Estimation and Statistical Learning

Farzad Farnoud
University of Virginia

Dec. 2020

Contents (Summary)

0 Review of Probability 6

1 Probability, Inference, and Learning 24

2 Frequentist Parameter Estimation 34

3 Bayesian Parameter Estimation 45

4 Multivariate Random Variables 56

5 Linear Regression 60

6 Linear Classification 74

7 Expectation-Maximization * 85

8 Basics of Graphical Models 96

9 Independence in Graphical Models 105

10 Parameter Estimation in Graphical Models 114

11 Inference in Graphical Models 122

12 Inference in Hidden Markov Models 135

13 Factor Graphs and Sum/Max-product Algorithms ** 143

14 Markov Chains 154

15 Sampling Methods 162

16 Appendix 176

Contents

0 Review of Probability **6**

0.1 What is probability? 6

0.2 Sets and their sizes 8

0.3 Random variables and distributions 9

0.3.1 Discrete distributions 9

0.3.2 Continuous distributions 10

0.3.3 Cumulative distribution functions 11

0.3.4 Expected value 11

0.3.5 Common distributions 14

0.4 Joint probability distributions 15

0.4.1 Expectation, correlation, and covariance 15

0.4.2 Independence 17

0.4.3 Conditional probability and conditional distributions 18

0.4.4 Bayes' rule 19

0.5 Inequalities and limits 20

0.5.1 Inequalities 20

0.5.2 Limits 21

1 Probability, Inference, and Learning **24**

1.1 Introduction 24

1.2 Relationships and joint probability distributions 24

1.3 Inference and decision making 26

1.4 Machine Learning and Probability 28

1.4.1 Empirical Risk Minimization (ERM) 29

1.4.2 Density estimation 29

1.4.3 Decomposition of error for mean squared error 30

1.5 Quantifying uncertainty 31

1.6 Conditional entropy* 32

2 Frequentist Parameter Estimation **34**

2.1 Parameter Estimation 34

2.2 Maximum likelihood: Introduction and Examples 34

2.3 Properties of Estimators 38

2.3.1 Estimation error and bias 38

2.3.2 Mean squared error and variance 40
2.3.3 Consistency 42
2.4 The Cramer-Rao lower bound* 43
2.5 Asymptotic normality of the MLE 44

3 Bayesian Parameter Estimation **45**
3.1 From Prior to Posterior 45
3.2 Bayesian Point Estimates 49
3.3 Posterior Predictive Distribution 50
3.4 Gaussian Prior and Likelihood 51
3.5 Conjugate Priors 53
3.6 The Exponential Family (EF)** 54

4 Multivariate Random Variables **56**
4.1 Review of Linear Algebra 56
4.2 Random vectors 57
4.3 Gaussian Random Vectors (Joint Gaussian Distribution) 58
4.4 Maximum likelihood for Gaussian Random Vectors 59

5 Linear Regression **60**
5.1 Introduction 60
5.2 Least-squares 61
5.3 Probabilistic Models for Regression 63
5.4 Regularized Linear Regression 67
5.5 Bias-Variance Trade-off 69
5.6 Stochastic Gradient Descent 72

6 Linear Classification **74**
6.1 Overview of probabilistic models 74
6.2 Generative Probabilistic Models 75
6.2.1 Gaussian Class-Conditionals 75
6.2.2 Linear Discriminant Analysis 75
6.2.3 Quadratic Discriminant Analysis 77
6.2.4 Maximum Likelihood Solution to LDA 77
6.2.5 Generative Model for Discrete Features ** 78
6.2.6 Class-conditionals from the exponential family 79
6.3 Discriminative Models and Logistic Regression 80
6.4 Risk minimization and loss functions for classification 80

7 Expectation-Maximization * **85**
7.1 Overview 85
7.2 Clustering with EM 88
7.3 EM with general missing data ** 91
7.4 The MM Algorithm 92
7.4.1 Rank Aggregation from Pairwise Comparisons via MM 93

