrictive since it apparently excludes dependence on, for example, x_j^2 . Imagine, in our marketing example, that buyers are likely to purchase a product if they see both TV ads and Web ads. In other words, y is large when $x_1 x_3$ is large. It seems that this case is not covered well by linear regression. However, this is not the case since we can transform the input variable using a set of functions g_1, \dots, g_e and reformulate our assumption as $\hat{y} = \sum_{j=1}^e \theta_j g_j(\mathbf{x})$, where g_j are any

functions of \mathbf{x} , such as x_1^2 and x_1x_3 . (But finding appropriate features is a challenging problem.) Note that the expression $\hat{y} = \sum_{j=1}^e \theta_j g_j(\mathbf{x})$ is still linear in $\boldsymbol{\theta}$, which is what matters, since we need to optimize $\boldsymbol{\theta}$.

Notation. Define $\mathbf{X} \in \mathbb{R}^{n \times d}$ and \mathbf{y} as

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad (5.3)$$

For a given value of $\boldsymbol{\theta}$, we let $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta}$ to be the predicted value. Furthermore, let $\boldsymbol{\epsilon}$ be the error vector such that

$$\mathbf{y} = \hat{\mathbf{y}} + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}. \quad (5.4)$$

Example 5.1. Suppose

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{x}_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad \mathbf{x}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad (5.5)$$

$$y_1 = -1, \quad y_2 = 1, \quad y_3 = 0. \quad (5.6)$$

Then

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \\ 1 & 1 \end{pmatrix}, \quad \hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta} = \theta_1 \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} + \theta_2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{\epsilon} = \mathbf{y} - \hat{\mathbf{y}} = \begin{pmatrix} -1 - \theta_2 \\ 1 - 2\theta_1 \\ -\theta_1 - \theta_2 \end{pmatrix}. \quad (5.7)$$

△

5.2 Least-squares

A common choice for the loss function is

$$L(y_i, \mathbf{x}_i^T \boldsymbol{\theta}) = (y_i - \mathbf{x}_i^T \boldsymbol{\theta})^2. \quad (5.8)$$

The empirical risk can be written as

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\theta})^2 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2, \quad (5.9)$$

where we have dropped the $1/n$ factor present in (5.2) as it does not affect our choice of $\boldsymbol{\theta}$. Denote

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}), \quad (5.10)$$

and define $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\theta}}$ as the predicted value or estimate based on the model.

Least-squares is relatively easy to deal with from a computational perspective. It also has the same solution as the MLE for a common probabilistic model as we will see, thus providing an additional rationale for the resulting approach.

Projection onto the column space of \mathbf{X} . Our first observation is that $\hat{\mathbf{y}}$ is in the column space of \mathbf{X} , i.e., it is a linear combination of the columns of \mathbf{X} . We can thus restate our goal as finding $\hat{\mathbf{y}}$ in the column space of \mathbf{X} such that $\|\mathbf{y} - \hat{\mathbf{y}}\|$ is minimized. Hence, $\hat{\mathbf{y}}$ is the projection of \mathbf{y} onto the column space of \mathbf{X} as shown in Figure 5.1. Then, from the Projection Lemma in the Appendix, $\mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to each column of \mathbf{X} .

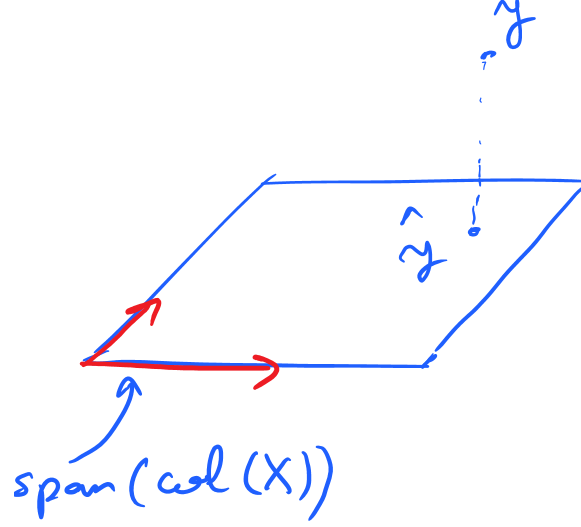


Figure 5.1: Error is minimized by projecting \mathbf{y} onto the column space of \mathbf{X} , $\text{Span}(\text{col}(\mathbf{X}))$.

This orthogonality of the error to columns of \mathbf{X} can be written as $\mathbf{X}^T(\mathbf{y} - \hat{\mathbf{y}}) = \mathbf{0}$. We have

$$\mathbf{X}^T(\mathbf{y} - \hat{\mathbf{y}}) = \mathbf{0} \iff \mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}}) = \mathbf{0} \quad (5.11)$$

$$\iff \mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\theta}} \quad (5.12)$$

$$\iff \hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (5.13)$$

Here we have assumed that $\mathbf{X}^T\mathbf{X}$ is invertible. This holds if the columns of \mathbf{X} are linearly independent. To see this, we will show $\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} = \mathbf{0}$ implies that $\boldsymbol{\alpha} = \mathbf{0}$ if the columns of \mathbf{X} are linearly independent:

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} = \mathbf{0} \Rightarrow \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\alpha} = 0 \Rightarrow (\mathbf{X}\boldsymbol{\alpha})^T(\mathbf{X}\boldsymbol{\alpha}) = 0 \Rightarrow \mathbf{X}\boldsymbol{\alpha} = \mathbf{0} \Rightarrow \boldsymbol{\alpha} = \mathbf{0}, \quad (5.14)$$

where the last step follows from the fact that the columns of \mathbf{X} are linearly independent.

Example 5.2. From Example 5.1, we have

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}, \quad (5.15)$$

and so

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} 5 & 1 \\ 1 & 2 \end{pmatrix}, \quad (\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{9} \begin{pmatrix} 2 & -1 \\ -1 & 5 \end{pmatrix} \quad (5.16)$$

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \frac{1}{9} \begin{pmatrix} -1 & 4 & 1 \\ 5 & -2 & 4 \end{pmatrix} \quad \hat{\boldsymbol{\theta}} = \begin{pmatrix} 5/9 \\ -7/9 \end{pmatrix} \quad (5.17)$$

$$\hat{\mathbf{y}} = \begin{pmatrix} -7/9 \\ 10/9 \\ -2/9 \end{pmatrix}, \quad \mathbf{y} - \hat{\mathbf{y}} = \begin{pmatrix} -2/9 \\ -1/9 \\ 2/9 \end{pmatrix}. \quad (5.18)$$

△

Gradient descent. The closed-form solution provided in (5.13) for finding $\hat{\boldsymbol{\theta}}$ requires taking matrix inverses of possibly very large matrices, which could be computationally expensive. A less expensive solution is gradient descent, where we take the derivative of the loss to minimize it. Let $\nabla\mathcal{L}(\boldsymbol{\theta}) = \left(\frac{d\mathcal{L}}{d\boldsymbol{\theta}}\right)^T$ be the

gradient of \mathcal{L} . Recall that the direction of the gradient indicates the direction of maximum increase and its magnitude represents the slope of the increase. We have

$$\mathcal{L} = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}), \quad (5.19)$$

$$\nabla \mathcal{L} = 2[(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(-\mathbf{X})]^T = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}). \quad (5.20)$$

(Setting the gradient equal to 0 again gives $\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. Note that the Hessian is $\mathbf{X}^T\mathbf{X}$, which is positive-semi-definite.) In gradient descent, we start from an arbitrary value $\boldsymbol{\theta}^{(0)}$ and move towards the solution in steps:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \rho \mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^{(t)}) \quad (5.21)$$

$$= \boldsymbol{\theta}^{(t)} + \rho \sum_{i=1}^n \mathbf{x}_i(y_i - \mathbf{x}_i^T \boldsymbol{\theta}^{(t)}), \quad (5.22)$$

where ρ is the learning rate. This approach gets to the lowest point by moving in the direction of the *steepest descent* as shown in figure below for Example 5.1.

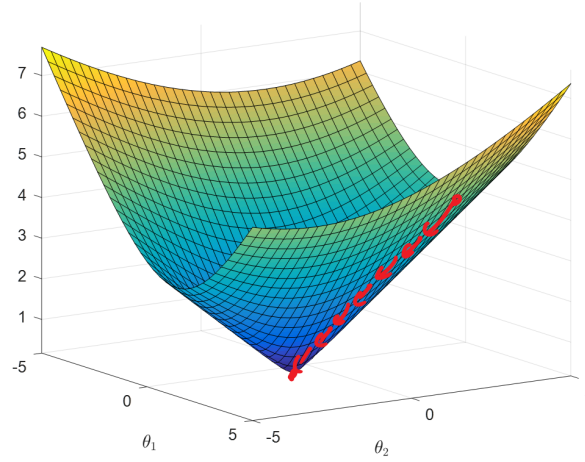


Figure 5.2: Gradient descent for linear regression

Standardization

We sometimes assume that \mathbf{X} is standardized, meaning that each column \mathbf{v} is shifted and scaled such that $\mathbf{v}^T\mathbf{1} = 0$ and $\mathbf{v}^T\mathbf{v} = 1$ and that \mathbf{y} is centered so that $\mathbf{y}^T\mathbf{1} = 0$. Standardization of the inputs puts different features under the same scale and can help to reduce the correlation between features when having polynomial/interaction terms. Standardizing inputs can also be shown to be equivalent to minimizing the squared loss with an intercept term.

Example 5.3 (†). We show that standardizing inputs and then finding the solution with no intercept term is equivalent to minimizing the squared loss with an intercept term. By including an intercept term, the loss becomes

$$\mathcal{L}(\boldsymbol{\theta}, \theta_0) = \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\theta} - \theta_0)^2. \quad (5.23)$$

Such formulation has the advantage of allowing y to be nonzero when \mathbf{x} is zero. Let the solution to this new

loss formulation be $\hat{\boldsymbol{\theta}}, \hat{\theta}_0$:

$$\hat{\theta}_0, \hat{\boldsymbol{\theta}} = \arg \min_{\theta_0, \boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \theta_0) \quad (5.24)$$

and let $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\theta}} + \hat{\theta}_0\mathbf{1}$ be the predictions.

Let the columns of \mathbf{X} be $\mathbf{v}_1, \dots, \mathbf{v}_m$. We provide an analysis that the optimal prediction $\hat{\mathbf{y}}$ found by considering the intercept term is the same as minimizing the normal squared loss (5.9) over the standardized inputs $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{y}}$:

$$\tilde{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n (\tilde{y}_i - \tilde{\mathbf{x}}_i^T \boldsymbol{\theta})^2, \quad (5.25)$$

$$\tilde{\mathbf{y}} = \tilde{\mathbf{X}}\tilde{\boldsymbol{\theta}} + \bar{y}\mathbf{1}, \quad (5.26)$$

where

$$\tilde{\mathbf{X}} = \begin{pmatrix} \tilde{\mathbf{x}}_1^T \\ \vdots \\ \tilde{\mathbf{x}}_n^T \end{pmatrix} = (\tilde{\mathbf{v}}_1, \dots, \tilde{\mathbf{v}}_m), \quad \tilde{\mathbf{v}}_j = (\mathbf{v}_j - \beta_j \mathbf{1})/\alpha_j, \quad \beta_j = \frac{1}{n} \sum_{i=1}^n v_{ji}, \quad \alpha_j = \|\mathbf{v}_j - \beta_j \mathbf{1}\|_2, \quad (5.27)$$

$$\tilde{\mathbf{y}} = \mathbf{y} - \bar{y}\mathbf{1}, \quad \bar{y} = \frac{1}{n} \sum_{j=1}^n y_j. \quad (5.28)$$

Let's first minimize the new loss (5.23). We can fix $\boldsymbol{\theta}$ for now. Then, the loss function is quadratic in θ_0 and the optimal choice for θ_0 can be found as

$$\hat{\theta}_0(\boldsymbol{\theta}) = \bar{y} - \sum_{j=1}^m \beta_j \theta_j. \quad (5.29)$$

Substituting θ_0 by $\hat{\theta}_0(\boldsymbol{\theta})$ in the loss (5.23) gives

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\theta} - \bar{y} + \sum_{j=1}^m \beta_j \theta_j)^2. \quad (5.30)$$

After rewriting (5.25) as

$$\tilde{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n (\tilde{y}_i - \tilde{\mathbf{x}}_i^T \boldsymbol{\theta})^2 = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - \bar{y} - \sum_{j=1}^m \frac{(x_{ij} - \beta_j)}{\alpha_j} \theta_j)^2 \quad (5.31)$$

$$= \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - \sum_{j=1}^m x_{ij} \frac{\theta_j}{\alpha_j} - \bar{y} + \sum_{j=1}^m \beta_j \frac{\theta_j}{\alpha_j})^2, \quad (5.32)$$

we can observe an analogy between (5.30) and (5.32). It follows that $\tilde{\theta}_j = \alpha_j \hat{\theta}_j$ for all $1 \leq j \leq m$.

Hence, for all $1 \leq i \leq n$,

$$\tilde{y}_i = \tilde{\mathbf{x}}_i^T \tilde{\boldsymbol{\theta}} + \bar{y} = \sum_{j=1}^m \tilde{x}_{ij} \tilde{\theta}_j + \bar{y} = \sum_{j=1}^m (x_{ij} - \beta_j) \hat{\theta}_j + \bar{y} = \mathbf{x}_i^T \hat{\boldsymbol{\theta}} + \hat{\theta}_0 = \hat{y}_i, \quad (5.33)$$

i.e., the predictions we obtained from (5.24) and (5.25) are equal. Note that the second last equality follows from $\hat{\theta}_0(\hat{\boldsymbol{\theta}}) = \hat{\theta}_0$. \triangle

5.3 Probabilistic Models for Regression

So far we haven't made any assumptions regarding the statistics of the data. In this section, we consider two models: i) a model that only characterizes the mean and covariance of the error vector and ii) a Gaussian model.

5.3.1 General model

Let us now assume that

$$Y = \mathbf{x}^T \boldsymbol{\theta}^* + \epsilon, \quad \mathbb{E}[\epsilon] = 0, \quad \text{Var}(\epsilon) = \sigma^2.$$

We have n samples (\mathbf{x}_i, y_i) . For simplicity, we will assume that \mathbf{x}_i are deterministic. We will further assume for any $i, j, i \neq j$, ϵ_i and ϵ_j are uncorrelated. In vector form, we have

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta}^* + \boldsymbol{\epsilon}, \quad \mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}, \quad \text{Cov}(\boldsymbol{\epsilon}) = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \sigma^2 \mathbf{I}.$$

At this point, we are not making any other assumptions related to the distribution of $\boldsymbol{\epsilon}$.¹

Frequentist evaluation: Consider the estimator $\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$. Let us investigate the behavior of $\hat{\boldsymbol{\theta}}$ by viewing it as a random variable $\hat{\boldsymbol{\Theta}}$ under this model. We have

$$\mathbb{E}[\hat{\boldsymbol{\Theta}}] = \mathbb{E}[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}] \quad (5.34)$$

$$= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbb{E}[\mathbf{Y}] \quad (5.35)$$

$$= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbb{E}[\mathbf{X}\boldsymbol{\theta}^* + \boldsymbol{\epsilon}] \quad (5.36)$$

$$= \boldsymbol{\theta}^*, \quad (5.37)$$

indicating that $\hat{\boldsymbol{\theta}}$ is an unbiased estimate of $\boldsymbol{\theta}^*$. In particular, each dimension is estimated without bias, i.e., $\mathbb{E}[\hat{\Theta}_i] = \theta_i^*$.

We also find

$$\text{Cov}(\hat{\boldsymbol{\Theta}}) = \text{Cov}((\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}) \quad (5.38)$$

$$= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \text{Cov}(\mathbf{Y}) \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \quad (5.39)$$

$$= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \text{Cov}(\boldsymbol{\epsilon}) \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \quad (5.40)$$

$$= (\mathbf{X}^T \mathbf{X})^{-1} \sigma^2. \quad (5.41)$$

In particular, element i of the diagonal of $\text{Cov}(\hat{\boldsymbol{\Theta}})$ is the variance of $\hat{\Theta}_i$ and also its MSE, as the estimator is unbiased.

The Gauss-Markov theorem. The Gauss-Markov theorem states that under the assumptions that $\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}$ and $\text{Cov}(\boldsymbol{\epsilon}) = \sigma^2 \mathbf{I}$, $\hat{\boldsymbol{\theta}}$ is the best *linear* unbiased estimator. Here, linear means that $\hat{\boldsymbol{\theta}}$ is linear in \mathbf{y} , i.e., $\hat{\boldsymbol{\theta}} = a_1 y_1 + a_2 y_2 + \dots + a_m y_m$ for some scalars a_i^m . The Gauss-Markov theorem implies that for any² vector \mathbf{u} , $\mathbf{u}^T \hat{\boldsymbol{\theta}}$ is an unbiased estimator of $\mathbf{u}^T \boldsymbol{\theta}^*$ with the smallest possible variance.

5.3.2 Gaussian model

Let us further assume that ϵ_i are iid, with distribution $\mathcal{N}(0, \sigma^2)$, i.e., $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. In other words, we have:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta}^* + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \quad (5.42)$$

$$p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\theta}^*, \sigma^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\theta}^*, \sigma^2 \mathbf{I}). \quad (5.43)$$

¹For ϵ , we will not follow the convention that random variables are shown as capital letters since capital ϵ can be confused with Latin E .

²This isn't entirely precise!

Exercise 5.4. Prove that if $p(\mathbf{y}; \boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\theta}, \sigma^2 I)$, then for all i , $p(y_i; \boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(x_i^T \boldsymbol{\theta}, \sigma^2)$ and the y_i are independent. \triangle

Now we have a probabilistic model with unknown parameters $\boldsymbol{\theta}$ and σ^2 .

Maximum Likelihood

Given that the covariance matrix is $\sigma^2 I$ and assuming that \mathbf{y} is n -dimensional, the density and the likelihood are

$$p(\mathbf{y}; \boldsymbol{\theta}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})\right) \quad (5.44)$$

$$\propto \frac{1}{\sigma^n} \exp\left(-\frac{\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2}{2\sigma^2}\right) \quad (5.45)$$

$$\ell(\boldsymbol{\theta}, \sigma^2) \doteq -n \ln(\sigma) - \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2. \quad (5.46)$$

So maximizing for $\boldsymbol{\theta}$ leads to minimizing $\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 = \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\theta})^2$ which we already know the solution to:

$$\hat{\boldsymbol{\theta}}_{ML} = \hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (5.47)$$

We can similarly show that

$$\hat{\sigma}_{ML}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \hat{\boldsymbol{\theta}})^2. \quad (5.48)$$

The mean and covariance of $\hat{\boldsymbol{\theta}}$ are the same as in §5.3.1. But now we also know that $\hat{\boldsymbol{\theta}}$ is *Gaussian*. This is because the linear combination of Gaussian variables is Gaussian. Hence,

$$\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}). \quad (5.49)$$

Cramer-Rao Lower Bound. With the additional Gaussian assumption in this section, using Cramer-Rao lower bound, a stronger result compared to the Gauss-Markov theorem can be obtained. Namely, $\hat{\boldsymbol{\theta}}$ is the best unbiased estimator (not just the best linear unbiased estimator).

Example 5.5 (†). For an unbiased vector estimator $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$, the CRLB has the form

$$\text{Cov}(\hat{\boldsymbol{\theta}}) \succcurlyeq I^{-1}(\boldsymbol{\theta}), \quad (5.50)$$

where $A \succcurlyeq B$ denotes that $A - B$ positive semidefinite. Let us find the CRLB for $\hat{\boldsymbol{\theta}}$ of (5.47). We have

$$\ell(\boldsymbol{\theta}, \sigma^2) \doteq -n \ln(\sigma) - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}). \quad (5.51)$$

$$\nabla_{\boldsymbol{\theta}} \ell = \left(-\frac{1}{\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T (-\mathbf{X}) \right)^T \quad (5.52)$$

$$= \frac{1}{\sigma^2} \mathbf{X}^T (\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \quad (5.53)$$

$$\mathbf{H}_{\boldsymbol{\theta}} \ell = \frac{d \nabla_{\boldsymbol{\theta}} \ell}{d \boldsymbol{\theta}} = -\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X}. \quad (5.54)$$

and so $I(\boldsymbol{\theta}) = \frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X}$. Hence, $I(\boldsymbol{\theta})^{-1} = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$, which matches the covariance of $\hat{\boldsymbol{\theta}}$ found in (5.41). \triangle

Bayesian Linear Regression

In Bayesian linear regression, the Gaussian likelihood

$$\mathbf{y}|\boldsymbol{\theta}, \sigma^2 \sim \mathcal{N}(\mathbf{X}\boldsymbol{\theta}, \sigma^2 \mathbf{I}) \quad (5.55)$$

is a common choice. But we also need to choose priors for $\boldsymbol{\theta}$ and σ^2 . A possible non-informative choice is

$$p(\boldsymbol{\theta}, \sigma^2) \propto 1/\sigma^2, \quad (5.56)$$

or equivalently, $p(\sigma^2) \propto \frac{1}{\sigma^2}$, $p(\boldsymbol{\theta}) \propto 1$ and σ^2 , and $\boldsymbol{\theta}$ are independent.

We are interested in finding

$$p(\boldsymbol{\theta}, \sigma^2|\mathbf{y}) = p(\boldsymbol{\theta}|\sigma^2, \mathbf{y})p(\sigma^2|\mathbf{y}) \quad (5.57)$$

We start with

$$p(\boldsymbol{\theta}|\mathbf{y}, \sigma^2) = \frac{p(\boldsymbol{\theta}, \mathbf{y}|\sigma^2)}{p(\mathbf{y}|\sigma^2)} \propto p(\boldsymbol{\theta}, \mathbf{y}|\sigma^2) = p(\mathbf{y}|\boldsymbol{\theta}, \sigma^2)p(\boldsymbol{\theta}|\sigma^2) \propto \exp\left(-\frac{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}{2\sigma^2}\right). \quad (5.58)$$

Note that in the above expression, we are viewing \mathbf{y} and σ^2 as given. So the expression $p(\mathbf{y}|\sigma^2)$ is treated as a constant and discarded. Furthermore, $p(\boldsymbol{\theta}|\sigma^2) = p(\boldsymbol{\theta}) \propto 1$.

The right-hand expression in (5.58) is quadratic in $\boldsymbol{\theta}$. So we'll try to see if we can write it in terms of a Gaussian distribution. With foresight, let the mean and the covariance of this distribution be denoted $\hat{\boldsymbol{\theta}}$ and $K\sigma^2$. We need

$$(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T K^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \doteq (\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}). \quad (5.59)$$

Ignoring terms that are constant in $\boldsymbol{\theta}$, we require

$$\boldsymbol{\theta}^T K^{-1} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T K^{-1} \hat{\boldsymbol{\theta}} \doteq \boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y}, \quad (5.60)$$

which is satisfied by $K^{-1} = \mathbf{X}^T \mathbf{X}$ and

$$-2\boldsymbol{\theta}^T K^{-1} \hat{\boldsymbol{\theta}} = -2\boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y}, \quad (5.61)$$

$$-2\boldsymbol{\theta}^T \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\theta}} = -2\boldsymbol{\theta}^T \mathbf{X}^T \mathbf{y}, \quad (5.62)$$

$$\mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\theta}} = \mathbf{X}^T \mathbf{y}, \quad (5.63)$$

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (5.64)$$

So it suffices to set $\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ and $K = (\mathbf{X}^T \mathbf{X})^{-1}$,

$$p(\boldsymbol{\theta}|\mathbf{y}, \sigma^2) \sim \mathcal{N}(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}, K\sigma^2). \quad (5.65)$$

Now we need to find $p(\sigma^2|\mathbf{y})$. Using the fact that $p(\sigma^2|\mathbf{y}) = p(\boldsymbol{\theta}, \sigma^2|\mathbf{y})/p(\boldsymbol{\theta}|\sigma^2, \mathbf{y})$, it can be shown that $p(\sigma^2|\mathbf{y})$ has a scaled inverse- χ^2 distribution,

$$p(\sigma^2|\mathbf{y}) \sim \text{Inv-}\chi^2(n - m, s^2), \quad (5.66)$$

where m is the dimension of \mathbf{x}_i and

$$s^2 = \frac{1}{n - m} (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}})^T (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}}). \quad (5.67)$$

While we can continue analytically and find $p(\boldsymbol{\theta}|\mathbf{y})$, in practice, we proceed computationally by generating samples from $p(\sigma^2|\mathbf{y})$ and then $p(\boldsymbol{\theta}|\mathbf{y}, \sigma^2)$. With this sampling approach we can also perform prediction for a given input vector \mathbf{x}_{n+1} of by producing samples from $p(y_{n+1}|\boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(\mathbf{x}_{n+1}^T \boldsymbol{\theta}, \sigma^2)$.

5.4 Regularized Linear Regression

Sometimes we are interested in reducing the flexibility of the model to avoid over-fitting, especially when the size of the data set is small. Alternatively, we may be interested in putting restrictions (e.g., forcing small coefficients to become 0) so that only the most important aspects of the data appear in the learned model, thus increasing its interpretability. These can be done by altering the loss function by adding a regularization term.

Ridge Regression

Ridge regression adds a penalty for the magnitude of the coefficients. Specifically, the loss function is

$$\mathcal{L}(\boldsymbol{\theta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2, \quad (5.68)$$

where λ is a parameter determining the relative importance of the square error versus the regularization loss term $\|\boldsymbol{\theta}\|_2^2$. The problem of minimizing this loss,

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2, \quad (5.69)$$

can be shown to be equivalent to

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2, \quad (5.70)$$

$$\text{subject to : } \|\boldsymbol{\theta}\|_2^2 \leq t, \quad (5.71)$$

for some t . There is a one-to-one correspondence between λ and t . The second form is perhaps easier to understand because of the explicit constraints on $\|\boldsymbol{\theta}\|_2^2$.

From (5.69),

$$\nabla \mathcal{L}(\boldsymbol{\theta}) = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) + 2\lambda\boldsymbol{\theta}, \quad (5.72)$$

$$\nabla \mathcal{L}(\hat{\boldsymbol{\theta}}) = 0 \iff \mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}}) = \lambda\hat{\boldsymbol{\theta}} \quad (5.73)$$

$$\iff \hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}. \quad (5.74)$$

Exercise 5.6. Prove that for $\lambda > 0$, $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ is invertible, even if the columns of \mathbf{X} are not linearly independent. \triangle

Bayesian Interpretation

We will now view the regularization penalty from a Bayesian point of view. As before, assume the Gaussian likelihood

$$\mathbf{y}|\boldsymbol{\theta}, \sigma^2 \sim \mathcal{N}(\mathbf{X}\boldsymbol{\theta}, \sigma^2\mathbf{I}). \quad (5.75)$$

For simplicity, we focus on estimating only $\boldsymbol{\theta}$ and not σ^2 . For the prior on $\boldsymbol{\theta}$, let

$$p(\boldsymbol{\theta}|\sigma^2) \sim \mathcal{N}(0, (\sigma^2/\lambda)\mathbf{I}) \propto e^{-\frac{\lambda\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\sigma^2}}. \quad (5.76)$$

Then

$$p(\boldsymbol{\theta}|\mathbf{y}, \sigma^2) \propto p(\mathbf{y}|\boldsymbol{\theta}, \sigma^2)p(\boldsymbol{\theta}|\sigma^2) \propto \exp\left(-\frac{(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) + \lambda\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\sigma^2}\right). \quad (5.77)$$

Based on the previous discussion, it is immediately clear that the mode of the posterior distribution for $\boldsymbol{\theta}$ is $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$. Furthermore, since the distribution is quadratic, and hence Gaussian, this is also the mean of the posterior. Hence the formulation for ridge regression is equivalent to assuming a zero-mean

Gaussian distribution for θ , which assigns high prior probabilities to smaller length of θ .

Lasso

In lasso, the regularization penalty has the form of the ℓ_1 norm,

$$\|\theta\|_1 = \sum_{i=1}^m |\theta_i|, \quad (5.78)$$

where m is the length of θ . The problem is to find

$$\hat{\theta} = \arg \min_{\theta} \|\mathbf{y} - \mathbf{X}\theta\|_2^2 + \lambda \|\theta\|_1, \quad (5.79)$$

or equivalently

$$\hat{\theta} = \arg \min_{\theta} \|\mathbf{y} - \mathbf{X}\theta\|_2^2, \quad (5.80)$$

$$\text{subject to : } \|\theta\|_1 \leq t. \quad (5.81)$$

Lasso does not have a closed form solution but efficient computational methods exist.

From a Bayesian point of view, lasso is equivalent to finding the *mode* of the posterior for θ assuming the same model as above but with the double exponential (Laplace) prior

$$p(\theta|\sigma^2) \propto e^{-\frac{\lambda \|\theta\|_1}{2\sigma^2}}. \quad (5.82)$$

Discussion and generalization

In general we could choose the regularization penalty to be of the form³

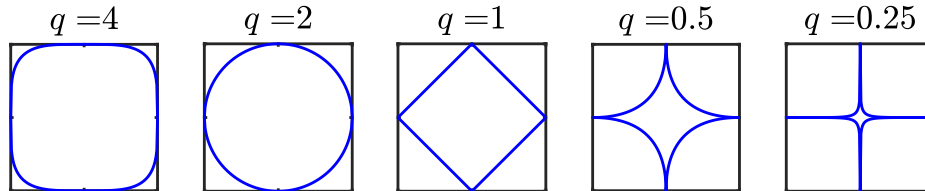
$$\|\theta\|_q^q = \sum_{i=1}^m |\theta_i|^q, \quad (5.83)$$

where m is the length of θ . For $q = 1$ and $q = 2$, we get lasso and ridge regression, respectively.

The effect of the regularization can be viewed from a Bayesian framework, by setting the prior

$$\exp\left(-\frac{\lambda}{2\sigma^2} \|\theta\|_q^q\right). \quad (5.84)$$

The contours for the priors for different values of q are given below.



In all cases, as we get further from the origin, the prior probability drops. But when q is small, the probability falls slower along the axes, encouraging solutions in which some of the coordinates are small or zero.

³ $\|\theta\|_q = (\sum_{i=1}^m |\theta_i|^q)^{1/q}$ is called the ℓ_q -norm of θ .

5.5 Error analysis and model selection

If our goal is to minimize the square of the prediction error, why would we use a different loss function for empirical risk minimization, as we did for ridge regression and lasso? How does this choice affect the error? Given that we have different choices for the form of the model and its parameters, how do we choose?

5.5.1 Bias-variance trade-off for quadratic error

Let us consider a general regression problem where we want to predict a value Y given an input vector \mathbf{x} . Let the prediction/estimate \hat{y} for Y given \mathbf{x} be denoted by $\hat{y} = f(\mathbf{x})$, where f is the function predicting Y given \mathbf{x} . For linear regression this is of the form $f(\mathbf{x}) = \mathbf{x}^T \hat{\boldsymbol{\theta}}$, so finding the predictor is the same as finding $\hat{\boldsymbol{\theta}}$.

For a *specific* estimator f (e.g., one found based on a given data set $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$), the expected value of the quadratic loss for the next data point is

$$\mathcal{L}(f) = \mathbb{E}[(Y - f(\mathbf{X}))^2 | \mathbf{X} = \mathbf{x}] \quad (5.85)$$

$$= \mathbb{E}[(Y - f(\mathbf{x}))^2 | \mathbf{X} = \mathbf{x}], \quad (5.86)$$

which is called the **test error** for \mathbf{x} . We can view this as the loss for the $n + 1$ st data point, where we are given $\mathbf{x} = \mathbf{x}_{n+1}$ and are interested in the loss of predicting Y_{n+1} . So, we are interested in evaluating f for a given input. For instance, in our marketing example from the beginning of the chapter, \mathbf{x}_{n+1} would indicate a specific budget, e.g., $\mathbf{x}_{n+1} = (\$10k, \$20k, \$5k)$.

Let $\bar{y}(\mathbf{x}) = \mathbb{E}[Y | \mathbf{X} = \mathbf{x}]$. Then, using the fact that $\mathbb{E}[(Z - c)^2] = \text{Var}(Z) + (\mathbb{E}[Z] - c)^2$, we have

$$\mathcal{L}(f) = \mathbb{E}[(Y - f(\mathbf{x}))^2 | \mathbf{X} = \mathbf{x}] \quad (5.87)$$

$$= \text{Var}(Y | \mathbf{X} = \mathbf{x}) + (\bar{y}(\mathbf{x}) - f(\mathbf{x}))^2. \quad (5.88)$$

Note that the error has two parts: **an irreducible part, referred to as intrinsic error, which is not under our control**, and **a part that depends on the choice of the predictor**. The intrinsic error results from the noise in “nature,” i.e., the fact that \mathbf{X} does not have enough information to fully determine Y . In other words, this term can be viewed as the accumulated effect of all factors that are not included in \mathbf{X} . Having a larger dataset or choosing a better f does not affect this term. The reducible part compares the performance of our predictor with the best possible. This error is minimized by setting $f(\mathbf{x}) = \bar{y}(\mathbf{x}) = \mathbb{E}[Y | \mathbf{X} = \mathbf{x}]$. However, doing so exactly is only possible if we have the distribution or an infinite amount of data.

To summarize, we can write **the test error for given f and \mathbf{x}** as

$$\mathcal{L}(f) = \text{irred.} + (f(\mathbf{x}) - \bar{y}(\mathbf{x}))^2. \quad (5.89)$$

We should choose f to minimize the above quantity. Let us consider how f is chosen through empirical risk minimization.

1. Determine a set \mathcal{F} from which f can be chosen, e.g., all linear functions.
2. Define an empirical loss. Typically, this reflects the loss function in the expected loss (5.86), but may include a regularization term, i.e., $\sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2 + \text{Reg.}$
3. Collect data, $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, and find $f \in \mathcal{F}$ that minimizes the empirical loss.

Consider a thought experiment in which this process is repeated many times. In each trial, the set \mathcal{F} and the definition of the empirical loss stay the same, while \mathcal{D} and consequently, f are random. Since f is a function of \mathcal{D} , let us denote it as $f_{\mathcal{D}}$. Let \mathcal{M} denote the fixed components of this process, i.e., the set \mathcal{F} and the definition of the empirical loss. We are interested to find the loss as a function of \mathcal{M} , which is under our control, averaged over all possible datasets (which is outside our control). This is called the **expected test**

error

$$\mathcal{L}(\mathcal{M}) = \mathbb{E}[\mathcal{L}(f_{\mathcal{D}})] = \text{irred.} + \mathbb{E}[(f_{\mathcal{D}}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2], \quad (5.90)$$

where the expectation is taken over all possible datasets. Note that the irreducible part, $\text{Var}(y|\mathbf{x})$ is a constant. With a similar trick as above, we have

$$\mathcal{L}(\mathcal{M}) - \text{irred.} = \mathbb{E}[(f_{\mathcal{D}}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2] \quad (5.91)$$

$$= \mathbb{E}[(f_{\mathcal{D}}(\mathbf{x}) - \mathbb{E} f_{\mathcal{D}}(\mathbf{x}))^2] + (\mathbb{E} f_{\mathcal{D}}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2, \quad (5.92)$$

where the last equality follows from the fact that $\mathbb{E} f_{\mathcal{D}}(\mathbf{x})$ and $\bar{y}(\mathbf{x})$ are constants for a given \mathbf{x} . The first term on the last line is the square of the bias and the second term is the variance of $f_{\mathcal{D}}(\mathbf{x})$. So

$$\mathcal{L}(\mathcal{M}) = \text{irred.} + (\text{bias})^2 + \text{variance} \quad (5.93)$$

Now, the loss is written as the sum of squared bias term, which compares the average prediction across all possible datasets with the best possible predictor, and a variance term, which quantifies how different the estimate for each dataset is from the average, across all datasets.

Typically, as model complexity/flexibility⁴ increases, bias decreases, while variance increases, since it has more freedom to vary based on the dataset. Simple/rigid models on the other hand typically have high bias and low variance. The bottom line is that neither unbiased models nor low variance predictors are necessary the best in terms of minimizing prediction error.

Another factor that affects the error is the size of the data. In many situations, the size of the data does affect the bias term but it may not significant since this term is averaged over data sets. The variance terms is affected because with more data, we get more information, approaching the situation in which $p(y|x)$ is known. So for very large datasets, the prediction reflects the distribution, which is constant.

5.5.2 Model Selection

As discussed above, the complexity of the model can affect the error. How can we choose the best model? In the past, we have used minimizing the empirical risk (training error) to optimize the parameters of a model. Can we again use the same strategy?

The **training error** for a given predictor f , i.e.,

$$\frac{1}{n} \sum L(y_i, f(\mathbf{x}_i)). \quad (5.94)$$

The training error itself is difficult to study. Averaged over all datasets, the **expected training error** is

$$\frac{1}{n} \sum \mathbb{E}[L(Y_i, f_{\mathcal{D}}(\mathbf{x}_i))], \quad (5.95)$$

where for simplicity, we have assumed that that the \mathbf{x}_i are fixed.

A typically behavior is in the table below, (this is not universally true). Training error is usually smaller for more complex models but this is not necessarily true for the test error. This makes choosing the best model based on training error difficult.

	Expected Train Err	Expected Test Err			
		Irred.	Bias ²	Var.	Total
More complex model	↓	—	↓	↑	?
More data	↑	—	↓	↓	↓

⁴By flexibility, I mean its responsiveness to changes in the data, i.e., the extent to which the results change when data changes.

5.5.2.1 Overfitting and Underfitting

Suppose the true relationship between two scalar variables x and Y is

$$y = ax + w, \quad w \sim \mathcal{N}(0, \sigma^2). \quad (5.96)$$

We assume that $\sigma < ax$ for typical values of x since otherwise, we cannot predict Y accurately even if a is known (the irreducible error is large relative to the best predicted value).

The data available to us consists of two points

$$\mathcal{D} = \{(x_1 = 1, y_1), (x_2 = 2, y_2)\}. \quad (5.97)$$

We consider predictors of the forms

- $\hat{y}(x) = 0$,
- $\hat{y}(x) = \theta x$,
- $\hat{y}(x) = \theta_1 x + \theta_2 x^2$.

For each predictor, we will find the parameter values $(\theta, \theta_1, \theta_2)$ that minimize the square loss for our data,

$$\frac{1}{2} [(y_1 - \hat{y}(x_1))^2 + (y_2 - \hat{y}(x_2))^2]. \quad (5.98)$$

For the third predictor, we will also consider the regularized version with loss

$$\frac{1}{2} [(y_1 - \hat{y}(x_1))^2 + (y_2 - \hat{y}(x_2))^2] + b\theta_2^2, \quad (5.99)$$

where b is a constant. Here, I chose the form $b\theta_2^2$ instead of the ℓ_2 norm, $b(\theta_1^2 + \theta_2^2)$ to simplify the derivation. We will still be able to see the effect of regularization.

We then find the expected error for the training data and for a test data point (x_3, y_3) , where we assume $x_3 = 3$. The expectation is taken over the randomness in y_1, y_2, y_3 . The results are given in the table below.

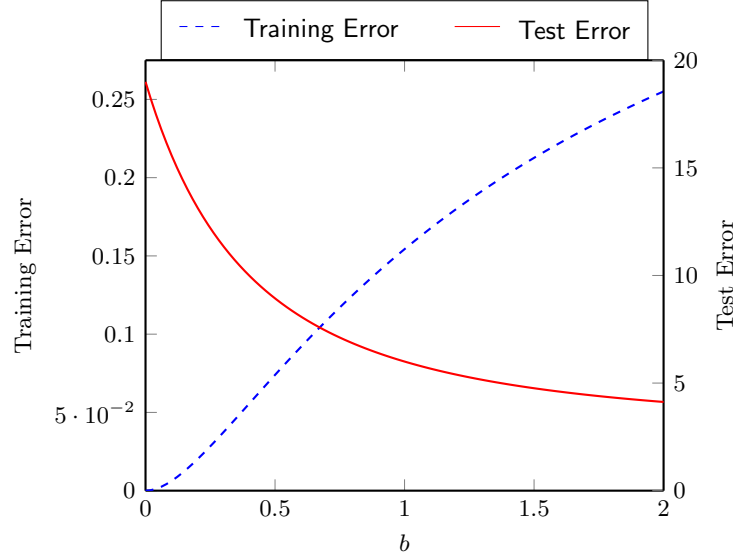
Prediction	Expected Train Err	Expected Test Error for $x_3 = 3$			
		Irr.	Bias ²	Var.	Total
$\hat{y}(x) = 0$	$\frac{5a^2}{2} + \sigma^2$	σ^2	$9a^2$	0	$9a^2 + \sigma^2$
$\hat{y}(x) = \frac{y_1 + 2y_2}{5}x$	$\frac{\sigma^2}{2}$	σ^2	0	$\frac{9}{5}\sigma^2$	$\frac{14}{5}\sigma^2$
$\hat{y}(x) = \frac{4y_1 - y_2}{2}x - \frac{2y_1 - y_2}{2}x^2$	0	σ^2	0	$18\sigma^2$	$19\sigma^2$
$\hat{y}(x) = \frac{b(y_1 + 2y_2) + 8y_1 - 2y_2}{5b + 4}x - \frac{4y_1 - 2y_2}{5b + 4}x^2$	$\frac{\sigma^2}{2} \left(1 - \frac{1}{5b/4 + 1}\right)^2$	σ^2	0	$\frac{9\sigma^2}{5} \left(1 + \frac{9}{(5b/4 + 1)^2}\right)$	$\frac{\sigma^2}{5} \left(14 + \frac{81}{(5b/4 + 1)^2}\right)$

As we go down the table, the model complexity increases. This allows the model to fit the training data better, leading to smaller expected training (square) error. The irreducible component of the test error stays the same, regardless of the model. The prediction bias for the test data point decreases, while its variance increases.

Given the assumption that σ is small relative to a , the smallest total error is obtained by the middle predictor. The zero predictor is not complex enough to be able to fit even the training data well. This situation is referred to as **underfitting**. The quadratic predictor is so complex that it can fit the training data, including the noise in the data, perfectly. But it does not generalize well due to its susceptibility to noise and high variance. This is called **overfitting**. In other words, the model memorizes this specific dataset rather than looking for patterns in it.

For the predictor with regularization, the graph below shows how the training and test errors change as a function of b . In this case, the predictor without regularization is overfitting the data. As b increases, over-

fitting decreases and we obtain a better test error. Note however that here the specific form of regularization prevents underfitting for large b , something that may occur in practice.



It is important to note models could perform poorly for reasons other than over- and under-fitting. For example, if the true distribution of the data is $y = a \sin x + w$, no polynomial predictor will perform well for a wide range of inputs due to the poor match between the true distribution and the learning model.

5.5.2.2 Training, validation, and test sets

In the previous subsection, we could identify the best model because we knew the true model for (\mathbf{x}, y) , which nature uses to produce them. In practice, however, the true model is not known and we cannot compute the expected test error. We have also seen that the training error is not necessarily a good estimate for the test error.

If we have sufficient data, a good solution is to divide it into three parts, a **training set**, a **validation set**, and a **test set**. For each model, the training set is used to optimize its parameters. Then all optimized models are evaluated on the validation set. Since the validation set is not used in training, this reduces the risk of over-fitting and, so, the errors for the validation set are better estimates for the test error. We choose the best model based on the validation set. We perform a final assessment using the test set, which should provide a good estimate of the error of the selected model for future practical use. Note that the test set cannot be used for any other purpose. If it is used in training or validation, it will not provide a reliable estimate of the error in the wild.

Example 5.7 (Regularization bias-variance trade-off). Regularization allows us to control the flexibility of the model. In ridge regression as λ increases, the model becomes more constrained. For $\lambda > 0$ it can be shown to be biased. With $\mathcal{D} = (\mathbf{X}, \mathbf{y})$,

$$\mathbb{E}[\hat{y}_{n+1}] = \mathbb{E}[\mathbf{x}_{n+1}^T \hat{\boldsymbol{\theta}}] \quad (5.100)$$

$$= \mathbf{x}_{n+1}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbb{E}[\mathbf{y}] \quad (5.101)$$

$$= \mathbf{x}_{n+1}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{X} \boldsymbol{\theta}. \quad (5.102)$$

Noting that $\mathbb{E}[y_{n+1}] = \mathbf{x}_{n+1}^T \boldsymbol{\theta}$, we see that the estimate of \hat{y}_{n+1} is biased. In particular, if $\mathbf{X}^T \mathbf{X} = \mathbf{I}$, then

$$\mathbb{E}[\hat{y}_{n+1}] = \frac{\mathbf{x}_{n+1}^T \boldsymbol{\theta}}{1 + \lambda} < \mathbf{x}_{n+1}^T \boldsymbol{\theta} = \mathbb{E}[y_{n+1}]. \quad (5.103)$$

But it can be shown to have lower variance. If the choice of λ is appropriate, it will have a smaller total

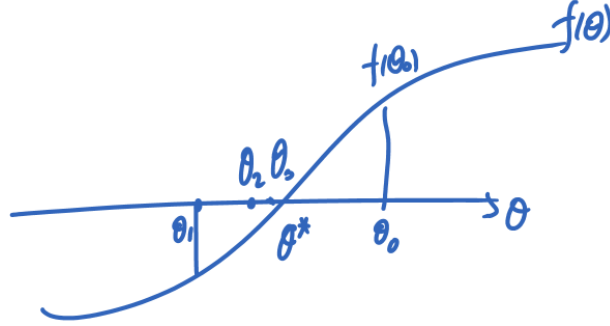
loss.

△

5.6 Stochastic Gradient Descent

Even though that gradient descent is sometimes less computationally expensive than directly finding the solution, its cost may still be high. In such cases, using *stochastic gradient descent* (SGD) may be helpful. SGD tries to improve the estimate by considering one data point (or a small batch of data points) at a time.

First, let's consider finding the root of a function $f(\theta)$ with a simple method. We assume that $f(\theta)$ is bounded and there is a unique root θ^* such that f is increasing at θ^* .



Suppose that we start from a point $\theta^{(0)}$ that is appropriately close to θ^* . We proceed iteratively as

$$\theta^{(t+1)} = \theta^{(t)} - a_t f(\theta^{(t)}), \quad (5.104)$$

where a_t satisfies

$$\sum_{t=1}^{\infty} a_t = \infty, \quad \sum_{t=1}^{\infty} a_t^2 < \infty. \quad (5.105)$$

For example, $a_t = 1/t$ is a good choice while $a_t = 1/t^2$ isn't. It can then be shown that $\theta^{(t)}$ converges to θ^* .

But what if we cannot compute $f(\theta)$ but instead we have access to a noisy version $F(\theta)$ that satisfies $f(\theta) = \mathbb{E}[F(\theta)]$, where $F(\theta)$ is bounded. It turns out that if we let

$$\theta^{(t+1)} = \theta^{(t)} - a_t F(\theta^{(t)}), \quad (5.106)$$

where in each iteration we sample $F(\theta)$, then $\theta^{(t)}$ again converges to θ^* .

Now let us consider the loss function for linear regression (note that we are using the expected loss as opposed to empirical loss)

$$\mathcal{L}(\theta) = \mathbb{E}[(y - \mathbf{x}^T \theta)^2], \quad (5.107)$$

where we are also assuming that \mathbf{x} is random with some distribution. To minimize this loss, we compute the gradient:

$$\nabla \mathcal{L}(\theta) = \mathbb{E}[-2(y - \mathbf{x}^T \theta) \mathbf{x}] \quad (5.108)$$

We would like to find θ such that the gradient above is zero.

Let

$$f(\theta) = \mathbb{E}[-2(y - \mathbf{x}^T \theta) \mathbf{x}] \quad (5.109)$$

$$F(\theta) = -2(y - \mathbf{x}^T \theta) \mathbf{x}, \quad (5.110)$$

so that $f(\theta) = \mathbb{E}[F(\theta)]$. Now the elements of the data set $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ can be used to produce

samples for $F(\boldsymbol{\theta})$. So we let

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + a_t(y_i - \mathbf{x}_i^T \boldsymbol{\theta}^{(t)})\mathbf{x}_i, \quad (5.111)$$

which is the stochastic gradient descent algorithm for linear regression.

Chapter 6

Linear Classification

In a classification problem, we have an input vector \mathbf{x} together with a corresponding label y . Based on a data set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, our goal is to predict y given a new value for \mathbf{x} . If y is a continuous variable the problem is that of regression, whereas in classification problems, y will represent a set of discrete class labels. For example, we may wish to classify images of handwritten digits. In this case, \mathbf{x} is a vector providing the values of pixels of the image and $y \in \{0, 1, \dots, 9\}$ is the label indicating what digit the image represents.

6.1 Overview of probabilistic models

The probabilistic approach to classification requires us to learn the distribution $p(y|\mathbf{x})$, which for any given \mathbf{x} provides the probability of belonging to different classes. We can identify the class for a given \mathbf{x} as the class that has the maximum probability,

$$\hat{y}(\mathbf{x}) = \arg \max_j p(y = j|\mathbf{x}).$$

This choice *minimizes the probability of predicting the wrong class*

$$\mathcal{L} = \Pr(\hat{y}(\mathbf{x}) \neq Y) = \mathbb{E}[I(Y \neq \hat{y}(\mathbf{x}))].$$

To find the distribution $p(y|\mathbf{x})$, our first step is developing a model that relates \mathbf{x} and y . There are two possible approaches.

We may develop a **generative model**, i.e., a model that is capable of *generating* data and also helping us predict y for a given \mathbf{x} . A generative model has two components, both of which must be learned from data:

- Prior class probabilities: $p(y)$
- Class-conditional probabilities: $p(\mathbf{x}|y)$

From these, using Bayes' theorem we can find $p(y|\mathbf{x})$ as

$$p(y|\mathbf{x}) = \frac{p(y)p(\mathbf{x}|y)}{p(\mathbf{x})} \propto p(y)p(\mathbf{x}|y).$$

We can often estimate $p(y = j)$ simply by computing the fraction of class j in our training data. For $p(\mathbf{x}|y)$, a common approach is to represent it parametrically and then learn the parameters from the data. For example, we may assume that given class j , \mathbf{x} is distributed normally with mean $\boldsymbol{\mu}_j$ and covariance matrix K_j and then learn these parameters from data.

Alternatively, we can develop a **discriminative model**. In this case, we directly model $p(y|\mathbf{x})$ since this is the distribution that we need to decide which class \mathbf{x} belongs to.

6.2 Generative Probabilistic Models

6.2.1 Gaussian Class-Conditionals

Let us denote

$$p(Y = j) = \pi_j.$$

We further assume $p(\mathbf{x}|Y = j)$ is Gaussian with mean $\boldsymbol{\mu}_j$ and covariance matrix Σ_j ,

$$p(\mathbf{x}|Y = j) = \frac{1}{\sqrt{2\pi|\Sigma_j|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j)\right)$$

For our purpose, it suffices to consider $\ln p(\mathbf{x}|y = j)$, which, after dropping the constant terms, becomes

$$\ln p(\mathbf{x}|Y = j) \doteq -\frac{1}{2} \ln |\Sigma_j| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j).$$

From these, we can find $\ln p(y = j|\mathbf{x})$ and then decide $\hat{y}(\mathbf{x})$ as

$$\hat{y}(\mathbf{x}) = \arg \max_j \ln p(y = j|\mathbf{x}).$$

More specifically,

$$\begin{aligned} \ln p(y = j|\mathbf{x}) &\doteq \ln \pi_j - \frac{1}{2} \ln |\Sigma_j| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) \\ &= \mathbf{x}^T \left(-\frac{1}{2} \Sigma_j^{-1} \right) \mathbf{x} + (\boldsymbol{\mu}_j^T \Sigma_j^{-1}) \mathbf{x} + \left(-\frac{1}{2} \boldsymbol{\mu}_j^T \Sigma_j^{-1} \boldsymbol{\mu}_j - \frac{1}{2} \ln |\Sigma_j| + \ln \pi_j \right) \\ &= \mathbf{x}^T A_j \mathbf{x} + \boldsymbol{\beta}_j^T \mathbf{x} + \gamma_j, \end{aligned} \quad (6.1)$$

For an appropriately defined symmetric matrix A_j , vector $\boldsymbol{\beta}_j$, and scalar γ_j . In the next two sections, we will consider two cases based on whether the covariance matrix depends on the class.

6.2.2 Linear Discriminant Analysis

First, let us suppose all classes have the same covariance matrix $\Sigma_j = \Sigma$. Then, the terms $\mathbf{x}^T (-\frac{1}{2} \Sigma^{-1}) \mathbf{x}$ and $-\frac{1}{2} \ln |\Sigma_j|$ in (6.1) become independent of the class and we thus have

$$\ln p(y = j|\mathbf{x}) \doteq \boldsymbol{\beta}_j^T \mathbf{x} + \gamma_j, \quad (6.2)$$

where

$$\boldsymbol{\beta}_j^T = \boldsymbol{\mu}_j^T \Sigma^{-1}, \quad \gamma_j = -\frac{1}{2} \boldsymbol{\mu}_j^T \Sigma^{-1} \boldsymbol{\mu}_j + \ln \pi_j,$$

Suppose we have only two classes, $y = 0$ and $y = 1$, with $p(y = 1) = \pi = 1 - p(y = 0)$. Equivalent to finding $\arg \max_j \ln p(y = j|\mathbf{x})$ for each \mathbf{x} , we can divide the space into two regions,

$$\ln p(y = 1|\mathbf{x}) \underset{\hat{y}=0}{\overset{\hat{y}=1}{\geq}} \ln p(y = 0|\mathbf{x}).$$

What is the decision boundary between them? We can find it by solving $\ln p(y = 1|\mathbf{x}) = \ln p(y = 0|\mathbf{x})$,

$$\boldsymbol{\beta}_1^T \mathbf{x} + \gamma_1 = \boldsymbol{\beta}_0^T \mathbf{x} + \gamma_0 \iff (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0)^T \mathbf{x} + \gamma_1 - \gamma_0 = 0 \iff \boldsymbol{\beta}^T \mathbf{x} + \gamma = 0,$$

where

$$\boldsymbol{\beta}^T = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \Sigma^{-1}, \quad \gamma = \ln \frac{\pi}{1 - \pi} - \frac{1}{2} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \Sigma^{-1} (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0). \quad (6.3)$$

Hence, the decision boundary is the hyperplane $\boldsymbol{\beta}^T \mathbf{x} + \gamma = 0$. On one side of this plane, we predict class 1

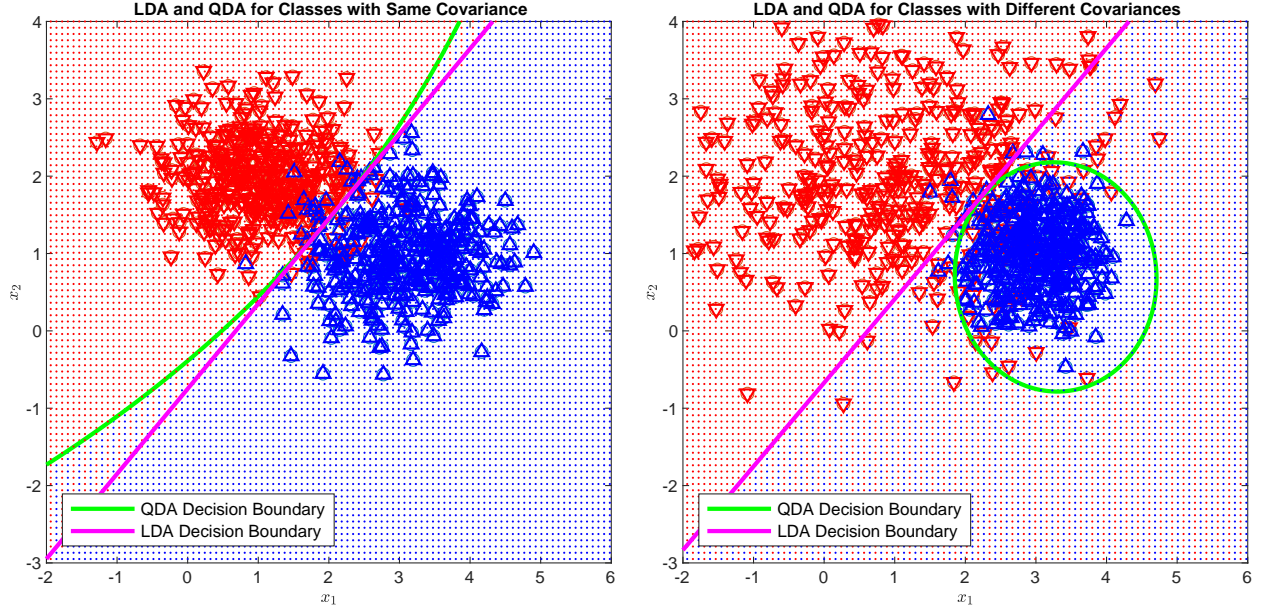


Figure 6.1: LDA vs QDA for when $\Sigma_1 = \Sigma_2$ (left) and when $\Sigma_1 \neq \Sigma_2$ (right).

(we let $\hat{y}(\mathbf{x}) = 1$) and on the other side, we declare class 0:

$$\hat{y}(\mathbf{x}) = \begin{cases} 1, & \beta^T \mathbf{x} + \gamma > 0 \\ 0, & \beta^T \mathbf{x} + \gamma < 0 \end{cases}$$

Since the boundary is linear (i.e., a hyperplane such as a line, 2-D plane, etc), this method is called **Linear Discriminant Analysis** (LDA).

As a special case, consider, $\pi = \frac{1}{2}$, $\Sigma = I$. The the boundary becomes

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \left(\mathbf{x} - \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0}{2} \right) = 0,$$

which implies that the boundary is the plane that passes through the midpoint of the line connecting $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_0$ and is perpendicular to it.

What about the probability $p(y|\mathbf{x})$ of each class for a given \mathbf{x} , which can tell us about the certainty of belonging to each class? From (6.2), we have $p(y = j|\mathbf{x}) \propto e^{\beta_j^T \mathbf{x} + \gamma_j}$ and so for two classes

$$p(y = 1|\mathbf{x}) = \frac{e^{\beta_1^T \mathbf{x} + \gamma_1}}{e^{\beta_1^T \mathbf{x} + \gamma_1} + e^{\beta_0^T \mathbf{x} + \gamma_0}} = \frac{1}{1 + e^{-(\beta^T \mathbf{x} + \gamma)}} = \sigma(\beta^T \mathbf{x} + \gamma),$$

where β and γ are given in (6.3), and $\sigma(u) = \frac{1}{1+e^{-u}}$ is the sigmoid (logistic) function.

If there are $c > 2$ classes, decision hyperplanes between pairs of classes will divide the space into c regions. And the conditional probability of class j is given by

$$p(y = j|\mathbf{x}) = \frac{e^{\beta_j^T \mathbf{x} + \gamma_j}}{\sum_{k=1}^c e^{\beta_k^T \mathbf{x} + \gamma_k}} = \sigma_j(\beta_1^T \mathbf{x} + \gamma_1, \dots, \beta_c^T \mathbf{x} + \gamma_c),$$

where $\sigma_j(\mathbf{v}) = \frac{e^{v_j}}{\sum_k e^{v_k}}$ is the softmax function.

6.2.3 Quadratic Discriminant Analysis

Let us now assume that each class has a different covariance matrix Σ_j . To decide between two classes, say $y = 0$ and $y = 1$, the decision boundary is given by $\ln p(y = 1|\mathbf{x}) = \ln p(y = 0|\mathbf{x})$. This will lead to a quadratic equation of the form $\mathbf{x}^T A \mathbf{x} + \boldsymbol{\beta}^T \mathbf{x} + \gamma = 0$, which leads to a nonlinear decision boundary. As a result, this method is called **Quadratic Discriminant Analysis** (QDA).

Figure 6.1 demonstrates LDA and QDA when $\Sigma_1 = \Sigma_2$ (left) and when $\Sigma_1 \neq \Sigma_2$ (right). Here the boundaries are learned from data (see Section 6.2.2). On the left the data is generated by distributions that match the assumption made by LDA and so LDA and QDA perform similarly. However, on the right the covariances are different and so, as expected, QDA performs better. Note however that we could augment our feature vectors as $(x_1, x_2, x_1x_2, x_1^2, x_2^2)$ instead of just (x_1, x_2) and then apply LDA, allowing a decision boundary that is not linear in x_1, x_2 . In that case, the performance of LDA would generally be similar to that of QDA (Hastie et al., Elements of Statistical Learning).

6.2.4 Maximum Likelihood Solution to LDA

Once we specified a parametric form for the class-conditional densities $p(\mathbf{x}|y = j)$, we can determine the values of the parameters, together with the prior class probabilities $p(y = j)$, using maximum likelihood.

Data: Our data set comprises of observations of \mathbf{x} along with their corresponding class labels. Let the n independent samples be denoted by $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{0, 1\}$ for all i .

Model:

$$p(y = j) = \begin{cases} \pi, & j = 1 \\ 1 - \pi, & j = 0 \end{cases}$$

$$\begin{aligned} p(\mathbf{x}|y = 0) &\sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma), \\ p(\mathbf{x}|y = 1) &\sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma), \end{aligned}$$

for some $\pi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1$ and diagonal matrix $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2)$. The diagonal covariance matrix implies conditional independence among the elements of \mathbf{x} . That is, for a given data point (\mathbf{x}_i, y_i) , depending on the value of y_i , we have one of the following cases

$$p(\mathbf{x}_i|y_i = 0) = \prod_{j=1}^m p(x_{ij}|y_i = 0) = \prod_{j=1}^m \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_{ij} - \mu_{0j})^2}{2\sigma_j^2}\right) \quad (6.4)$$

$$p(\mathbf{x}_i|y_i = 1) = \prod_{j=1}^m p(x_{ij}|y_i = 1) = \prod_{j=1}^m \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_{ij} - \mu_{1j})^2}{2\sigma_j^2}\right) \quad (6.5)$$

Note that since we assume both classes have the same covariance matrix, the decision boundary will be linear (i.e., LDA). Also, we have assumed given the class, features are independent (since Σ is diagonal); this is called the **Naive Bayes** model.

Likelihood:

$$\begin{aligned}
 p(\mathcal{D}|\pi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \Sigma) &= \prod_{i=1}^n p(y_i) p(\mathbf{x}_i|y_i) \\
 &= \left(\prod_{i:y_i=0} p(y_i=0) p(\mathbf{x}_i|y_i=0) \right) \left(\prod_{i:y_i=1} p(y_i=1) p(\mathbf{x}_i|y_i=1) \right) \\
 &= \left(\prod_{i:y_i=0} (1-\pi) \prod_{j=1}^m \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_{ij}-\mu_{0j})^2}{2\sigma_j^2}\right) \right) \left(\prod_{i:y_i=1} \pi \prod_{j=1}^m \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_{ij}-\mu_{1j})^2}{2\sigma_j^2}\right) \right),
 \end{aligned}$$

where $x_{i,j}$ is the j^{th} component of \mathbf{x}_i and $\mu_{0,j}, \mu_{1,j}$ are the j^{th} components of $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$, respectively. The maximum likelihood solution is (exercise)

$$\begin{aligned}
 \hat{\pi}_{ML} &= \frac{\sum y_i}{n}, \\
 (\hat{\mu}_{0,j})_{ML} &= \frac{\sum_{i:y_i=0} x_{i,j}}{\sum (1-y_i)}, \quad (\hat{\mu}_{1,j})_{ML} = \frac{\sum_{i:y_i=1} x_{i,j}}{\sum y_i}, \\
 (\hat{\sigma}_j^2)_{ML} &= \frac{1}{n} \left(\sum_{i:y_i=0} (x_{i,j} - \hat{\mu}_{0,j})^2 + \sum_{i:y_i=1} (x_{i,j} - \hat{\mu}_{1,j})^2 \right)
 \end{aligned}$$

6.2.5 Generative Model for Discrete Features **

If a features is categorical, for example, type of a vehicle or genre of a movie, we can encode them as binary vectors. For example, if there are three categories, with the vector $(1,0,0)$ we can indicate belonging to the first category. This is called *one-hot* or *dummy encoding*. In this case, our data is still denoted by $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, where each \mathbf{x}_i is composed of vectors, that is¹

$$\mathbf{x}_i = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{im}),$$

and each $\mathbf{x}_{ij} = (x_{ij1}, \dots, x_{ijl}, \dots)$ is a binary vector of finite length which represents a one-hot encoding of a feature.

Example 6.1 (One-hot encoding). Suppose $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ provide information about a set of movies, where $\mathbf{x}_1 = (\mathbf{x}_{11}, \dots, \mathbf{x}_{1m})$, $\mathbf{x}_2 = (\mathbf{x}_{21}, \dots, \mathbf{x}_{2m})$, ..., with $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ denoting in order the genre of the movie, the director of the movie, etc. Explicitly, for the genre, if we order them as (comedy, horror, drama, scifi, action), and for five directors A, B, C, D, E , order them as (A, B, C, D, E) , then $\mathbf{x}_{11} = (0, 0, 1, 0, 0)$, $\mathbf{x}_{12} = (0, 0, 0, 1, 0)$ means movie 1 is a drama directed by director D , and $\mathbf{x}_{21} = (1, 0, 0, 0, 0)$, $\mathbf{x}_{22} = (1, 0, 0, 0, 0)$ means that movie 2 is a comedy directed by director A . \triangle

Model: We model this classification problem in the following way:

$$p(x_{ijl} = 1|y_i = k) = \eta_{kjl}, \quad \sum_l \eta_{kjl} = 1,$$

and all x_{ijl} are independent from one another. For two vectors \mathbf{a}, \mathbf{b} with the same length, we define $\mathbf{a}^{\mathbf{b}} = \prod_{i=1}^{|\mathbf{a}|} a_i^{b_i}$. Let $\boldsymbol{\eta}_{kj} = (\eta_{kj1}, \dots)$. We have

$$p(\mathbf{x}_{ij}|y_i = k) = \boldsymbol{\eta}_{kj}^{\mathbf{x}_{ij}},$$

¹All vectors in this section are column vectors and all concatenations are also along the vertical dimension. However, for simplicity of notation, we write $\mathbf{x}_i = (\mathbf{x}_{i1}, \dots, \mathbf{x}_{im})$ instead of $\mathbf{x}_i^T = (\mathbf{x}_{i1}^T, \dots, \mathbf{x}_{im}^T)^T$

and then

$$p(\mathbf{x}_i|y_i = k) = \prod_{j=1}^m p(\mathbf{x}_{ij}|y_i = k) = \prod_{j=1}^m \eta_{kj}^{\mathbf{x}_{ij}} = \boldsymbol{\eta}_k^{\mathbf{x}_i},$$

where $\boldsymbol{\eta}_k = (\eta_{k1}, \dots, \eta_{km})$. It follows that

$$p(y_i = k|\mathbf{x}_i) \propto p(\mathbf{x}_i|y_i = k)p(y_i = k) = \pi_k \boldsymbol{\eta}_k^{\mathbf{x}_i} \propto \exp(\ln \pi_k + \mathbf{x}_i^T \ln \boldsymbol{\eta}_k).$$

For a new data point (\mathbf{x}, y) , we similarly have

$$\ln p(y = k|\mathbf{x}) \doteq \boldsymbol{\beta}_k^T \mathbf{x} + \gamma_k, \quad (6.6)$$

where $\boldsymbol{\beta}_k = \ln \boldsymbol{\eta}_k$ and $\gamma_k = \ln \pi_k$. The log-probabilities are again linear in \mathbf{x} , an fact that as we will see contributes to the motivation for logistic regression.

6.2.6 Class-conditionals from the exponential family

The exponential family of distributions includes common distributions such as Gaussian, exponential, gamma, beta, Dirichlet, Bernoulli, Poisson, and geometric. Distributions from this family have the following form

$$p(\mathbf{x}|\boldsymbol{\theta}) = \exp[\mathbf{b}(\boldsymbol{\theta})^T \mathbf{a}(\mathbf{x}) + f(\mathbf{x}) + g(\boldsymbol{\theta})].$$

Let us consider the case in which $\mathbf{a}(\mathbf{x}) = \mathbf{x}$, and parameters are functions of class y . So instead of $\boldsymbol{\theta}$ we write $\boldsymbol{\theta}_j$, when considering the j th class. Then the class-conditional distribution will become

$$p(\mathbf{x}|y = j) = \exp[\mathbf{b}(\boldsymbol{\theta}_j)^T \mathbf{x} + f(\mathbf{x}) + g(\boldsymbol{\theta}_j)].$$

Furthermore, let $p(y = j) = \pi_j$. Given \mathbf{x} , the log-probability of each class is given as

$$\ln p(y = j|\mathbf{x}) \doteq \ln \pi_j + \ln p(\mathbf{x}|y = j) \doteq \ln \pi_j + \mathbf{b}(\boldsymbol{\theta}_j)^T \mathbf{x} + g(\boldsymbol{\theta}_j) \doteq \boldsymbol{\beta}_j^T \mathbf{x} + \gamma_j, \quad (6.7)$$

where $\boldsymbol{\beta}_j = \mathbf{b}(\boldsymbol{\theta}_j)$ and $\gamma_j = \ln \pi_j + g(\boldsymbol{\theta}_j)$. So for a large class of class-conditional probabilities, the log-probabilities of classes given the feature vector \mathbf{x} is linear in \mathbf{x} .

6.3 Discriminative Models and Logistic Regression

In the discriminative approach, we model $p(y = j|\mathbf{x})$ directly. But what is a good model for this conditional distribution? As we have seen in (6.2), (6.6) and (6.7), in many generative cases, the log-probabilities of classes given data is linear in \mathbf{x} ,

$$\ln p(y = j|\mathbf{x}) \doteq \boldsymbol{\beta}_j^T \mathbf{x} + \gamma_j.$$

And based on Section 6.2.2, this form leads to linear class boundaries and posterior class probabilities of the logistic form for two classes,

$$p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\boldsymbol{\beta}^T \mathbf{x} + \gamma)}}, \quad p(y = 0|\mathbf{x}) = \frac{e^{-(\boldsymbol{\beta}^T \mathbf{x} + \gamma)}}{1 + e^{-(\boldsymbol{\beta}^T \mathbf{x} + \gamma)}},$$

where $\boldsymbol{\beta} = \boldsymbol{\beta}_1 - \boldsymbol{\beta}_2$ and $\gamma = \gamma_1 - \gamma_2$.

Limiting ourselves to two classes, this observation raises the following question: ‘Why not assume from the beginning that $p(y|\mathbf{x})$ is of the logistic form and learn this distribution instead of learning first $p(\mathbf{x}|y)$ and $p(y)$?’ Doing so leads to a *discriminative model* resulting in *logistic regression*.

Let $h(\mathbf{x}) = p(y = 1|\mathbf{x})$ and assume that the data consists of n iid samples, $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$. We have

$$p(\mathcal{D}; \boldsymbol{\beta}, \gamma) = \prod_{i=1}^n h(\mathbf{x}_i)^{y_i} (1 - h(\mathbf{x}_i))^{1-y_i}$$

and the negative log-likelihood loss is given by

$$-\mathcal{L}(\beta, \gamma) = \sum_{i=1}^n \left(y_i \ln \frac{1}{h(\mathbf{x}_i)} + (1 - y_i) \ln \frac{1}{1 - h(\mathbf{x}_i)} \right). \quad (6.8)$$

We can use gradient descent to minimize this loss (maximize the likelihood). For simplicity, let $\theta = \begin{pmatrix} \beta \\ \gamma \end{pmatrix}$, $\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$, and $h_\theta = p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-\theta^T \tilde{\mathbf{x}}}}$. Then

$$\theta^{(t+1)} = \theta^{(t)} + \rho_t \nabla_\theta \mathcal{L}(\theta),$$

where

$$\nabla_\theta \mathcal{L}(\theta) = \sum_{i=1}^n (y_i - h_\theta(\tilde{\mathbf{x}}_i)) \tilde{\mathbf{x}}_i.$$

When we find θ , and thus β, γ , we have the decision boundary as $\beta^T \mathbf{x} + \gamma = 0$. Points \mathbf{x} for which $\beta^T \mathbf{x} + \gamma > 0$ are classified as class $y = 1$.

6.4 Risk minimization and loss functions for classification

An alternative approach to generative models and logistic regression we discussed before is directly minimizing an empirical loss,

$$\frac{1}{n} \sum_{i=1}^n L(y_i, \mathbf{x}_i, \hat{y}(\mathbf{x}_i)),$$

where $\hat{y}(\mathbf{x})$ is the predictor of the class for input vector \mathbf{x} . For ease of exposition, instead of assuming $y \in \{0, 1\}$, we assume $y \in \{-1, 1\}$.

Our attention will be limited to linear classifiers, determined by a vector β and a constant γ , which define the hyperplane $\beta^T \mathbf{x} + \gamma = 0$. On one side of the hyperplane, we decide class 1 and the other side class -1,

$$\hat{y}(\mathbf{x}) = \text{sign}(\beta^T \mathbf{x} + \gamma) = \begin{cases} 1, & \text{if } \beta^T \mathbf{x} + \gamma > 0, \\ -1, & \text{if } \beta^T \mathbf{x} + \gamma < 0, \end{cases}$$

where the dependence of \hat{y} on β, γ is implicit.

One such linear classifier is shown in Figure 6.2. Below, we will use the fact that for any point \mathbf{x}_i with label y_i and prediction $\hat{y}(\mathbf{x}_i)$, the loss contributed by it can often be viewed as a function of its signed distance d_i to the decision hyperplane. Without loss of generality, assume that $y_i = 1$ and $\mathbf{x}_i = \mathbf{x}_0 + d_i \beta / \|\beta\|$ for some \mathbf{x}_0 on the decision boundary. If d_i is positive, then this point is classified correctly, since $\beta^T \mathbf{x}_i + \gamma > 0$. The distance between \mathbf{x}_i and the decision boundary equals $|d_i|$.

6.4.1 Zero-one loss

The most natural loss function for classification is the **0-1 loss**,

$$L_{01}(y, \hat{y}(\mathbf{x})) = \begin{cases} 1, & \text{if } y \neq \hat{y}(\mathbf{x}) \end{cases} = \begin{cases} 1, & \text{if } y(\beta^T \mathbf{x} + \gamma) < 0, \\ 0, & \text{if } y(\beta^T \mathbf{x} + \gamma) > 0. \end{cases}$$

Figure 6.3 shows the 0-1 loss for a point in the positive class. Note that how far the point is from the boundary does not affect how much it contributes to the loss.

Unfortunately, minimizing this loss function is computationally difficult (NP-hard) [1]. So in practice, we use differentiable loss-functions for which efficient algorithms exist. Here we will consider two such loss functions.

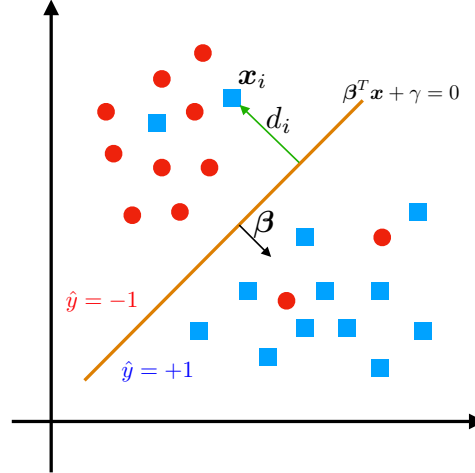


Figure 6.2: A linear classifier defined by the vector β and scalar γ . Squares represents points with $y = +1$ and circles $y = -1$. For a point \mathbf{x}_i , many loss functions can be viewed as a function of the signed distance d_i of \mathbf{x}_i to the decision hyperplane.

First, we view logistic regression in terms of empirical risk minimization and then we will consider the hinge loss in the context of support vector machine (SVM) classifiers.

6.4.2 Logistic regression

Let us re-examine the logistic regression loss function (6.8). The loss incurred by a data point \mathbf{x}_i at signed distance d_i from the decision hyperplane (i.e., $\mathbf{x}_i = \mathbf{x}_0 + d_i\beta/\|\beta\|$) is

$$\ln(1 + e^{-(\beta^T \mathbf{x}_i + \gamma)}) = \ln(1 + e^{-d_i \|\beta\|}).$$

The figure below shows this loss: For $d_i < 0$, where the input is misclassified, the loss is larger, and it increases as the point gets farther from the boundary. But even for points that are classified correctly, there is a loss, which decreases as we get farther from the boundary.

6.4.3 Hinge loss (SVM)

Hinge loss results from penalizing misclassified points as well as those that are classified correctly, but are within a certain margin close to the decision boundary. The expression for hinge loss is

$$\max(0, 1 - y_i(\beta^T \mathbf{x}_i + \gamma)).$$

Letting $y_i = 1$ and $\mathbf{x}_i = \mathbf{x}_0 + d_i\beta/\|\beta\|$ as before, results in

$$\max(0, 1 - d_i \|\beta\|).$$

which is shown in Figure 6.3. So the penalty for misclassified points is larger the farther away they are from the boundary. In addition, even points classified correctly are penalized if they are within a **margin** of width $1/\|\beta\|$ of the decision boundary.

In addition to penalizing points within the margin, we would like to ensure that the margin is not very small. This can be done by ensuring $1/\|\beta\|$ is large or equivalently $\|\beta\|^2$ is small. Both of these goals can be achieved with the loss

$$\frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i(\beta^T \mathbf{x}_i + \gamma)) + \lambda \|\beta\|^2, \quad (6.9)$$

where λ is a constant that balances the two components of the loss. This results in the so called **support**

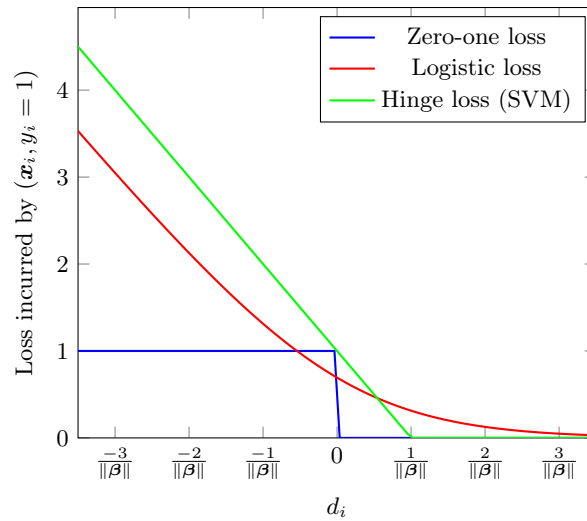


Figure 6.3: Loss functions for a data point $(\mathbf{x}_i, y_i = 1)$ as a function of the distance of \mathbf{x}_i from the boundary (in terms of the length of $\boldsymbol{\beta}$).

vector machine classifier (SVM).

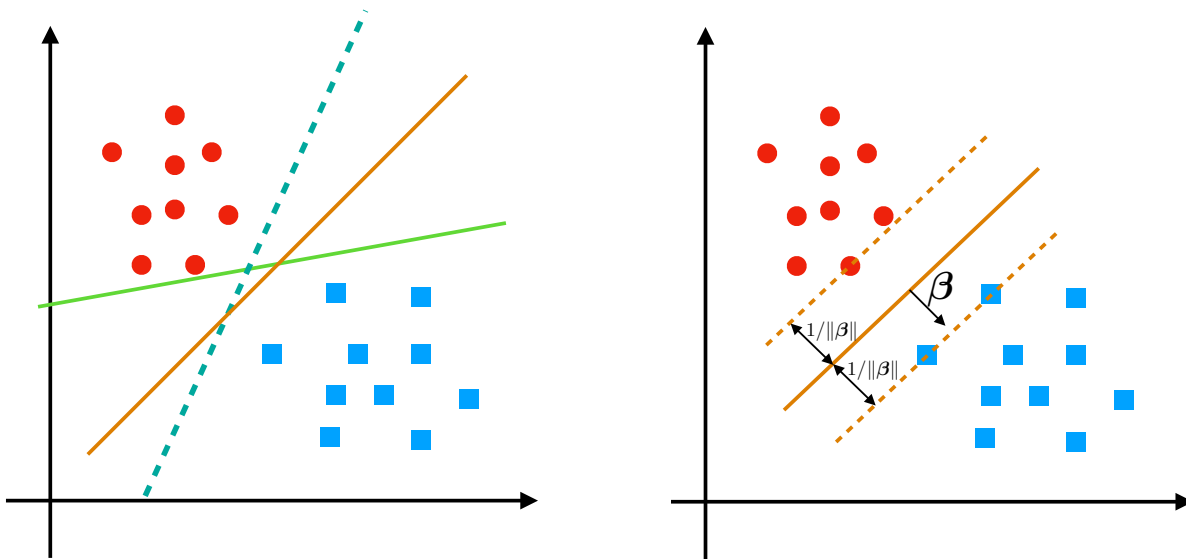
SVM as maximum-margin classifier. Let us consider the case in which the data is linearly separable, i.e., there exists a hyperplane that correctly classifies all data points. In such a case, as shown in Figure 6.4a, there are typically an infinite number of separating hyperplanes. This leads to the question of which one should be chosen. The SVM loss given in (6.9) provides a solution. Assume λ is positive but very small. So we are primarily concerned about the first term in the loss, i.e., the hinge loss. Between choices that incur the same hinge loss, we must pick the one that maximizes the margin, i.e., minimizes $\|\boldsymbol{\beta}\|^2$. Thus:

- We can make the hinge loss term zero by choosing any separating hyperplane that makes no mistakes and choosing any margin (length of $\|\boldsymbol{\beta}\|$) that is small enough such that there no points within the margin.
- Now the second term ensures that among the hyperplanes that perfectly separate the data, we should pick the one that has the maximum margin, as shown in Fig. 6.4b.

In nonlinear cases, SVM can use “kernel functions” to transform the input space into a higher-dimensional space where it is easier to find a linear separating hyperplane. This, along with a related computational technique, is known as the kernel trick, allowing SVM to effectively perform in complex, nonlinear classification tasks.

References

- [1] Tan T. Nguyen and Scott Sanner. “Algorithms for direct 0-1 loss optimization in binary classification.” In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML’13. Atlanta, GA, USA: JMLR.org, June 2013, pp. III–1085–III–1093. (Visited on 03/23/2020).



(a) For two linearly separable classes, there are an infinite number of classifiers that perfectly separate the training data. Which one should we pick?

(b) The maximum-margin classifier is the classifier that maximizes the distance between the decision boundary and the closest points to it.

Figure 6.4: SVM for separable data

Chapter 7

Expectation-Maximization *

7.1 Overview

Expectation-maximization (EM) is a method for dealing with missing data/hidden variables. In other words, part of the variables in the assumed model do not have associated data points. For example, for classification, the complete data consists of the features \mathbf{x} and labels y , as shown in the left panel of Figure 7.1. With a probabilistic model for this data, we can find the parameters for each class through maximum likelihood, where the log-likelihood function is

$$\log p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}),$$

where $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ and $\boldsymbol{\theta}$ represents the parameters of class-conditional distributions for each of the classes.

But what if the class labels are not given as in the right panel of Figure 7.1? The problem becomes more difficult, but doesn't seem hopeless as we can still distinguish two clusters and assign points to these with various degrees of confidence.

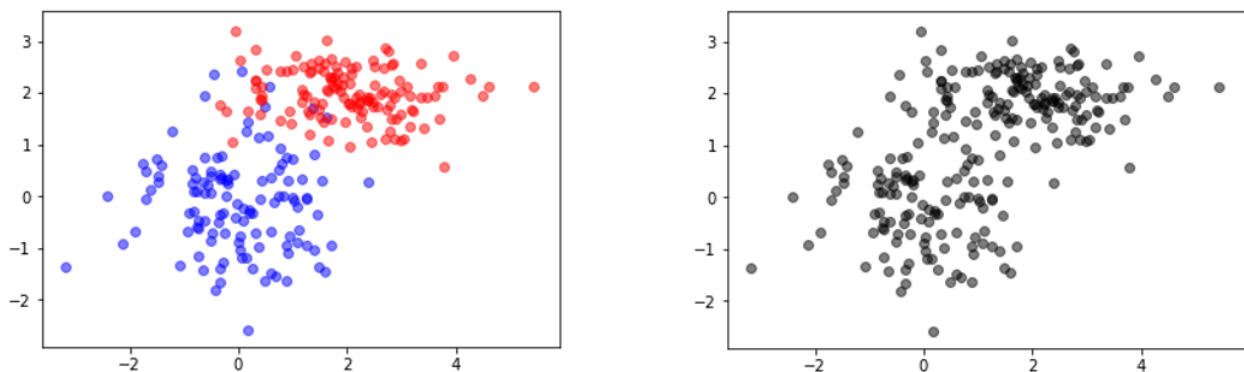


Figure 7.1: Data from two classes, with labels given as colors (left) and not given (right).