8 Basics of Graphical Models **96**

8.1 Introduction . . . 96
8.2 Bayesian Networks . . . 96
8.2.1 Markov Model . . . 99
8.2.2 Why graphical models? . . . 100
8.3 Markov Random Fields . . . 100
8.3.1 Energy-based models . . . 102
8.4 Moralization: Converting BNs to MRFs . . . 103
8.5 Latent Dirichlet Allocation** . . . 104

9 Independence in Graphical Models 105
9.1 Independence for sets of random variables . . . 105
9.2 Independence in Bayesian Networks . . . 106
9.2.1 Simple Bayesian networks . . . 106
9.2.2 d-separation . . . 108
9.2.3 Markov Blanket in Bayesian Networks . . . 110
9.3 Independence in MRFs . . . 110
9.4 In-class activity . . . 111

10 Parameter Estimation in Graphical Models 114
10.1 Introduction . . . 114
10.2 Maximum Likelihood Estimation in Bayesian Networks . . . 114
10.3 Bayesian Parameter Estimation in Bayesian Networks . . . 115
10.3.1 Bayesian Estimation formulated as Bayesian Networks . . . 116
10.3.2 Estimating Parameters of CPDs in Bayesian Networks . . . 117
10.4 Parameter Estimation in MRFs . . . 120

11 Inference in Graphical Models 122
11.1 Introduction . . . 122
11.2 The Elimination Algorithm . . . 123
11.3 The Sum-Product Algorithm . . . 125
11.4 The Max-Product Algorithm . . . 127
11.5 Sum-product Example . . . 129

12 Inference in Hidden Markov Models 135

13 Factor Graphs and Sum/Max-product Algorithms ** 143

14 Markov Chains 154
14.1 Introduction . . . 154
14.2 State distribution as a function of time . . . 157
14.3 Long-term Behavior of Markov Chains . . . 157
14.3.1 How often does the Markov Chain visit each state? . . . 159
14.4 Balance Properties and Finding the Stationary Distribution . . . 159
14.4.1 Global Balance . . . 159
14.4.2 Detailed Balance . . . 161

15 Sampling Methods 162

15.1 Introduction 162
15.2 Basic Sampling Techniques 163
15.2.1 Deterministic Integration 163
15.2.2 Rejection Sampling 164
15.2.3 Importance Sampling 166
15.3 Metropolis Monte Carlo 167
15.4 Gibbs Sampling 171
15.5 Hamiltonian Monte Carlo ** 173

16 Appendix 176
16.1 Vector and matrix differentiation 176
16.2 Properties of Expectation, Correlation, and Covariance for Vectors 178

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. For an excellent review of probability see [1], which also has many examples.

0.1 What is probability?

Probability is a branch of mathematics that deals with sets, and functions that assign real values to those sets, in a way that certain axioms are satisfied. Note that this may or may not correspond to our models of the real world. In that sense, probability is similar to geometry, number theory, etc.

0.1.0.1 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.

0.1.0.2 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 $E_1 \cap E_2 = \varnothing$.

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

[1] These 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 $E_1, E_2, \ldots$, we require that $\Pr(\bigcup_{i=1}^{\infty}) = \sum_{i=1}^{\infty} \Pr(E_i)$.

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

More definitions for basic concepts:

- Two events A and B are *independent*, denoted $A \perp\!\!\!\perp B$, 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.

0.1.0.3 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 $\Pr(E) = 1/3$?

- Frequentist interpretation: Assuming that there is a "random" experiment that can be repeated many times for which E is an event, the relative "frequency" of E occurring is 1/3.
  - Probability of heads for a fair coin: $\Pr(H) = 1/2$. As odds this is represented as 1:1 (happening : not happening)
  - Probability **distribution** of the number N of children (≤ 18) of a randomly chosen American household:

	$\Pr(N = 0)$	$\Pr(N = 1)$	$\Pr(N = 2)$	$\Pr(N \ge 3)$
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(Heads) = 1/2 (both Bayesian and frequentist)
  - Pr(Stock market will hit a certain threshold this year)
  - Pr(Nuclear war this century)
  - Pr(A certain person is guilty of a given crime)

Different interpretations lead to different approaches to problems, potentially leading to different real-life 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 of desirable outcomes A, 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. 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 (doing this is pretty difficult if not impossible), 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 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).