We thus formulate this problem as finding $\boldsymbol{\theta}$ that maximizes

$$\log p(\mathbf{x}; \boldsymbol{\theta}) = \log \sum_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$$

In this case, (\mathbf{x}, \mathbf{y}) is the complete data, for which computing the likelihood is easy, but a component of this data, namely \mathbf{y} , is missing. Now computing the likelihood is difficult because of the summation, which is typically over a large number of possibilities. Expectation-maximization is a method for handling missing

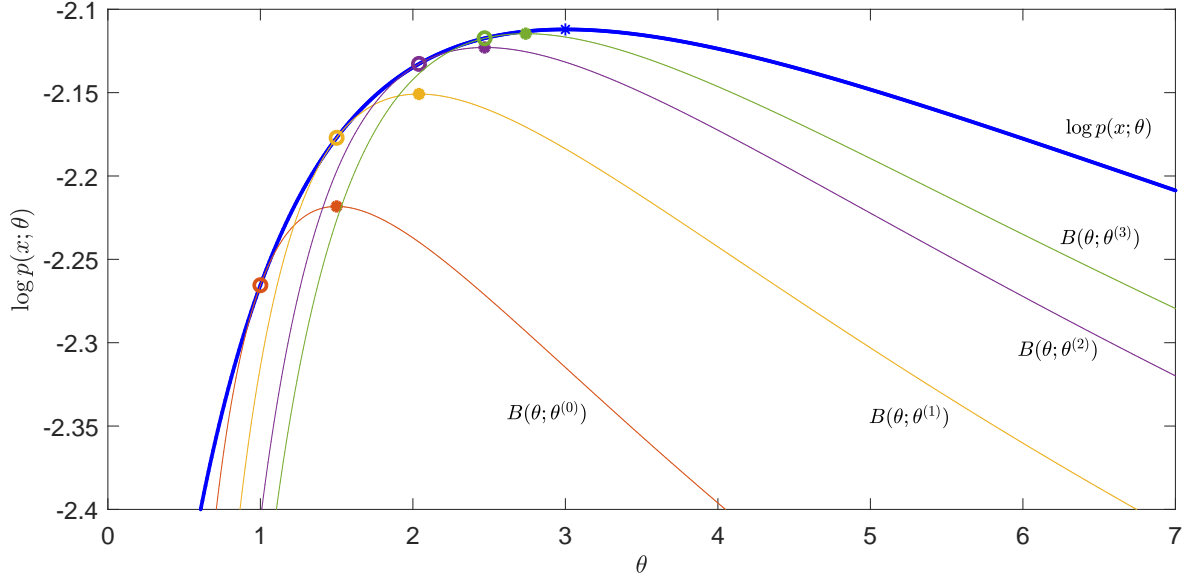


Figure 7.2: The log-likelihood of the observation and consecutive EM lower bounds and estimates. In each iteration, the current value of θ is denoted by \circ and the new value by $*$. Here, $\theta^{(0)} = 1$, $\theta^{(1)} = 1.5$, $\theta^{(2)} = 2.04$, $\theta^{(3)} = 2.472$. Continuing in the same manner, we would obtain estimates 2.740, 2.880, 2.946, 2.976, \dots , where 3 is the true maximum.

data.

EM is an iterative method that given the current estimate for the parameter, finds a new estimate. The idea behind EM is finding lower bounds on the log-likelihood of the observed data and maximizing these lower bounds. This is illustrated in Figure 7.2 (see Example 7.1). Suppose our current estimate of θ is θ' . In each iteration, we find a lower bound $B(\theta, \theta')$ on $\log p(\mathbf{x}; \theta)$ that coincides with it at $\theta = \theta'$, i.e.,

$$\begin{aligned} \log p(\mathbf{x}; \theta) &\geq B(\theta, \theta'), & \text{for all } \theta, \\ \log p(\mathbf{x}; \theta) &= B(\theta, \theta'), & \text{for } \theta = \theta'. \end{aligned} \quad (7.1)$$

Now let our new estimate be

$$\theta'' = \arg \max_{\theta} B(\theta, \theta').$$

Note that we have not used $\log p(\mathbf{x}; \theta)$ to find θ'' . Since

$$\log p(\mathbf{x}, \theta'') \geq B(\theta'', \theta') \geq B(\theta', \theta') \geq \log p(\mathbf{x}, \theta'),$$

we have

$$\log p(\mathbf{x}; \theta'') \geq \log p(\mathbf{x}; \theta').$$

So our new estimate is at least as good as the old one, and under certain conditions, it is going to be strictly better. We then use θ'' in place of θ' and repeat. Note that if $\log p(\mathbf{x}; \theta)$ is bounded, since the sequence $\log p(\mathbf{x}; \theta')$ is non-decreasing, it will converge. Under appropriate conditions, this means that θ' also converges to a stationary point of $p(\mathbf{x}; \theta)$. See [1] for details.

It remains to find a lower bound that satisfies (7.1). For any \mathbf{y} such that $p(\mathbf{y}|\mathbf{x}; \theta) > 0$,

$$\ell(\theta) = \ln p(\mathbf{x}; \theta) = \ln \frac{p(\mathbf{x}, \mathbf{y}; \theta)}{p(\mathbf{y}|\mathbf{x}; \theta)}.$$

Then, for any distribution q for the missing data \mathbf{y} ,

$$\begin{aligned}
 \ell(\boldsymbol{\theta}) &= \sum_{\mathbf{y}} q(\mathbf{y}) \ln p(\mathbf{x}; \boldsymbol{\theta}) \\
 &= \sum_{\mathbf{y}} q(\mathbf{y}) \ln \frac{p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})} \\
 &\geq \sum_{\mathbf{y}} q(\mathbf{y}) \ln \frac{p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})} - D(q(\mathbf{y})||p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})) \\
 &= \sum_{\mathbf{y}} q(\mathbf{y}) \ln \frac{p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})} - \sum_{\mathbf{y}} q(\mathbf{y}) \ln \frac{q(\mathbf{y})}{p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})} \\
 &= \sum_{\mathbf{y}} q(\mathbf{y}) \ln p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) - \sum_{\mathbf{y}} q(\mathbf{y}) \ln q(\mathbf{y}),
 \end{aligned}$$

where for two distribution p_1 and p_2 , $D(p_1(z)||p_2(z))$ is the *relative entropy* (also called the Kullback–Leibler divergence or KL divergence) between p_1 and p_2 defined as

$$\sum_z p_1(z) \log \frac{p_1(z)}{p_2(z)}.$$

Relative entropy is a measure of dissimilarity between distributions and can be shown to be non-negative and is equal to 0 if and only if $p_1 = p_2$.

Thus for any distribution q , we have a lower bound on $\ell(\boldsymbol{\theta})$. Suppose our current guess for $\boldsymbol{\theta}$ is $\boldsymbol{\theta}^{(t)}$. We would like this lower bound to be equal to $\ell(\boldsymbol{\theta})$ at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$. For this to occur, we need

$$D(q(\mathbf{y})||p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^{(t)})) = 0 \iff q(\mathbf{y}) = p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^{(t)}),$$

resulting in

$$\ell(\boldsymbol{\theta}) \geq \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) - \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^{(t)}) = B(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}).$$

Now instead of maximizing ℓ , we can maximize B . We note however that the second term in B is not a function of $\boldsymbol{\theta}$. So we instead define the following expectation

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}^{(t)}) \ln p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}),$$

and find

$$\boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}).$$

For simplicity of notation, I often use $\boldsymbol{\theta}'$ to denote $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}''$ to denote $\boldsymbol{\theta}^{(t+1)}$. Also, let \mathbb{E}' be expected value assuming the value of $\boldsymbol{\theta}'$. We can then describe the EM algorithm as

- The E-step:

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}') = \sum_{\mathbf{y}} p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}') \ln p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \mathbb{E}[\ln p(\mathbf{x}, \mathbf{Y}; \boldsymbol{\theta})|\mathbf{x}; \boldsymbol{\theta}'] = \mathbb{E}'[\ln p(\mathbf{x}, \mathbf{Y}; \boldsymbol{\theta})|\mathbf{x}]$$

- The M-step:

$$\boldsymbol{\theta}'' = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}').$$

Update $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}''$ and repeat.

Roughly speaking, EM can be viewed as alternatively finding an estimate for the missing data through expectation by assuming a value for the parameters (the E-step) and finding a new estimate for the parameter based on the estimate of the data.

7.2 Clustering with EM

For classification the complete data is $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. When the labels y_i are missing, the problem becomes *clustering*.

We assume Gaussian class-conditionals:

$$\begin{aligned} p(Y_i = 1) &= \pi, & (\mathbf{x}_i | Y_i = 1) &\sim \mathcal{N}(\boldsymbol{\mu}_1, \mathbf{K}_1) \\ p(Y_i = 0) &= 1 - \pi, & (\mathbf{x}_i | Y_i = 0) &\sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{K}_0) \end{aligned}$$

Let $\boldsymbol{\theta} = (\pi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \mathbf{K}_0, \mathbf{K}_1)$. Ideally, we would want to maximize the likelihood for the observed data $\{(\mathbf{x}_i)\}_{i=1}^n$,

$$\ell(\boldsymbol{\theta}) = \ln p(\mathbf{x}_1^n | \boldsymbol{\theta}) = \ln \sum_{y_1^n} p(\mathbf{x}_1^n, y_1^n | \boldsymbol{\theta}).$$

But this is difficult to do because of a lack of an analytical solution due to the summation. Instead, we can use a computational method such as EM.

We will proceed as follows:

- **Set-up:** It is helpful to start with the log-likelihood of the complete data and simplify it before proceeding to the EM algorithm. We have

$$\ln p(\mathbf{x}_1^n, y_1^n; \boldsymbol{\theta}) = \sum_{i=1}^n \ln p(\mathbf{x}_i, y_i; \boldsymbol{\theta}),$$

and for each term in this sum,

$$\begin{aligned} \ln p(\mathbf{x}_i, y_i; \boldsymbol{\theta}) &= \ln \left((\pi p(\mathbf{x}_i | Y_i = 1; \boldsymbol{\theta}))^{y_i} ((1 - \pi) p(\mathbf{x}_i | Y_i = 0; \boldsymbol{\theta}))^{1 - y_i} \right) \\ &= y_i \ln(\pi p(\mathbf{x}_i | Y_i = 1; \boldsymbol{\theta})) + (1 - y_i) \ln((1 - \pi) p(\mathbf{x}_i | Y_i = 0; \boldsymbol{\theta})). \end{aligned}$$

- **The E-step:** Let $\boldsymbol{\theta}'$ be the current estimate for $\boldsymbol{\theta}$ and let \mathbb{E}' denote expected value operator with respect to the distribution $p(\mathbf{y} | \mathbf{x}; \boldsymbol{\theta}')$. We have

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}') &= \mathbb{E}'[\ln p(\mathbf{x}_1^n, Y_1^n; \boldsymbol{\theta}) | \mathbf{x}_1^n] \\ &= \mathbb{E}' \left[\sum_{i=1}^n \ln p(\mathbf{x}_i, Y_i; \boldsymbol{\theta}) | \mathbf{x}_1^n \right] \\ &= \sum_{i=1}^n \mathbb{E}'[\ln p(\mathbf{x}_i, Y_i; \boldsymbol{\theta}) | \mathbf{x}_i] \end{aligned}$$

And for each term in the sum,

$$\begin{aligned} \mathbb{E}'[\ln p(\mathbf{x}_i, Y_i; \boldsymbol{\theta}) | \mathbf{x}_i] &= \mathbb{E}'[Y_i \ln(\pi p(\mathbf{x}_i | Y_i = 1; \boldsymbol{\theta})) + (1 - Y_i) \ln((1 - \pi) p(\mathbf{x}_i | Y_i = 0; \boldsymbol{\theta})) | \mathbf{x}_i] \\ &= \mathbb{E}'[Y_i | \mathbf{x}_i] \ln(\pi p(\mathbf{x}_i | Y_i = 1; \boldsymbol{\theta})) + \mathbb{E}'[1 - Y_i | \mathbf{x}_i] \ln((1 - \pi) p(\mathbf{x}_i | Y_i = 0; \boldsymbol{\theta})) \\ &= \gamma'_i (\ln \pi + \ln p(\mathbf{x}_i | Y_i = 1; \boldsymbol{\theta})) + (1 - \gamma'_i) (\ln(1 - \pi) + \ln p(\mathbf{x}_i | Y_i = 0; \boldsymbol{\theta})), \end{aligned}$$

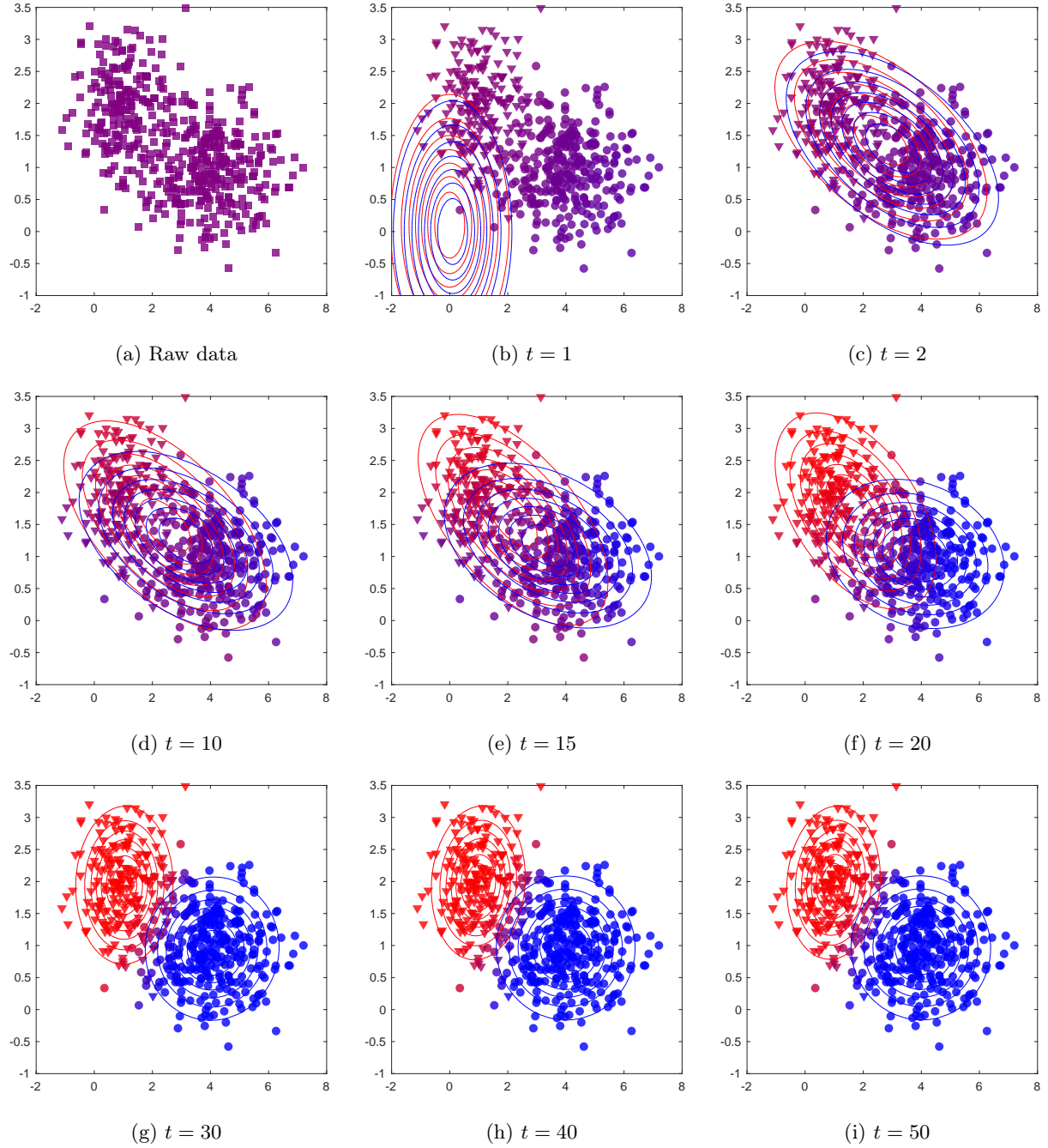


Figure 7.3: EM clustering of a mixture of two Gaussian datasets. In (a) the raw data is shown and in (b-i), steps of the EM algorithm are shown. To compare with the underlying distributions and clusters, the points from each of the Gaussian distributions are shown with triangles and circles. However, the EM algorithm does not have access to this data. The contour plots represent the current estimate for the parameters of each of the Gaussian distributions and the color of each data point represents the estimate of the EM algorithm for the probability that the point belongs to the clusters ($\gamma'_i = p(Y_i = 1 | \mathbf{x}_i; \boldsymbol{\theta}')$). A video of the clustering can be found [here](#).

where

$$\begin{aligned}
 \gamma'_i &= \mathbb{E}[Y_i | \mathbf{x}_i] \\
 &= p(Y_i = 1 | \mathbf{x}_i; \boldsymbol{\theta}') \\
 &= \frac{p(\mathbf{x}_i, Y_i = 1; \boldsymbol{\theta}')}{p(\mathbf{x}_i, Y_i = 1; \boldsymbol{\theta}') + p(\mathbf{x}_i, Y_i = 0; \boldsymbol{\theta}')} \\
 &= \frac{\pi' \mathcal{N}(\mathbf{x}_i; \mu'_1, K'_1)}{\pi' \mathcal{N}(\mathbf{x}_i; \mu'_1, K'_1) + (1 - \pi') \mathcal{N}(\mathbf{x}_i; \mu'_0, K'_0)}.
 \end{aligned}$$

Here, γ'_i has a significant meaning. It represents the probability that a given point \mathbf{x}_i belongs to class 1 given the current estimate of the parameters. Instead of computing the likelihood based on a known value for y_i , in the E-step, we compute the likelihood by partially assigning \mathbf{x}_i to class 1 and to class 0.

- **The M-step:** To find π'' :

$$\frac{\partial Q}{\partial \pi} = \sum_{i=1}^n \left(\frac{\gamma'_i}{\pi} - \frac{1 - \gamma'_i}{1 - \pi} \right) = 0 \Rightarrow \pi'' = \frac{\sum_{i=1}^n \gamma'_i}{n}.$$

To find μ''_1 :

$$\begin{aligned}
 \frac{\partial Q}{\partial \mu_1} &= \frac{\partial}{\partial \mu_1} \sum_{i=1}^n \gamma'_i \ln p(\mathbf{x}_i | Y_i = 1; \boldsymbol{\theta}) \\
 &= \frac{\partial}{\partial \mu_1} \sum_{i=1}^n \gamma'_i \left(-\frac{1}{2} (\mathbf{x}_i - \mu_1)^T K_1^{-1} (\mathbf{x}_i - \mu_1) \right) \\
 &= \sum_{i=1}^n \gamma'_i K_1^{-1} (\mathbf{x}_i - \mu_1) = 0 \Rightarrow \mu''_1 = \frac{\sum_{i=1}^n \gamma'_i \mathbf{x}_i}{\sum_{i=1}^n \gamma'_i}.
 \end{aligned}$$

To find K''_1 :

$$\begin{aligned}
 \frac{\partial Q}{\partial K_1^{-1}} &= \frac{\partial}{\partial K_1^{-1}} \sum_{i=1}^n \gamma'_i \left(\frac{1}{2} \ln |K_1^{-1}| - \frac{1}{2} (\mathbf{x}_i - \mu_1)^T K_1^{-1} (\mathbf{x}_i - \mu_1) \right) \\
 &= \frac{1}{2} K_1 \sum_{i=1}^n \gamma'_i - \frac{1}{2} \sum_{i=1}^n \gamma'_i (\mathbf{x}_i - \mu_1) (\mathbf{x}_i - \mu_1)^T = 0 \\
 &\Rightarrow K''_1 = \frac{\sum_{i=1}^n \gamma'_i (\mathbf{x}_i - \mu''_1) (\mathbf{x}_i - \mu''_1)^T}{\sum_{i=1}^n \gamma'_i}.
 \end{aligned}$$

Several steps of an EM clustering of a dataset are shown in Figure 7.3. In essence, the EM algorithm uses the current estimates of posterior class probabilities of a point as labels and updates the distributions. Having updated the distributions, it updates the posterior class probabilities and repeats.

7.3 EM with general missing data **

So far, we have considered problems in which data can be divided into an observed component x and a hidden component y , with the expectation given by

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}') = \sum_y p(y | x; \boldsymbol{\theta}') \ln p(\mathbf{x}, y; \boldsymbol{\theta})$$

But we can use EM to solve a more general class of problems, where this division may not be possible. Specifically, we assume that the complete data is given by z and the observed data is given by x , where x is a function of z . In this case, the expectation is given by

$$Q(\theta; \theta') = \sum_z p(z|x; \theta') \ln p(z; \theta)$$

Example 7.1 ([1]). Let

$$\begin{aligned} x &= s + \epsilon, \\ s &\sim \mathcal{N}(0, \theta), \quad \theta \geq 0 \\ \epsilon &\sim \mathcal{N}(0, \sigma^2) \quad \sigma^2 > 0, \end{aligned}$$

where s and ϵ are independent, σ is known, and θ is unknown. Our goal is to estimate θ . In this case, the complete data is $z = (s, \epsilon)$ and observed data is $x = s + \epsilon$.

We can solve this problem directly by noting that

$$x \sim \mathcal{N}(0, \theta + \sigma^2),$$

where we have used

$$\text{Var}(x) = \text{Cov}(s + \epsilon, s + \epsilon) = \sigma^2 + \theta.$$

The maximum likelihood estimate for the variance of x is then

$$\hat{\theta}_{ML} = \begin{cases} x^2 - \sigma^2 & \text{if } x^2 \geq \sigma^2, \\ 0 & \text{if } x^2 < \sigma^2. \end{cases}$$

With EM:

- The E-step:

$$\begin{aligned} Q(\theta; \theta') &= \mathbb{E}'[\ln p(z; \theta)|x] \\ &= \mathbb{E}'[\ln p(s; \theta) + \ln p(\epsilon; \theta)|x] \\ &\doteq \mathbb{E}'[\ln p(s; \theta)|x] \\ &\doteq \mathbb{E}'\left[-\frac{\ln \theta}{2} - \frac{s^2}{2\theta} | x\right] \\ &= -\frac{\ln \theta}{2} - \frac{\mathbb{E}'[s^2|x]}{2\theta} \end{aligned}$$

- The M-step:

$$\frac{\partial Q}{\partial \theta} = -\frac{1}{2\theta} + \frac{\mathbb{E}'[s^2|x]}{2\theta^2} = 0 \Rightarrow \theta'' = \mathbb{E}'[s^2|x].$$

This is a very intuitive result.

With some manipulation (HW), this results in

$$\theta'' = \left(\frac{\theta'}{\theta' + \sigma^2}\right)^2 x^2 + \frac{\theta' \sigma^2}{\theta' + \sigma^2}.$$

The plot for the log-likelihood and the EM estimates, starting from $\theta^{(0)} = 1$, is given in Figure 7.2, where $\sigma^2 =$ and $x = 2$ and thus $\hat{\theta}_{ML} = 3$. △

7.4 The MM Algorithm **

The idea behind the EM algorithm, i.e., finding a lower bound with certain properties, can be generalized, leading to the Minorization-Maximization (MM) algorithm. Specifically, EM provides a certain way of finding a lower bound, but if we find a lower bound by another method that still satisfies appropriate equality and inequality conditions, we can still maximize the function we are interested in. We illustrate this by applying MM to rank aggregation.

7.4.1 Rank Aggregation from Pairwise Comparisons via MM

Rank aggregation refers to combining a set of full or partial rankings of a set of alternatives in order to obtain a consensus ranking. For example, we may be interested in ranking sport teams based on match results. In this case, the input data is a set of pairwise comparisons (i.e., a partial ranking involving two items) and the desired output is a ranking of all the teams.

The data: There are n teams. We are given a dataset $\mathcal{D} = \{w_{12}, w_{13}, \dots, w_{n-1,n}\}$, where w_{ij} is the number of times team i beats team j . It will be helpful to assume $w_{ii} = 0$ rather than leaving it undefined.

The model: For two teams i and j , we assume

$$\Pr(i \text{ beats } j) = \frac{e^{s_i}}{e^{s_i} + e^{s_j}},$$

where s_i is a score reflecting the strength of team i . Denote $\mathbf{s} = (s_1, \dots, s_n)$.

This leads to the log-likelihood

$$\mathcal{L}(\mathbf{s}) = \sum_{i,j} w_{ij} (s_i - \ln(e^{s_i} + e^{s_j}))$$

As an aside, note that for a differentiable convex function $f(x)$, we have

$$\begin{aligned} f(x) &\geq f(x') + f'(x')(x - x'), & \text{for all } x', \\ f(x) &= f(x') + f'(x')(x - x'), & \text{for } x' = x. \end{aligned}$$

Since $-\ln x$ is a convex function,

$$-\ln x \geq -\ln x' - \frac{x - x'}{x'} = -\ln x' - \frac{x}{x'} + 1.$$

Hence, if we define

$$Q(\mathbf{s}, \mathbf{s}') = \sum_{i,j} w_{ij} \left(s_i - \ln(e^{s'_i} + e^{s'_j}) - \frac{e^{s_i} + e^{s_j}}{e^{s'_i} + e^{s'_j}} + 1 \right),$$

then,

$$\begin{aligned} \mathcal{L}(\mathbf{s}) &\geq Q(\mathbf{s}, \mathbf{s}'), & \text{for all } \mathbf{s}', \\ \mathcal{L}(\mathbf{s}) &= Q(\mathbf{s}, \mathbf{s}'), & \text{for } \mathbf{s} = \mathbf{s}'. \end{aligned}$$

We can simplify Q by ignoring terms that do not involve \mathbf{s} , and then separating the parameters (the latter

was not possible for \mathcal{L})

$$\begin{aligned}
Q(\mathbf{s}, \mathbf{s}') &= \sum_{i,j} w_{ij} \left(s_i - \frac{e^{s_i} + e^{s_j}}{e^{s'_i} + e^{s'_j}} \right) \\
&= \sum_i s_i \sum_j w_{ij} - \sum_i e^{s_i} \sum_j \frac{w_{ij}}{e^{s'_i} + e^{s'_j}} - \sum_i \sum_j w_{ij} \frac{e^{s_j}}{e^{s'_i} + e^{s'_j}} \\
&= \sum_i s_i \sum_j w_{ij} - \sum_i e^{s_i} \sum_j \frac{w_{ij}}{e^{s'_i} + e^{s'_j}} - \sum_i \sum_j w_{ji} \frac{e^{s_i}}{e^{s'_i} + e^{s'_j}} \\
&= \sum_i s_i \sum_j w_{ij} - \sum_i e^{s_i} \sum_j \frac{w_{ij} + w_{ji}}{e^{s'_i} + e^{s'_j}}.
\end{aligned}$$

Given the current estimate \mathbf{s}' , we can now find the next estimate \mathbf{s}'' by differentiating Q , and setting it equal to 0,

$$\begin{aligned}
\frac{\partial Q}{\partial s_i} &= \sum_j w_{ij} - e^{s_i} \sum_j \frac{w_{ij} + w_{ji}}{e^{s'_i} + e^{s'_j}} = 0 \\
s''_i &= \ln \frac{\sum_j w_{ij}}{\sum_j \frac{w_{ij} + w_{ji}}{e^{s'_i} + e^{s'_j}}}.
\end{aligned}$$

This allows us to estimate the scores s_i . When convergence is achieved or after a set number of iterations, we sort the scores and thus find a ranking of the n teams. [1]

References

- [1] Bruce Hajek. *Random Processes for Engineers*. Illinois, 2014. URL: <http://hajek.ece.illinois.edu/Papers/randomprocJuly14.pdf> (visited on 01/30/2017).

Chapter 8

Basics of Graphical Models

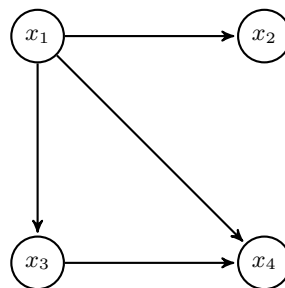
8.1 Introduction

Graphical models (GMs) are used to represent distributions on graphs. They enable us to *represent conditional independencies* and *factorization of distributions* facilitate probabilistic inference through message passing algorithms. There are different types of GMs:

- Bayesian Networks (BN, aka Directed Graphical Models): Natural for representing causal relationships
- Markov Random Fields (MRF, aka Undirected Graphical Models): Suitable for representing co-influence or non-causal relationships among a subsets of variables, e.g., friendship in social networks and pixels in an image (adjacent pixels are likely to have similar colors).
- Factor Graphs: A flexible type of GM that can represent distributions represented by BNs and MRFs.

8.2 Bayesian Networks

A Bayesian network is a **directed acyclic graph** (DAG) with some additional attributes. A DAG is a graph whose edges have direction and in which there is no cycle if one follows the edges based on their direction. In a DAG, a **parent** of a node y is a node x such that there is an edge from x to y . A **child** of y is a node z such that y is the parent of z . An **ancestor** is a parent, parent of a parent, etc., and a **descendant** is a child, child of a child, etc. A **complete DAG** is a DAG such that with an edge between each pair of vertices. An example of a DAG with four nodes is shown below.



In a Bayesian network represented by a DAG G :

- Nodes x_1, \dots, x_m represent variables or quantities (can be scalar or vector)
- Edges represent causal relationships

- The probability distribution over $x_1^m = x_1, \dots, x_m$ can be expressed as:

$$p(x_1^m) = \prod_{i=1}^m p(x_i | \text{pa}(x_i))$$

where $\text{pa}(x_i)$ are the parents of x_i in G , i.e., nodes with an edge to x_i .

We then say that the distribution p **factorizes** with respect to G . For example, for a distribution p that factorizes with respect to the graph shown above, we have

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_1, x_3). \quad (8.1)$$

What does (8.1) tell us about the distribution? Recall that based on the chain rule of probability, we always have

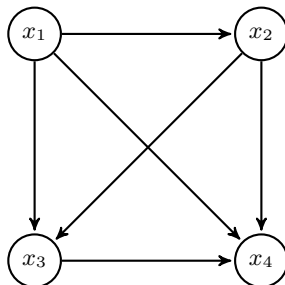
$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3). \quad (8.2)$$

It is straightforward to show that (8.1) is equivalent to

$$\begin{aligned} p(x_3|x_1, x_2) &= p(x_3|x_1), \\ p(x_4|x_1, x_2, x_3) &= p(x_4|x_1, x_3). \end{aligned} \quad (8.3)$$

These two expressions are conditional independence statements, which we can restate as $x_3 \perp\!\!\!\perp x_2 | x_1$ and $x_4 \perp\!\!\!\perp x_2 | x_1, x_3$. Thus saying that p factorizes with respect to the graph above is equivalent to assuming (8.3). This is in general true. The set of missing incoming edges for each node in the graph represents a conditional independence assumption.

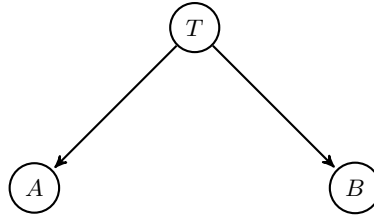
The complete graph, shown for four nodes below, represents the factorization given in (8.2), which holds for any distribution and thus the graph can represent any distribution. But such a graph is not particularly useful since the power of graphical models results from the independence assumptions that they encode.



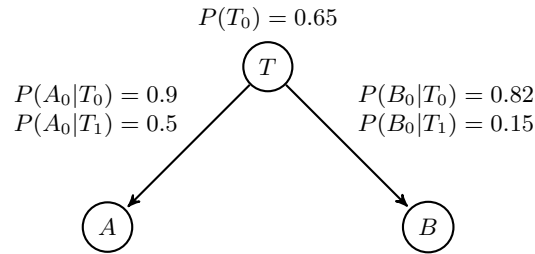
Note that the complete graph is acyclic as it imposes an ordering over the nodes (in this case, x_1, x_2, x_3, x_4). We can view any Bayesian network as being obtained from a complete DAG by removing edges. So every Bayesian network is also acyclic.

Example 8.1. Alice and Bob are employees of a business in Charlottesville, both of whom take 29S to get to work. We are interested in whether they arrive on time or late. We assume their arrival time is affected by traffic, which leads to dependence, but there aren't any other factors that can affect both of them. Let $A = 0$ and $A = 1$ denote Alice being on time and being late A_1 , respectively and similarly for Bob ($B = 0$ and B_1). Traffic is either normal ($T = 0$) or heavy ($T = 1$). We use X_0 and X_1 as shorthand for $X = 0$ and $X = 1$ for our random variables.

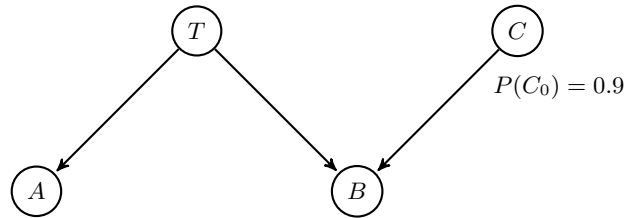
The Bayesian Network that models the probability distribution is shown below.



This graph implies that $A \perp\!\!\!\perp B|T$ and that $p(ABT) = p(T)p(A|T)p(B|T)$. We now have the **structure** of our model. But we still need the **conditional probability distributions** to complete the model. Suppose these distributions are as below:



Taking the example a step further, suppose that Bob has a son, Charlie (C_0 and C_1) who has to be dropped off at school. Charlie being late has an effect on Bob being late. We will adjust the Bayesian Network below and use the joint probability distribution in the following table.



CT	$P(B_0 CT)$	$P(B_1 CT)$
C_0T_0	0.9	0.1
C_0T_1	1/6	5/6
C_1T_0	0.1	0.9
C_1T_1	0	1

Note that this new conditional distribution does not change any previously calculated probabilities involving Traffic, Alice, and Bob, but the numbers were chosen specifically to achieve this—this is not always the case.

Based on this graph, the joint probability distribution is:

$$p(ABTC) = p(T)p(C)p(A|T)p(B|CT).$$

It is easy to show that $T \perp\!\!\!\perp C$ but as we will see below $T \not\perp\!\!\!\perp C|B$.

Bayesian networks facilitate certain kinds of reasoning. In **causal reasoning**, we draw conclusions about unobserved effects base on observed causes. For example, if we know there was heavy traffic, then it is more likely that Bob was late, $p(B_1|T_1) = 0.85 > p(B_1) = 0.41$. **Evidential reasoning** allows us to say something about the cause by observing the effects. For example,

$$p(T_1|B_1) = \frac{p(B_1|T_1)p(T_1)}{p(B_1)} = 0.7177 > p(T_1) = 0.35,$$

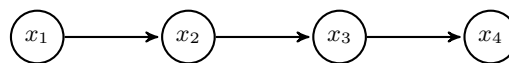
tells us that heavy traffic is more likely when Bob is late, even though we have no direct information about the traffic.

We also have $p(T_1|B_1C_1) < P(T_1|B_1)$, which makes intuitive sense. Bob being late provides evidence for traffic being heavy. But if we know Charlie is late, then we have an alternative explanation for Bob being late, lessening the need for traffic being heavy as a reason for Bob's tardiness. This type of reasoning, where given an effect, occurrence of one cause lessens the probability of another cause, is called **explaining away**. \triangle

8.2.1 Markov Model

A **Markov Model** or a **Markov chain** is a Bayesian network whose graph consists of a single path. Such a model can, for example, represent the total winning of a gambler as a function of time, where each game is independent. The main assumption is that *given the present, the future is independent of the past*: how much money you'll have after the next game is independent of past games, if your current worth is known. Another, idealized example is weather forecast: Given that we know today's weather, past weather is irrelevant for the purpose of forecasting tomorrow's weather.

A Markov chain with four nodes is given below

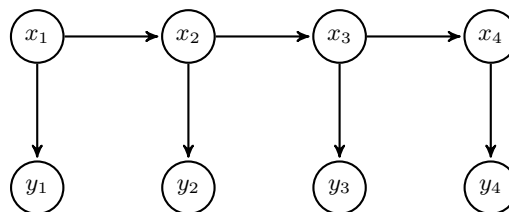


with an associated joint probability distribution that factorizes as

$$p(x_1^4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3).$$

Consider a set of n random variables each of which can take on k different values. The most general probability distribution over these variables will have $k^n - 1$ parameters (the -1 comes from the fact that we know the probabilities must sum to one). In practice, this is such a huge number even for $k = 2$ and relatively small n , e.g., $n = 100$, that we can't even store the distribution, let alone learn it from data. The Markov model, however, has $(k - 1) + (n - 1)k(k - 1)$ parameters, which is much more manageable. This is an example of graphical models making modeling more feasible.

A closely related model is the **hidden Markov model (HMM)**:



An HMM is used when the true state of the system cannot be directly observed but we can observe some function of the state. For example, x_i can represent if cancer is in remission or not and y_i can represent observations from medical tests.

Like Markov random fields, Markov and hidden Markov models are named after Russian mathematician Andrey Markov, but Markov models are Bayesian Networks and not Markov Random Fields.

8.2.2 Why graphical models?

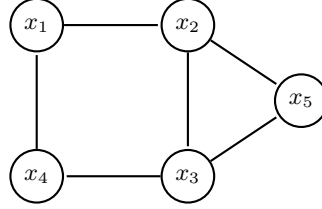
Graphical models, such as Bayesian networks are useful for several reasons.

- They provide a simple but flexible way to encode conditional independencies, enabling us to answer questions about independence based on graphs.
- GMs help constructing tractable models. As an example, see the number of parameters for a Markov chain versus an unrestricted model described above.
- Restriction to GMs has computational benefits, allowing us to draw conclusions about hidden quantities based on observations efficiently using algorithms such as belief propagation.

8.3 Markov Random Fields

Definition 8.2 (Clique and maximal clique). The following definitions from graph theory will be used in this section. In an undirected graph, a *clique* is a subset of nodes such that there is an edge between any two of them. A *maximal clique* is a clique such that there are no nodes not in the clique that connected to all the nodes already in the clique.

Suppose that we are interested in developing a political party affiliation model for a group of 5 people (or millions of people if we have social network data). Let's assume their friendships are given by the following graph



in which each node x_i represents the party of person i and an edge between x_i and x_j means that i and j are friends. How can we develop a probability distribution that can help us in this task?

We would like to encode the following observations in our distribution. We know that if two people are friends (e.g., 1 and 2), then it is more likely for them to have a common political alignment. Furthermore, for three people who are all friends (2,3,5), it is perhaps even more likely that they share the same political views. Let party affiliation be denoted by 0 or 1. We define

$$\psi_{ij}(x_i, x_j) = \begin{cases} 1, & x_i = x_j \\ 1/2, & x_i \neq x_j \end{cases} \quad (8.4)$$

and

$$\psi_{ijk}(x_i, x_j, x_k) = \begin{cases} 1, & x_i = x_j = x_k \\ 1/2, & \text{if two of the three are equal} \end{cases} \quad (8.5)$$

So agreements are assigned a higher value. Now we can define a probability distribution as

$$p(x_1, \dots, x_5) \propto \psi_{12}(x_1, x_2) \psi_{14}(x_1, x_4) \psi_{34}(x_3, x_4) \psi_{235}(x_2, x_3, x_5), \quad (8.6)$$

which assigns higher probability to configurations in which cliques of friends are in the same parties, as we wanted. For example, the probability of the left configuration is 16 times as likely to occur as the one on the right.



Note that there is no guarantee that the right side of (8.6) sums to 1 when going over all possible configurations so we need a normalization factor, which in this context is called the partition function,

$$Z = \sum_{x_1^5} \psi_{12}(x_1, x_2) \psi_{14}(x_1, x_4) \psi_{34}(x_3, x_4) \psi_{235}(x_2, x_3, x_5).$$

We can then write

$$p(x_1, \dots, x_5) = \frac{1}{Z} \psi_{12}(x_1, x_2) \psi_{14}(x_1, x_4) \psi_{34}(x_3, x_4) \psi_{235}(x_2, x_3, x_5).$$

In our example, it turns out that $Z = 8.5$, and thus $p(1, 1, 1, 1, 1) = 0.11765$ while $p(1, 0, 1, 0, 1) = 0.0073529$.

Finally, we note while we chose the potential function for each pair and triple to be the same regardless of the identity of the nodes, this is not a necessity; for example, we could have chosen different functions for ψ_{12} and ψ_{34} .

We can now consider the general case. A **Markov random field (MRF)** or an undirected graphical model consists of an undirected graph G with nodes $x_1^m = x_1, \dots, x_m$, and a probability distribution p that *factorizes with respect to G* , i.e.,

$$p(x_1^m) = \frac{1}{Z} \prod_{C \text{ is a clique in } G} \psi_C(x_C), \quad (8.7)$$

where for each clique C in G , x_C is the set of nodes in that clique, ψ_C is a *potential function*, which assigns non-negative values to all configurations of x_C , and Z is the *partition function*, which ensures that the right side is a proper distribution. Without loss of generality, we may assume the cliques are maximal by absorbing the potential functions for smaller cliques into the maximal clique. For our political party example above, for the clique with nodes x_2, x_3, x_5 , we can either have 4 potential functions over all the sub-cliques,

$$\psi'(x_2, x_3)\psi'(x_3, x_5)\psi'(x_2, x_5)\psi'(x_2, x_3, x_5)$$

or a single potential function

$$\psi(x_2, x_3, x_5).$$

Both are valid and equally powerful in terms of representation.

When designing an MRF we incorporate local information into the potential functions, but the final result is that we learn about the global view of the entire system. Also, in an MRF, the relationships between nodes are symmetric rather than causal or directed.

8.3.1 Energy-based models

When for all configurations $\mathbf{x} = x_1^m$, the probability $p(\mathbf{x})$ is positive, it is helpful to represent the distribution as

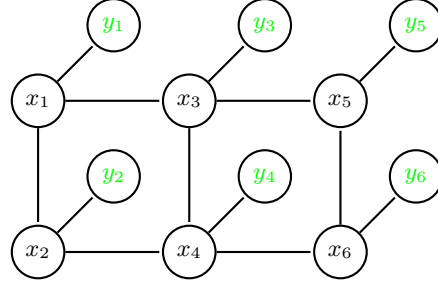
$$p(\mathbf{x}) \propto e^{-E(\mathbf{x})},$$

where $E(\cdot)$ is called the **energy function**. Such a distribution is also called a **Boltzmann distribution**. The terminology comes from statistical physics. In that context, lower energy corresponds to higher stability and thus higher probability for a system. For a graphical model, the energy function can be written as the sum of terms each of which correspond to a clique in the graph,

$$E(\mathbf{x}) = \sum_{C \text{ is a clique in } G} -\phi_C(\mathbf{x}_C) \Rightarrow p(\mathbf{x}) \propto \prod e^{\phi_C(\mathbf{x}_C)}$$

A **Boltzmann machine** is such a graphical model, typically with both nodes that can be observed and nodes that are hidden (latent).

Example 8.3 (An MRF for denoising Images). The figure below shows an MRF for a noisy black and white image. Here x_1, x_2, \dots, x_6 represent the true B/W status of the pixels and y_1, y_2, \dots, y_6 the noisy values (e.g., due to noise of a camera). We denote ‘Black’ = -1 and ‘White’ = 1.



The energy function can be written as

$$E(\mathbf{x}, \mathbf{y}) = - \sum_i^m \alpha_i x_i - \sum_{(i,j) \in \mathcal{E}(G)} \beta_{i,j} x_i x_j - \sum_i^m \zeta_i x_i y_i,$$

where $\mathcal{E}(G)$ is the set of edges between neighboring pixels and $\beta_{i,j} > 0$ and $\zeta_i > 0$. The α_i control how likely a pixel is to be white without considering other pixels. The interaction between neighboring pixels is controlled by $\beta_{i,j}$; since each is positive, it is more likely for adjacent pixels to have the same status. We assume that it is more likely for the noisy pixel to match the true pixel and so $\zeta_i > 0$ as well.

In a denoising task, we are given \mathbf{y} and our goal is to recover \mathbf{x} . A reasonable solution is

$$\arg \max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y}).$$

If we can output fractional values (if the denoised image can be grayscale), another possible solution is

$$\mathbb{E}[\mathbf{X}|\mathbf{y}].$$

△

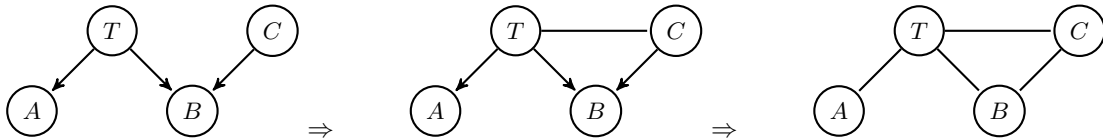
8.4 Moralization: Converting BNs to MRFs

In a BN, there is a term for each node x_i of the form

$$p(x_i | \text{pa}(x_i)).$$

To be able to have the same term in an MRF, we need to have a clique containing x_i and its parents. So to design an MRF that can represent the same distribution as the BN, we first connect all the parents of each nodes with each other and then remove all directions from the edges.

Example 8.4. As an example, consider:



△

We have

$$p(A, B, T, C) = p(T)p(C)p(A|T)p(B|T, C) \quad \Rightarrow \quad p(A, B, T, C) = \psi(T)\psi(C)\psi(A, T)\psi(B, T, C),$$

where, for example, $\psi(B, T, C) = p(B|T, C)$.

Chapter 9

Independence in Graphical Models

Graphical models encode independence assumptions. In this chapter, we will study algorithms that enable us to answer questions of the form “Is $S_1 \perp\!\!\!\perp S_2 \mid S_3$?” where S_1, S_2, S_3 are subsets of the nodes in the graph.

Recall that we construct Bayesian network by assuming certain independence assumptions that allow us to remove edges from a complete DAG. The topic of this section is study of all independence properties, which is more general than assumptions used to construct Bayesian networks.

9.1 Independence for sets of random variables

We know that for three random variables x, y, z , x is independent of z given y , denoted $x \perp\!\!\!\perp z \mid y$, if and only if

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

This extends to sets of random variables and random vectors. For example, $\{x, y\} \perp\!\!\!\perp \{z, w\} \mid \{t, u\}$, or simply $x, y \perp\!\!\!\perp z, w \mid t, u$, if and only if

$$p(x, y, z, w \mid t, u) = p(x, y \mid t, u)p(z, w \mid t, u)$$

Using this we can show that if $x, y \perp\!\!\!\perp z, w \mid t, u$, then $x \perp\!\!\!\perp z \mid t, u$ and $x, y \perp\!\!\!\perp z \mid t, u$. For example,

$$\begin{aligned} p(x, z \mid t, u) &= \sum_{y', w'} p(x, y', z, w' \mid t, u) \\ &= \sum_{y', w'} p(x, y' \mid t, u)p(z, w' \mid t, u) \\ &= \sum_{y'} p(x, y' \mid t, u) \sum_{w'} p(z, w' \mid t, u) \\ &= p(x \mid t, u)p(z \mid t, u). \end{aligned}$$

Note however that if $x \perp\!\!\!\perp z$ and $y \perp\!\!\!\perp z$ it does not follow that $x, y \perp\!\!\!\perp z$. For a counter-example, set $x \sim \text{Ber}(1/2), y \sim \text{Ber}(1/2)$ and $z = x + y$.

Exercise 9.1. Show that for three disjoint sets of random variables A, B, C , if for some functions f and g ,

$$p(A, B \mid C) \propto f(A, C)g(B, C),$$

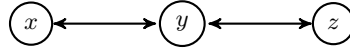
where the constant of proportionality may depend on C , then $A \perp\!\!\!\perp B \mid C$. △

9.2 Independence in Bayesian Networks

In the last chapter we saw that we can obtain a Bayesian Network by starting from a complete graph, representing the chain rule of probability, and then relying on independence assumptions, remove certain edges. Conversely, these independence assumptions are implied by the graphical model. But, in addition, to these, many other independence statements are implied by the network. In this section, we will introduce the concept of *d-separation*, using which we can find all independence statements satisfied by every distribution that factorizes with respect to the Bayesian network. We start by considering several simple networks that will help us describe d-separation.

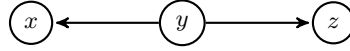
9.2.1 Simple Bayesian networks

Independence analysis in BNs relies on determining when information flows along paths in the graph. As a preliminary step, we study whether information about x affects our belief about z in the graphs of the form given below



with various directions on the edges and with y or one of its descendants being known or unknown.

Example 9.2. Given three random variables x, y , and z with relationships shown below, is $x \perp\!\!\!\perp z$?

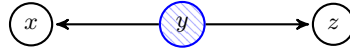


The answer: not in general. The only thing we know from the GM is $p(x, y, z) = p(y)p(x|y)p(z|y)$. We thus have

$$p(x, z) = \sum_y p(x, y, z) = \sum_y p(y)p(x|y)p(z|y)$$

and this is not necessarily equal to $p(x)p(z)$. *Exercise:* Find a counter example, i.e., find p such that it factorizes with respect to the graph but $x \not\perp\!\!\!\perp z$. \triangle

Example 9.3. Is $x \perp\!\!\!\perp z \mid y$ in the graph below?

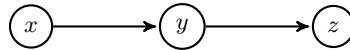


The answer: yes. We need to show $p(x, z \mid y) = p(x|y)p(z|y)$,

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

\triangle

Example 9.4. Is $x \perp\!\!\!\perp z$ in the graph below?

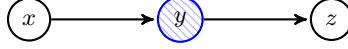


The answer: not in general since

$$p(x, z) = \sum_y p(x)p(y|x)p(z|y) = p(x) \sum_y p(y|x)p(z|y)$$

is not necessarily equal to $p(x)p(z)$. *Exercise:* Provide a counter example for $x \perp\!\!\!\perp z$. \triangle

Example 9.5. Is $x \perp\!\!\!\perp z \mid y$ in the graph below?

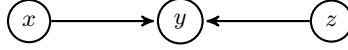


The answer: yes. We have

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

△

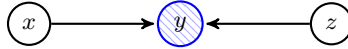
Example 9.6. Is $x \perp\!\!\!\perp z$ in the graph below?



Yes: $p(x, z) = \sum_y p(x, y, z) = \sum_y p(x)p(z)p(y \mid x, z) = p(x)p(z) \sum_y p(y \mid x, z) = p(x)p(z)$.

△

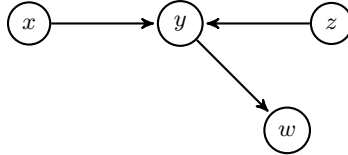
Example 9.7. Is $x \perp\!\!\!\perp z \mid y$ in the graph below?



Not in general. *Exercise:* Verify that for $x \sim \text{Ber}(\frac{1}{2})$, $z \sim \text{Ber}(\frac{1}{2})$ and $y = x + z$, $p(x, y, z)$ factorizes with respect to the graph above and $x \not\perp\!\!\!\perp z \mid y$. △

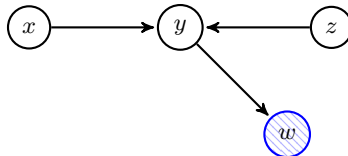
In graphs of Examples 9.6 and 9.7, if y has a descendant, that will also affect the independence relationship between x and z . These cases are considered next.

Example 9.8. Is $x \perp\!\!\!\perp z$ in the graph below?



Yes: $p(x, z) = \sum_{y,w} p(x, y, z, w) = \sum_{y,w} p(x)p(z)p(y \mid x, z)p(w|y) = p(x)p(z) \sum_{y,w} p(y \mid x, z)p(w|y) = p(x)p(z)$. △


Example 9.9. Is $x \perp\!\!\!\perp z \mid y$ in the graph below?

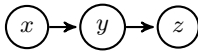
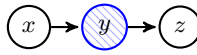
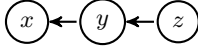
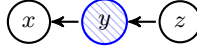
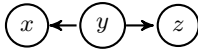
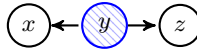
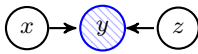
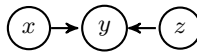
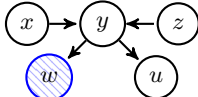
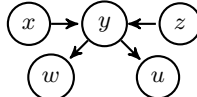


Not in general. *Exercise:* Verify that for $x \sim \text{Ber}(\frac{1}{2})$, $z \sim \text{Ber}(\frac{1}{2})$, $y = x + z$, and $w = y$, $p(x, y, z, w)$ factorizes with respect to the graph above and $x \not\perp\!\!\!\perp z \mid y$. △

9.2.2 d-separation

Based on our analysis in the previous section, we can summarize whether information flows from x to z in a graph of the form $x - y - z$ in Table 9.1. The table is organized by the direction of edges at y , with H (Head) representing an incoming edge and T (Tail) representing an outgoing edge. We can see that for the HT, TH, and TT configurations, y blocks the path from x to z if it is known (given) and for HH, it blocks the path if it is not known and neither are any of its descendants.

Table 9.1: Flow of information between x and z . Nodes with style  are known.

	Passing through	Blocked
HT/TH		
		
TT		
HH		
		

We can generalize this observation to decide, for three disjoint sets A , B , and C , of nodes, whether $A \perp\!\!\!\perp B \mid C$.

Definition 9.10. For a set C of known/observed nodes, a path is said to be blocked if it has a node v such that the nodes incident to v are:

- HT, TH, or TT and $v \in C$;
- HH, and neither v nor its descendants are in C .

Definition 9.11. For disjoint sets A , B , and C , we say that A and B are **d-separated** given C if every path between a node in A and a node in B is blocked if we assume the nodes in C are known.

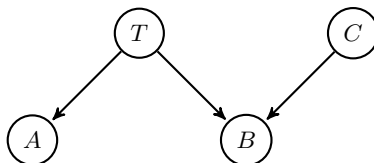
Theorem 9.12. For three disjoint sets of nodes, A , B , and C , in a graph G , such that A and B are d-separated given C , then $A \perp\!\!\!\perp B \mid C$ according to any probability p that factorize with respect to G .

Remark ** The converse of the theorem also holds in the sense that any distribution p that satisfies all independencies implied by d-separation factorizes with respect to the graph.

Remark ** Could a distribution p that factorizes with respect to G satisfy independencies that are not implied by d-separation? Indeed, yes. The distribution $p = \prod_{i=1}^n p(x_i)$ factorizes with respect to any graph G and for any non-trivial G , p satisfies independencies that are not implied by d-separation in G . However, for any independency $A \perp\!\!\!\perp B \mid C$ not implied by d-separation, there is a probability distribution factorizing with respect to G for which $A \not\perp\!\!\!\perp B \mid C$.

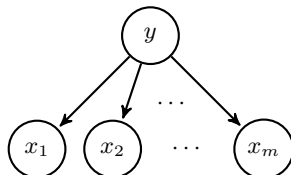
Example 9.13. In the traffic graphic from last chapter, shown below, we want to find all independencies of the form $x \perp\!\!\!\perp y$ and $x \perp\!\!\!\perp y \mid z$ for vertices x, y, z . For those that do not follow from d-separation, we write $x \not\perp\!\!\!\perp y$ and $x \not\perp\!\!\!\perp y \mid z$. We have

- No conditioning: $T \perp\!\!\!\perp C$, $T \not\perp\!\!\!\perp A$, $T \not\perp\!\!\!\perp B$, $C \perp\!\!\!\perp A$, $C \not\perp\!\!\!\perp B$, $A \not\perp\!\!\!\perp B$.
- Given T : $A \perp\!\!\!\perp B \mid T$, $A \perp\!\!\!\perp C \mid T$, $B \not\perp\!\!\!\perp C \mid T$.
- Given C : $A \not\perp\!\!\!\perp B \mid C$, $A \not\perp\!\!\!\perp T \mid C$, $B \not\perp\!\!\!\perp T \mid C$.
- Given A : $T \not\perp\!\!\!\perp B \mid A$, $T \perp\!\!\!\perp C \mid A$, $B \not\perp\!\!\!\perp C \mid A$.
- Given B : $T \not\perp\!\!\!\perp A \mid B$, $T \not\perp\!\!\!\perp C \mid B$, $A \not\perp\!\!\!\perp C \mid B$.



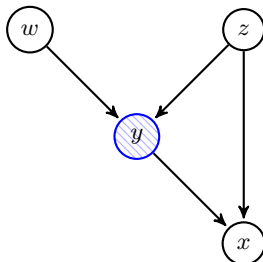
In addition, we have $A \perp\!\!\!\perp \{B, C\} \mid T$ but $\{T, A\} \not\perp\!\!\!\perp C \mid B$. \triangle

Example 9.14 (The Naive Bayes model). The graph for the naive Bayes classification model is



where y denotes the class and x_1, \dots, x_m denote the dimensions of the input vector. Given y the dimensions are independent, i.e., $x_i \perp\!\!\!\perp x_j \mid y$ for $i \neq j$. But if the class y is not known, generally speaking, $x_i \not\perp\!\!\!\perp x_j$. \triangle

Example 9.15. For four nodes w, x, y , and z , shown below, assume y is given. We can determine that none of the independencies $w \perp\!\!\!\perp z \mid y, x \perp\!\!\!\perp z \mid y, x \perp\!\!\!\perp w \mid y$ follow from d-separation. In fact, we can find a counter example, i.e., a distribution that factorizes with respect to the graph below and does not satisfy these independencies. Specifically, let $w \sim \text{Ber}(1/2), z \sim \text{Ber}(1/2), y = w + z$ and $x = y + z$. Note however that $y \perp\!\!\!\perp n \mid y$ for $n \in \{x, w, z\}$ by the definition of independence.



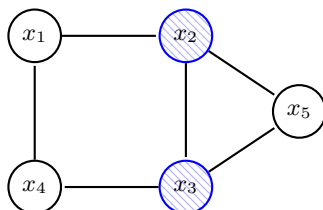
\triangle

9.2.3 Markov Blanket in Bayesian Networks

In a graphical model, the **Markov blanket** of a node y is the set of nodes S such that $y \perp\!\!\!\perp U \mid S$ for any set U . In other words, the set S isolates y from the rest of the graph. In a Bayesian network, the Markov blanket of y consists of its parents, its children, and the immediate parents of its children. The proof of this statement is left as an exercise. An example is shown in Figure 9.1.

9.3 Independence in MRFs

The set of independencies implied by an MRF are more straightforward as separation is the naive graph-theoretic separation. As an example, consider the friendship graph of the previous chapter and assume we know the political affiliation of x_2, x_3 .



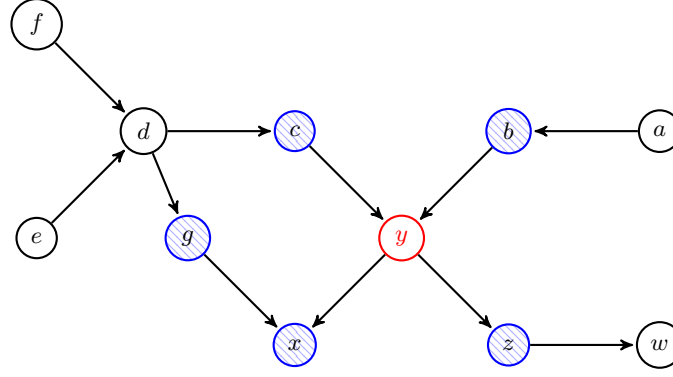


Figure 9.1: The Markov blanket of node y are the set of nodes colored red.

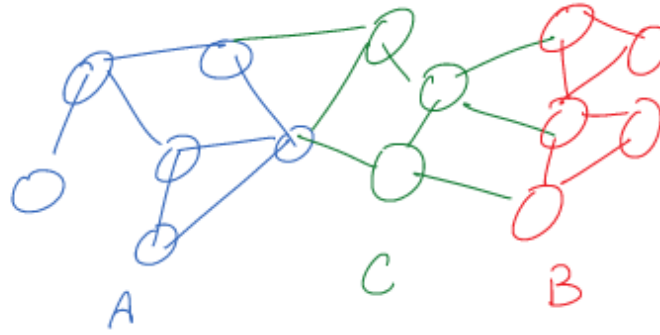


Figure 9.2: MRF Theorem

Then intuitively, we can expect that knowing x_5 does not provide any relevant information about x_1, x_4 and so we must have $x_1, x_4 \perp\!\!\!\perp x_5 \mid x_2, x_3$.

In an MRF G , suppose x_A, x_B , and x_C are disjoint subsets of vertices such that $x_A \cup x_B \cup x_C = G$, as shown in Figure 9.2. If every path from x_A to x_B travels through x_C , then $x_A \perp\!\!\!\perp x_B \mid x_C$. To see that this is the case, note that

$$\begin{aligned} p(x_A, x_B \mid x_C) &= P(x_A \mid x_C)P(x_B \mid x_C) \\ &= \frac{p(x_A, x_B, x_C)}{p(x_C)} \propto p(x_A, x_B, x_C) \\ &\propto \prod_{Q \text{ is a clique in } G} \psi_Q(x_Q) = \prod_{Q \in x_A \cup x_C} \psi_Q(x_Q) \prod_{Q \in x_B \cup x_C} \psi_Q(x_Q). \end{aligned}$$

The last equality follows from the fact that there is no clique in G that has a node in both x_A , and x_B since x_C separates x_A and x_B . The result follows from Exercise 9.1.

Examples are given in Figure 9.3.

The **Markov Blanket** of a node in an MRF is the set of neighbors as shown in Figure 9.4.

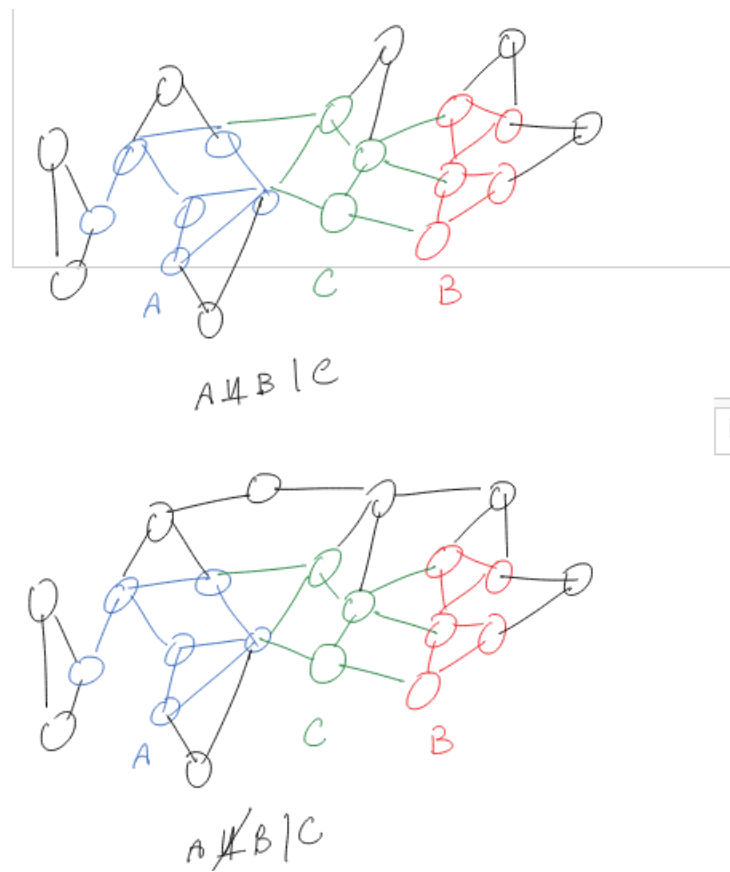


Figure 9.3: Two examples of MRFs

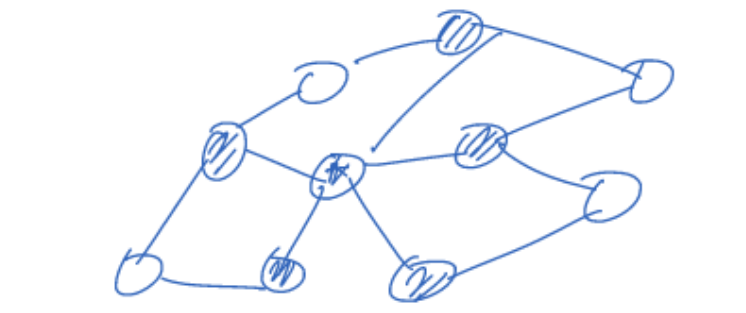


Figure 9.4: As example of a Markov Blanket

Chapter 10

Parameter Estimation in Graphical Models

A graphical model has two components: the graph structure (the nodes and their connections), and the conditional probability distributions/potential functions, which are usually expressed in parametric form. In this chapter:

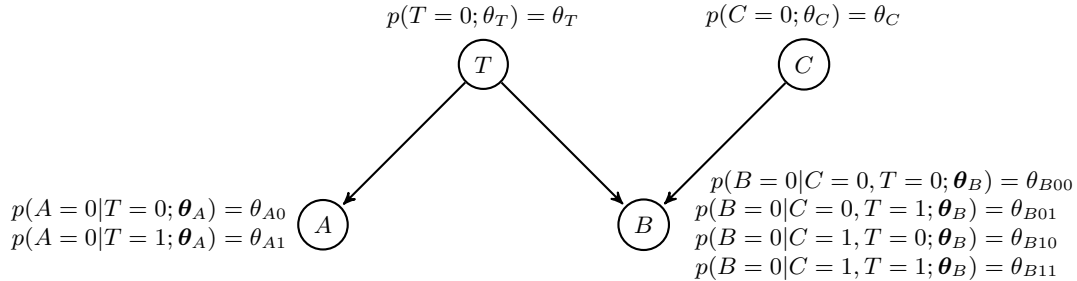
- We will consider the problem of estimating the parameters in graphical models. The problem is simpler in the case of Bayesian networks and for simplicity, that is where our attention will be focused.
- We will not consider the more challenging problem of learning the structure of a network. The best case scenario is that you have good reason to design a graph in a certain way, e.g., based on causality.

Consider a BN with a known graph with m nodes x_1, \dots, x_m in which the parameters of the conditional distribution are unknown. There are m conditional probability distributions (CPDs)¹, one for each node, and each of these has an unknown parameter vector. We denote the concatenated vector of all parameters as $\theta = (\theta_1, \dots, \theta_m)$. We collect a dataset $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of n iid samples, where $\mathbf{x}_i = (x_{i1}, \dots, x_{im})$. Our goal is to estimate θ and possibly also to predict the next outcome $\mathbf{x}_{n+1} = (x_{n+1,1}, \dots, x_{n+1,m})$.

10.1 MLE for Parameters of Bayesian Networks

We will start with maximum likelihood estimation via an example.

Example 10.1. Consider the network from previous chapters with the vector of parameters $\theta = (\theta_T, \theta_C, \theta_A, \theta_B)$.



To collect data, on n days, we record whether there is heavy traffic and whether Alice, Bob, and/or Charlie

¹Some of the nodes do not have any parents so their distribution is not conditioned on any other nodes. We view these as conditioned on the empty set and thus refer to all probability distributions in a Bayesian Network as *conditional* probability distributions.

are late, resulting in $\mathbf{x}_1, \dots, \mathbf{x}_n$, where $\mathbf{x}_i = (T_i, A_i, B_i, C_i)$. Then we maximize the likelihood

$$\arg \max_{\boldsymbol{\theta}} p(\mathcal{D}; \boldsymbol{\theta}) = \arg \max_{\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B, \theta_C} p(\mathbf{x}_1, \dots, \mathbf{x}_n; (\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B, \theta_C)) \quad (10.1)$$

△

Note that in the above example, the maximization evidently involve 6 dimensions. In real-world problems the networks have many more parameters. This would create computational difficulties since it would require maximizing a function of many variables. Fortunately, in the case of Bayesian networks, the problem decomposes to estimating the parameters for each nodes separately, as we will see.

Decomposability of likelihood. For a network with m nodes, parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m)$ and data $\mathcal{D} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, the likelihood function is

$$p(\mathcal{D}; \boldsymbol{\theta}) = \prod_{i=1}^n p(\mathbf{x}_i; \boldsymbol{\theta}),$$

where for the i th data sample, we have

$$p(\mathbf{x}_i; \boldsymbol{\theta}) = \prod_{j=1}^m p(x_{ij} | \text{pa}(x_{ij}); \boldsymbol{\theta}_j)$$

and thus the log-likelihood of the whole dataset is

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^n \ln p(\mathbf{x}_i; \boldsymbol{\theta}) = \sum_{i=1}^n \sum_{j=1}^m \ln p(x_{ij} | \text{pa}(x_{ij}); \boldsymbol{\theta}_j) = \sum_{j=1}^m \sum_{i=1}^n \ln p(x_{ij} | \text{pa}(x_{ij}); \boldsymbol{\theta}_j).$$

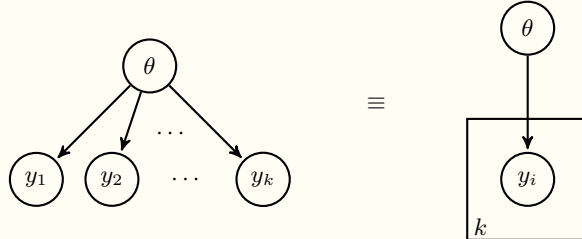
Thus for a given j , $\boldsymbol{\theta}_j$ only appears in the term $\sum_{i=1}^n \ln p(x_{ij} | \text{pa}(x_{ij}); \boldsymbol{\theta}_j)$ and no other $\boldsymbol{\theta}_k$ appears in this term. So each $\boldsymbol{\theta}_j$, and thus each conditional probability distribution, can be learned independently of the others, which *significantly* reduces the complexity.

Exercise 10.2. For the TABC network above, what would our data look like? What is the ML estimate for each parameter based on this data? △

10.2 Bayesian Parameter Estimation for Bayesian Networks

An alternative approach is using Bayesian inference. Since in the Bayesian view, parameters are considered random, we can augment the Bayesian network by adding the parameters as nodes.

Side note 1: the plate notation. Before proceeding, we introduce the plate notation which is helpful for simplifying repeated elements in graphical models, especially a set of iid nodes. Specifically, instead of repeating a node k times, we enclose one instance and indicate how many times that segment of the graph is repeated. Both of the following graphs represent the factorization $p(y_1, \dots, y_k, \theta) = \prod_{i=1}^k p(y_i | \theta)$.



Side note 2: Conditioning for sets of nodes. Consider a BN with nodes y_1, \dots, y_m . Assume that the set of nodes can be partitioned into two sets $S_1 = \{y_1, \dots, y_r\}$ and $S_2 = \{y_{r+1}, \dots, y_m\}$ such that there are no edges from S_2 to S_1 . Then the following hold

$$p(S_1) = p(y_1, \dots, y_r) = \prod_{i=1}^r p(y_i | \text{pa}(y_i)), \quad (10.2)$$

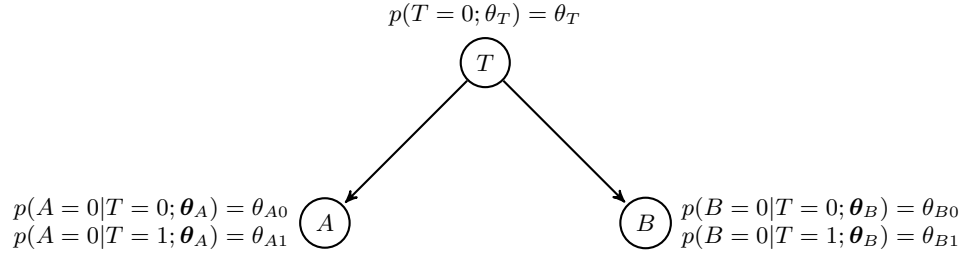
$$p(S_2 | S_1) = p(y_{r+1}, \dots, y_m | y_1, \dots, y_r) = \prod_{i=r+1}^m p(y_i | \text{pa}(y_i)). \quad (10.3)$$

However, in general,

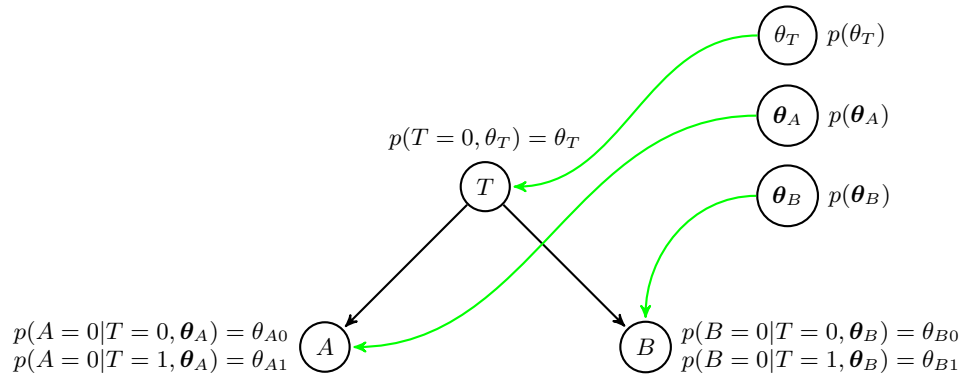
$$p(S_2) = p(y_{r+1}, \dots, y_m) \neq \prod_{i=r+1}^m p(y_i | \text{pa}(y_i)), \quad (10.4)$$

$$p(S_1 | S_2) = p(y_1, \dots, y_r | y_{r+1}, \dots, y_m) \neq \prod_{i=1}^r p(y_i | \text{pa}(y_i)) \quad (10.5)$$

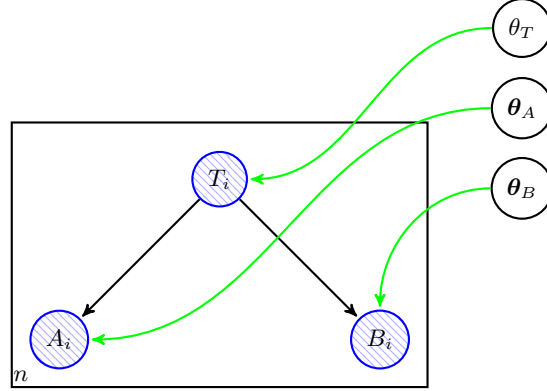
Bayesian networks with explicit representation of the parameters and data. Let us consider a simpler version of the network given in Example 10.1, with unknown parameter vector $\theta = (\theta_T, \theta_A, \theta_B)$:



At this point, we are still viewing the parameters as unknown constants. Now to formulate the Bayesian estimation of these parameters, we need to view them as random and add nodes for them to the graph, estimating the parameters as the network:



Incorporating the n samples $\mathcal{D} = \{(T_1, A_1, B_1), \dots, (T_n, A_n, B_n)\}$, we represent the problem of estimating the parameters as the network, where the CPDs are omitted for clarity:



Posterior distributions of the parameters. We are interested in finding $p(\boldsymbol{\theta}|\mathcal{D}) = p(\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B|\mathcal{D})$. Note that

$$p(\boldsymbol{\theta}|\mathcal{D}) = p(\theta_T|\mathcal{D})p(\boldsymbol{\theta}_A, \boldsymbol{\theta}_B|\mathcal{D}) \quad (10.6)$$

$$= p(\theta_T|\mathcal{D})p(\boldsymbol{\theta}_A|\mathcal{D})p(\boldsymbol{\theta}_B|\mathcal{D}) \quad (10.7)$$

$$= p(\theta_T|T_1^n)p(\boldsymbol{\theta}_A|T_1^n, A_1^n)p(\boldsymbol{\theta}_B|T_1^n, B_1^n), \quad (10.8)$$

where the first equality follows from the fact that given $T_1^i \subset \mathcal{D}$, θ_T is independent of all other nodes, including $\boldsymbol{\theta}_A, \boldsymbol{\theta}_B$. In other words, T_1^i is the Markov blanket of θ_T . The second equality also holds because \mathcal{D} contains the Markov blanket for $\boldsymbol{\theta}_A$ and $\boldsymbol{\theta}_B$. Similarly, the last equality follows from a Markov blanket-type argument.

In other words, to estimate each parameter, we need to only consider the part of the data that is in the parameter's Markov blanket. This also makes intuitive sense: For example, to estimate the probability of Alice being late given the state of traffic, only the part of data that deals with Alice's arrival time and traffic is relevant. The fact that the posterior for each parameter can be determined separately significantly reduces the computational complexity.

Example 10.3. Let us find $p(\boldsymbol{\theta}_A|\mathcal{D})$, assuming that the prior satisfies $p(\boldsymbol{\theta}_A) = p(\theta_{A0})p(\theta_{A1})$,

$$\begin{aligned} p(\boldsymbol{\theta}_A|\mathcal{D}) &= p(\boldsymbol{\theta}_A|T_1^n, A_1^n) \propto p(\boldsymbol{\theta}_A)p(T_1^n, A_1^n|\boldsymbol{\theta}_A) \stackrel{(*)}{\propto} p(\boldsymbol{\theta}_A) \prod_{i=1}^n p(A_i|T_i, \boldsymbol{\theta}_A) \\ &= \left(p(\theta_{A0}) \prod_{i:T_i=0} p(A_i|T_i=0, \theta_{A0}) \right) \left(p(\theta_{A1}) \prod_{i:T_i=1} p(A_i|T_i=1, \theta_{A1}) \right). \end{aligned}$$

(Why does the relation shown as $\stackrel{(*)}{\propto}$ hold?) Since the terms depending on θ_{A0} and θ_{A1} separate, they are conditionally independent and we can estimate them separately: Hence, the estimators of θ_{A0}^0 and θ_{A1}^1 are

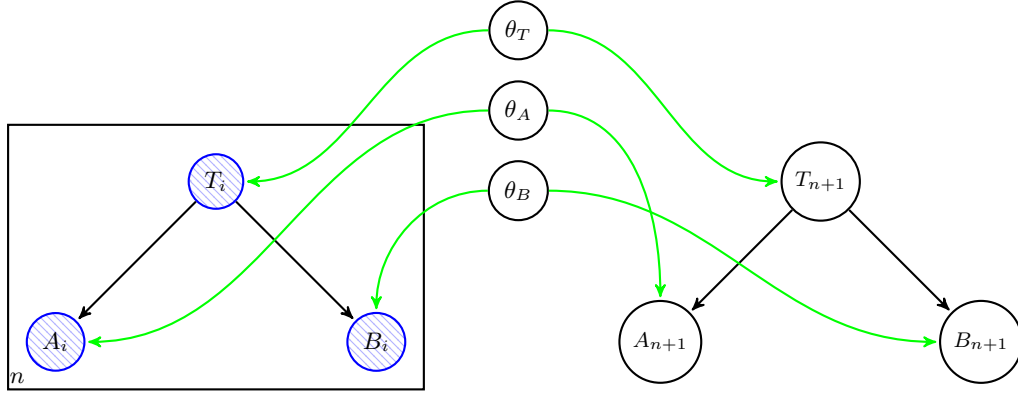
$$\begin{aligned} p(\theta_{A0}|\mathcal{D}) &\propto p(\theta_{A0}) \prod_{i:T_i=0} p(A_i|T_i=0, \theta_{A0}), \\ p(\theta_{A1}|\mathcal{D}) &\propto p(\theta_{A1}) \prod_{i:T_i=1} p(A_i|T_i=1, \theta_{A1}). \end{aligned}$$

Suppose $p(\theta_{A0}^0) \sim \text{Beta}(1, 1)$ and out of 100 days with no traffic, in 40 days Alice was on time. Hence,

$$\theta_{A0}|\mathcal{D} \sim \text{Beta}(41, 61).$$

△

Predicting future outcomes. We can also add future outcomes to predict their value to the network:



Let $\mathbf{x}_{n+1} = (T_{n+1}, A_{n+1}, B_{n+1})$. We have

$$p(\mathbf{x}_{n+1}, \boldsymbol{\theta} | \mathcal{D}) = p(\boldsymbol{\theta} | \mathcal{D}) p(\mathbf{x}_{n+1} | \mathcal{D}, \boldsymbol{\theta}) = p(\boldsymbol{\theta} | \mathcal{D}) p(\mathbf{x}_{n+1} | \boldsymbol{\theta}). \quad (10.9)$$

We have already seen how to find $p(\boldsymbol{\theta} | \mathcal{D})$. We can decompose $p(\mathbf{x}_{n+1} | \boldsymbol{\theta})$ as given by the Bayesian network:

$$p(\mathbf{x}_{n+1} | \boldsymbol{\theta}) = p(T_{n+1} | \theta_T) p(A_{n+1} | \theta_A, T_{n+1}) p(B_{n+1} | \theta_B, T_{n+1}). \quad (10.10)$$

Note that the terms on the right are known probability distributions.

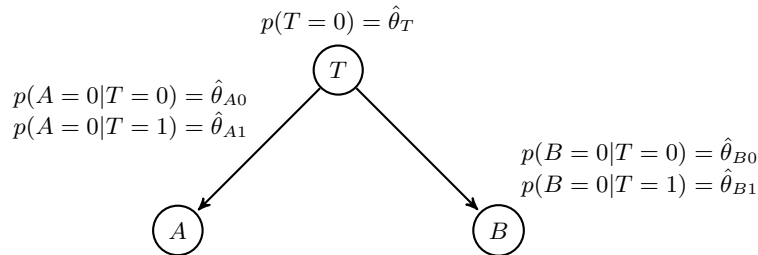
Finally, if we are interested in a specific future outcome, e.g., $p(A_{n+1} | \mathcal{D})$, we can find it through an appropriate integration/summation of $p(\mathbf{x}_{n+1}, \boldsymbol{\theta} | \mathcal{D})$.

Example 10.4. The posterior probability of the next sample (A_{n+1}, B_{n+1}) is

$$p(A_{n+1}, B_{n+1} | \mathcal{D}) = \int_{\boldsymbol{\theta}} \sum_{T_{n+1}} p(A_{n+1}, B_{n+1}, T_{n+1}, \boldsymbol{\theta} | \mathcal{D}) d\boldsymbol{\theta},$$

where we can find the integrand/summand as described in (10.9). In general, such integrals may be difficult to find analytically. In practice, we rely on computational methods such as Markov Chain Monte Carlo (MCMC).

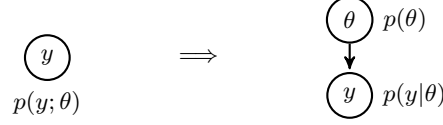
Alternatively, to predict future values, we can use a Bayesian point estimate for $\boldsymbol{\theta}$, and then assume that they are known as shown below.



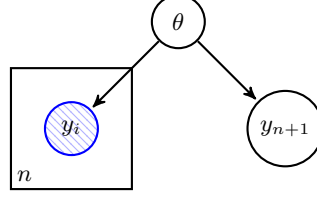
△

We can use graphical models to represent some of the estimation/learning problems we have already discussed in previous chapters.

Example 10.5. Let y have a distribution p with an unknown parameter θ . Below, on the left a graphical model for y is shown without explicit representation of θ and on the right, θ is added as a node:



We have n independent samples, $\mathcal{D} = \{y_1, y_2, \dots, y_n\}$, from the distribution and our goal is to predict the next outcome y_{n+1} . We can augment the graph to represent the problem as follows:



with a joint distribution that can be written as $p(\theta, y_1^n, y_{n+1}) = p(\theta)p(y_1^n|\theta)p(y_{n+1}|\theta)$.

We can perform similar analysis as we have done in the Bayesian Estimation chapter, using d-separation to verify independence relations. We have

$$p(y_{n+1}|y_1^n) = \int p(y_{n+1}, \theta|y_1^n) d\theta = \int p(\theta|y_1^n) p(y_{n+1}|\theta, y_1^n) d\theta = \int p(\theta|y_1^n) p(y_{n+1}|\theta) d\theta$$

where in the last step we have used $y_{n+1} \perp\!\!\!\perp y_1^n \mid \theta$, which follows from d-separation. Furthermore,

$$\mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta, y_1^n]|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n]. \quad (10.11)$$

Roughly speaking, to learn about y_{n+1} given y_1^n , we must first learn about θ since this is the node that connects y_1^n and y_{n+1} .

For example, assume $p(\theta) \propto 1$, $y_i|\theta \sim \text{Ber}(\theta)$, and that out of the n samples y_i , there s 1s and f 0s. Then

$$p(y_{n+1} = 1|y_1^n) = \mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n] = \mathbb{E}[\theta|y_1^n] = \frac{s+1}{s+f+2}.$$

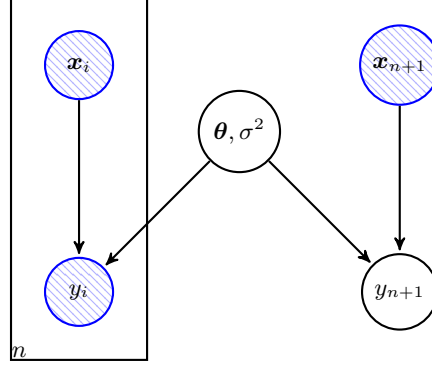
△

In more general cases, some of the “future outcomes” may also be known. But the same principles discussed above, still apply.