Examples: a random variable that is 1 if heads shows when a given coin is filliped and is 0 otherwise (discrete, finite); the arrival time of a plane in seconds from midnight; the number of people buying a specific product; ...

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

$$\sum_x p_X(x) = 1. \tag{0.1}$$

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). △

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. △

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. △

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,

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

For two real numbers a, b,

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

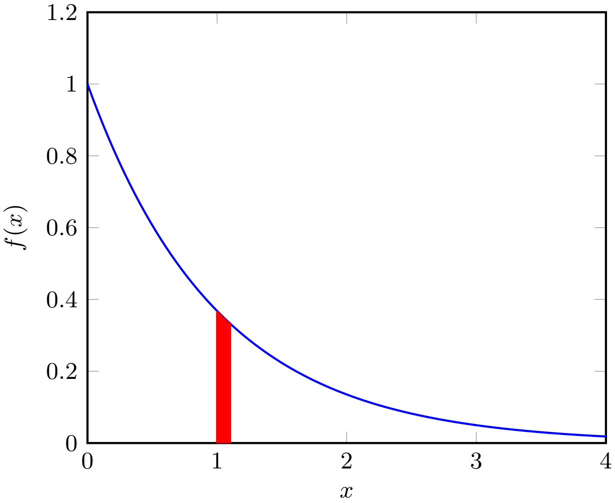
For any pdf, we have $p_X(x) \ge 0$ and

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

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

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

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

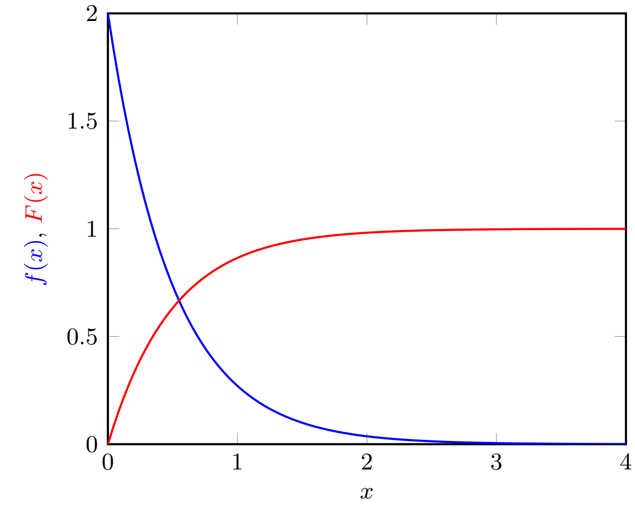
Cumulative distribution functions (CDFs) are defined for both discrete and continuous RVs as $F_X(x) = p_X(X \leq x)$ and can be found via summation or integration:

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

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

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

$$\mathbb{E}[X] = \sum_{x} xp(x),$$

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x)dx.$$

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

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

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

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

Find $\mathbb{E}\,X$. Let $Y = X^2$. Find $\mathbb{E}\,Y$, both by finding the distribution of Y and by using (0.6). △

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

Consider a collection of random variables $X_1, X_2, \ldots, X_n$. 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

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

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,

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

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. △

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 $\mathrm{Var}[X]$,

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

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

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

and the variance is usually denoted as σ_X^2.

Exercise 0.17. Prove that

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

△

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)

△

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

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

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

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

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 $g(\lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \propto \lambda^x e^{-\lambda}$.

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

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 $\text{Beta}(x+1, n-x+1)$. △

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

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 type,

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

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

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

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 (0.6), we have

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

and similarly for discrete variables.

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

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

It can be shown that $-1 \le \rho_{X,Y} \le 1$. 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,[3] 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 $1-\rho^2$.