Example 10.6 (Bayesian Linear Regression). Consider the regression problem

$$p(y_i|\mathbf{x}_i, \boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(\boldsymbol{\theta}^T \mathbf{x}_i, \sigma^2),$$

with data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$. We are interested in determining $p(y_{n+1}|\mathbf{x}_{n+1}, \mathcal{D})$. The problem can be represented as the graph



We note that

$$p(y_{n+1}, \theta, \sigma^2 | \mathbf{x}_{n+1}, \mathcal{D}) = p(\theta, \sigma^2 | \mathbf{x}_{n+1}, \mathcal{D}) p(y_{n+1} | \theta, \sigma^2, \mathbf{x}_{n+1}, \mathcal{D}) \quad (10.12)$$

$$= p(\theta, \sigma^2 | \mathcal{D}) p(y_{n+1} | \mathbf{x}_{n+1}, \theta, \sigma^2), \quad (10.13)$$

where we have used the following facts: $\theta, \sigma^2 \perp \mathbf{x}_{n+1} | \mathcal{D}$ and $y_{n+1} \perp \mathcal{D} | \mathbf{x}_{n+1}, \theta, \sigma^2$. We know how to find $p(\theta, \sigma^2 | \mathcal{D})$ and $p(y_{n+1} | \mathbf{x}_{n+1}, \theta, \sigma^2)$ is given by assumption. While we can find $p(y_{n+1} | \mathbf{x}_{n+1})$ through integration analytically, as discussed in the linear regression chapter, we normally produce samples for θ, σ^2 and then proceed to produce samples for $p(y_{n+1} | \mathbf{x}_{n+1}, \theta, \sigma^2)$.

△

10.3 Parameter Estimation in MRFs

Recall that for an MRF G , the probability distribution is given as

$$p(\mathbf{x}; \theta) = \prod_{c \text{ is a clique in } G} \psi_{\theta}(\mathbf{x}_c) / Z(\theta),$$

where $Z(\theta) = \sum_{\mathbf{x}} \prod_c \psi_{\theta}(\mathbf{x}_c)$ is the partition function. Let us consider the frequentist estimation of θ , e.g., maximum likelihood. Unfortunately, the log-likelihood function does not decompose into terms each depending on one component of θ . This is due to the presence of the partition function, which generally depends on all the components of θ , leading to a high-dimensional problem. Furthermore, computing the partition function is a computationally difficult task since it involves computing a sum with possibly exponentially many terms. We will discuss computational approaches to this problem later in the course.

Helpful references: [2, 3, 1]

References

- [1] Christopher M. Bishop. *Pattern Recognition And Machine Learning*. New York: Springer, 2006. URL: <http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%5C%20202006.pdf> (visited on 02/14/2017).
- [2] Michael I Jordan. *An Introduction to Probabilistic Graphical Models (Preprints and Course Notes)*. University of California, Berkeley, 2003.
- [3] David JC MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge university press, 2003.

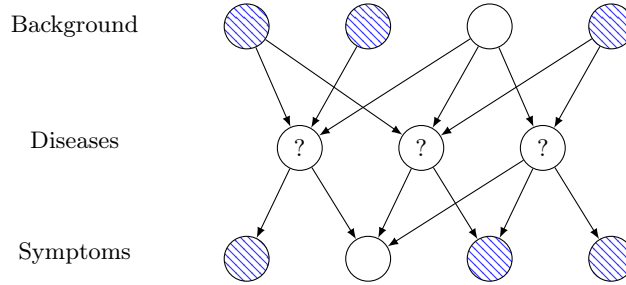
Chapter 11

Inference in Graphical Models

11.1 Introduction

Inference refers to drawing conclusions about unknown quantities based on observations and a model. In the context of graphical models assume, our goal is to learn about a set of query nodes given observed nodes.

For example, consider the following graph with nodes for background information about a patient (e.g., diet, exercise, genetics, etc.), diseases (diabetes, hypertension, etc.), and symptoms/test results (blood pressure, etc). Our goal is assign probabilities to disease based on our observations. Alternatively, we may be interested in identifying the disease that is most likely.



In such a graph, we deal with three types of nodes, evidence (observed) nodes, x_E , query nodes, x_Q , and other nodes, x_O .

Without having made any observations, we can find the probability of the query nodes through *marginalization*:

$$p(x_Q) = \sum_{x_O, x_E} p(x_Q, x_O, x_E),$$

and with observations, through *conditioning*:

$$p(x_Q|x_E) \propto \sum_{x_O} p(x_Q, x_O, x_E).$$

Since we can view the latter case as doing summation over x_E that only consists of a single set of values, from this point on, we will only consider marginalization. Note that we need to compute $\sum_{x_O} p(x_Q, x_O, x_E)$ for all values of x_Q to be able to find $p(x_Q|x_E)$.

11.2 The Elimination Algorithm

Suppose that in a Markov chain $x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5$, we need to find $p(x_4)$,

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

Assume each node can take k different values. In the naive approach, we need to compute and add $O(k^4)$ terms, and we need to do so for each possible value of x_4 . So finding the distribution of x_4 has complexity $O(k^5)$.

Alternatively, we could eliminate each variable, which can be done in different orders. The equalities below represent computation performed by an algorithm:

$$\begin{aligned} p(x_4) &= \sum_{x_1} \sum_{x_2} \sum_{x_3} \sum_{x_5} p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3)p(x_5|x_4) \\ &= \sum_{x_1} \sum_{x_2} \sum_{x_3} p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3) \sum_{x_5} p(x_5|x_4) \\ &= \sum_{x_1} \sum_{x_2} \sum_{x_3} p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3) \\ &= \sum_{x_1} \sum_{x_2} p(x_1)p(x_2|x_1) \sum_{x_3} p(x_3|x_2)p(x_4|x_3) \\ &= \sum_{x_1} \sum_{x_2} p(x_1)p(x_2|x_1) M_1(x_2, x_4) \\ &= \sum_{x_1} p(x_1) \sum_{x_2} p(x_2|x_1) M_1(x_2, x_4) \\ &= \sum_{x_1} p(x_1) M_2(x_1, x_4) \\ &= p(x_4) \end{aligned}$$

The function $M_1(x_2, x_4)$ is *defined* as the result of the sum $\sum_{x_3} p(x_3|x_2)p(x_4|x_3)$, and a similar statement holds for M_2 . We can think of $M_1(x_2, x_4)$ as a table stored in computer memory after it is computed. Computing $M_1(x_2, x_4)$ needs to be done for k different values of x_2 and each of these requires computing and adding k terms, one for each possible value of x_3 . The computational complexity for a specific value of x_4 is $O(k^2)$, i.e., we need of the order of k^2 computations. The total computational complexity of finding the distribution $p(x_4)$ is $O(k^3)$ since we need to repeat all operations for the k different values that x_4 can take. More generally, for a Markov chain with n nodes, the complexity is $O(nk^3)$ for computing the distribution $p(x_n)$. But with the naive approach it is $O(k^n)$.

Note that in Bayesian networks, we can ignore downstream nodes since their probability marginalizes to 1 (but not in MRFs).

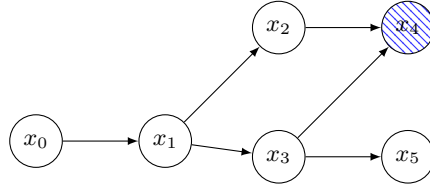
We could also choose the following ordering, which would lead to a different complexity:

$$\begin{aligned}
 p(x_4) &= \sum_{x_1} \sum_{x_3} \sum_{x_2} p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3) \\
 &= \sum_{x_1} \sum_{x_3} p(x_1)p(x_4|x_3) \sum_{x_2} p(x_2|x_1)p(x_3|x_2) \\
 &= \sum_{x_1} \sum_{x_3} p(x_1)p(x_4|x_3) T_1(x_1, x_3) \\
 &= \sum_{x_1} p(x_1) \sum_{x_3} p(x_4|x_3) T_1(x_1, x_3) \\
 &= \sum_{x_1} p(x_1) T_2(x_1, x_4) \\
 &= p(x_4).
 \end{aligned}$$

Here, computing $T_1(x_1, x_3)$ has complexity $O(k^3)$, which is also the complexity for one value of x_4 . For the distribution, the complexity is $O(k^4)$ for this ordering.

The problem of finding the best ordering for elimination is NP-hard (i.e., computationally difficult) for general graphs.

Now let us find $p(x_0|x_4)$ in the following network:



We have

$$\begin{aligned}
 p(x_0|x_4) &\propto \sum_{x_1, x_2, x_3} p(x_0)p(x_1|x_0)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3) \\
 &= \sum_{x_1, x_3} p(x_0)p(x_1|x_0)p(x_3|x_1) \sum_{x_2} p(x_2|x_1)p(x_4|x_2, x_3) \\
 &= \sum_{x_1, x_3} p(x_0)p(x_1|x_0)p(x_3|x_1) K_1(x_1, x_3, x_4) \\
 &= \sum_{x_1} p(x_0)p(x_1|x_0) \sum_{x_3} p(x_3|x_1) K_1(x_1, x_3, x_4) \\
 &= \sum_{x_1} p(x_0)p(x_1|x_0) K_2(x_1, x_4) \\
 &= p(x_0) \sum_{x_1} p(x_1|x_0) K_2(x_1, x_4) \\
 &= p(x_0) K_3(x_0, x_4).
 \end{aligned}$$

The complexity is dominated by $K_1(x_1, x_3, x_4)$, which is $O(k^3)$, assuming each node can take on k values, leading to a total complexity of $O(k^4)$ for the conditional distribution of x_0 .

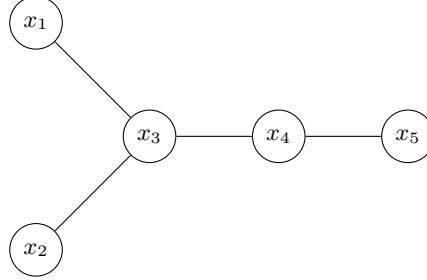
11.3 The Sum-Product Algorithm

The *sum-product algorithm*, also known as *belief propagation* and *sum-product message passing*, provides a simple way of doing exact inference on trees. It is also commonly used on graphs that are not trees since it often provides good approximations.

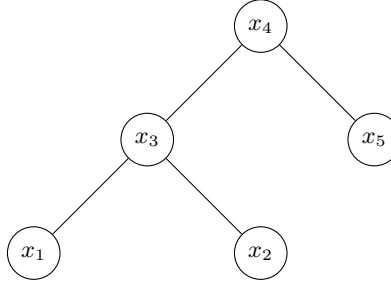
We need to clarify what we mean by trees. For Markov random fields, the algorithm works on trees, in the graph-theoretic sense. But for Bayesian networks, it works for graphs whose equivalent MRF (the moralized graph) is a tree. In particular, no node can have more than one parent. Given the straightforward equivalence between these two categories, we only consider Markov random field trees.

Consider the the following MRF, where we are interested in $p(x_4)$, with

$$p(x_4) \propto \psi(x_1, x_3) \psi(x_2, x_3) \psi(x_3, x_4) \psi(x_4, x_5)$$



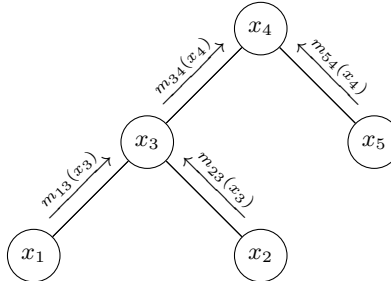
Let's look at this graph as a rooted tree,



and perform elimination starting from the leaves to the roots:

$$\begin{aligned}
 p(x_4) &\propto \sum_{x_1, x_2, x_3, x_5} \psi(x_1, x_3) \psi(x_2, x_3) \psi(x_3, x_4) \psi(x_4, x_5) \\
 &= \sum_{x_3} \psi(x_3, x_4) \psi(x_4) \left(\sum_{x_1} \psi(x_1, x_3) \right) \left(\sum_{x_2} \psi(x_2, x_3) \psi(x_3) \right) \left(\sum_{x_5} \psi(x_4, x_5) \right) \\
 &= \sum_{x_3} \psi(x_3, x_4) \psi(x_4) m_{13}(x_3) m_{23}(x_3) m_{54}(x_4) \\
 &= \psi(x_4) m_{54}(x_4) \sum_{x_3} \psi(x_3, x_4) m_{13}(x_3) m_{23}(x_3) \\
 &= \psi(x_4) m_{54}(x_4) m_{34}(x_4)
 \end{aligned} \tag{11.1}$$

We can view this computation as being done on each node and then messages being passed to neighbors:



where

$$\begin{aligned} m_{13}(x_3) &= \sum_{x_1} \psi(x_1, x_3), \\ m_{23}(x_3) &= \sum_{x_2} \psi(x_2, x_3) \psi(x_2), \\ m_{54}(x_4) &= \sum_{x_5} \psi(x_4, x_5), \\ m_{34}(x_4) &= \sum_{x_3} \psi(x_3, x_4) m_{13}(x_3) m_{23}(x_3). \end{aligned}$$

and then at the root, we can find $p(x_4)$ as

$$p(x_4) \propto \psi(x_4) m_{54}(x_4) m_{34}(x_4).$$

Recall that this also works for conditioning. Specifically, if we are interested in the conditional probability $p(x_4 | x_3 = a)$, we would compute

$$\begin{aligned} m_{34}(x_4) &= \psi(x_3 = a, x_4) m_{13}(a) m_{23}(a), \\ p(x_4) &\propto \psi(x_4) m_{54}(x_4) m_{34}(x_4). \end{aligned}$$

We can state the sum-product algorithm for a rooted tree as follows. At each node x_j with parent x_k ,

- Product step: After receiving messages $m_{ij}(x_j)$ from all children x_i of x_j , compute the product of all messages and any potential functions containing x_j ,

$$\psi(x_j) \psi(x_j, x_k) \prod_i m_{ij}(x_j).$$

Note that not all potentials are always present. Do this for each possible pair of values for (x_j, x_k) .

- Sum step: Sum over all possible values of x_j to produce the message

$$m_{jk}(x_k) = \sum_{x_j} \psi(x_j) \psi(x_j, x_k) \prod_i m_{ij}(x_j), \quad (11.2)$$

and send to x_k . Do this for each possible value for x_k .

A critical point in the correctness of the sum-product algorithm is that the messages received by each node are functions of the value of that node. This is easy to see by induction. After the product step, we get a function of both the current node x_j and its parent x_k . The sum eliminates the current node and so the parent node x_k receives a message that is only a function of x_k .

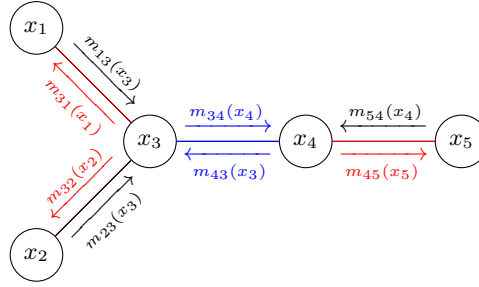
Complexity of computing each message: Suppose each node can take on K different values, namely $\{1, 2, \dots, K\}$. So the sum in (11.2) contains K terms. Furthermore, $m_{jk}(x_k)$ needs to be computed for $x_k = 1, 2, \dots, K$. We can imagine a vector

$$\mathbf{m}_{jk} = \begin{pmatrix} m_{jk}(1) \\ m_{jk}(2) \\ \vdots \\ m_{jk}(K) \end{pmatrix}$$

being sent to the node x_k . So the complexity at each node is $O(K^2)$ and for n nodes the complexity is $O(nK^2)$.

Computing marginals at all nodes. We can easily extend this algorithm to computing all marginals rather than a single node. We note that the messages sent by the nodes do not depend on the location of the

root. Each node sends a message when it receives messages from all but one of its neighbors. We can extend this by not sending a message only once, but sending a message to each neighbor based on the messages received by the other neighbors:



Here the order of messages is color-coded: 1, 2, 3. We can now find the marginal at each node. For example,

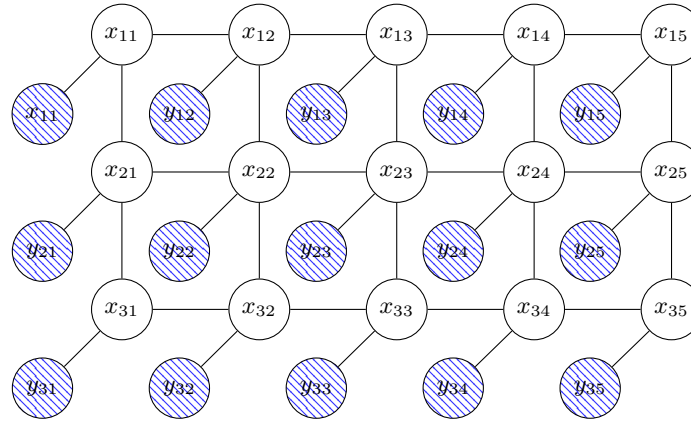
$$p(x_2) \propto m_{32}(x_2)\psi(x_2),$$

$$p(x_3) \propto m_{13}(x_3)m_{23}(x_3)m_{43}(x_3).$$

Example 11.1. An example for the sum-product algorithm is given at the end of the document. △

11.4 The Max-Product Algorithm

The max-product algorithm is used to identify the configuration that has the maximum probability. Examples include part-of-speech tagging, voice recognition, decoding (communication), and image denoising. The last example is shown below:



where x_{ij} are true image pixels and y_{ij} are observed pixels, e.g., from a camera. Our goal is to find

$$\arg \max_{\mathbf{x}} p(\mathbf{x}, \mathbf{y}).$$

Note that the local maximum-probability configuration does not necessarily agree with the global maximum-probability configuration. As an example, consider

$p(x_1, x_2)$	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$.3	.4
$x_2 = 1$.3	0

We have

$$\begin{aligned}\arg \max_{x_1, x_2} p(x_1, x_2) &= (\mathbf{1}, 0) \\ \arg \max_{x_1} p(x_1) &= \arg \max_{x_1} (p(x_1, x_2 = 0) + p(x_1, x_2 = 1)) = \mathbf{0}.\end{aligned}$$

To see the max-product algorithm, suppose we want to find

$$\arg \max_{x_1^5} p(x_1^5)$$

for the tree given in the previous section. To solve this problem, let us start with solving

$$\max_{x_1^5} p(x_1^5)$$

We proceed similar to (11.1). For clarity, we make the partition function Z explicit, but we don't actually need to find it. We replace each summation in the previous derivation with max and write:

$$\begin{aligned}\max p(x_1^5) &= \max_{x_1, x_2, x_3, x_4, x_5} Z \psi(x_1, x_3) \psi(x_2, x_3) \psi(x_2) \psi(x_3, x_4) \psi(x_4) \psi(x_4, x_5) \\ &= Z \max_{x_4} \max_{x_3} \psi(x_3, x_4) \psi(x_4) \left(\max_{x_1} \psi(x_1, x_3) \right) \left(\max_{x_2} \psi(x_2, x_3) \psi(x_2) \right) \left(\max_{x_5} \psi(x_4, x_5) \right) \\ &= Z \max_{x_4} \max_{x_3} \psi(x_3, x_4) \psi(x_4) m_{13}(x_3) m_{23}(x_3) m_{54}(x_4) \\ &= Z \max_{x_4} \psi(x_4) m_{54}(x_4) \max_{x_3} \psi(x_3, x_4) m_{13}(x_3) m_{23}(x_3) \\ &= Z \max_{x_4} \psi(x_4) m_{54}(x_4) m_{34}(x_4)\end{aligned}$$

This is the same as the sum-product algorithm, except that we take the max of product terms. We can again view this as message-passing, but using max instead of sum, with the following messages:

$$\begin{aligned}m_{13}(x_3) &= \max_{x_1} \psi(x_1, x_3), \\ m_{23}(x_4) &= \max_{x_2} \psi(x_2, x_3) \psi(x_2), \\ m_{54}(x_4) &= \max_{x_5} \psi(x_4, x_5), \\ m_{34}(x_4) &= \max_{x_3} \psi(x_3, x_4) m_{13}(x_3) m_{23}(x_3).\end{aligned}$$

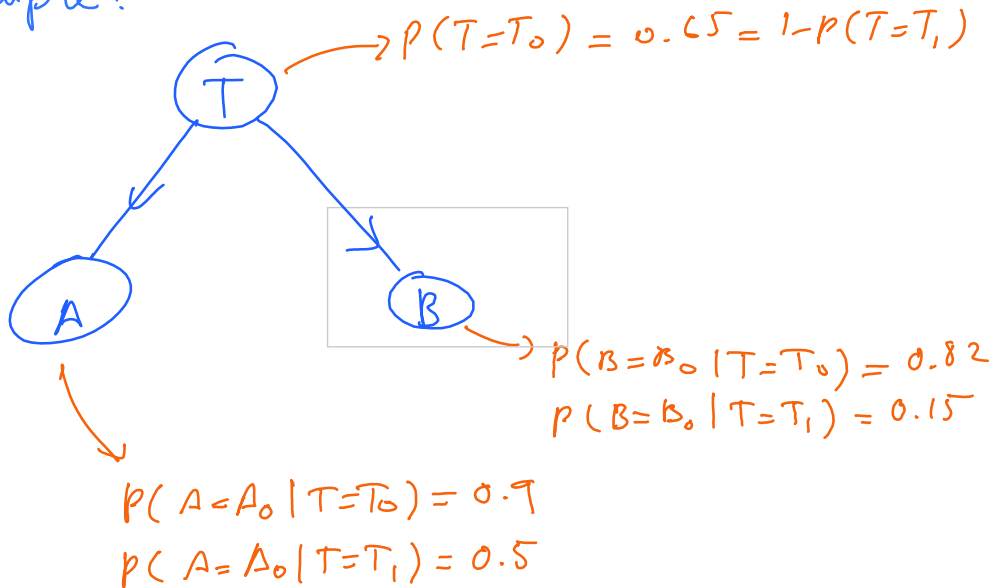
If we have Z , we can find the maximum probability. But we are interested in the values x^* of x that achieve this maximum probability (also we don't have Z). To find x^* , we simply need to keep track of which values of x_i maximize the message. Specifically, for a message $m_{ij}(x_j)$, we should know for each value of x_j what value of x_i was used to obtain the maximum value of the message. Then, when we find what value of x_4 maximizes the probability at the last step, we backtrack and find all the other x_i s.

11.5 Sum-product Example

In the example below, we are interested in the probability of each node given that $B = 0$, i.e., Bob's on time. Specifically, we are after $p(T|B = 0), p(A|B = 0), p(B|B = 0)$.

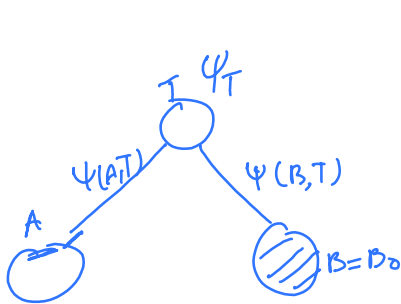
The sum-product algorithm example

Example:



$$P(ABT) = P(T) P(A|T) P(B|T)$$

We first convert this to an MRF



$$\psi_T(T) = P(T) = \begin{array}{c|c} T=0 & T=1 \\ \hline 0.65 & 0.35 \end{array}$$

$$\psi_{BT}(B, T) = P(B|T) =$$

	B=0	B=1
T=0	0.82	0.18
T=1	0.15	0.85

$$\psi_{AT}(A, T) = P(A|T) =$$

	A=0	A=1
T=0	0.9	0.1
T=1	0.5	0.5

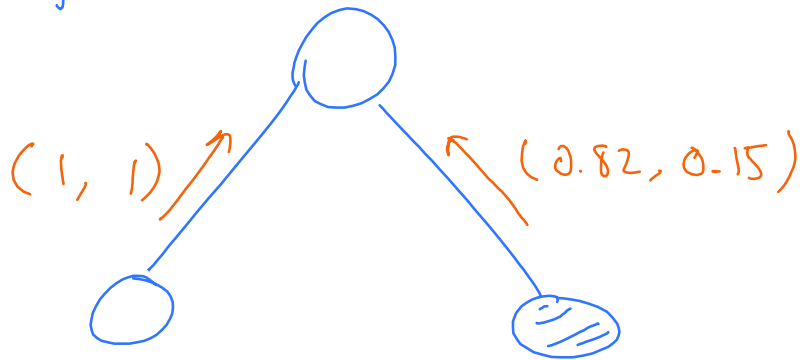
$$\mu_{BT}(T=0) = \sum_{B \in \{0\}} \psi(B, T=0) = 0.82$$

$$\mu_{BT}(T=1) = \sum_{B \in \{0\}} \psi(B, T=1) = 0.15$$

$$\psi(A, T=0) = 0.9 + 0.1 = 1$$

$$\mu_{AT}(T=0) = \sum_{A \in \{0,1\}}$$

$$\mu_{AT}(T=1) = \sum_{A \in \{0,1\}} \psi(A, T=1) = 0.5 + 0.5 = 1$$



$$P(T | B=B_0) \propto \mu_{AT}(T) \mu_{BT}(T) \psi_T(T)$$

$$T=0 : 1 \times 0.82 \times 0.65$$

$$T=1 : 1 \times 0.15 \times 0.35$$

Normalization :

$$P(T=0 | B=0) = \frac{0.82 \times 0.65}{0.82 \times 0.65 + 0.15 \times 0.35} = 0.91$$

$$M_{TA}(A=0) = \sum_{T \in \{0,1\}} M_{BT}(T) \psi(T, A=0) \psi(T)$$

$$= 0.65 \times 0.82 \times 0.9 + 0.35 \times 0.15 \times 0.5$$

$$= 0.506$$

$$M_{TA}(A=1) = \sum_{T \in \{0,1\}} M_{BT}(T) \psi(T, A=1) \psi(T)$$

$$= 0.65 \times 0.82 \times 0.1 + 0.35 \times 0.15 \times 0.5$$

$$= 0.08$$

$$P(A | B=0) \propto M_{TA}(A)$$

$$A = A_0 : 0.506$$

$$A = A_1 : 0.08$$

Normalization

$$P(A=0 | B=0) = \frac{0.506}{0.506 + 0.08} = \frac{0.506}{0.586} = 0.863$$

$$\begin{aligned}
 \mu_{TB}(B=0) &= \sum_{T \in \{0,1\}} \mu_{AT}(T) \psi(T, B=0) \\
 &= 1 \times 0.82 + 1 \times 0.15 \\
 &= 0.97
 \end{aligned}$$

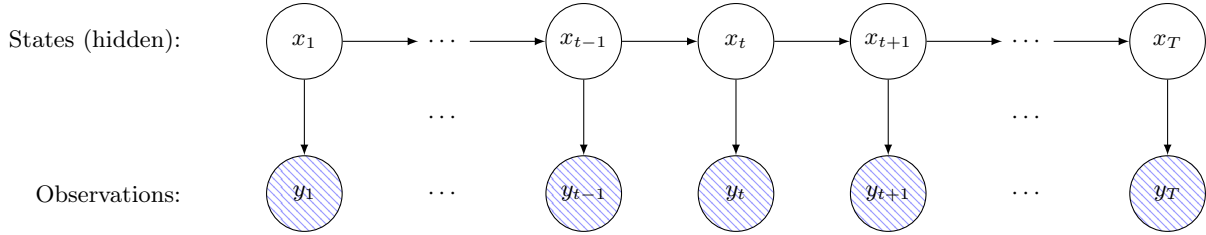
But the only case is $B=0$, so after normalization regardless of the value of $\mu_{TB}(B=0)$, we have

$$P(B=0|B=0) = 1$$

Chapter 12

Inference in Hidden Markov Models

A hidden Markov model (HMM) is a graphical model of the form shown below. The top chain is a Markov chain representing the state of some system. Typically the state cannot be observed directly. However, we can observe some (probabilistic) function of the state. For example, the Markov chain can represent the health status of a patient and the observations are symptoms such as temperature, blood pressure, etc. As another example, the Markov chain can represent the part of speech of words in a text, and the observation is the actual word.



The probability distribution for this model factorizes as

$$p(x_1^T, y_1^T; \theta) = p(x_1) \prod_{t=2}^T p(x_t | x_{t-1}) \prod_{t=1}^T p(y_t | x_t).$$

Assuming the Markov chain and the observations are both on discrete spaces, we can complete the model by specifying $\theta = (\pi, A, B)$, where:

- The probability distribution π for x_1 ,

$$\pi_i = p(x_1 = i).$$

- The *transition matrix* A of the Markov chain,

$$A_{ij} = p(x_{t+1} = j | x_t = i).$$

- The *emission matrix* B describing the probabilities of the observations given the state,

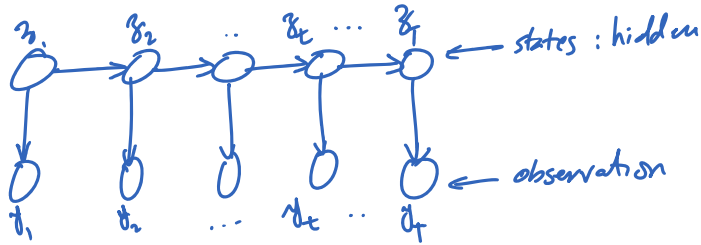
$$B_{ij} = p(y_t = j | x_t = i).$$

Below are three common inference problems associated with HMMs and the methods for solving them. We will not derive the solutions but they can be found in [2].

- Evaluation: $p(x_t | y_1^T; \theta) \rightarrow$ *forward-backward algorithm* (sum-product).
- Decoding: $\arg \max_{x_1^T} p(x_1^T | y_1^T; \theta) \rightarrow$ *Viterbi algorithm* (max-product).

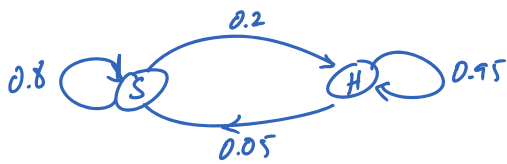
- Learning: $\arg \max_{\theta} p(y_1^T; \theta) \rightarrow \text{Baum-Welch algorithm (EM)}$.

Below are HMM notes from a previous class. Unless I get a chance to go over these in class, they are not part of the course material and are here for self-study. But note that the methods are sum-product, max-product, and EM algorithms, which are part of the course and so reviewing the material below can be helpful in understanding those.



* A person can be either sick or healthy : hidden state

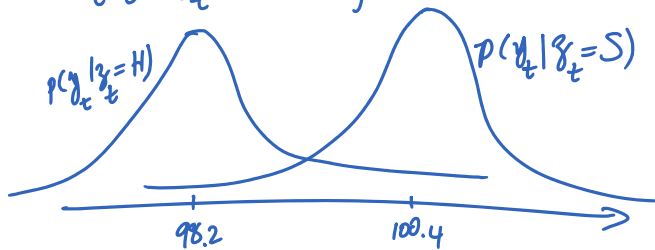
temperature : observation



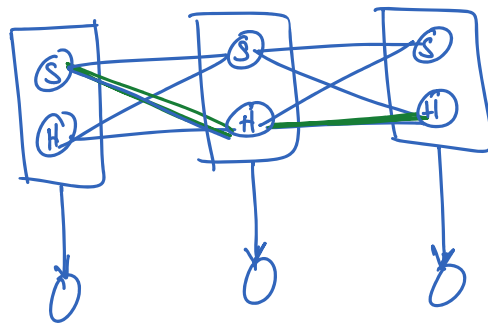
$$p(z_{t+1}=j | z_t=i) = A_{ij} : \text{transition probs.}$$

prob. distr. over initial state $p(z_1=i) = \pi_i$

$$p(y_t=j | z_t=i) = B_{ij} : \text{emission probs.}$$



Trellis :



paths : configurations
of hidden states

$$p(z_1^T, y_1^T | \theta) = p(z_1) \left(\prod_{t=2}^T p(z_t | z_{t-1}) \right) \left(\prod_{t=1}^T p(y_t | z_t) \right)$$

$$\theta = (\pi, A, B)$$

$$\pi_i = p(z_1 = i | \theta)$$

$$A_{ij} = p(z_{t+1} = j | z_t = i, \theta)$$

$$B_{ij} = p(y_t = j | z_t = i, \theta)$$

Three HMM problems:

* Evaluation: $p(z_t | y^T, \theta)$

- Forward-Backward (Sum-product)

* Decoding: $\arg \max_{z_1^T} p(z_1^T | y^T, \theta)$

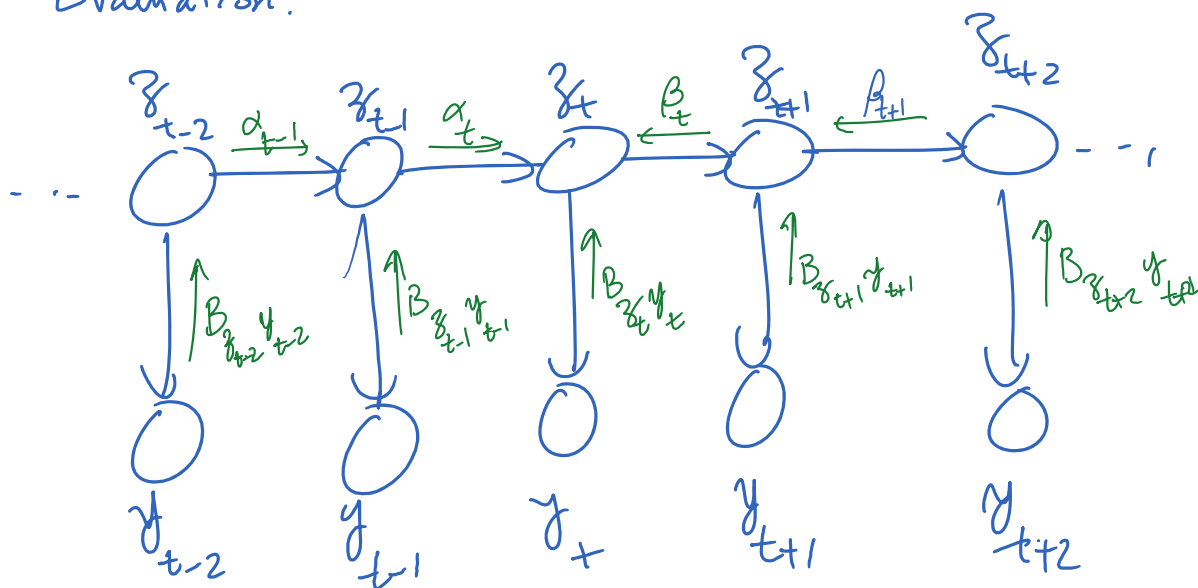
- Viterbi Alg (Max-product)

Copeland & Qualcomm

* Learning: $\arg \max_{\theta} p(y^T | \theta)$

- Baum-Welch alg (EM)

Evaluation:



Define

$$\dots \quad \pi_i = p(z_1 = i) \quad t \geq 2 \quad | \quad A_{ij} = p(z_t = i) \quad t \leq T-1$$

Define

$$\alpha_t(i) = \mu_{z_{t-1} z_t}(z_t=i) \quad t \geq 2 \quad \left| \quad \beta_t(i) = \mu_{z_{t+1} z_t}(z_t=i) \quad t \leq T-1\right.$$

$$\alpha_1(i) = \pi_i \quad \left| \quad \beta_T(i) = 1\right.$$

It can be shown (by induction) that:

$$\alpha_t(i) = p(z_t=i, y_1^{t-1} | \theta), \quad \beta_t(i) = p(y_{t+1}^T | z_t=i, \theta)$$

$$\alpha_t(i) = \sum_j \alpha_{t-1}(j) B_{j y_{t-1}} A_{ji}$$

$$\beta_t(i) = \sum_j \beta_{t+1}(j) B_{j y_{t+1}} A_{ij}$$

Marginals:

$$p(z_t=i | y_1^T, \theta) = \gamma_t(i) \propto \alpha_t(i) \beta_t(i) B_{i y_t}$$

$$p(z_{t-1}=i, z_t=j | y_1^T, \theta) = \zeta_t(i, j) \propto p(y_1^T, z_{t-1}=i, z_t=j | \theta)$$

$$= p(y_1^{t-2}, z_{t-1}=i) \underbrace{p(y_{t-1} | z_{t-1}=i) p(z_t=j | z_{t-1}=i) p(y_t | z_t=j)}_{\downarrow = p(y_{t-1} | y_1^{t-2}, z_{t-1}=i)} p(y_{t+1}^T | z_t=j)$$

$$= \alpha_{t-1}(i) B_{i y_{t-1}} A_{ij} B_{j y_t} \beta_t(j)$$

In traditional form of Forward-Backward, forward mgs
included $B_{i y_t}$.

$$\bar{\alpha}_t(i) = p(z_t=i, y_1^t | \theta) = \alpha_t(i) B_{i y_t}$$

$$\bar{\alpha}_t(i) = \sum_j \bar{\alpha}_{t-1}(j) A_{ji} B_{ij_t}$$

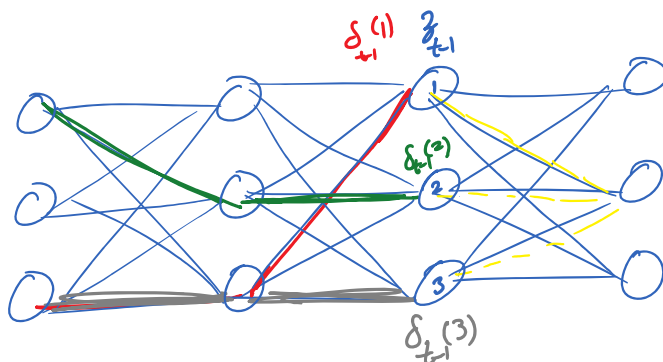
$$\gamma_t(i) \propto \bar{\alpha}_t(i) \beta_t(i)$$

Max-product: choose z_T as root.

$$\text{Define } \delta_t(i) = \mu_{z_{t-1} z_t} (z_t = i), \quad t \geq 2 \quad \delta_1(i) = \pi_i$$

Can be shown that

$$\delta_t(i) = \max_{z_1^{t-1}} p(z_1^{t-1}, z_t = i, y_1^{t-1} | \theta)$$



$$\delta_t(i) = \max_j \delta_{t-1}(j) B_{jy_{t-1}} A_{ji}$$

$$\text{prob of the max-prob path} = \max_j \delta_T(j) B_{jy_T}$$

* Learning: EM / Baum-Welch

Assume complete data: z_1^T, y_1^T

Estimate $\theta = (\pi, A, B)$

$$\hat{\pi}_i = \begin{cases} 1 & z_1 = i \\ 0 & \text{else} \end{cases}$$

$$\hat{A}_{ij} = \frac{\sum_{t=1}^{T-1} I(z_t = i, z_{t+1} = j)}{\sum_{t=1}^{T-1} I(z_t = i)}$$

$$\hat{A}_{ij} = \frac{\sum_{t=1}^T I(z_t = i, y_t = j)}{\sum_{t=1}^T I(z_t = i)}$$

log likelihood of the complete data

$$\ln p(z_1^T, y_1^T | \theta) = \ln \pi_{z_1} + \sum_{t=2}^T \ln A_{z_{t-1} z_t} + \sum_{t=1}^T \ln B_{z_t y_t}$$

E-step

$$Q(\theta | \theta') = E[\ln p(z_1^T, y_1^T | \theta) | y_1^T, \theta']$$

$$E[\ln \pi_{z_1} | y_1^T, \theta'] = \sum_i (\ln \pi_i) \underbrace{p(z_1 = i | y_1^T, \theta')}_{\gamma'_1(i)} = \sum_i \gamma'_1(i) \ln \pi_i$$

$$E\left[\sum_{t=2}^T \ln A_{z_{t-1} z_t} | y_1^T, \theta'\right] = \sum_{t=2}^T \sum_i \sum_j (\ln A_{ij}) \underbrace{p(z_t = j, z_{t-1} = i | y_1^T, \theta')}_{\sum_t \gamma'_t(i, j)}$$

$$= \sum_i \sum_j \left(\sum_{t=2}^T \gamma'_t(i, j) \right) \ln A_{ij}$$

$$E\left[\sum_{t=1}^T \ln B_{z_t y_t} | y_1^T, \theta'\right] = \sum_i \sum_{t=1}^T \underbrace{p(z_t = i | y_1^T, \theta')}_{\gamma'_t(i)} \ln B_{i y_t}$$

ML for p if the LL = $\sum_j n_j \ln p_j \Rightarrow p_j \propto n_j$

$$\pi_i \propto \gamma'_1(i) \quad A_{ij} \propto \sum_{t=2}^T \gamma'_t(i, j)$$

$$\Rightarrow \sum_i \sum_j \left(\sum_{t=1}^T \gamma'_t(i) I(y_t = j) \right) \ln B_{ij} \Rightarrow B_{ij} \propto \sum_{t=1}^T \gamma'_t(i) I(y_t = j)$$

Helpful references: [2, 1]

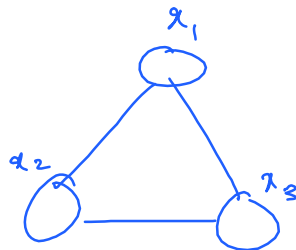
References

- [1] Christopher M. Bishop. *Pattern Recognition And Machine Learning*. New York: Springer, 2006. URL: <http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%5C%20-%5C%20Pattern%5C%20Recognition%5C%20And%5C%20Machine%5C%20Learning%5C%20-%5C%20Springer%5C%20%5C%202006.pdf> (visited on 02/14/2017).
- [2] Bruce Hajek. *Random Processes for Engineers*. Illinois, 2014. URL: <http://hajek.ece.illinois.edu/Papers/randomprocJuly14.pdf> (visited on 01/30/2017).

Chapter 13

Factor Graphs and Sum/Max-product Algorithms **

This MRF implies the factorization



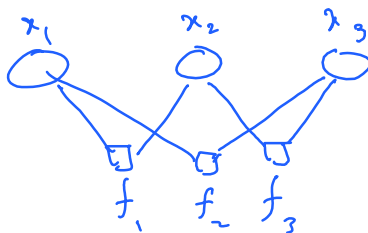
$$p(x_1, x_2, x_3) \propto \psi(x_1, x_2, x_3)$$

But suppose we actually want to represent

$$p(x_1, x_2, x_3) \propto f(x_1, x_2) f(x_2, x_3) f(x_3, x_1)$$

Is there a way to do this with a graph?