[3] Minimizing the Mean Square Error

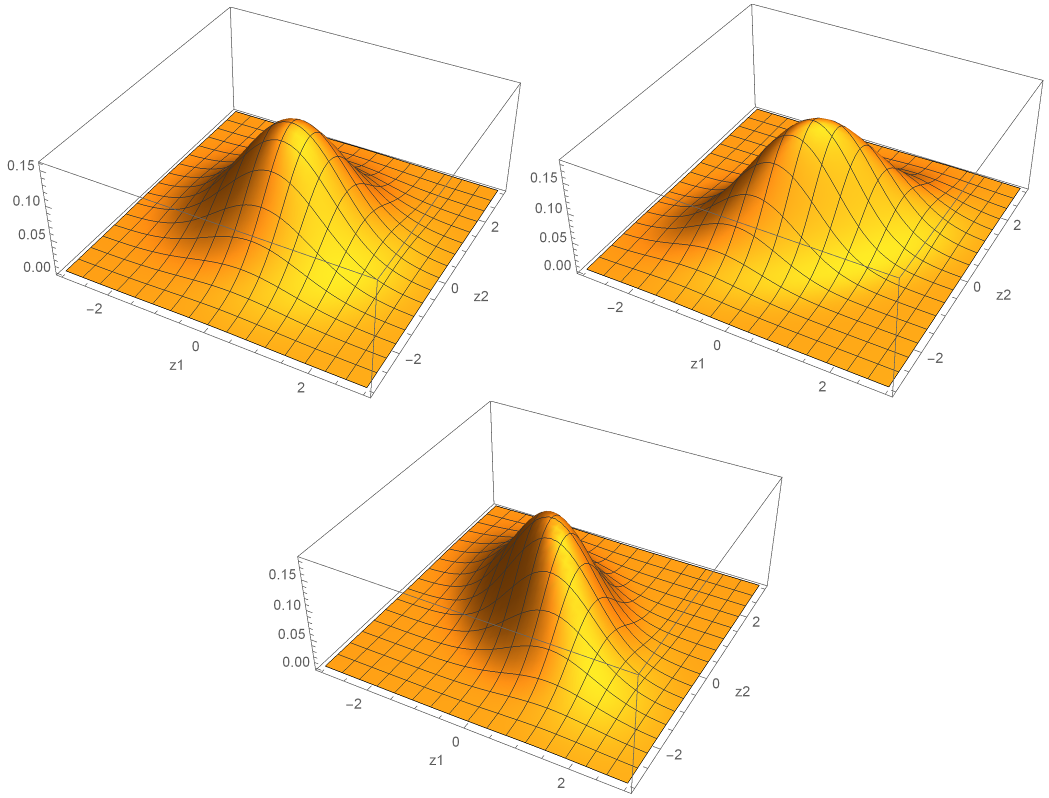


Figure 1: Bivariate Normal pdfs with $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, with $\rho = 0$ (uncorrelated), $\rho = .5$ (positively correlated), and $\rho = -.5$ (negatively correlated), respectively.

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

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

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. △

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

- $\mathrm{Var}(X) = \mathrm{Cov}(X, X)$
- $\mathrm{Cov}(X + Y, Z) = \mathrm{Cov}(X, Z) + \mathrm{Cov}(Y, Z)$
- $\mathrm{Cov}(aX, Y) = a\,\mathrm{Cov}(X, Y)$

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

△

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

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

$$p(x, y) = p(x)p(y). \tag{0.7}$$

For two independent random variables, we have

$$\mathbb{E}[XY] = \mathbb{E}[X]\,\mathbb{E}[Y] \tag{0.8}$$

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

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

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

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

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

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

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

△

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

$$\mathbb{E}[\bar{X}] = \mu, \qquad \mathrm{Var}[\bar{X}] = \frac{\sigma^2}{n}. \tag{0.11}$$

△

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

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

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 $A_i \cap A_j = \varnothing$. 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

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

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

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. So we can consider $g(X) = \mathbb{E}[Y|X]$. 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 $\mathbb{E}[\mathbb{E}[Y|X]]$. △

It can be shown that

$$\mathbb{E}[\mathbb{E}[Y|X]] = \mathbb{E}[Y], \qquad \mathbb{E}[\mathbb{E}[Y|X, Z]|Z] = \mathbb{E}[Y|Z]. \tag{0.12}$$

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

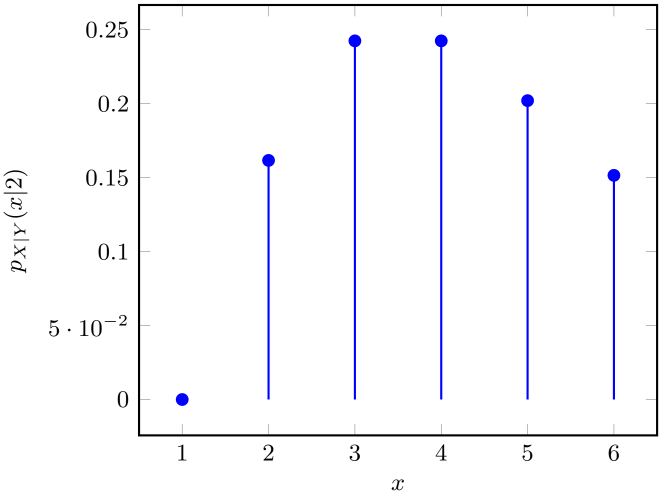
$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')},$$

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 $3, 4$.



△

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 22 m 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 ≥ 30m? For example, is it possible that this probability is 0.8 or larger? No, since in that case, the average would be $\geq 0.8 \times 30 = 24$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,

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

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

Exercise 0.35 (optional)**.** 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 $k \times$ 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 $X_1, \ldots, X_n$ 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, it is sufficient for them to be independent). Also, let $\bar{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$. Then, for any $\epsilon > 0$,

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

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 $\mathbb{E}\,X$,

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

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. △

Example 0.39. Suppose $X_i \sim \text{Poi}(2)$, $1 \le i \le 500$, and let $\bar{X}_n$ be the average of the first n X_is. Figure 2 shows the plot for $\bar{X}_n$ for a realization of X_is 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. △

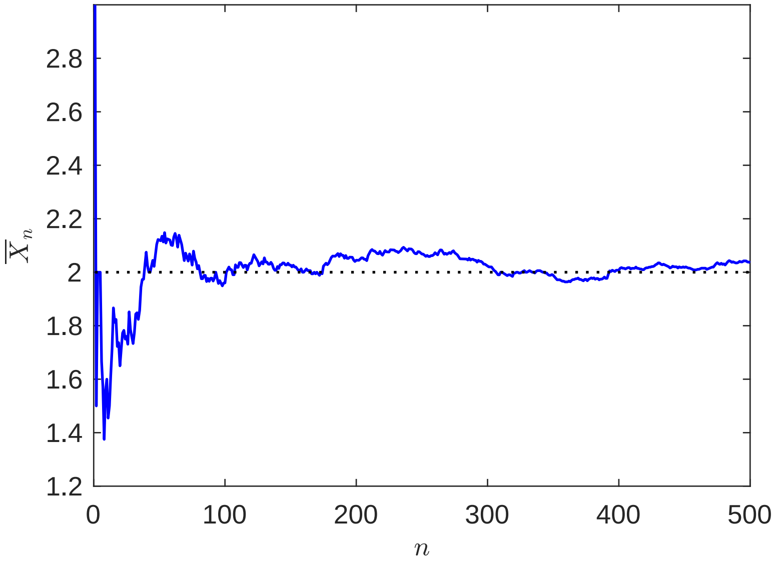


Figure 2: $\bar{X}_n$ based on $X_i \sim \text{Poi}(2)$ as a function of n.

0.5.2.2 Central limit theorem

Let $X_1, X_2, \ldots$ 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 $n \to \infty$. The **Central Limit Theorem (CLT)** states that