Factor graph:



Two types of nodes:

V : variables: x_1, x_2, x_3

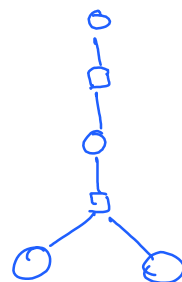
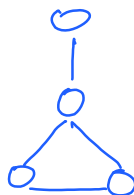
F : factors: f_1, f_2, f_3

$$p(x_1^m) = \prod_{f_i \in F} f_i(x_{f_i})$$

variable nodes adjacent to f_i

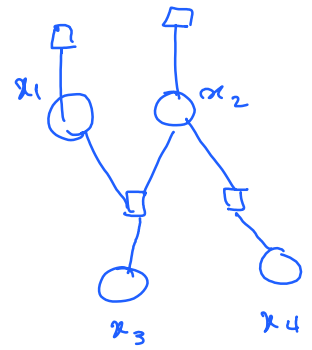
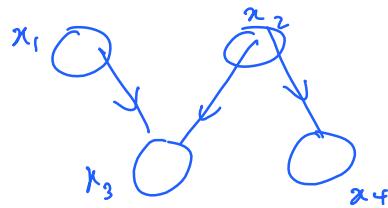
MRF \rightarrow FG

nodes \rightarrow variable nodes
maximal cliques \rightarrow factor nodes



BN \rightarrow FG

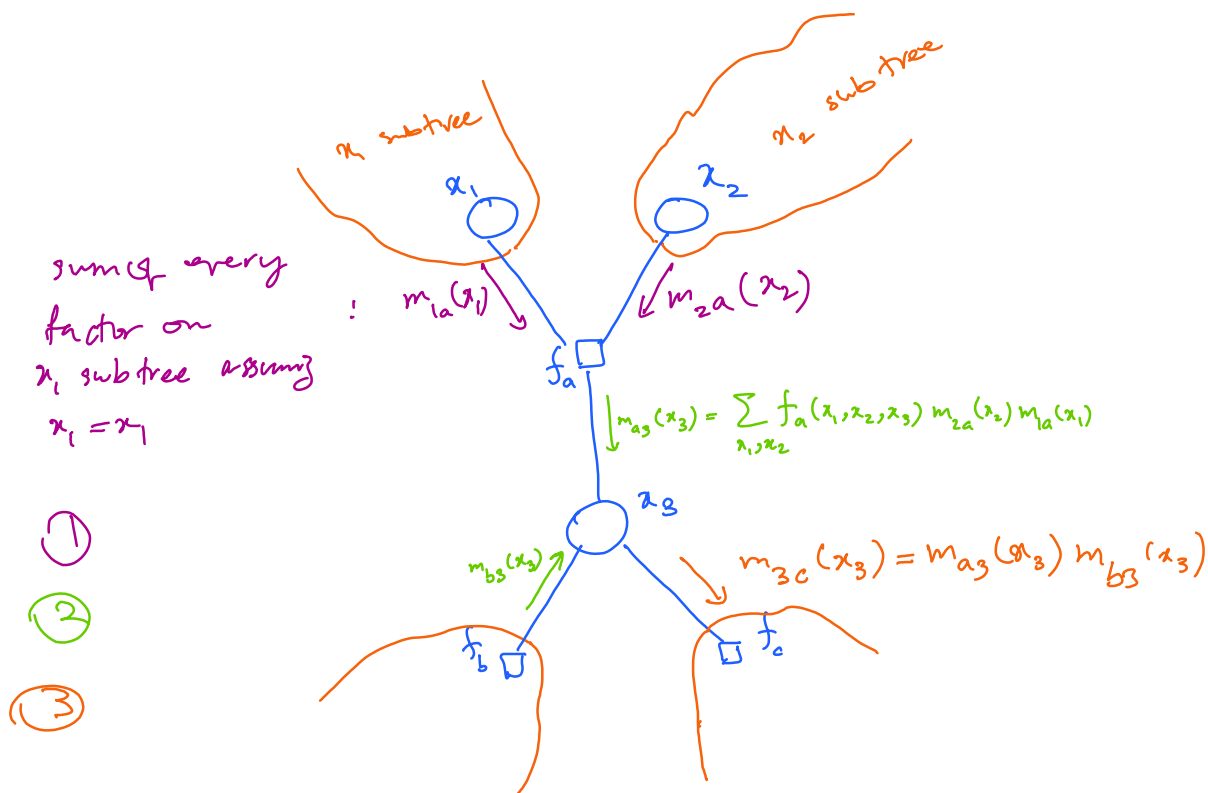
nodes \rightarrow variable nodes
 (CPDs at) nodes \rightarrow factor nodes



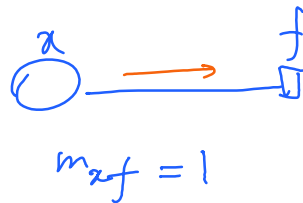
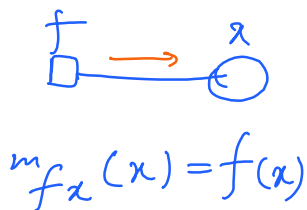
Note: even if the original MRF is not a tree, or the original BN has an MRF which is not a tree, the factor graph may still be a tree: that is discarding the types of the nodes in FG leads to a tree.

this is good because \rightarrow

Sum-product for factor tree graphs:



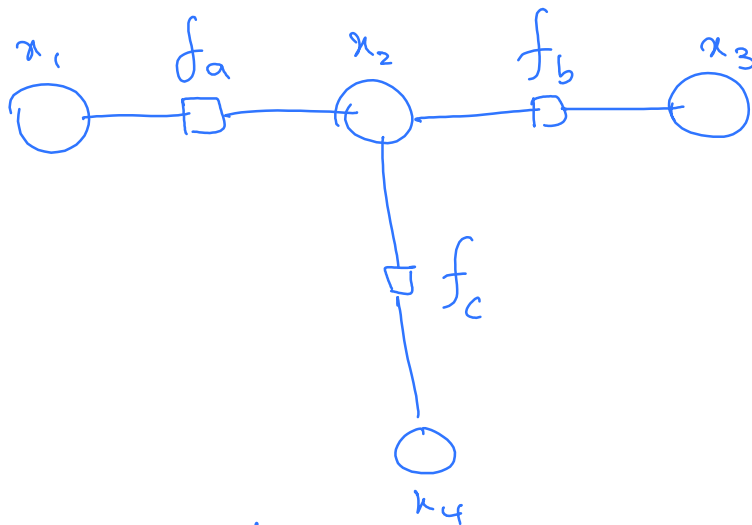
Starting steps at leaves:



Marginal at each node: product of msgs it receives

marginals at all nodes takes twice as much of that of a single node : two msgs per link as opposed to one.

Example:



round 1

round 2

$$\mu_{1a}(x_1) = 1$$

$$\mu_{4c}(x_4) = 1$$

$$\mu_{3b}(x_3) = 1$$

$$\mu_{a2}(x_2) = \sum_{x_1} f_a(x_1, x_2) \mu_{1a}(x_1)$$

$$\mu_{b2}(x_2) = \sum_{x_3} f_b(x_3, x_2) \mu_{3b}(x_3)$$

$$\mu_{c2}(x_2) = \sum_{x_4} f_c(x_2, x_4) \mu_{4c}(x_4)$$

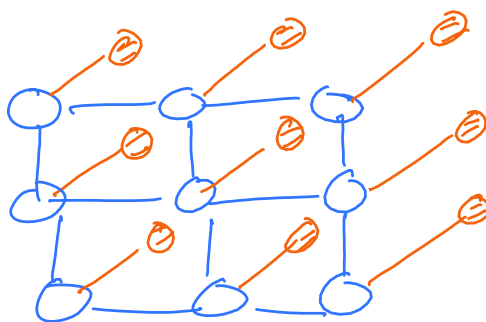
Problem: Identify the most likely configuration:

Find a set of values x_1^*, \dots, x_m^* s.t.

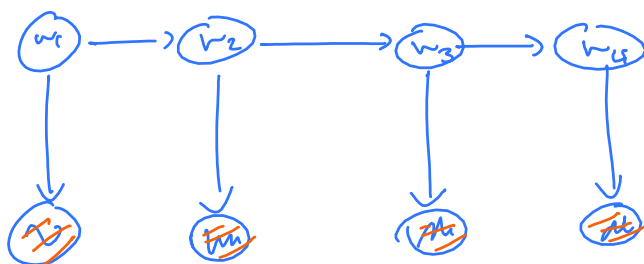
$$P(x_1^*, \dots, x_m^*) \geq P(x_i^m) \text{ for all } x_i^m$$

Applications:

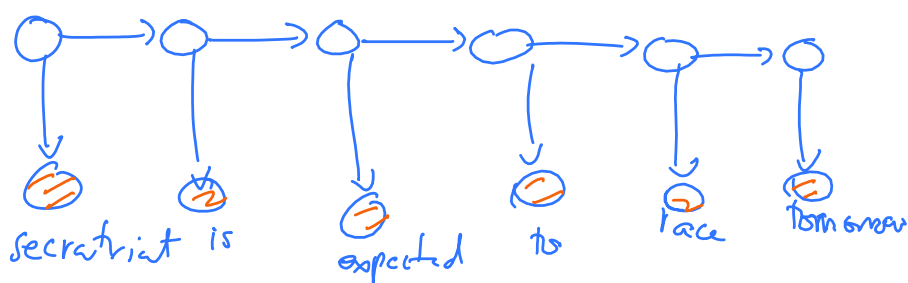
- Image denoising (Lab 1)



- Voice recognition



- Part of speech tagging



Do we already know how to solve this?

Finding most probable state for a node:

run sum-product and find
state with max prob.

Finding most probable configuration for graph:

max-product

$$x_{\max} = \arg \max_x p(x)$$

May not be the same.

	$x=0$	$x=1$
$y=0$	0.3	0.4
$y=1$	0.3	0

$x=0$ max for x

$y=0$ max for y

$(x,y)=(0,1)$ max for (x,y)

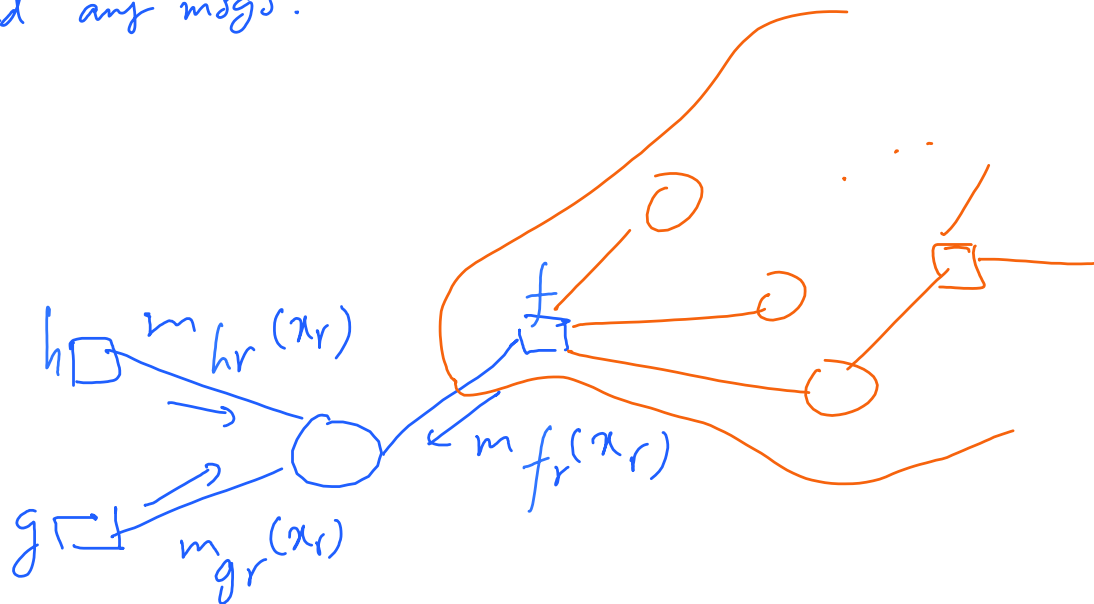
Let's instead try to find $\max_{x_1^n} p(x_1^n)$

$$\max_{x_1^n} p(x_1^n) = \max_{x_1} \max_{x_2} \dots \max_{x_m} p(x_1^n)$$

* We can use a similar approach to elimination
except that Σ is replaced with max.

* Similarly sum-product can be turned
to max-product to find $\max_x p(x)$

* Here, we pick an arbitrary root, which does not send any msgs.



The message from f $\mu_{fr}(x_r)$

Given the particular value for x_r

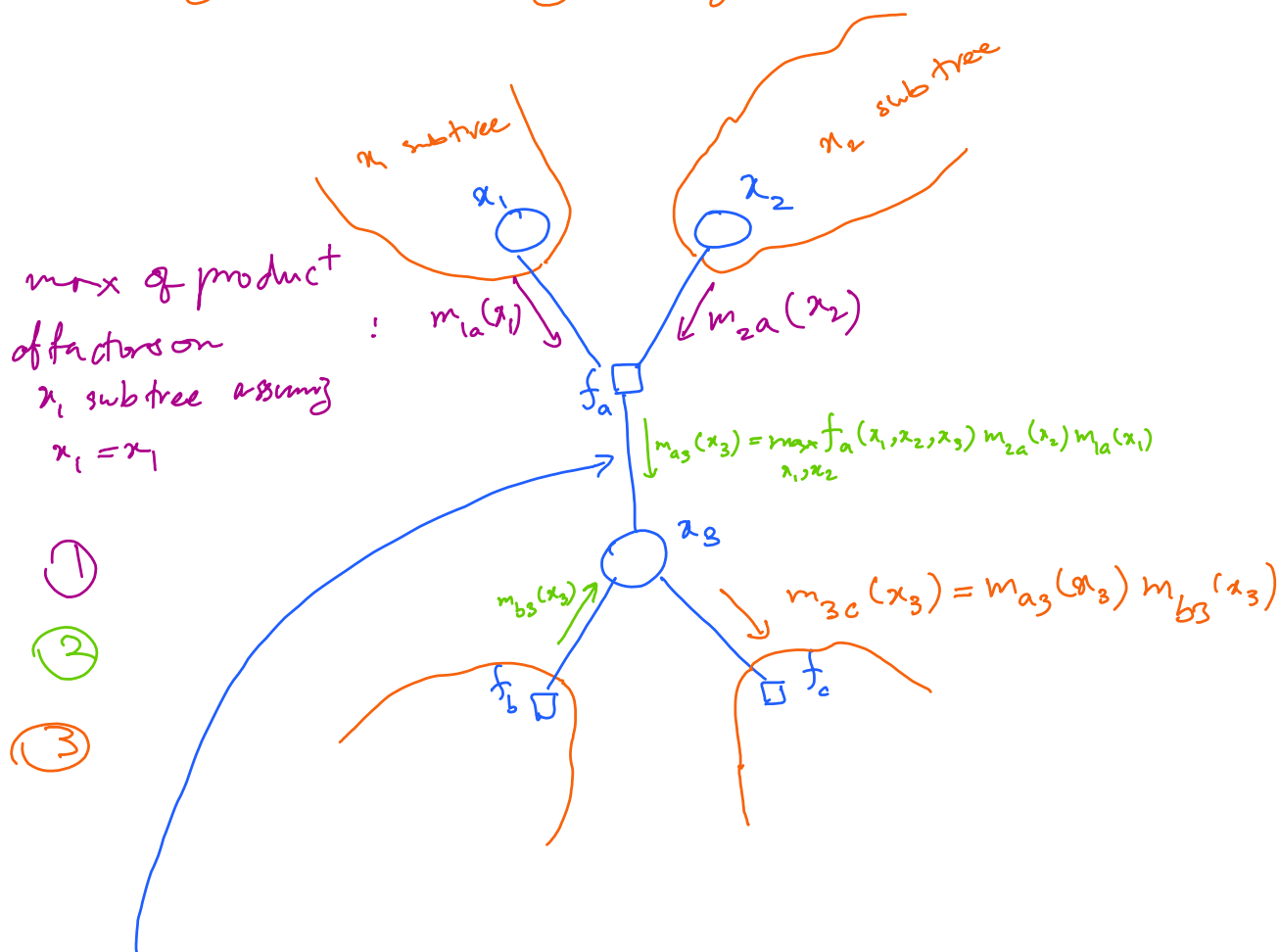
* What is the most likely configuration for the subtree of f

* What is the "probability" of this configuration

$$\max p(\alpha_i^m) = \max_{x_r} \mu_{fr}(x_r) \mu_{gr}(x_r) \mu_{hr}(x_r)$$

from the value of x_r that maximizes this sum and messages we find the most likely configuration

* Finding the maximizing configuration:



For each value of x_3 , we also record which values of x_1, x_2 achieved the max.

At the root, we find x_r^* that achieves the max. Then we back-track and find maximizing values for all nodes.

* It may be more convenient to maximize

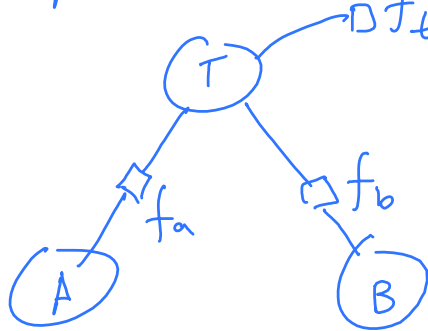
$\ln p(x) \Rightarrow$ max-sum algorithm

$$\sum f_i(x_{f_i})$$

$$\max_y p(x) = \max_{x_1 \dots x_n} \dots$$

note: why not continue the alg so that we can find x_i^* for all nodes just like how we found it for x_r ? We will find maximizing values, but they may belong to different maximizing configurations.

Example: most-likely configuration



$$f_t(T=0) = 0.65$$

$$f_t(T=1) = 0.35$$

$$f_a(A=0, T=0) = 0.9$$

$$f_a(A=0, T=1) = 0.5$$

$$f_a(A=1, T=0) = 0.1$$

$$f_a(A=1, T=1) = 0.5$$

$$f_b(B=0, T=0) = 0.82$$

$$f_b(B=0, T=1) = 0.15$$

$$f_b(B=1, T=0) = 0.18$$

$$f_a(B=1, T=0) = 0.12$$

$$\mu_{A^*} : \max \quad |$$

$$A=0 \longrightarrow |$$

$$A=1 \longrightarrow |$$

$$\mu_{aT} : \max_A f_a(A, T)$$

$$T=0 \longrightarrow 0.9 \text{ for } A=0$$

$$T=1 \longrightarrow 0.5 \text{ for } A=0 \text{ \& } A=1$$

$$\mu_{tT} : \max_T f_t(T)$$

$$T=0 \longrightarrow 0.65$$

$$T=1 \longrightarrow 0.35$$

$$\mu_{Tb} : \max_T \mu_{tT}(T) \mu_{a\bar{1}}(T) f_b(B, T)$$

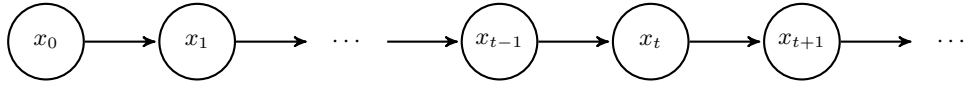
...

Chapter 14

Markov Chains

14.1 Introduction

A **Markov chain (MC)** is a stochastic process whose future is independent from its past and can be represented as the following Bayesian network:



The value of x_t is called the **state** of the Markov chain at time t . The set of all possible states is the **state space**. For example,

- We may represent daily weather with the state space: {sunny, cloudy, rainy}
- The state of the disease in a patient may be represented by a MC with two states: {remission, relapse}.
- The number of animals of a certain species can be represented with states $\{0, 1, 2, \dots\}$.

Uncountable state spaces are also possible (e.g., temperature) and we will rely on them for sampling later. But for simplicity, we focus on finite-state MCs. Also, note that a MC is usually an approximation of the world since we like to have a small number of states.

To complete the characterization of a MC, we also need to know the CPDs,

$$p(x_0 = i), \quad p(x_{t+1} = j | x_t = i).$$

We refer to $p(x_0)$ as the **initial distribution** and to the CPD $p(x_{t+1} = j | x_t = i)$ as **transition probabilities**. We are interested in **time-homogeneous** MCs only, in which $p(x_{t+1} = j | x_t = i)$ is independent of t , i.e., the same for all time instances. In such MCs, we can represent the transition probabilities as a transition matrix A with

$$P_{ij} = p(x_{t+1} = j | x_t = i),$$

which is particularly useful if the state space is a finite set.

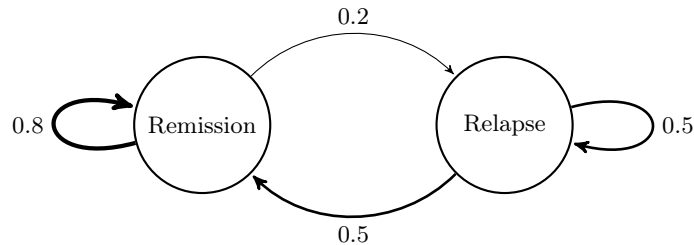
Example 14.1. In a Markov chain representing the health of a patient, if we let 1 represent ‘remission’ and 2 represent ‘relapse,’ we may have

$$P = \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix}.$$

△

Given that the important features of a time-homogeneous MC are its state space and transition probabilities,

it is useful to represent the chain as graph, called the **state-transition graph**, whose nodes are the states and edges represent transitions and their probabilities. For example, for a disease, we may have

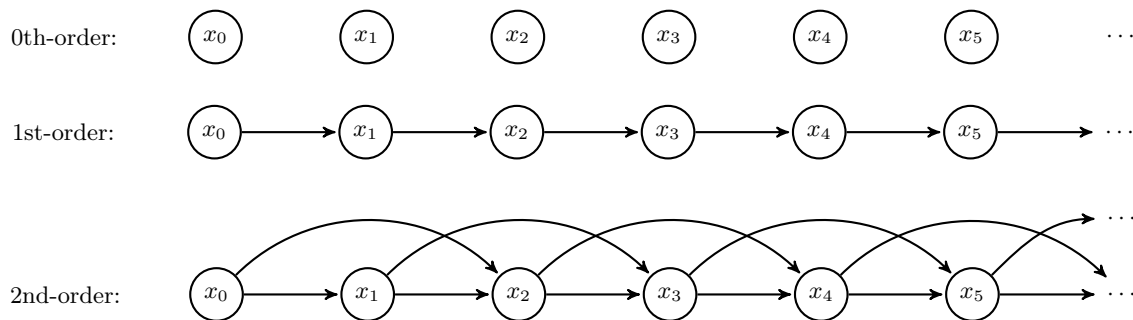


Here are some other examples of common MCs:

- **Random walk** on a grid (1D, 2D, ...). For example, in the 1-dimensional case, we can move left or right at random. This extends to n dimensions. In this context, “a drunk man will find his way home, but a drunk bird may get lost forever.”
- **Page-rank**. This is closely related to the previous chain, except that this time the states are webpages, and we click on a link in the current page to transition to another one. This was the main idea behind Google search’s ranking of web pages, using stationary probabilities (more on these below).
- **DNA mutations**. There are four states $\{A, C, G, T\}$ and due to mutations, a position in the genome may change from one state to another. Several variations are used in phylogenetics.

As stated before, MCs are usually approximations of real phenomena because we cannot include all relevant information in the state. As an example, consider a MC for weather. Suppose our chain represents a short period where seasonal effects are negligible and so we can assume the chain to be time-homogeneous. Each state of the MC could be the total amount of precipitation. This is already useful since a rainy day is more likely after a rainy day than after a sunny day. But if we add information about temperature, cloud cover, air pressure, etc., the model becomes more accurate and useful.

Another way that MCs can be extended is by allowing dependence on more than previous state, i.e., allowing the **order** to be larger than 1. Graphical examples of zeroth-order, first-order, and second-order MCs are shown below:



Example 14.2 ([2]). More accurate models can produce more realistic data, as shown in the following example from Shannon on modeling English text as a MC.

1. Zero-order approximation with uniform distribution (symbols are independent and equally probable).

XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGXYD QPAAMKBZAACIB-
ZLHJQD

2. Zero-order approximation (symbols independent but their probability is the same as English text).

OCRO ILI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA
NAH BRL

3. First-order approximation (digram structure; the conditional probability of each symbol given the previous is like English).

ON IE ANTSOUTINY S ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE
TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

4. Second-order approximation (trigram structure as in English).

IN NO 1ST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMON-
STURES OF THE REPTAGIN IS REGOACTIONA OF CRE

5. Zero-Order Word approximation; words are chosen independently but with their appropriate frequencies.

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT
NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO
FURNISHES THE LINE MESSAGE HAD BE THESE.

6. First-Order Word approximation; the word transition probabilities are as in English text.

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE
CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LET-
TERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEX-
PECTED

△

14.2 State distribution as a function of time

Consider a MC with m states. Let $\pi_t = (\pi_{t1}, \pi_{t2}, \dots, \pi_{tm})$ denote the probability distribution over the states at time t , where $\pi_{tj} = p(x_t = j)$. Usually, π_0 , or equivalently, $p(x_0)$ is given. We have the following recursion,

$$\pi_{tj} = \sum_{i=1}^m p(x_{t-1} = i) p(x_t = j | x_{t-1} = i) = \sum_{i=1}^m \pi_{t-1,i} P_{ij},$$

or more compactly

$$\pi_t = \pi_{t-1} P \quad \text{and} \quad \pi_t = \pi_0 P^t.$$

Furthermore, the ij th element of P^t , shown as $(P^t)_{ij}$, is the probability of ending up in state j in t steps if we start from state i .

Example 14.3 (Example 14.1 continued). Suppose $\pi_0 = (1, 0)^T$, i.e., the patient starts in remission. Then,

$$\begin{aligned} \pi_1 &= (1, 0) \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix} = (0.8, 0.2), & \pi_2 &= \pi_1 \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix} = (0.74, 0.26) \\ \pi_5 &= \pi_0 P^5 = (0.71498, 0.28502), & \pi_{10} &= \pi_0 P^{10} = (0.71429, 0.28571) \end{aligned}$$

So after 10 days, the probability of being in remission is about 71%.

Now suppose the patient starts in relapse. Then

$$\begin{aligned} \pi_1 &= (0.5, 0.5), & \pi_2 &= (0.65, 0.35) \\ \pi_5 &= (0.71255, 0.28745), & \pi_{10} &= (0.71428, 0.28572) \end{aligned}$$

We can see that, interestingly, π_5 and π_{10} are very close to each other and almost independent of π_0 . We will study this further in the next section. △

14.3 Long-term Behavior of Markov Chains

What happens to a MC if we let it run for a long time? This problem is of interest in a variety of contexts, e.g., the Page-rank algorithm above and sampling methods discussed later. We saw in the previous example that as t grows the distribution over the states appears to settle down on a certain distribution, which is called the **limiting distribution**. In the example, the limiting distribution was independent of the initial distribution. In this section, we will study when and why this happens.

A **stationary distribution** of a MC is a distribution σ that satisfies

$$\sigma = \sigma P.$$

Any finite-state Markov chain has at least one stationary distribution [1]. The limiting distribution, if it exists, must be a stationary distribution.

Example 14.4 (Example 14.3 continued). The stationary distribution $\sigma = (\sigma_1, \sigma_2)$ is obtained by solving $(\sigma_1, \sigma_2) = (\sigma_1, \sigma_2)P$ and $\sigma_1 + \sigma_2 = 1$. It can be shown that the unique solution to these equations is

$$\sigma = (5/7, 2/7) = (0.71429, 0.28571),$$

which indeed appears to be the limiting distribution regardless of the initial distribution. \triangle

Graph vs. transition matrix. Whether or not a MC converges to a unique limiting distribution is determined by P . This dependence is only on P_{ij} being zero or nonzero but not how large the values are otherwise. The zero/positive status of each transition probability is given by the MC graph—an edge from states i to state j exists if and only if $P_{ij} > 0$. So the graph is sufficient to decide whether the MC will converge to a unique stationary distribution.

First, let us see some examples when the stationary distribution is not unique:



On the left, the limiting distribution depends on the initial distribution. This arises because of a lack of connectivity between the states. On the right a limiting distribution does not exist because the chain is *periodic* in a certain sense.

We can eliminate both of these possibilities by defining regular Markov chains. A Markov chain is **regular** if there is a positive integer k such that for all i and j it is possible to go from state i to state j in k steps. This is equivalent to $(P^k)_{ij} > 0$ for all i, j and also equivalent to the existence of a path of length k between any two states. In Example 14.1, we have $k = 1$.

Theorem 14.5. *If a MC with transition matrix P is regular, then there exists a unique distribution σ such that $\sigma = \sigma P$ and for any π_0 , we have $\pi_t = \pi_0 P^t \rightarrow \sigma$ as $t \rightarrow \infty$.*

The above theorem guarantees that regular MCs converge to their unique stationary distributions. Furthermore, since we can choose π_0 to have a 1 in any position, the theorem also implies that each row of P^t converges to σ .

Example 14.6 (Example 14.4 continued). Indeed, $\sigma = (5/7, 2/7) = (0.71429, 0.28571)$ is the stationary distribution of

$$P = \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix}$$

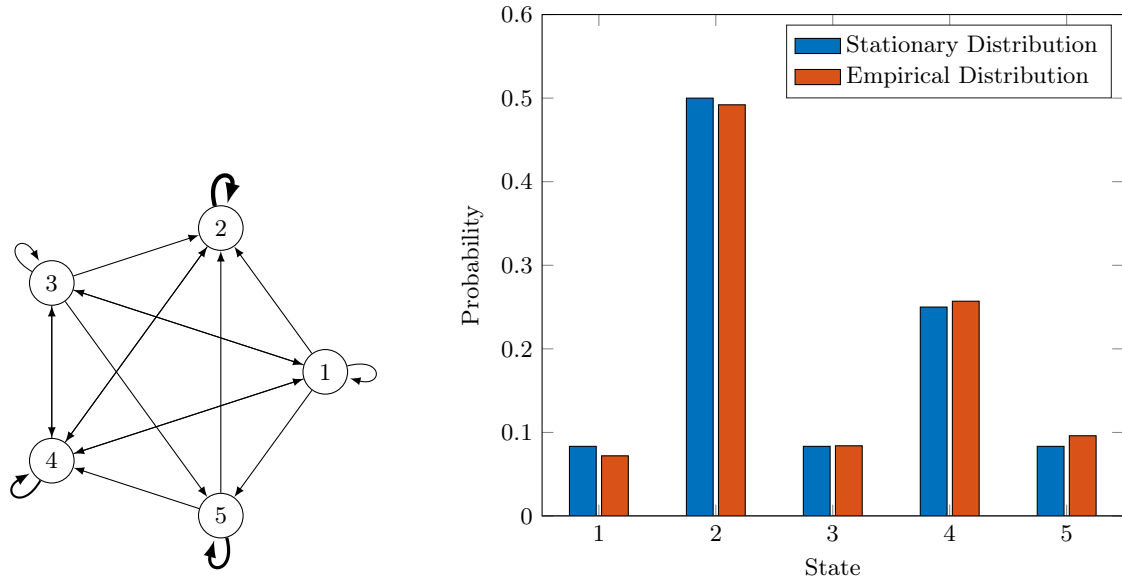


Figure 14.1: In the Markov chain (left), edges between different nodes have probability $1/5$ and the probability of self loops is such that the outgoing probabilities sum to 1. The stationary distribution and an empirical (time-averaged) distribution are given on the right.

and $\pi_t \rightarrow \sigma$ regardless of π_0 as we saw in Example 14.1. Furthermore,

$$P^2 = \begin{pmatrix} 0.74 & 0.26 \\ 0.65 & 0.35 \end{pmatrix}, \quad P^5 = \begin{pmatrix} 0.71498 & 0.28502 \\ 0.71255 & 0.28745 \end{pmatrix}, \quad P^{10} = \begin{pmatrix} 0.71429 & 0.28571 \\ 0.71428 & 0.28572 \end{pmatrix}$$

△

14.3.1 How often does the Markov Chain visit each state?

For a regular MC with stationary distribution σ , we know if t is large, at time t , the probability of being in state j is σ_j . But in a time period of length N , how many times state j is visited? The answer is approximately $N\sigma_j$ if N is large. (While this seems natural, similar statements do not necessarily hold for other random processes.)

For example, for a chain with transition matrix,

$$P = \frac{1}{5} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 4 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 0 \\ 0 & 1 & 0 & 1 & 3 \end{pmatrix},$$

whose graph is shown in Figure 14.1 (left), a simulation of length 1000 time units produced an empirical distribution close to the stationary distribution. The first 20 samples are as follows: 32244322242222244122.

14.4 Balance Properties and Finding the Stationary Distribution

14.4.1 Detailed Balance

A distribution π satisfies the **Detailed Balance Property (DBP)** if

$$\pi_i P_{ij} = \pi_j P_{ji}.$$

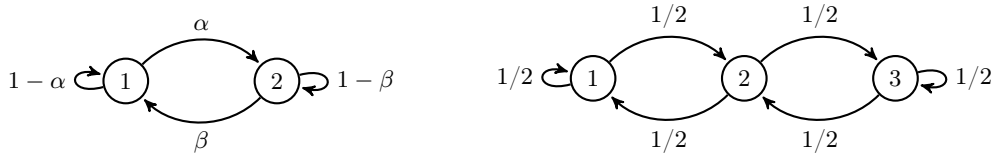
Theorem 14.7. *For a regular MC, if a vector π satisfies the detailed balance property, then π is the unique stationary distribution ($\pi = \sigma$).*

Proof. From Theorem 14.5, we know that the stationary distribution is unique, i.e., there is a unique σ satisfying $\sigma = \sigma P$. So it suffices to show that π satisfies the equation $\pi = \pi P$, where P is the transition matrix. For all j ,

$$\pi_j = \pi_j \sum_i P_{ji} = \sum_i \pi_j P_{ji} = \sum_i \pi_i P_{ij}.$$

Hence, $\pi = \pi P$. □

Exercise 14.8. Using DBP, find the stationary distribution for the following MCs.

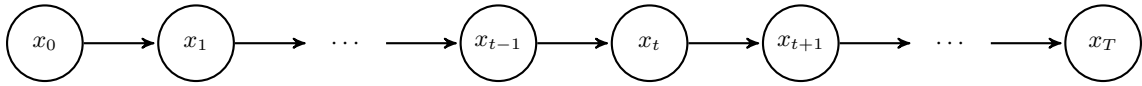


△

If our MC is regular and DBP holds, then we have the stationary distribution. This approach, if possible, is an easy way to find the stationary distribution. For this reason, DBP is commonly used in Markov Chain Monte Carlo (MCMC) methods which we discuss later.

14.4.2 Time-Reversibility **

Consider the Markov chain



and assume that $\pi_t = \sigma$, where σ is a stationary distribution. Suppose that we run the chain backward in time (or play a movie of it backward). Note that the Markov property still holds as

$$p(x_t | x_{t+1}, \dots, x_T) = p(x_t | x_{t+1})$$

So what are the transition probabilities P^- for the reversed MC? We have

$$P_{ij}^- = p(x_t = j | x_{t+1} = i) = \frac{p(x_t = j, x_{t+1} = i)}{p(x_{t+1} = i)} = \frac{\pi_j P_{ji}}{\pi_i}.$$

The MC is called **time-reversible** if $P^- = P$, which is equivalent to $\pi_i P_{ij} = \pi_j P_{ji}$ for all i, j , which are the detailed balance equations.

14.4.3 Global Balance **

A distribution π over the states of the MC satisfies the **Global Balance Property (GBP)** if for any partition¹ $\{R, L\}$ of the states of the MC, we have

$$\sum_{i \in L} \pi_i \sum_{j \in R} P_{ij} = \sum_{j \in R} \pi_j \sum_{i \in L} P_{ji}.$$

In particular, for any node i ,

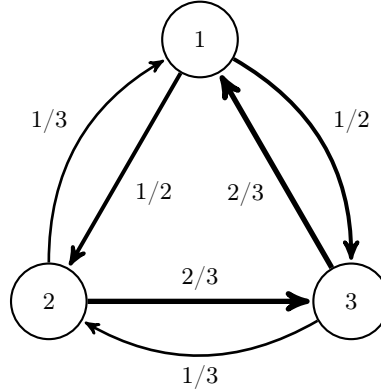
$$\pi_i \sum_{j \neq i} P_{ij} = \sum_{j \neq i} \pi_j P_{ji}.$$

It is not difficult to show mathematically that any stationary distribution σ of the Markov chain satisfies the global balance property. To see this intuitively, imagine Alice performs a random walk over the state-transition graph, going from state to state according to the transition probabilities P . Assume that $\pi_0 = \sigma$, i.e., Alice chooses her initial position according to σ . It follows that $\pi_t = \sigma$. During N steps, where N is large, the number of times that Alice goes from a state in L to a state in R is approximately $N \sum_{i \in L} \pi_i \sum_{j \in R} P_{ij}$. Similarly, the number of times that Alice goes from R to L is about $\sum_{j \in R} \pi_j \sum_{i \in L} P_{ji}$. Since Alice cannot disappear, we must have $\sum_{i \in L} \pi_i \sum_{j \in R} P_{ij} = \sum_{j \in R} \pi_j \sum_{i \in L} P_{ji}$.

We can use the GBP to find the stationary distribution as shown in the next example.

Example 14.9. Consider a chain with

$$P = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix}.$$



The DBP equations are

$$\begin{aligned} \pi_1 \cdot \frac{1}{2} &= \pi_2 \cdot \frac{1}{3} \\ \pi_1 \cdot \frac{1}{2} &= \pi_3 \cdot \frac{2}{3} \\ \pi_2 \cdot \frac{2}{3} &= \pi_3 \cdot \frac{1}{3}, \end{aligned}$$

which are not satisfiable. Visually, from the diagram above we may also have guessed that the flow of probability is more counterclockwise than clockwise, and so each pair of states is unbalanced.

¹A partition of a set S is a collection of disjoint sets whose union is equal to S .

The GBP equations are

$$\begin{aligned}\pi_1 \cdot \left(\frac{1}{2} + \frac{1}{2}\right) &= \pi_2 \cdot \frac{1}{3} + \pi_3 \cdot \frac{2}{3} \\ \pi_2 \cdot \left(\frac{1}{3} + \frac{2}{3}\right) &= \pi_1 \cdot \frac{1}{2} + \pi_3 \cdot \frac{1}{3} \\ \pi_3 \cdot \left(\frac{2}{3} + \frac{1}{3}\right) &= \pi_1 \cdot \frac{1}{2} + \pi_2 \cdot \frac{2}{3}\end{aligned}$$

which are satisfied for $\pi_2 = \frac{12}{14}\pi_1$ and $\pi_3 = \frac{15}{14}\pi_1$. Taking into account the fact that the probabilities must sum to 1, we find $\pi_1 = \frac{14}{41}, \pi_2 = \frac{12}{41}, \pi_3 = \frac{15}{41}$. \triangle

References

- [1] Alex Furman. “WHAT IS . . . a Stationary Measure?” In: *Notices of the AMS* 58.9 (). URL: <https://www.ams.org/journals/notices/201109/rtx110901276p.pdf> (visited on 04/19/2020).
- [2] C.E. Shannon. “A mathematical theory of communication.” In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: [10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).

Chapter 15

Sampling Methods

15.1 Introduction

In Bayesian inference, **distributions** are the ultimate tool for representing knowledge about unknown quantities. This is the reason that we try to find $p(\boldsymbol{\theta}|\mathcal{D})$. If we have the distribution, we can find the **expected value** of various functions of the unknown quantity and in this way find point estimates or the probability of an event,

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{Bayes} &= \mathbb{E}[\boldsymbol{\Theta}|\mathcal{D}], \\ p(\boldsymbol{\theta} \in A|\mathcal{D}) &= \mathbb{E}[\mathbb{1}(\boldsymbol{\Theta} \in A)|\mathcal{D}],\end{aligned}$$

where A is an event and $\mathbb{1}(\text{condition})$ equals 1 if the condition holds and is 0 otherwise.

If we find the posterior distribution in closed form and it turns out to be one of the common distributions, e.g., Gaussian, Poisson, etc, then typically, we can easily compute expected values. However, this is not always the case, and we may face two difficulties:

1. Sometimes all we have is a function $q(\boldsymbol{\theta})$ that is proportional to $p(\boldsymbol{\theta}|\mathcal{D})$,

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta}) = q(\boldsymbol{\theta})$$

and we are not even able to compute $p(\boldsymbol{\theta}|\mathcal{D})$ for a given $\boldsymbol{\theta}$ because the normalization factor is not known.

2. Even if we can compute $p(\boldsymbol{\theta}|\mathcal{D})$, computing expected values requires integration, which may be computationally infeasible.

In such cases, sampling from this distribution will be useful because sampling allows us to find expected values. For a function h , by the law of large numbers we have the following approximation

$$\mathbb{E}[h(X)] \simeq \sum_{i=1}^N h(x_i),$$

where x_i are independent samples drawn from the distribution p_X with respect to which the expected value is to be computed.

For example, recall that in Bayesian linear regression, a common likelihood is

$$\mathbf{y}|\boldsymbol{\theta}, \sigma^2 \sim \mathcal{N}(X\boldsymbol{\theta}, \sigma^2 I),$$

with prior

$$p(\boldsymbol{\theta}, \sigma^2) \propto 1/\sigma^2.$$

For this model, we found $p(\boldsymbol{\theta}|\mathcal{D}, \sigma^2)$ and $p(\sigma^2|\mathcal{D})$ and stated that while it is possible to obtain $p(\boldsymbol{\theta}|\mathcal{D})$ analytically, doing so is complicated. In practice, we proceed computationally by generating samples from $p(\sigma^2|\mathbf{y})$ and then $p(\boldsymbol{\theta}|\mathbf{y}, \sigma^2)$. With this sampling approach we can also perform prediction for a given input vector \mathbf{x}_{n+1} of by producing samples from $p(y_{n+1}|\boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(\mathbf{x}_{n+1}^T \boldsymbol{\theta}, \sigma^2)$, and answer question such as finding $p(y_{n+1} > a|\boldsymbol{\theta}, \sigma^2)$ for a given constant a .

In this chapter, we will discuss methods for generating samples from a distribution $p(\boldsymbol{\theta})$ which we can only compute up to a multiplicative constant. The approach is identical for conditional distributions such as $p(\boldsymbol{\theta}|\mathcal{D})$. To emphasize the fact that the constant may not be known, we use p to refer to the true distribution and q to the “distribution” without the constant. We will use \mathbb{E}_p to denote expectation with respect to distribution p . For a non-normalized distribution q we define $\mathbb{E}_q = \mathbb{E}_p$.

15.2 Basic Sampling Techniques

In this section, we will review some basic but useful sampling techniques.

15.2.1 Deterministic Integration

This method is not actually a sampling method but rather tries to approximate the expected value by approximating the corresponding integral over a grid,

$$\mathbb{E}_q[h(\boldsymbol{\theta})] = \int h(\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta} \simeq \frac{\sum_{i=1}^N h(\boldsymbol{\theta}_i)q(\boldsymbol{\theta}_i)}{\sum_{i=1}^N q(\boldsymbol{\theta}_i)},$$

where $\boldsymbol{\theta}_i$ form a uniform grid covering the support of q . This method becomes computationally prohibitive if the number of dimensions of $\boldsymbol{\theta}$ is large.

15.2.1.1 The Inverse-CDF Method

Suppose θ is one dimensional and that we have the CDF $F(\theta)$. First, assume θ is continuous and $F(\theta)$ is invertible. Inverse-CDF sampling relies on sampling from the uniform distribution to generate samples for potentially more complex distributions. For $i = 1, \dots, N$,

1. Generate $U_i \sim \text{Uni}[0, 1]$;
2. Let $\theta_i = F^{-1}(U_i)$.

Claim: If $U \sim \text{Uni}[0, 1]$, then $\theta = F^{-1}(U)$ has CDF F . To see this observe that:

$$p(\theta \leq c) = p(F^{-1}(u) \leq c) = p(U \leq F(c)) = F(c).$$

The algorithm is slightly modified if F has discontinuities or is not invertible. Specifically, we define $F^{-1}(u) = \min\{x : F(x) \geq u\}$.

15.2.2 Rejection Sampling

In rejection sampling, to produce samples for a distribution q , we first produce samples from another distribution g but then only keep some of the samples produced in a way that the resulting distribution is q . The distribution g needs to satisfy

$$\begin{aligned} g(\boldsymbol{\theta}) &> 0, & \text{if } q(\boldsymbol{\theta}) > 0, \\ q(\boldsymbol{\theta}) &\leq M g(\boldsymbol{\theta}) & \text{for some known } M \text{ and for all } \boldsymbol{\theta}. \end{aligned}$$

We also need to sample $u \sim \text{Uni}(0, 1)$.

Rejection Sampling

1. Sample $\boldsymbol{\theta}' \sim g$.

2. Sample $U \sim \text{Uni}(0, 1)$.

3. $\theta \leftarrow \theta'$ if $U \leq \frac{q(\theta')}{Mg(\theta')}$. (Accept θ' as a new sample if $U \leq \frac{q(\theta')}{Mg(\theta')}$; else reject the sample.)

We define the normalizing constants for the distribution,

$$Z_q = \int q(\theta) d\theta, \quad Z_g = \int g(\theta) d\theta.$$

Note that the probability of a sample being accepted is

$$\begin{aligned} p(\text{accepted}) &= \int p(\theta', \text{accepted}) d\theta' = \int p(\theta') p(\text{accepted}|\theta') d\theta' \\ &= \int \frac{g(\theta')}{Z_g} \cdot \frac{q(\theta')}{Mg(\theta')} d\theta' = \frac{Z_q}{MZ_g}. \end{aligned}$$

Let us now find the distribution for an accepted sample,

$$\begin{aligned} p(\theta) &= p(\theta'|\text{accepted}) \\ &= \frac{p(\theta') p(\text{accepted}|\theta')}{p(\text{accepted})} \\ &= \frac{\frac{g(\theta')}{Z_g} \cdot \frac{q(\theta')}{Mg(\theta')}}{\frac{Z_q}{MZ_g}} \\ &= \frac{q(\theta')}{Z_q}, \end{aligned}$$

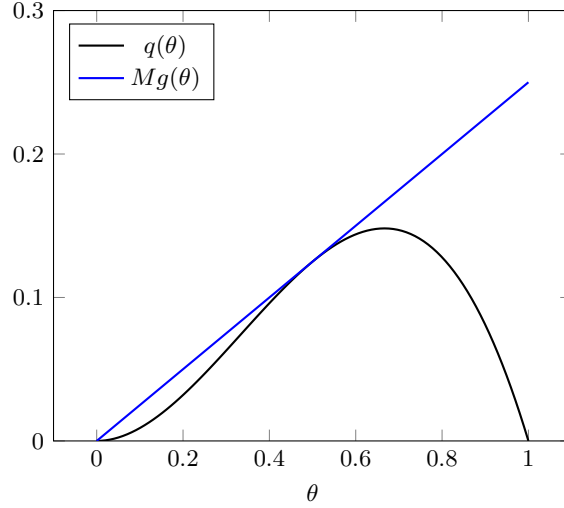
which is the desired distribution.

Rejection sampling does not take advantage of all the samples, unlike importance sampling that we will see next, so in some sense it is inefficient. In particular, if $Z_q = Z_g = 1$, then only a fraction of $\frac{1}{M}$ of the samples will be accepted. If M is large, i.e., g is not a good match for q , then we lose a lot of samples. But rejection sampling has a very important property: it is *self-evaluating*. If we are doing poorly, it is easy to find out by considering the number of samples that are rejected. This is a property that importance sampling lacks.

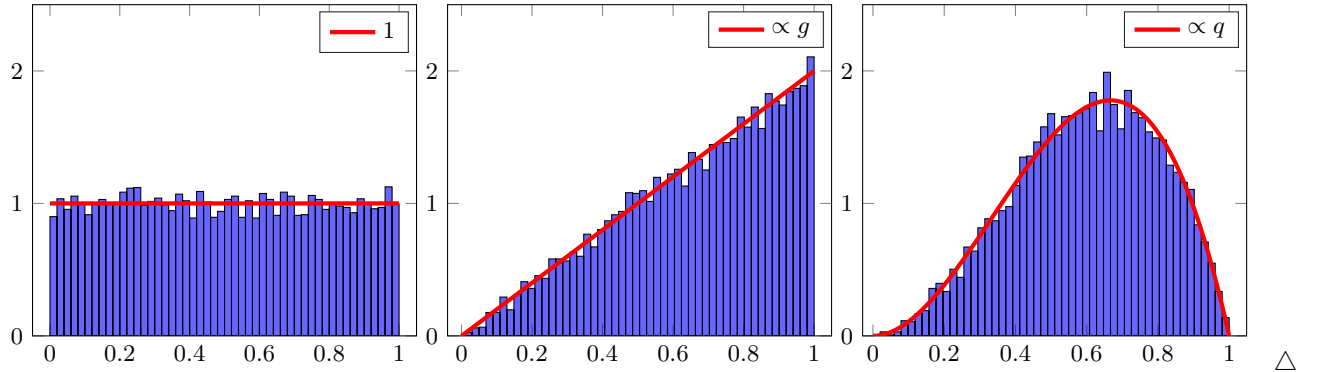
Example 15.1. Suppose we need to sample from $\text{Beta}(3, 2)$ so we let $q(\theta) = \theta^2(1 - \theta)$. We would like to do this by sampling from $g(\theta) = \theta$, which we can do using inverse-CDF sampling. First, let us find the required value for M . Observe that

$$Mg(\theta) \geq q(\theta) \iff M\theta \geq \theta^2(1 - \theta) \iff M \geq \theta(1 - \theta).$$

So the smallest valid value for M is $1/4$, which is what we will choose. Note that in practice, we don't need to find the smallest possible M . For example, here we could argue that the $\theta(1 - \theta) \leq 1$ and so it would have been sufficient to let $M = 1$. The plots of q, g are shown below.



To generate samples, first we generate samples from $\text{Uni}(0, 1)$, obtaining $S_1 = \{x_1, \dots, x_N\}$. To generate samples from g , we use the inverse CDF method. The CDF of g is θ^2 and its inverse is $\sqrt{\theta}$. So, our samples become $S_2 = \{\theta'_1, \dots, \theta'_N\}$, where $\theta'_i = \sqrt{x_i}$. We then accept/reject these based on the rejection sampling rule to obtain S_3 , which are samples with distribution q . Specifically, for a sample θ'_i , we accept it with probability $4\theta'_i(1 - \theta'_i)$. Note that this step again requires generating uniform samples, from $\text{Uni}(0, 1)$. The graphs below show histograms for x_i , θ'_i and θ_i , as well as the corresponding normalized pdfs. The histograms are normalized so that they are valid pdfs. In this experiment, out of the $N = 1000$ generated samples, 6692 were accepted.



15.2.3 Importance Sampling

Again, suppose we are interested in finding

$$\mathbb{E}_q[h(\Theta)],$$

where \mathbb{E}_q denotes expectation with respect to distribution q . Now if q is a complicated distribution, we may have a hard time sampling from it. Even if we can sample from q , another issue may arise. The values of θ such that $h(\theta)q(\theta)$ are large contribute to the expectation significantly. But $h(\theta)q(\theta)$ may be large in places where $q(\theta)$ is small. So unless we generate a lot of samples, we may not produce one for which $h(\theta)q(\theta)$ is large, and thus miss significant contributions to the expectation from such points.

Suppose we have a second (possibly unnormalized) distribution $g(\theta)$, which is simpler and from which we

can produce samples. Ideally, $g(\boldsymbol{\theta})$ is large if $q(\boldsymbol{\theta})h(\boldsymbol{\theta})$ is large. We have

$$\begin{aligned}\mathbb{E}_q[h(\boldsymbol{\theta})] &= \frac{\int h(\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta}}{\int q(\boldsymbol{\theta})d\boldsymbol{\theta}} \\ &= \frac{\int h(\boldsymbol{\theta})[q(\boldsymbol{\theta})/g(\boldsymbol{\theta})]g(\boldsymbol{\theta})d\boldsymbol{\theta}}{\int [q(\boldsymbol{\theta})/g(\boldsymbol{\theta})]g(\boldsymbol{\theta})d\boldsymbol{\theta}} \\ &= \frac{\mathbb{E}_g[h(\boldsymbol{\theta})(q(\boldsymbol{\theta})/g(\boldsymbol{\theta}))]}{\mathbb{E}_g[q(\boldsymbol{\theta})/g(\boldsymbol{\theta})]}.\end{aligned}$$

So we have converted the problem into expectation with respect to g . Define $w(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}$ as the importance weight or ratio at $\boldsymbol{\theta}$. Then we can estimate $\mathbb{E}_q[h(\boldsymbol{\theta})]$ as

$$\mathbb{E}_q[h(\boldsymbol{\theta})] = \frac{\mathbb{E}_g[h(\boldsymbol{\theta})w(\boldsymbol{\theta})]}{\mathbb{E}_g[w(\boldsymbol{\theta})]} \simeq \frac{\frac{1}{N} \sum_{i=1}^N h(\boldsymbol{\theta}_i)w(\boldsymbol{\theta}_i)}{\frac{1}{N} \sum_{i=1}^N w(\boldsymbol{\theta}_i)}, \quad \text{with } \boldsymbol{\theta}_i \sim g(\boldsymbol{\theta}),$$

by producing samples from g rather than q .

Of course, if g is small where $h \times q$ is large, we may miss samples for which $h(\boldsymbol{\theta})g(\boldsymbol{\theta})$ makes significant contributions to the expectation; and this is a drawback of importance sampling.

Example 15.2. Let $h(x) = 1 - x$ and $q(x) = x$ for $0 \leq x \leq 1$. Then

$$\mathbb{E}_q[h(x)] = \int_0^1 (1-x)(2x)dx = (x^2 - 2x^3/3)_0^1 = 1/3.$$

To estimate this computationally, let $g(x) = 1$. The weights become $w(x) = x$. Generating $N = 100$ samples $x_i \sim \text{Uni}(0, 1)$ using MATLAB, we find

$$\mathbb{E}_q[h(x)] \simeq \frac{\sum_{i=1}^N (1-x_i)x_i}{\sum_{i=1}^N x_i} = 0.34623,$$

which is close to $0.33 \dots$. Of course, for such a simple q we wouldn't resort to importance sampling. \triangle

15.3 Metropolis Monte Carlo

To generate samples from a distribution $p(\boldsymbol{\theta})$, one possible approach is to design a Markov chain whose state space includes all possible values for $\boldsymbol{\theta}$ and its stationary distribution $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\theta})$ is equal to the target distribution $p(\boldsymbol{\theta})$. *In the long term, the number of times that the MC spends in a given state is proportional to the probability of that state.* Hence, we can generate samples from the states of the Markov process by letting it run for a long time and record the states that are visited as samples. The distribution of these samples is approximately the same as $\boldsymbol{\sigma}$ and thus the same as $p(\boldsymbol{\theta})$. This is called Markov Chain Monte Carlo (MCMC).

In this section, we present elegant solutions to the challenging problem of finding a MC satisfying $\boldsymbol{\sigma}(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$. In fact, these methods only need $q \propto p$. While MCs can generate samples with the same distribution, we note that the samples are not independent.

We will first discuss the Metropolis algorithm. This algorithm requires a *jump distribution*, $J(\boldsymbol{\theta}'|\boldsymbol{\theta})$, which proposes a new state $\boldsymbol{\theta}'$ given that we are in state $\boldsymbol{\theta}$. We then either move to $\boldsymbol{\theta}'$ or stay at the current state. The jump distribution is chosen in a way that it guarantees $\boldsymbol{\sigma}(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$. We next describe the Metropolis algorithm more formally. We assume θ is one dimensional for simplicity of notation but this is not a requirement.

Metropolis Algorithm:

1. Choose θ^0 such that $q(\theta) > 0$.

2. For $t = 1, 2, 3, \dots$, do

- (a) Generate a proposal θ' based on the jump distribution $J(\theta'|\theta^{t-1})$.
- (b) Move to θ' with probability

$$r = \frac{q(\theta')}{q(\theta^{t-1})}.$$

Specifically:

- i. Generate $u \sim \text{Uni}[0, 1]$.
- ii. The next state of the MC, θ^t , is given by

$$\theta^t = \begin{cases} \theta', & u \leq r \\ \theta^{t-1}, & u > r \end{cases} \quad (15.1)$$

The transition probabilities. The rule (15.1) has interesting implications. Note that if $r > 1$, or equivalently if $q(\theta') > q(\theta^{t-1})$, then we will definitely move to θ' . Otherwise, we move to θ' with probability $r = \frac{q(\theta')}{q(\theta^{t-1})} = \frac{p(\theta')}{p(\theta^{t-1})}$. Define $D = \{\theta : p(\theta) > 0\}$ as the set of all possible values of θ based on target distribution p . If the transition probability of $\theta_a \rightarrow \theta_b$ in the MC is denoted by $\Pr(\theta_a \rightarrow \theta_b)$, we have

$$\Pr(\theta_a \rightarrow \theta_b) = J(\theta_b|\theta_a) \min\left(1, \frac{p(\theta_b)}{p(\theta_a)}\right).$$

The jump distribution. In the Metropolis algorithm, it is not necessary for the jump distribution to have $p(\theta)$ as a stationary distribution. However, the jump distribution $J(\theta'|\theta^{t-1})$ should satisfy certain constraints, discussed below.

1. *Reachability.* To ensure that the MC is regular, we require that

$$J(\theta'|\theta) > 0, \quad \forall \theta, \theta' \in D. \quad (15.2)$$

2. *Symmetry.* For $\theta_a, \theta_b \in D$, the detailed balance property with distribution $\pi(\theta) = p(\theta)$ can be written as

$$p(\theta_a) \Pr(\theta_a \rightarrow \theta_b) = p(\theta_b) \Pr(\theta_b \rightarrow \theta_a).$$

Assume without loss of generality that $p(\theta_a) < p(\theta_b)$. Then, the DBP can be written as

$$p(\theta_a) J(\theta_b|\theta_a) = p(\theta_b) J(\theta_a|\theta_b) \frac{p(\theta_a)}{p(\theta_b)}.$$

which is satisfied if the jump distribution is symmetric, i.e.,

$$J(\theta'|\theta) = J(\theta|\theta'), \quad \forall \theta, \theta' \in D. \quad (15.3)$$

If the jump distribution satisfies (15.2) and (15.3), then the MC is regular and $\sigma(\theta) = p(\theta)$ satisfies the DBP. Hence, $p(\theta)$ is the unique stationary distribution of the Markov chain.

Example 15.3. Consider a Bayesian regression problem where the data as in Figure 15.1a. The data is generated using the distribution

$$y_i|\theta, \sigma \sim \mathcal{N}(\theta x_i, \sigma^2),$$

where the true values are $\theta = 2, \sigma = 1$. The figure provides a plot for the samples $\mathcal{D} = \{(x_i, y_i)_{i=1}^N\}$, where $x_i = 0, 0.1, 0.2, \dots, 5$ as well as the line $y = 2x$.

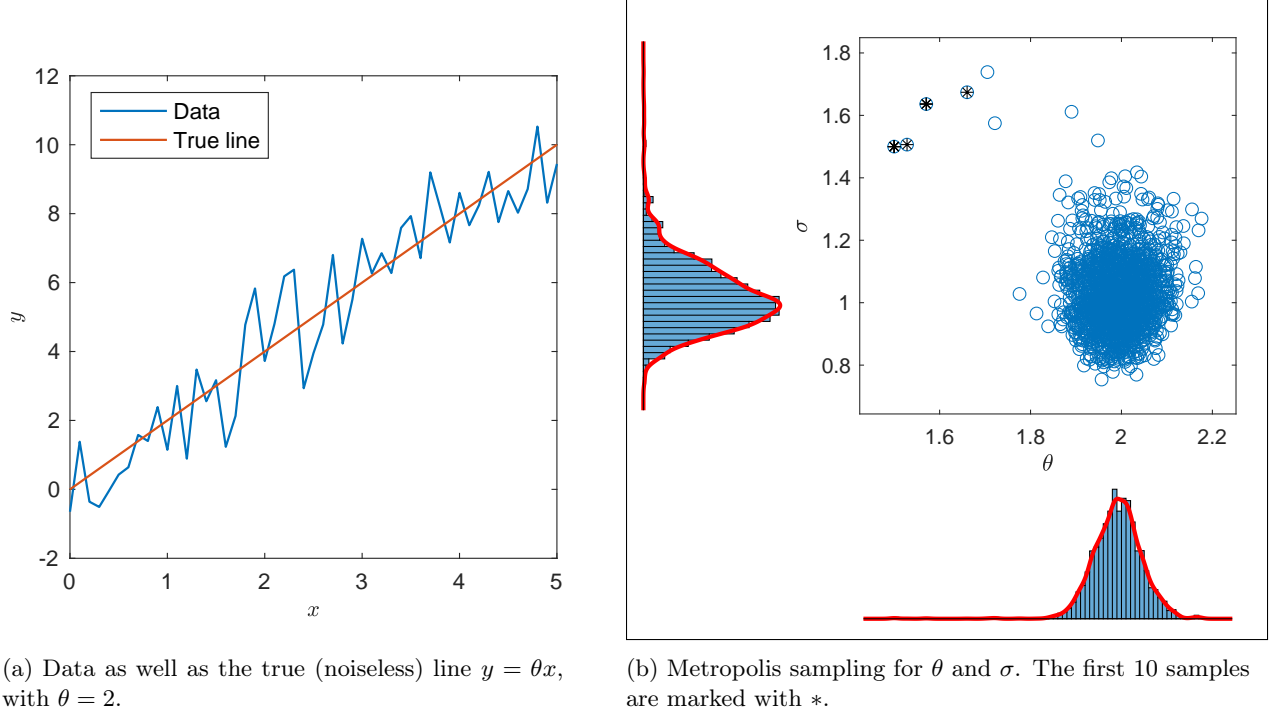


Figure 15.1: Metropolis sampling for 1-D Bayesian linear regression.

As we have seen, the Bayesian posteriors for this problem are rather complicated. But it is straightforward to obtain estimates using Metropolis sampling. Assuming the prior $p(\theta, \sigma) \propto 1/\sigma^2$, the posterior is

$$\ln p(\theta, \sigma | \mathcal{D}) \propto -(2 + N) \ln \sigma - \frac{1}{2\sigma^2} (\mathbf{y} - \theta \mathbf{x})' (\mathbf{y} - \theta \mathbf{x}).$$

We use log-probability because probabilities may be very small and for numerical precision, it is better to work with logs. We can convert these to probabilities if we need to. But in this problem, since we are only interested in the samples, we can keep probabilities in log scale.

The samples produced by Metropolis are given in Figure 15.1b. As the jump proposal, we use a product of independent Gaussians:

$$\begin{aligned} J(\theta', \sigma' | \theta, \sigma) &= J(\theta' | \theta) J(\sigma' | \sigma), \\ J(\theta' | \theta) &\sim \mathcal{N}(\theta, 0.01), \\ J(\sigma' | \sigma) &\sim \mathcal{N}(\sigma, 0.01). \end{aligned}$$

Based on these samples, the posterior mean for θ is 1.9911 with posterior std 0.055163. The posterior mean for σ is found to be 1.0198. It is also worth noting that the ML estimate for θ is 1.9896. In this example, the estimates are very accurate, which is probably the result of a combination of low noise in the data and chance. \triangle

Metropolis-Hastings algorithm. We can eliminate the symmetry property of the jump distribution if we modify r in the Metropolis algorithm as

$$r = \frac{p(\theta')/J(\theta'|\theta^{t-1})}{p(\theta^{t-1})/J(\theta^{t-1}|\theta')}.$$

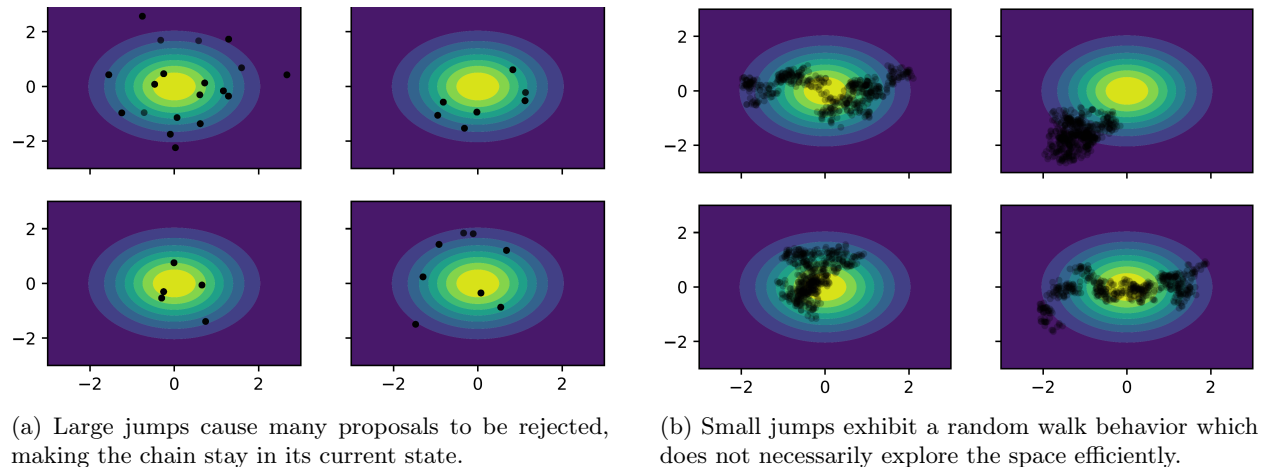


Figure 15.2: Metropolis sampling with poorly designed jump distributions (4 runs for each case).

Exercise 15.4. Prove that with this definition for r , DBP holds even if J is not symmetric. \triangle

Sampling from a MC. Ideally, we should keep only one sample from every m samples for m “large enough” to ensure that the samples are nearly independent. However, there are two issues here:

- It is not easy to determine how large is “large enough.”
- If m is too large, the process is inefficient.

However, as long as the empirical distribution (e.g., the histogram) is close to the target distribution, we are not too concerned about independence and the sampling algorithm does not need to throw away any samples, since the order of the samples is not considered. Because the samples at the start states don’t satisfy the stationary distribution, it is a good idea to discard the samples produced by the chain at the beginning.

Strong dependence between samples that are close to each other in time could be problematic. For example, suppose we get N samples from a chain whose samples are strongly dependent during intervals of duration not much smaller than N . While each of the N samples may individually have the target distribution, due to strong dependence they all may be from the same area of the probability space and thus the empirical distribution may not look like the target distribution, necessitating obtaining a larger number of samples. This problem can be caused by choosing a poor jump distribution as discussed next.

The Jump distribution. The jumps should be neither too small nor too large!

- When the jumps are large, a large number of proposals will be rejected (we’ll stay in the current state) because it is likely that with a large jump, we’ll end up with a low probability proposal. In this case, strong dependence manifests as many samples being likely to be equal. An example is shown in Figure 15.2a, where most of the proposals are rejected, resulting in a small number of distinct samples.
- If the jumps are too small, the sampling process is similar to a random walk, because most proposals are accepted but we move only a small step. This means that the MC does not explore the probability space efficiently, again necessitating a large number of samples. An example is given in Figure 15.2b. To see why random walk behavior is not good, consider a random walk with step size ε . How far from the starting point will we be after T steps? For the random walk, let X_i be the movement in one step:

$$X_i = \begin{cases} \varepsilon, & \text{with } p = \frac{1}{2}; \\ -\varepsilon, & \text{with } p = \frac{1}{2}. \end{cases}$$

After T steps, the expected distance $L = \mathbb{E}\left[\left|\sum_{i=1}^T X_i\right|\right]$ is difficult to find. But we can approximate the distance as

$$L^2 \simeq \mathbb{E}\left[\left(\sum_{i=1}^T X_i\right)^2\right] = T\varepsilon^2 \text{ (exercise).}$$

In conclusion, after T steps, we will be approximately at distance $\sqrt{T}\varepsilon$, which is a case of diminishing returns, and not very efficient. In other words, we need $\frac{L^2}{\varepsilon^2}$ steps to move distance L . In the context of MCMC, this means if the probability space has a dimension in which there is a high probability region with length L , we need to run the chain for *at least* $\frac{L^2}{\varepsilon^2}$ steps.

15.4 Gibbs Sampling

At each iteration of the Metropolis algorithm, all the components of $\boldsymbol{\theta}$ are updated at the same time. In Gibbs sampling, for $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_d)$, at each iteration, components are updated one-by-one as

$$\theta_j^t \sim p(\theta_j | \theta_1^t, \dots, \theta_{(j-1)}^t, \theta_{(j+1)}^{(t-1)}, \dots, \theta_d^{(t-1)}), \quad \text{for } j = 1, \dots, d.$$

Gibbs sampling may be simpler and more efficient than Metropolis sampling if the joint distribution is too complicated but we can easily sample from the conditional distributions. The components do not need to be one-dimensional necessarily; we can group several dimensions and update each the dimensions in each group simultaneously.

Example 15.5. Suppose $\boldsymbol{\theta} = (\theta_1, \theta_2)$ and the observation $\mathbf{y} = (y_1, y_2)$ are related by the likelihood

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} | \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right),$$

with the prior $p(\boldsymbol{\theta}) \propto 1$. The posterior distribution $p(\boldsymbol{\theta} | \mathbf{y})$ is:

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} | \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

We can use Gibbs sampling to produce samples for $\boldsymbol{\theta} | \mathbf{y}$. The following fact is of use:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}\right) \Rightarrow x_1 | x_2 \sim \mathcal{N}\left(\mu_1 + \frac{\rho\sigma_1}{\sigma_2}(x_2 - \mu_2), (1 - \rho^2)\sigma_1^2\right)$$

Then, in the t -th iteration, the θ_1^t is sampled by

$$\theta_1^t | \theta_2^{t-1} \sim \mathcal{N}(y_1 + \rho(\theta_2^{t-1} - y_2), (1 - \rho^2)).$$

Similarly, the θ_2^t can be updated by

$$\theta_2^t | \theta_1^t \sim \mathcal{N}(y_2 + \rho(\theta_1^t - y_1), (1 - \rho^2)).$$

So we produce a new sample using 1-D distributions. △

Stationary distribution. We prove that Gibbs sampling (with a caveat) satisfies the DBP with distribution $p(\boldsymbol{\theta})$.

Suppose we are in state $\boldsymbol{\theta}$ and we update the j th component to get $\boldsymbol{\theta}'$. We have

$$\theta'_j \sim p(\theta'_j | \boldsymbol{\theta}_{-j}),$$

where $\boldsymbol{\theta}_{-j} = (\theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_d)$. Furthermore, the $\boldsymbol{\theta}'_{-j} = \boldsymbol{\theta}_{-j}$.

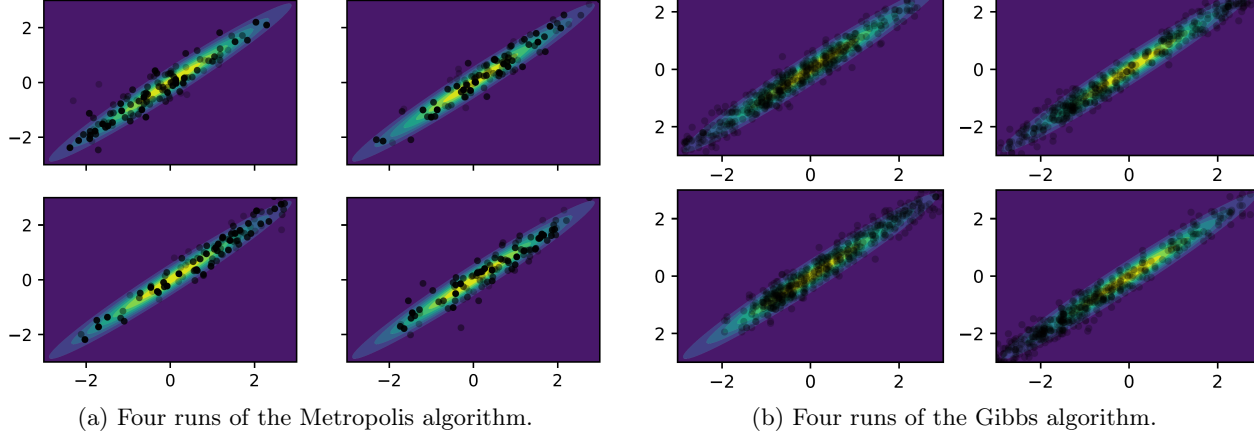


Figure 15.3: Metropolis and Gibbs sampling for highly correlated dimensions. Many proposals for Metropolis are rejected.

To prove DBP for this step, we need to prove $p(\theta) \Pr(\theta \rightarrow \theta') = p(\theta') \Pr(\theta' \rightarrow \theta)$, which holds since

$$\begin{aligned} p(\theta) \Pr(\theta \rightarrow \theta') &= p(\theta) p(\theta'_j | \theta_{-j}) = p(\theta_{-j}) p(\theta_j | \theta_{-j}) p(\theta'_j | \theta_{-j}), \\ p(\theta') \Pr(\theta' \rightarrow \theta) &= p(\theta') p(\theta_j | \theta'_{-j}) = p(\theta_{-j}) p(\theta'_j | \theta_{-j}) p(\theta_j | \theta_{-j}). \end{aligned}$$

Since the DBP holds for each sub-iteration. We can use this observation to prove that Gibbs sampling works if we choose a component to update at random to produce a new sample or if we update all components in a random order and after a full cycle produce a new sample. Updating the components in a predetermined order works in practice.

Gibbs sampling can be viewed as a special case of Metropolis-Hastings in which the proposal is always accepted and where we don't need to design a jump distribution. Gibbs can use the current state to provide better proposals. An example is shown in Figure 15.3. Here, the dimensions are highly correlated, with most of the probability concentrated in a narrow region. Because of this, many of the Metropolis proposals are rejected. Gibbs, which produces samples based on the conditional distribution given the current state, does not suffer from this.

Note that θ_j may be independent from some dimensions of θ_{-j} given others. In particular, if θ denotes the nodes in a graphical model, given its Markov blanket, θ_j is independent of other elements of θ_{-j} .

15.5 Hamiltonian Monte Carlo **

One problem with the Metropolis algorithm is that, in certain situations, the proposed θ' by the jump distribution may be rejected too often because $p(\theta')$ is much smaller than $p(\theta^{t-1})$, in which case we will let $\theta^t = \theta^{t-1}$. While the stationary distribution is still $p(\theta)$, too many rejection means that it will take a long time to get a sample whose empirical distribution is close to the true distribution.

Let us write our target distribution $p(\theta)$ as

$$p(\theta) \propto e^{-E(\theta)}$$

and suppose that we can also compute $\nabla_{\theta} E(\theta)$. Note that as $E(\theta)$ decreases, the probability increases.

Can we use the fact that we know the gradient to increase the chance of proposals being accepted? At first glance it may seem that we could let $\theta^t = \theta^{t-1} - \epsilon \nabla E(\theta)$, similar to gradient descent. But this is a deterministic path rather than a probabilistic MC.

A bit of (questionable) physics. Instead, we use an idea from Hamiltonian Mechanics. We can think of θ as location and of $E(\theta)$ as potential energy. Note that lower potential has a higher probability (a river flows down the valley). Now let us also include momentum (speed) ϕ , which has the same number of dimensions as θ , in our formulation and define the total energy as

$$H(\theta, \phi) = E(\theta) + K(\phi),$$

where $K(\phi)$ is the Kinetic energy

$$K(\phi) = \frac{1}{2} \phi^T \phi.$$

With this physical viewpoint, Hamilton's equations describing the motion of an object with position θ and momentum ϕ are

$$\begin{aligned}\dot{\theta} &= \frac{\partial H}{\partial \phi} = \phi \\ \dot{\phi} &= -\frac{\partial H}{\partial \theta} = -\nabla_{\theta} E(\theta)\end{aligned}$$

(A more familiar form of these equations are obtained by representing position with x and speed with v . Then, $\dot{x} = v, \dot{v} = -\nabla_x E(x)$.) It can then be shown that H , the total energy, stays constant in time.

Back to Sampling. Instead of sampling from $p(\theta)$, let us define and sample from

$$p(\theta, \phi) \propto e^{-H(\theta, \phi)} = e^{-E(\theta)} e^{-K(\phi)},$$

where $K(\phi) = \frac{1}{2} \phi^T \phi$. We will then discard the ϕ component of the samples.

The Hamiltonian Monte Carlo Algorithm is as follows:

1. Randomly choose θ^0 from the domain and choose ϕ^0 arbitrarily.
2. For $t = 1, 2, \dots$, do
 - (a) Pick a random momentum ϕ' according to the distribution $p(\phi) \propto e^{-K(\phi)}$.
 - (b) Starting from (θ^{t-1}, ϕ') , simulate the dynamic system for a certain amount of time according to

$$\begin{aligned}\dot{\theta} &= \phi, \\ \dot{\phi} &= -\nabla_{\theta} E(\theta).\end{aligned}$$

The final values of (θ, ϕ) are the new sample, (θ^t, ϕ^t) .

It can be shown this process leads to a Markov chain whose stationary distribution is $p(\theta, \phi)$. This hinges on step (a) being reversible and step (b) keeping the Hamiltonian and thus the probability constant.

In practice however, we cannot have a perfect simulation. So instead of step (b) above, we perform the following:

(2.b)' For $i = 1, 2, \dots, L$, perform the following steps, called leapfrog updates:

$$\begin{aligned}\phi &\leftarrow \phi - \frac{1}{2} \epsilon \nabla E(\theta) \\ \theta &\leftarrow \theta + \epsilon \phi \\ \phi &\leftarrow \phi - \frac{1}{2} \epsilon \nabla E(\theta)\end{aligned}$$

Let the final $(\boldsymbol{\theta}, \phi)$ be denoted by $(\boldsymbol{\theta}^*, \phi^*)$. If our simulation is perfect, then this can be accepted as the new state. But because $\epsilon > 0$, we have to perform an accept/reject check similar to Metropolis. That is, we let

$$r = \frac{e^{-H(\boldsymbol{\theta}^*, \phi^*)}}{e^{-H(\boldsymbol{\theta}^{t-1}, \phi^{t-1})}}.$$

If $r \geq 1$, we let $(\boldsymbol{\theta}^t, \phi^t) = (\boldsymbol{\theta}^*, \phi^*)$. If $r \leq 1$, then we let $(\boldsymbol{\theta}^t, \phi^t) = (\boldsymbol{\theta}^*, \phi^*)$ with probability r and with probability $1 - r$, we let $(\boldsymbol{\theta}^t, \phi^t) = (\boldsymbol{\theta}^{t-1}, \phi^{t-1})$.

If ϵ is too large, our simulation will be too rough, leading to many rejections. In this case, we decrease ϵ and increase L . On the other hand, if nearly all proposals are accepted, it may be a sign of being too conservative and not exploring the state space as fast as we can, in which case we can be more efficient by increasing ϵ and decreasing L .

Chapter 16

Variational Inference *

Consider the inference problem where our objective is to compute the probability distribution of unknown parameters (Bayesian inference) or latent variables \mathbf{Z} , conditioned on observations $\mathbf{X} = \mathbf{x}$. Mathematically, this is represented as

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}} \quad (16.1)$$

Here, the joint distribution $p(\mathbf{x}, \mathbf{z})$ is typically known, either explicitly defined by the model or through combining $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z})$. However, the integral $\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$, is challenging to compute, especially in high-dimensional spaces, as it sums over all possible configurations of the latent variables \mathbf{z} .

While Monte-Carlo methods enable sampling from the target distribution $p(\mathbf{z}|\mathbf{x})$, they are often computationally intensive. Variational inference, on the other hand, offers a computationally efficient alternative by approximating $p(\mathbf{z}|\mathbf{x})$ through optimization.

Variational inference simplifies the process by approximating the complex posterior distribution $p(\mathbf{z}|\mathbf{x})$ with a more tractable distribution $q(\mathbf{z})$ from a predefined family Q . The objective is to identify $q^* \in Q$ that is closest to $p(\mathbf{z}|\mathbf{x})$ as measured by the KL-divergence, $D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$. The optimization problem thus formulated is

$$q^* = \arg \min_{q \in Q} D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})), \quad (16.2)$$

where

$$D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} d\mathbf{z} = \mathbb{E}_q \left[\log \frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{x})} \right], \quad (16.3)$$

with \mathbb{E}_q denoting that the expectation assumes distribution q for \mathbf{Z} . In variational inference, minimizing $D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$ is preferred over minimizing $D_{KL}(p(\mathbf{z}|\mathbf{x})||q(\mathbf{z}))$ because the latter is typically intractable.

Evidence Lower Bound (ELBO): Reformulating the KL-divergence yields:

$$D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} d\mathbf{z} \quad (16.4)$$

$$= \int q(\mathbf{z}) \log \frac{q(\mathbf{z})p(\mathbf{x})}{p(\mathbf{z}, \mathbf{x})} d\mathbf{z} \quad (16.5)$$

$$= \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z}, \mathbf{x})} d\mathbf{z} + \log p(\mathbf{x}). \quad (16.6)$$

Minimizing the KL-divergence, in this context, is equivalent to maximizing the Evidence Lower Bound (ELBO), defined as:

$$\mathcal{L}(q) = \log p(\mathbf{x}) - D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) \quad (16.7)$$

$$= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} d\mathbf{z} \quad (16.8)$$

$$= \int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} + H(q). \quad (16.9)$$

Note that since $D_{KL} \geq 0$, ELBO is no greater than $\log p(\mathbf{x})$, i.e., $\log p(\mathbf{x}) \geq \mathcal{L}(q)$. As \mathbf{x} is sometimes referred to as *evidence*, this observation motivates the name “Evidence Lower Bound” or ELBO.

Why does maximizing $\mathcal{L}(q)$ make sense? Let us inspect each term in $\mathcal{L}(q)$. The distribution q that maximizes $\int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ is the one that puts all the probability mass on $\hat{\mathbf{z}} = \arg \max_{\mathbf{z}} \log p(\mathbf{x}, \mathbf{z}) = \arg \max_{\mathbf{z}} \log p(\mathbf{z}|\mathbf{x})$, i.e., the Bayesian mode point estimator. This is a degenerate distribution that tells us that \mathbf{Z} is equal to $\hat{\mathbf{z}}$ with probability 1. To balance this overconfidence, q that maximizes the second term, $H(q)$, must be high-entropy.

Alternatively, we can rewrite ELBO in the following way to gain more intuition about why maximizing the ELBO gives a reasonable approximation [1]. By (16.8),

$$\mathcal{L}(q) = \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} + \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z} \quad (16.10)$$

$$= -D_{KL}(q(\mathbf{z})||p(\mathbf{z})) + \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) d\mathbf{z}. \quad (16.11)$$

The two terms are now the negative KL divergence between $q(\mathbf{z})$ and the prior $p(\mathbf{z})$, and the expected likelihood assuming $\mathbf{Z} \sim q(\mathbf{z})$. For the divergence to be small, $q(\mathbf{z})$ is encouraged to be close to the prior. On the other hand, for the expected likelihood to be large, $q(\mathbf{z})$ should assign more mass to \mathbf{Z} that can better explain our observed data \mathbf{x} , i.e., $\arg \max_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})$. So, the solution balances closeness to the prior with the maximum-likelihood solution, similar to the true posterior.

16.1 Background on Calculus of Variations

Before proceeding further, we review the calculus of variations, a mathematical area that focuses on determining functions that optimize a *functional*. A functional $F : \mathcal{F} \rightarrow \mathbb{R}$ is a function that maps the elements of a specified family \mathcal{F} of functions to \mathbb{R} . In variational inference, the functional is $D_{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))$, which assigns a real number to each choice of $q \in \mathcal{Q}$.

A simple class of functionals are those of the form $\int_S J(x, f(x)) dx$ for some function J and set S . For entropy,

$$H[p] = \int_{\mathcal{X}} p(x) \log \frac{1}{p(x)} dx,$$

we have $J(x, p) = J(p) = p \log \frac{1}{p}$. For KL divergence

$$D_{KL}(q||p) = \int q(x) \log \frac{q(x)}{p(x)},$$

viewed as a functional of q and for fixed p , we have $J(x, q) = q \log \frac{q}{p(x)}$.

The **functional differential** of F at f in the direction of ϕ is defined as

$$\lim_{\epsilon \rightarrow 0} \frac{F[f + \epsilon\phi] - F[f]}{\epsilon} = \left. \frac{dF[f + \epsilon\phi]}{d\epsilon} \right|_{\epsilon=0}$$

This tells us how the functional changes at f if it is perturbed by moving infinitesimally in the “direction”

of ϕ . This quantity is useful for optimizing a functional, that is, for finding a function that maximizes or minimizes the functional. We will explore this concept through an analogy to multivariate calculus.