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

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. △

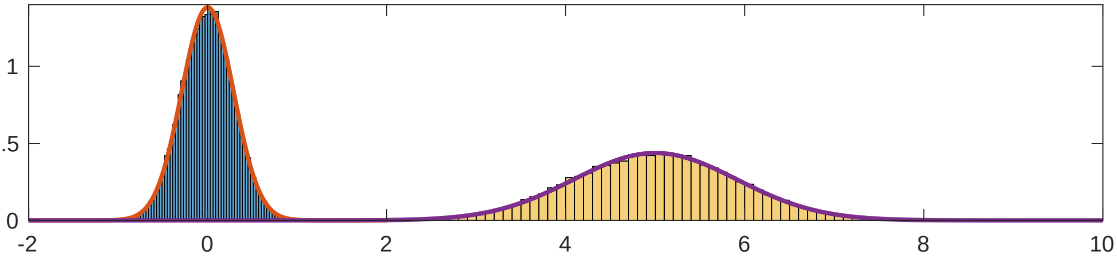


Figure 3: The normalized histograms for $\sqrt{n}(\bar{X}_n - \mu)$ and the pdf of $\mathcal{N}(0, \sigma^2)$ (on the left) and the normalized histogram for S_n and the pdf of $\mathcal{N}(n\mu, n\sigma^2)$ (on the right) for uniform X_i with $\mu = 1/2$ and $\sigma^2 = 1/12$ and with $n = 10$.

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 were 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.

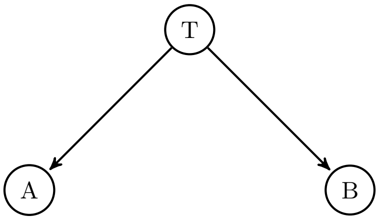
In our example, there's obviously a relationship, that is, the arrival times are not independent. For example, both are affected by traffic. Let

$$\begin{aligned}
&T_0 : \text{normal traffic}\\
&T_1 : \text{heavy traffic}\\
&A_0 : \text{Alice is on time}\\
&A_1 : \text{Alice is late}\\
&B_0, B_1 \text{ for Bob}
\end{aligned}$$

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 grater 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 $\Pr(E_1E_2) = \Pr(E_1)\Pr(E_2)$, or, if $\Pr(E_2) \neq 0$, $\Pr(E_1|E_2) = \Pr(E_1)$. We have

$$\begin{aligned}
\Pr(A_0|B_0) &= \Pr(A_0B_0)/\Pr(B_0)\\
\Pr(A_0B_0) &= (0.65 \times 0.82 \times 0.9) + (0.35 \times 0.15 \times 0.5) = 0.506\\
\Pr(A_0|B_0) &= 0.506/0.586 = 0.863 \neq \Pr(A_0)\\
\Pr(B_0|A_0) &= 0.506/0.76 = 0.6658 \neq \Pr(B_0)
\end{aligned}$$

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 and we are interested to know if $\theta > 0.01$, where 0.01 is the fraction of people in the general population with that disease. 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 devises statistical tests to decide if $\theta > 0.01$. Clearly, 2 out of 100 is larger than would be expected by chance. So this may be because the allele and the disease are related. On the other hand, maybe the allele doesn't have anything to do with the disease, but we have been unlucky enough to pick two people with the disease. So how do we decide?

  Our statistician may consider how likely it is to see *similar or stronger evidence by chance.* This probability is called the *p-value.*

  If the probability of the disease is 0.01, what is the probability of seeing 2 or more sick people in a sample of size 100?

  $$p = 1 - \left(\binom{100}{0} 0.99^{100} + \binom{100}{1} 0.99^{99} 0.01^{1} \right) = 1 - 0.37 - 0.37 = 0.26 > 0.05$$

  The smaller the p-value, the stronger the evidence. Typically, if the p-value is smaller than 0.05, we believe the evidence is strong enough to reject the hypothesis that the observation has occurred by chance.

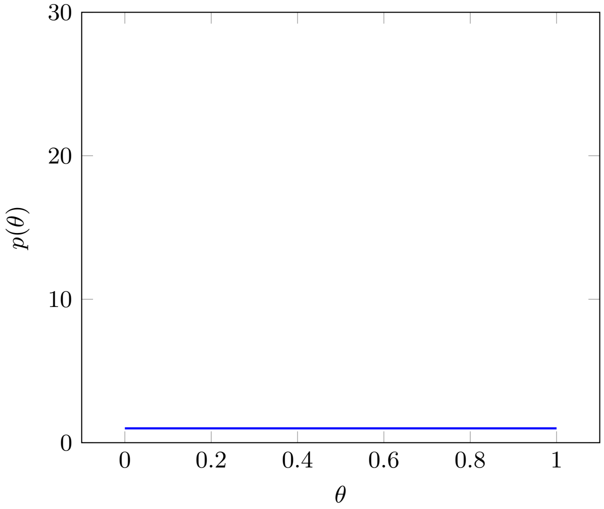
- A Bayesian thinks of θ as random and assigns to it a distribution, called the *prior*, before seeing the data. She then looks at the data and updates her distribution for θ, thus obtaining the *posterior* distribution. (We'll learn more about Bayesian methods.)

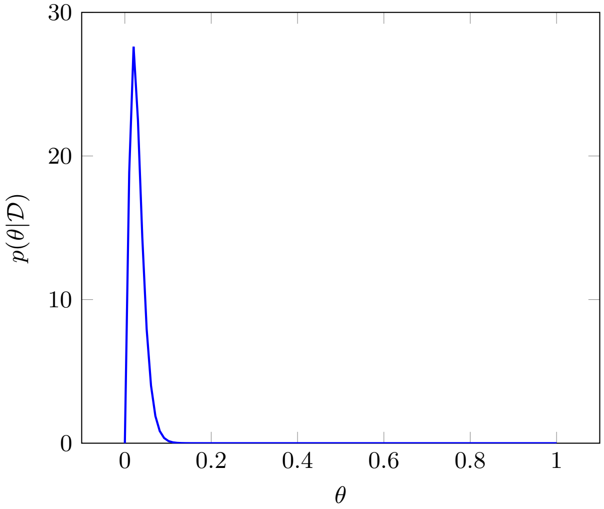
  Assume that before seeing the data, we believe that the distribution for θ is uniform, i.e., $p(\theta) \sim \text{Uni}[0,1] = \text{Beta}(1,1)$. This means that while we do not know what θ is, we believe it is equally likely to be any value between 0 and 1. When we see the data, we can update this belief,

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

It turns out $p(\theta|\mathcal{D}) \sim \text{Beta}(3, 99)$, and, as we will see,

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





What is the source of uncertainty in this case? Why can't we say for certain if $\theta > 0.01$? This is because of 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**: $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}, x_i \in \mathcal{X}, y_i \in \mathcal{Y}$. $\mathcal{X}$ 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, . . .), and y 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
$$\mathcal{L}(f) = \mathbb{E}[L(Y, f(X)], \tag{1.1}$$
where the expectation is over the distribution $p(x, y)$ of (X, Y). Our goal then becomes finding
$$f^* = \arg\min_f \mathcal{L}(f) = \arg\min_f \mathbb{E}[L(Y, f(X)]. \tag{1.2}$$
- **Learning Algorithm**: The algorithm that finds f^*, or tries to.

You may have noticed that $\mathcal{D}$ consists of samples from $p(x, y)$, but in (1.2), we need the joint distribution of X, Y. We can address this in two ways, either through the Empirical Risk Minimization framework discussed in §1.4.1, or through estimating the unknown distribution 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:
$$L(y, f(x)) = (y - f(x))^2. \tag{1.3}$$
For this choice, if the distribution is known, it can be shown that
$$\hat{y} = f(x) = \mathbb{E}[Y|X = x]. \tag{1.4}$$
- **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**:
$$L(y, f(x)) = \begin{cases} 1, & \text{if } y \neq f(x). \\ 0, & \text{if } y = f(x). \end{cases} \tag{1.5}$$

In this case, if the distribution is known, then the best classifier is $\hat{y} = \arg\max_{y\in\mathcal{Y}} p(y|x)$.

1.4.1 Empirical Risk Minimization (ERM)

Since we usually do not know the distribution but have access to data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, we cannot directly minimize the expected loss as in (1.2). Instead we can minimize the loss on observed data points,

$$f^* = \arg\min_f \mathbb{E}[L(Y, f(X)] \quad \rightarrow \quad f^* = \arg\min_f \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)). \tag{1.6}$$

This is, however, problematic, as it only provides a way for us to determine the value of $f(x)$ for $x \in \{x_1, \ldots, x_n\}$. 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, called the **hypothesis set**. This leads to the ERM formulation of the learning problem

$$f^* = \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)). \tag{1.7}$$

For example, we may choose $\mathcal{F}$ to be the set of linear or sigmoid functions.