Analogy from multivariate calculus: Consider $g : \mathbb{R}^n \rightarrow \mathbb{R}$, a function that assigns to each vector $\mathbf{x} \in \mathbb{R}^n$ a real number $g(\mathbf{x})$. If we are at \mathbf{x} , how does g change if we move in the direction of some vector \mathbf{v} ? The change in g can be quantified by

$$\lim_{\epsilon \rightarrow 0} \frac{g(\mathbf{x} + \epsilon \mathbf{v}) - g(\mathbf{x})}{\epsilon}$$

This is useful for optimizing g . For instance, if it is 0 for all \mathbf{v} , then we are at a local extremum. But for each vector \mathbf{v} , we would need to compute it from scratch. To address this, we define the gradient

$$\nabla g = \left(\lim_{\epsilon \rightarrow 0} \frac{g(\mathbf{x} + \epsilon \mathbf{i}_1) - g(\mathbf{x})}{\epsilon}, \dots, \lim_{\epsilon \rightarrow 0} \frac{g(\mathbf{x} + \epsilon \mathbf{i}_n) - g(\mathbf{x})}{\epsilon} \right),$$

where $\mathbf{i}_1, \dots, \mathbf{i}_n$ are unit vectors in the standard basis. Then we can find the rate of change for any vector \mathbf{v} as the inner product of the gradient and the \mathbf{v}

$$\lim_{\epsilon \rightarrow 0} \frac{g(\mathbf{x} + \epsilon \mathbf{v}) - g(\mathbf{x})}{\epsilon} = \langle \nabla g, \mathbf{v} \rangle.$$

Furthermore, for small ϵ ,

$$g(\mathbf{x} + \epsilon \mathbf{v}) \simeq g(\mathbf{x}) + \epsilon \langle \nabla g, \mathbf{v} \rangle.$$

Back to functionals: For functionals of the form $F[f] = \int_{\mathcal{S}} J(x, f(x)) dx$, we can find something similar to a gradient. Specifically, there is a function $\frac{\partial F}{\partial f}(x)$, called a **functional derivative** such that

$$\left. \frac{dF[f + \epsilon \phi]}{d\epsilon} \right|_{\epsilon=0} = \int_{\mathcal{X}} \frac{\partial F}{\partial f}(x) \phi(x) dx$$

This derivative measures how the functional $F[f]$ changes when the function f is perturbed infinitesimally at the point x .

The Defining the inner product in the space of functions as $\langle f(x), g(x) \rangle = \int_{\mathcal{X}} f(x) g(x) dx$ for some predetermined set \mathcal{X} , we can write

$$\begin{aligned} \left. \frac{dF[f + \epsilon \phi]}{d\epsilon} \right|_{\epsilon=0} &= \left\langle \frac{\partial F}{\partial f}(x), \phi(x) \right\rangle \\ F[f + \epsilon \phi] &\simeq F[f] + \epsilon \left\langle \frac{\partial F}{\partial f}(x), \phi(x) \right\rangle \end{aligned}$$

Observe that

$$\begin{aligned} \left. \frac{dF[f + \epsilon \phi]}{d\epsilon} \right|_{\epsilon=0} &= \left. \frac{d}{d\epsilon} \int J(x, f(x) + \epsilon \phi(x)) dx \right|_{\epsilon=0} \\ &= \left. \int \frac{d}{d\epsilon} J(x, f(x) + \epsilon \phi(x)) dx \right|_{\epsilon=0} \\ &= \int J_2(x, f(x)) \phi(x) dx, \end{aligned}$$

where J_2 is the partial derivative of J with respect to its second argument.

Hence,

$$\frac{\partial F}{\partial f}(x) = J_2(x, f(x))$$

Example 16.1. Let us find $\frac{\partial H[p]}{\partial p}(x)$ where H is the entropy function. Here, we have $J(x, p) = p \log \frac{1}{p}$. Hence,

$$\frac{\partial H[p]}{\partial p}(x) = \frac{\partial(p \log \frac{1}{p})}{\partial p}(x) = \log \frac{1}{p(x)} - 1.$$

△

Example 16.2. For fixed p , let us find $\frac{\partial D_{KL}(q||p)}{\partial q}(x)$. Here, we have $J(x, q) = q \log \frac{q}{p(x)}$. Hence,

$$\frac{\partial D_{KL}}{\partial q}(x) = \frac{\partial(q \log \frac{q}{p(x)})}{\partial q}(x) = 1 + \log \frac{q(x)}{p(x)}.$$

△

Optimization of Functionals: Now that we have functional derivatives, we can optimized functionals by setting the derivative to 0. When we have constrained, we can use Lagrange multipliers.

Example 16.3. We find the distribution with the highest possible entropy with variance at most σ^2 , i.e.,

$$\begin{aligned} & \text{maximize } H[p] \\ & \text{s.t. } S[p] = \int p(x) dx = 1 \\ & \quad V[p] = \int p(x) x^2 dx = 1 \end{aligned}$$

Using Lagrange multipliers:

$$\frac{\partial H}{\partial p}(x) + \lambda_1 \frac{\partial S}{\partial p}(x) + \lambda_2 \frac{\partial V}{\partial p}(x) = 0$$

Hence,

$$\log \frac{1}{p(x)} + \lambda_1 + \lambda_2 x^2 = 0 \Rightarrow p(x) = e^{\lambda_1 + \lambda_2 x^2}.$$

This is a Gaussian distribution. Since we know which Gaussian distribution has variance σ^2 , we have

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/\sigma^2}.$$

We could also find the constants by solving the constraint equations. Note that the mean is arbitrary. △

Example 16.4. For fixed p , let us find the distribution that minimizes $D_{KL}(q||p)$, i.e.,

$$\begin{aligned} & \text{maximize } D_{KL}(q||p) \\ & \text{s.t. } S[q] = \int q(x) dx = 1 \end{aligned}$$

Again, using Lagrange multipliers, we have

$$1 + \log \frac{q(x)}{p(x)} + \lambda_1 = 0 \Rightarrow q(x) = q(x) \propto p(x),$$

which, along with the constraint, leads to

$$q(x) = p(x).$$

△

16.2 Mean-field variational inference

In this chapter, we restrict the “nice” family \mathcal{Q} to be the family of distributions that factorize (being tractable is important!), i.e.,

$$q(\mathbf{z}) = \prod_{j=1}^J q_j(z_j), \quad (16.12)$$

where z_1, z_2, \dots, z_J form a partition of all hidden variables in \mathbf{z} . This is called the **mean-field** approximation and leads to

$$\mathcal{L}(q) = \int \prod_{j=1}^J q_j(z_j) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} + \sum_{j=1}^J H(q_j). \quad (16.13)$$

Coordinate ascent variational inference (CAVI)

$\mathcal{L}(q)$ is a functional of J functions. The most common way for optimizing (16.13) is coordinate ascent. In other words, we will take turns to optimize $\mathcal{L}(q)$ with respect to one component q_i while fixing the others $q_j, j \neq i$. Now, let us assume that we fix q_j for all $j \neq i$. We can write the ELBO as

$$\mathcal{L}(q) = \int \prod_{j=1}^J q_j(z_j) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} + \sum_{j=1}^J H(q_j) \quad (16.14)$$

$$= \sum_{j \neq i} H(q_j) + H(q_i) + \int q_i(z_i) \left(\int \prod_{j \neq i} q_j(z_j) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{-i} \right) dz_i \quad (16.15)$$

$$= \sum_{j \neq i} H(q_j) + H(q_i) + \int q_i(z_i) \tilde{f}_i(z_i) dz_i, \quad (16.16)$$

where $\mathbf{z}_{-i} = \{z_j\}_{j \neq i}$ and

$$\tilde{f}_i(z_i) = \int \prod_{j \neq i} q_j(z_j) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{-i} \quad (16.17)$$

$$= \int \mathbf{q}_{-i}(\mathbf{z}_{-i}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z}_{-i} \quad (16.18)$$

$$= \mathbb{E}_{\mathbf{z}_{-i} \sim \mathbf{q}_{-i}} [\log p(\mathbf{x}, \mathbf{z})], \quad (16.19)$$

where $\mathbf{q}_{-i} = \{q_j\}_{j \neq i}$.

Taking the derivative, we have

$$\frac{\partial \mathcal{L}(q)}{\partial q_i}(z_i) = \log \frac{1}{q_i(z_i)} - 1 + \tilde{f}_i(z_i) = 0 \Rightarrow q_i(z_i) \propto \exp(\tilde{f}_i(z_i)).$$

So we update q_i to

$$q_i^*(z_i) = \frac{\exp(\tilde{f}_i(z_i))}{\int \exp(\tilde{f}_i(z_i)) dz_i}. \quad (16.20)$$

So $q_i^*(z_i)$ is also a distribution over z_i and since $\tilde{f}_i(z_i)$ is a function of z_i and does not depend on q_i , neither does $\int \exp(\tilde{f}_i(z_i)) dz_i$.

We summarize the above process in the following algorithm.

Algorithm 1 Coordinate ascent variational inference (CAVI)

```

1: Input: visible variables  $\mathbf{x}$ ; latent variables  $\mathbf{z} = (z_1, \dots, z_J)$ ; joint distribution  $p(\mathbf{x}, \mathbf{z})$ ;
2: Output: an approximation for  $p(\mathbf{z}|\mathbf{x})$ ;
3: Initialize distributions  $q_1, \dots, q_J$  over  $z_1, \dots, z_J$ , respectively;
4: while not converged do
5:   for  $i = 1$  to  $J$  do
6:      $\tilde{f}_i(z_i) = \mathbb{E}_{\mathbf{z}_{-i} \sim \mathbf{q}_{-i}} [\log p(\mathbf{x}, \mathbf{z})]$ ;
7:      $q_i(z_i) = \frac{\exp(\tilde{f}_i(z_i))}{\int \exp(\tilde{f}_i(z_i)) dz_i}$ ;
8:   end for
9: end while

```

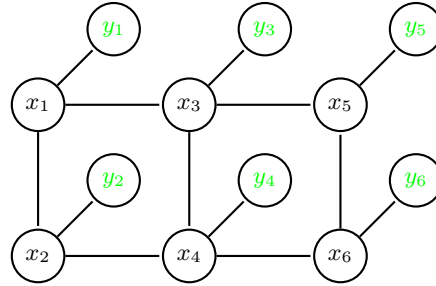
Note that the update rule (16.20) is given in the form of a function involving an integration. In actual implementation, we often derive a parametric form based on $q_i^*(z_i) \propto \exp(\tilde{f}_i(z_i))$ and perform update over the “variational” parameters. Especially when variables z_i are discrete, we can always represent q_i by $k - 1$ parameters, where k is the number of possible values that z_i can take.

16.3 Examples

Next, let us take a look at two examples, one in discrete case and the other in continuous case. The examples are adopted from [3] and [1].

16.3.1 CAVI on a MRF for image denoising

Consider the task of denoising an image using the following MRF



with energy function

$$E(\mathbf{x}, \mathbf{y}) = - \sum_{i=1}^m \alpha_i x_i - \sum_{(i,j) \in \mathcal{E}(G)} \beta_{i,j} x_i x_j - \sum_{i=1}^m \zeta_i x_i y_i,$$

where $\mathcal{E}(G)$ is the set of edges between neighboring pixels and $\beta_{i,j}, \zeta_i > 0$. In this task, the visible variables are the noisy pixels y_i and hidden variables are pixels x_i . All variables are discrete and take values in $\{+1, -1\}$.

To recover the original image based on its noisy version, let us apply CAVI to obtain the distribution of \mathbf{x} given \mathbf{y} . The joint distribution $p(\mathbf{x}, \mathbf{y})$ is

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} e^{-E(\mathbf{x}, \mathbf{y})}, \quad Z = \sum_{\mathbf{x}} \sum_{\mathbf{y}} e^{-E(\mathbf{x}, \mathbf{y})}. \quad (16.21)$$

We now assume a distribution $q(\mathbf{x})$ that factorizes:

$$q(\mathbf{x}) = \prod_{i=1}^m q_i(x_i). \quad (16.22)$$

Let $\mathbb{E}_{q_i}[x_i] = \mu_i$. Since every x_i takes two values, it suffices to optimize the ELBO over μ_i . We have

$$\log q_i^*(x_i) = \mathbb{E}_{q_{-i}}[\log p(\mathbf{x}, \mathbf{y})] + \text{const} \quad (16.23)$$

$$= \mathbb{E}_{q_{-i}}[-E(\mathbf{x}, \mathbf{y}) - \log Z] + \text{const} \quad (16.24)$$

$$= \mathbb{E}_{q_{-i}} \left[\sum_i^m \alpha_i x_i + \sum_{(i,j) \in \mathcal{E}(G)} \beta_{i,j} x_i x_j + \sum_i^m \zeta_i x_i y_i - \log Z \right] + \text{const} \quad (16.25)$$

$$= \mathbb{E}_{q_{-i}} \left[\alpha_i x_i + \sum_{j \in \mathcal{E}(x_i)} \beta_{i,j} x_i x_j + \zeta_i x_i y_i \right] + \text{const} \quad (16.26)$$

$$= \alpha_i x_i + \sum_{j \in \mathcal{E}(x_i)} \beta_{i,j} x_i \mu_j + \zeta_i x_i y_i + \text{const}, \quad (16.27)$$

where $\mathcal{E}(x_i)$ is the set of neighbors of x_i .

It follows that

$$q_i^*(x_i = 1) = \frac{e^{f_i}}{e^{f_i} + e^{-f_i}} = \frac{1}{1 + e^{-2f_i}}, \quad (16.28)$$

where $f_i = \alpha_i + \sum_{j \in \mathcal{E}(x_i)} \beta_{i,j} \mu_j + \zeta_i y_i$. Hence, the updating rules are given by

$$\mu_i^* = +1 \cdot q_i^*(x_i = 1) + (-1) \cdot q_i^*(x_i = -1) = \frac{1}{1 + e^{-2f_i}} - \frac{1}{1 + e^{2f_i}}. \quad (16.29)$$

16.3.2 Bayesian estimation of a univariate Gaussian [3]

Another application where we need to do inference about hidden variables given the visible ones is in Bayesian estimation. For a prior $p(\boldsymbol{\theta})$ and evidence $p(\mathcal{D}|\boldsymbol{\theta})$, we find an approximation for the posterior $p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})$ by maximizing the ELBO

$$\mathcal{L}(q) = \log p(\mathcal{D}) - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})) = \int q(\boldsymbol{\theta}) \log p(\mathcal{D}, \boldsymbol{\theta}) d\boldsymbol{\theta} + H(q). \quad (16.30)$$

univariate Gaussian Consider Bayesian modeling of a univariate Gaussian. Let our data x follow a Gaussian distribution $\mathcal{N}(\mu, \lambda^{-1})$, where λ is the precision. Here we use precision λ as the parameter instead of the variance to simplify our computation.

The likelihood is thus given by

$$p(\mathcal{D}|\mu, \lambda) = \left(\frac{\lambda}{2\pi} \right)^{N/2} \prod_{n=1}^N \exp\left(-\frac{\lambda}{2} (x_n - \mu)^2 \right). \quad (16.31)$$

We pick the conjugate *Gaussian-Gamma prior* of the form

$$p(\lambda; a_0, b_0) = \text{Gamma}(a_0, b_0) = \frac{\lambda^{a_0-1} \exp(-b_0 \lambda) b_0^{a_0}}{\Gamma(a_0)}, \quad (16.32)$$

$$p(\mu|\lambda; \mu_0, \kappa_0) = \mathcal{N}(\mu_0, (\kappa_0 \lambda)^{-1}) = \left(\frac{\kappa_0 \lambda}{2} \right)^{1/2} \exp\left(-\frac{\kappa_0 \lambda}{2} (\mu - \mu_0)^2 \right), \quad (16.33)$$

$$p(\mu, \lambda; \mu_0, \kappa_0, a_0, b_0) = \text{GaussGamma}(\mu_0, \kappa_0, a_0, b_0) \quad (16.34)$$

$$\propto \lambda^{a_0-1/2} \exp(-b_0 \lambda) \exp\left(-\frac{\kappa_0}{2} (\mu - \mu_0)^2 \lambda \right). \quad (16.35)$$

Then, $\mathbb{E} \lambda = a_0/b_0$, $\mathbb{E} \mu = \mu_0$, $\text{Var}[\lambda] = a_0/b_0^2$, $\text{Var}[\mu] = b_0/(\kappa_0(a_0 - 1))$.

We are interested in the posterior

$$p(\mu, \lambda | \mathcal{D}) \propto p(\mu, \lambda) p(\mathcal{D} | \mu, \lambda). \quad (16.36)$$

Exact posterior ** The exact posterior can be shown to be

$$p(\mu, \lambda | \mathcal{D}) = \text{GaussGamma} \left(\frac{\kappa_0 \mu_0 + N \bar{x}}{\kappa_0 + N}, \kappa_0 + N, a_0 + \frac{N}{2}, b_0 + \frac{1}{2} \left(N s + \frac{\kappa_0 N (\bar{x} - \mu_0)^2}{\kappa_0 + N} \right) \right), \quad (16.37)$$

where $\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n$, $s = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})^2$.

Approximate posterior Next, we approximate $p(\mu, \lambda | \mathcal{D})$ by

$$q(\mu, \lambda) = q_\mu(\mu) q_\lambda(\lambda). \quad (16.38)$$

Let us derive the updating rules needed by CAVI. Suppose we begin with two guesses $q_\mu(\mu)$ and $q_\lambda(\lambda)$. By (16.20),

$$\log q_\mu^*(\mu) = \mathbb{E}_{q_\lambda} [\log p(\mathcal{D}, \mu, \lambda)] + \text{const} \quad (16.39)$$

$$= \mathbb{E}_{q_\lambda} [\log p(\mathcal{D} | \mu, \lambda) + \log p(\mu | \lambda)] + \text{const} \quad (16.40)$$

$$= \mathbb{E}_{q_\lambda} \left[-\frac{\lambda}{2} \left(\sum_{n=1}^N (x_n - \mu)^2 + \kappa_0 (\mu - \mu_0)^2 \right) \right] + \text{const} \quad (16.41)$$

$$= -\frac{\mathbb{E}_{q_\lambda}[\lambda]}{2} \left(\sum_{n=1}^N (x_n - \mu)^2 + \kappa_0 (\mu - \mu_0)^2 \right) + \text{const} \quad (16.42)$$

$$\Rightarrow q_\mu^*(\mu) \sim \mathcal{N}(\nu, \tau^{-1}), \quad \nu = \frac{\kappa_0 \mu_0 + \sum_{n=1}^N x_n}{N + \kappa_0}, \quad \tau = (N + \kappa_0) \mathbb{E}_{q_\lambda}[\lambda]. \quad (16.43)$$

Further,

$$\log q_\lambda^*(\lambda) = \mathbb{E}_{q_\mu} [\log p(\mathcal{D}, \mu, \lambda)] + \text{const} \quad (16.44)$$

$$= \mathbb{E}_{q_\mu} [\log p(\mathcal{D} | \mu, \lambda) + \log p(\mu | \lambda) + \log p(\lambda)] + \text{const} \quad (16.45)$$

$$= \mathbb{E}_{q_\mu} \left[\frac{N}{2} \log \left(\frac{\lambda}{2\pi} \right) + \sum_{n=1}^N \left(-\frac{\lambda}{2} (x_n - \mu)^2 \right) + \frac{1}{2} \log \left(\frac{\kappa_0 \lambda}{2} \right) + \left(-\frac{\kappa_0 \lambda}{2} (\mu - \mu_0)^2 \right) \right] \quad (16.46)$$

$$+ (a_0 - 1) \log \lambda + (-b_0 \lambda) \Big] + \text{const} \quad (16.47)$$

$$= \mathbb{E}_{q_\mu} \left[\left(\frac{N+1}{2} + a_0 - 1 \right) \log \lambda + \left(-\frac{1}{2} \sum_{n=1}^N (x_n - \mu)^2 - \frac{\kappa_0}{2} (\mu - \mu_0)^2 - b_0 \right) \lambda \right] + \text{const} \quad (16.48)$$

$$= \left(\frac{N+1}{2} + a_0 - 1 \right) \log \lambda - \left(b_0 + \mathbb{E}_{q_\mu} \left[\frac{1}{2} \sum_{n=1}^N (x_n - \mu)^2 + \frac{\kappa_0}{2} (\mu - \mu_0)^2 \right] \lambda \right) + \text{const} \quad (16.49)$$

$$(16.50)$$

$$\Rightarrow q_\lambda^*(\lambda) \sim \text{Gamma}(a, b), \quad (16.51)$$

where

$$a = \frac{N+1}{2} + a_0, \quad b = b_0 + \frac{1}{2} \mathbb{E}_{q_\mu} \left[\sum_{n=1}^N (x_n - \mu)^2 + \kappa_0 (\mu - \mu_0)^2 \right]. \quad (16.52)$$

As we can see from (16.43) and (16.51), q_μ is a Gaussian and q_λ is a Gamma. Therefore, in practice, we can initialize with these parametric forms and do updating on their parameters. Note that we did not specify the Gaussian and Gamma parametric forms beforehand.

The updating rules for parameters ν, τ, a, b are thus

$$\nu = \frac{\kappa_0 \mu_0 + \sum_{n=1}^N x_n}{N + \kappa_0}, \quad (16.53)$$

$$\tau = (N + \kappa_0) \mathbb{E}_{q_\lambda}[\lambda] = (N + \kappa_0) \frac{a}{b}, \quad (16.54)$$

$$a = \frac{N + 1}{2} + a_0, \quad (16.55)$$

$$b = b_0 + \frac{1}{2} \mathbb{E}_{q_\mu} \left[\sum_{n=1}^N (x_n - \mu)^2 + \kappa_0 (\mu - \mu_0)^2 \right] \quad (16.56)$$

$$= b_0 + \frac{1}{2} \left(\left(\sum_{n=1}^N x_n^2 \right) + \kappa_0 \mu_0^2 - 2 \left(\sum_{n=1}^N x_n + \kappa_0 \mu_0 \right) \mathbb{E}_{q_\mu}[\mu] + (N + \kappa_0) \mathbb{E}_{q_\mu}[\mu^2] \right) \quad (16.57)$$

$$= b_0 + \frac{1}{2} \left(\left(\sum_{n=1}^N x_n^2 \right) + \kappa_0 \mu_0^2 - 2 \left(\sum_{n=1}^N x_n + \kappa_0 \mu_0 \right) \nu + (N + \kappa_0) (\nu^2 + \tau^{-1}) \right). \quad (16.58)$$

Figure 16.1 shows the updates when we apply CAVI to approximate the posterior of Gaussian parameters.

16.4 Factorized variational approximations are compact

The variational approximations $q(\mathbf{z})$ tend to be more compact than the actual posterior $p(\mathbf{z}|\mathbf{x})$. This is partly due to the natural asymmetry of KL-divergence. Consider that we approximate $p(x)$ using $q(x)$ by minimizing

$$D_{KL}(q(x)||p(x)) = \sum_x q(x) \log \frac{q(x)}{p(x)}. \quad (16.59)$$

It can be seen that when $p(x)$ is close to 0, $q(x)$ being large will contribute a large positive value to the KL. Therefore, to minimize $D_{KL}(q(x)||p(x))$, wherever $p(x)$ is small, $q(x)$ must also be small. $q(x)$ thus has a tendency of “shrinking” to only regions where $p(x)$ is not close to 0, shown in Figure 16.2a.

On the other hand, if we instead minimize

$$D_{KL}(p(x)||q(x)) = \sum_x p(x) \log \frac{p(x)}{q(x)}, \quad (16.60)$$

then wherever $p(x)$ is large, $q(x)$ must also be large. Therefore, $q(x)$ will have a tendency of “covering” regions where $p(x)$ is positive, shown in Figure 16.2b.

References

- [1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational inference: A review for statisticians.” In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.
- [2] Ian Goodfellow. “NIPS 2016 tutorial: Generative adversarial networks.” In: *arXiv preprint arXiv:1701.00160* (2016).
- [3] Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

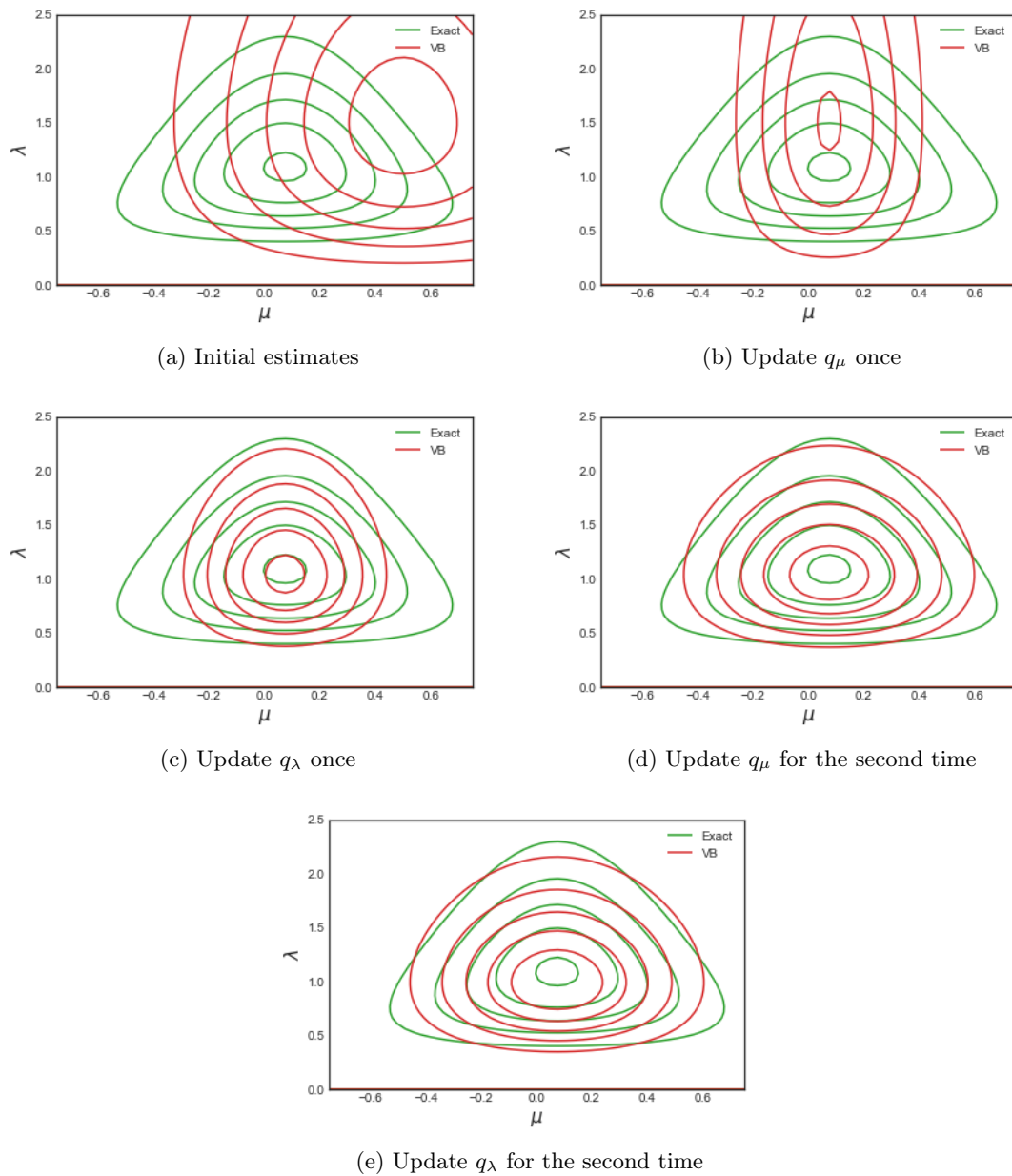


Figure 16.1: CAVI for the mean μ and precision λ of a univariate Gaussian distribution.

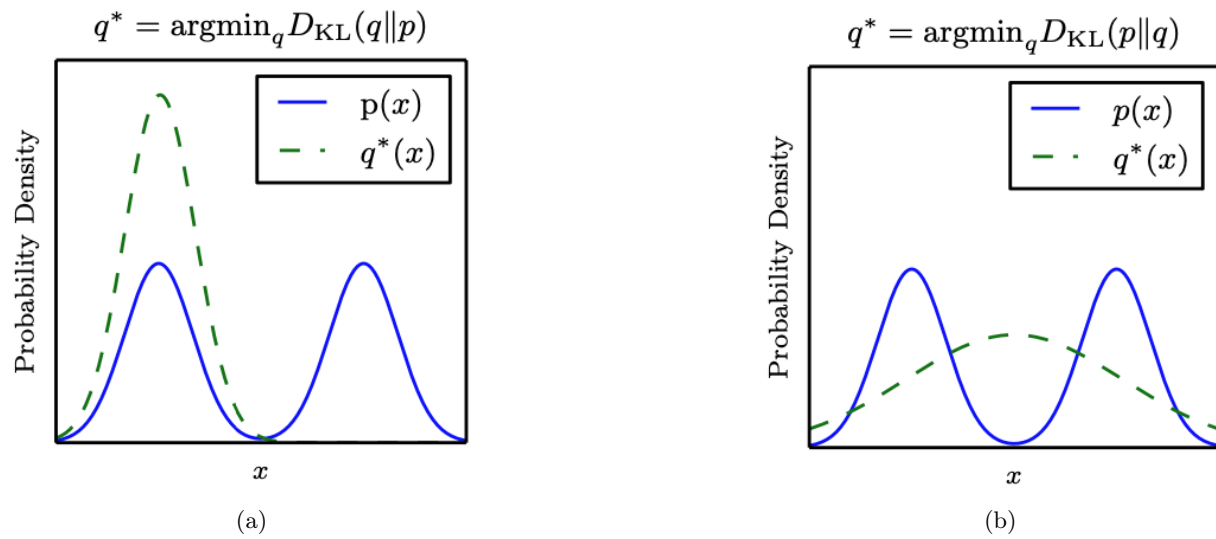


Figure 16.2: Approximating a bimodal distribution with a uni-modal distribution. Figures are from [2].

Chapter 17

Appendix

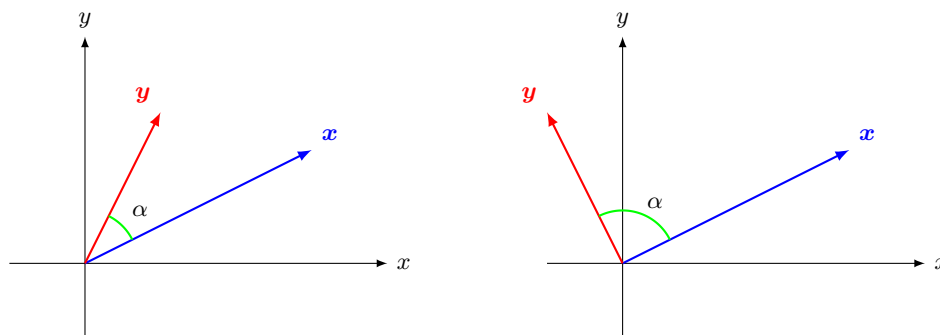
17.1 Review of Linear Algebra

For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the **inner product** $\langle \mathbf{x}, \mathbf{y} \rangle$ of \mathbf{x} and \mathbf{y} is

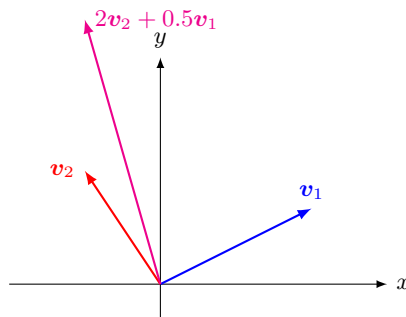
$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i y_i. \quad (17.1)$$

where \mathbf{x}^T is the transpose of \mathbf{x} .

The **length** or the ℓ_2 norm of a vector \mathbf{x} is $\|\mathbf{x}\| = \|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T \mathbf{x}}$ and we have $\|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x}$. Let α be the angle between \mathbf{x} and \mathbf{y} . Then $\mathbf{x}^T \mathbf{y} = \|\mathbf{x}\| \|\mathbf{y}\| \cos \alpha$. If $\mathbf{x}^T \mathbf{y} = 0$, then the two are called **orthogonal**.



For a collection of vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$, a **linear combination** of these is any vector of the form $a_1 \mathbf{v}_1 + \dots + a_m \mathbf{v}_m$, $a_i \in \mathbb{R}$. The set of all linear combinations of $\mathbf{v}_1, \dots, \mathbf{v}_m$ is their **span** and denoted as $\text{Span}\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$. This is a **subspace** (think line, plane, or the whole space). For a matrix \mathbf{A} , the span of the columns of \mathbf{A} is the **column space** of \mathbf{A} .

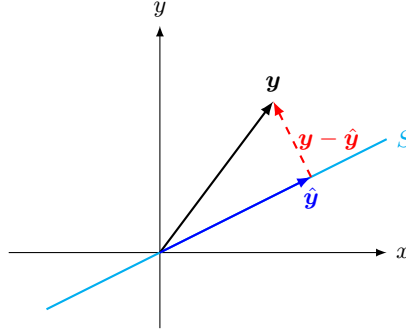


The vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ are **linearly independent** if there is no vector among them that can be written as a linear combination of the others, and linearly dependent otherwise. The vectors are linearly independent if and only if the only values for a_1, \dots, a_m satisfying $a_1\mathbf{v}_1 + \dots + a_m\mathbf{v}_m = \mathbf{0}$ are $a_1, \dots, a_m = 0$. In particular, the columns of a matrix \mathbf{A} are linearly independent if and only if the only vector \mathbf{a} satisfying $\mathbf{A}\mathbf{a} = \mathbf{0}$ is $\mathbf{a} = \mathbf{0}$.

The **inverse** of a square matrix \mathbf{A} is a matrix \mathbf{A}^{-1} such that $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, where \mathbf{I} is the **identity matrix**, which has 1s on the diagonal and 0s elsewhere. A matrix that has an inverse is called **invertible**. For a square matrix \mathbf{A} , the following are equivalent:

- It is invertible.
- For all distinct vectors \mathbf{a} and \mathbf{b} , we have $\mathbf{A}\mathbf{a} \neq \mathbf{A}\mathbf{b}$.
- The only solution to $\mathbf{A}\mathbf{x} = \mathbf{0}$ is $\mathbf{x} = \mathbf{0}$.
- Its columns are linearly independent.
- Its determinant $|\mathbf{A}|$ is nonzero. (We also have $|\mathbf{A}^{-1}| = \frac{1}{|\mathbf{A}|}$.)

Given a subspace S (e.g., a plane or the column space of a matrix) and a vector \mathbf{y} , let $\hat{\mathbf{y}}$ be the vector in the subspace that is closest to \mathbf{y} . That is, we find $\hat{\mathbf{y}} \in S$ such that $\|\mathbf{y} - \hat{\mathbf{y}}\|$ is minimized. Then $\hat{\mathbf{y}}$ is called the **projection** of \mathbf{y} onto the subspace S .



Lemma 17.1 (Projection Lemma). *Let $\hat{\mathbf{y}}$ be the projection of a vector \mathbf{y} onto a subspace S . Then $\mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to every vector in S .*

Proof. Suppose that this is not the case. Then there is a nonzero vector $\mathbf{v} \in S$ such that $(\mathbf{y} - \hat{\mathbf{y}})^T \mathbf{v} \neq 0$. We will show that this contradicts the minimality of $\|\mathbf{y} - \hat{\mathbf{y}}\|$. For any $a \in \mathbb{R}$,

$$\|\mathbf{y} - \hat{\mathbf{y}} - a\mathbf{v}\|_2^2 = (\mathbf{y} - \hat{\mathbf{y}} - a\mathbf{v})^T (\mathbf{y} - \hat{\mathbf{y}} - a\mathbf{v}) \quad (17.2)$$

$$= \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 - 2a\mathbf{v}^T (\mathbf{y} - \hat{\mathbf{y}}) + a^2\|\mathbf{v}\|_2^2. \quad (17.3)$$

This is a convex function in a . So setting the derivative to 0 gives the value of a that minimizes the error:

$$\frac{\partial}{\partial a} \|\mathbf{y} - \hat{\mathbf{y}} - a\mathbf{v}\|_2^2 = -2\mathbf{v}^T (\mathbf{y} - \hat{\mathbf{y}}) + 2a\|\mathbf{v}\|_2^2 = 0 \Rightarrow a = \frac{\mathbf{v}^T (\mathbf{y} - \hat{\mathbf{y}})}{\|\mathbf{v}\|_2^2} \neq 0. \quad (17.4)$$

Let

$$\hat{\mathbf{y}}' = \hat{\mathbf{y}} + \frac{\mathbf{v}^T (\mathbf{y} - \hat{\mathbf{y}})}{\mathbf{v}^T \mathbf{v}} \mathbf{v}, \quad (17.5)$$

and note that $\hat{\mathbf{y}}'$ is also in S but it is closer to \mathbf{y} , contradicting the optimality of $\hat{\mathbf{y}}$. \square

17.2 Vector and matrix differentiation

Definition 17.2 (The three derivatives). For a matrix \mathbf{A} , scalar z , and two vectors \mathbf{x}, \mathbf{y} (possibly one-dimensional), let

$$\frac{d\mathbf{A}}{dz} = \begin{pmatrix} \frac{\partial A_{11}}{\partial z} & \cdots & \frac{\partial A_{1n}}{\partial z} \\ \vdots & \ddots & \vdots \\ \frac{\partial A_{m1}}{\partial z} & \cdots & \frac{\partial A_{mn}}{\partial z} \end{pmatrix}, \quad \frac{dz}{d\mathbf{A}} = \begin{pmatrix} \frac{\partial z}{\partial A_{11}} & \cdots & \frac{\partial z}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z}{\partial A_{m1}} & \cdots & \frac{\partial z}{\partial A_{mn}} \end{pmatrix}, \quad \frac{d\mathbf{y}}{d\mathbf{x}} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_m} \end{pmatrix}$$

Lemma 17.3. For a scalar a , vectors $\mathbf{x}, \mathbf{y}, \mathbf{v}$, and constant matrices \mathbf{A} and \mathbf{S} ,

$$\begin{aligned} \frac{d\mathbf{y}}{d\mathbf{v}} &= \frac{d\mathbf{y}}{d\mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{v}}, \\ \frac{d}{d\mathbf{v}}(a\mathbf{x}) &= a \frac{d\mathbf{x}}{d\mathbf{v}} + \mathbf{x} \frac{da}{d\mathbf{v}}, \\ \frac{d}{d\mathbf{v}}(\mathbf{y}^T \mathbf{A} \mathbf{x}) &= \mathbf{y}^T \mathbf{A} \frac{d\mathbf{x}}{d\mathbf{v}} + \mathbf{x}^T \mathbf{A}^T \frac{d\mathbf{y}}{d\mathbf{v}}, \\ \frac{d}{d\mathbf{v}}(\mathbf{y}^T \mathbf{S} \mathbf{y}) &= 2\mathbf{y}^T \mathbf{S} \frac{d\mathbf{y}}{d\mathbf{v}}, \quad (\mathbf{S} \text{ is symmetric}) \\ \frac{d}{d\mathbf{v}}(\mathbf{A} \mathbf{x}) &= \mathbf{A} \frac{d\mathbf{x}}{d\mathbf{v}}. \end{aligned}$$

Lemma 17.4. For matrix \mathbf{A} and constant vector \mathbf{x} ,

$$\begin{aligned} \frac{d}{d\mathbf{A}}(\mathbf{x}^T \mathbf{A} \mathbf{x}) &= \mathbf{x} \mathbf{x}^T \\ \frac{d}{d\mathbf{A}} \ln |\mathbf{A}| &= \mathbf{A}^{-T} \end{aligned}$$

Definition 17.5. Let $f : \mathbb{R}^m \rightarrow \mathbb{R}$. The gradient of $f(\mathbf{x})$ with respect to \mathbf{x} is defined as

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \left(\frac{df(\mathbf{x})}{d\mathbf{x}} \right)^T = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_m} \end{pmatrix}$$

and the Hessian of $f(\mathbf{x})$ with respect to \mathbf{x} is defined as

$$\mathbf{H}_{\mathbf{x}}(f(\mathbf{x})) = \frac{d\nabla_{\mathbf{x}} f(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial f(\mathbf{x})}{\partial x_m \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_m} & \cdots & \frac{\partial f(\mathbf{x})}{\partial x_m \partial x_m} \end{pmatrix}$$

Chain rule. Consider $h : \mathbb{R}^m \rightarrow \mathbb{R}$, $g : \mathbb{R} \rightarrow \mathbb{R}$, and $f(\mathbf{x}) = g(h(\mathbf{x}))$. From Lemma 17.3,

$$\begin{aligned} \nabla f(\mathbf{x}) &= g'(h(\mathbf{x})) \nabla h(\mathbf{x}), \\ \mathbf{H} f(\mathbf{x}) &= g'(h(\mathbf{x})) \mathbf{H} h(\mathbf{x}) + g''(h(\mathbf{x})) \nabla h(\mathbf{x}) \nabla^T h(\mathbf{x}) \end{aligned}$$

since

$$\begin{aligned}
 \mathbf{H}f(\mathbf{x}) &= \frac{d\nabla f}{d\mathbf{x}} \\
 &= \frac{d(g'(h(\mathbf{x}))\nabla h(\mathbf{x}))}{d\mathbf{x}} \\
 &= g'(h(\mathbf{x}))\frac{d\nabla h(\mathbf{x})}{d\mathbf{x}} + \nabla h(\mathbf{x})\frac{d(g'(h(\mathbf{x})))}{d\mathbf{x}} \\
 &= g'(h(\mathbf{x}))\mathbf{H}h(\mathbf{x}) + \nabla h(\mathbf{x})\nabla^T h(\mathbf{x})g''(h(\mathbf{x}))
 \end{aligned}$$

Example 17.6. Let us find the derivatives of $f(\mathbf{x}) = \log \sum_{i=1}^m e^{x_i}$. Let $\mathbf{z} = (\exp(x_i))_{i=1}^m$ so that $f(\mathbf{x}) = \log \mathbf{1}^T \mathbf{z}$.

$$\begin{aligned}
 \nabla f(\mathbf{x}) &= \frac{\mathbf{z}}{\mathbf{1}^T \mathbf{z}}, \\
 \mathbf{H}f(\mathbf{x}) &= \frac{\text{diag}(\mathbf{z})}{\mathbf{1}^T \mathbf{z}} - \frac{\mathbf{z}\mathbf{z}^T}{(\mathbf{1}^T \mathbf{z})^2}.
 \end{aligned}$$

△

Chain rule. Let $\mathbf{h} = (h_1, \dots, h_n) : \mathbb{R}^m \rightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \rightarrow \mathbb{R}$, and $f(\mathbf{x}) = g(\mathbf{h}(\mathbf{x}))$. Then

$$\begin{aligned}
 \frac{\partial f}{\partial x_i} &= \sum_{j=1}^n \frac{\partial g}{\partial h_j} \frac{\partial h_j}{\partial x_i} = \frac{dg}{d\mathbf{h}} \cdot \frac{d\mathbf{h}}{dx_i} = \nabla^T g \cdot \frac{d\mathbf{h}}{dx_i}, \\
 \frac{df}{d\mathbf{x}} &= \frac{dg}{d\mathbf{h}} \frac{d\mathbf{h}}{d\mathbf{x}} = \nabla^T g \frac{d\mathbf{h}}{d\mathbf{x}}, \quad \nabla_{\mathbf{x}} f = \left(\frac{df}{d\mathbf{x}} \right)^T = \left(\frac{d\mathbf{h}}{d\mathbf{x}} \right)^T \nabla g
 \end{aligned}$$