The choice of $\mathcal{F}$ is critical to how well the predictor generalizes. On the one hand, it needs to be large enough to be able to produce a small loss. As an extreme example, setting $\mathcal{F}$ to contain only $f(x) = 0$ for all x is not a good choice. On the other hand, if $\mathcal{F}$ has too many degrees of freedom, we may get a predictor f that is tuned well to the dataset but does not generalize well, i.e., performs poorly for examples outside of the dataset. This is called **overfitting**. We check whether this is the case by setting aside part of the data, referred to as the **test set**, which is used only for evaluating performance but not for training. Data used for training is called the **training set**. (If we need to choose between different algorithms or tune hyper-parameters, we may further divide the training set to training and validation sets.)

1.4.2 Density estimation

As discussed, 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.

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,

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

Alternatively, we can formulate the problem as density estimation with maximum-likelihood loss to begin with. From the following equation, loss is minimized when log-likelihood is maximized.

$$\begin{aligned} L(P_\theta(x,y)) &= -\log P_\theta(x,y) \\ \mathcal{L}(\theta) &= -\mathbb{E}(\log(P_\theta(X,Y)) \end{aligned}$$

Again, before determining θ, we do not know the distribution and cannot evaluate the expected loss. So we minimize the empirical risk:

$$\begin{aligned} \mathcal{L}(\theta) &= -\sum_{i=1}^{n} \log P_\theta(x_i, y_i) \\ \hat{\theta} &= \arg\min_{\theta\in\Theta} \mathcal{L}(\theta), \end{aligned}$$

where Θ is the set of all valid parameters.

1.4.3 Decomposition of error for mean squared error

In (1.4), we claimed that for mean squared error and known joint distribution, the best predictor for Y given $X = x$ is $\mathbb{E}[Y|X = x]$. We start by proving this claim. First, let us consider: What is the best predictor for a (random) quantity Y when we know the distribution of Y but have no other information. Since we have no information, this predictor is a single constant value c and for the mean squared error we have

$$\begin{aligned} \mathbb{E}\big[(Y-c)^2\big] &= \mathbb{E}\Big[(Y-\mu+\mu-c)^2\Big] \\ &= \mathrm{Var}(Y) + 2\,\mathbb{E}[Y-\mu](\mu-c) + (\mu-c)^2 \\ &= \mathrm{Var}(Y) + (\mu-c)^2, \end{aligned}$$

where $\mu = \mathbb{E}[Y]$. This is minimized by letting $c = \mu = \mathbb{E}[Y]$.

Now let us consider the original problem: What is the best predictor $f(x)$ for Y if we know $X = x$ as well as the joint distribution of (X,Y)? Let $\bar{y}(x) = \mathbb{E}[Y|X = x]$. For the mean squared error for a given value of x, we have

$$\begin{aligned} \mathbb{E}\big[(Y-f(x))^2|X=x\big] &= \mathbb{E}\Big[(Y-\bar{y}(x)+\bar{y}(x)-f(x))^2|X=x\Big] \\ &= \mathbb{E}\Big[(Y-\bar{y}(x))^2|X=x\Big] + 2\,\mathbb{E}[Y-\bar{y}(x)|X=x](\bar{y}(x)-f(x)) + (\bar{y}(x)-f(x))^2 \\ &= \mathbb{E}\Big[(Y-\bar{y}(x))^2|X=x\Big] + (\bar{y}(x)-f(x))^2. \end{aligned}$$

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 our model and not lack of enough data. The reducible part, and thus the error, is minimized by setting $f(x) = \bar{y}(x) = \mathbb{E}[Y|X = x]$. However, doing so exactly is only possible if we have the distribution or an infinite amount of data. When f is determined based on a finite sample $\mathcal{D}$, the term $(\bar{y}(x) - f(x))^2$ can be decomposed into bias and variance components, which we will discuss later.

1.5 Quantifying uncertainty

Suppose we know the distribution for a random variable. How do we measure how uncertain we are? Alternatively how much information will we gain when we find out the outcome or how surprised will we be when we see the outcome?

First, we observe that the lower the probability of a statement, the higher the surprise/information content:

- The sun will rise tomorrow: Very likely, low information content
- It's raining in Seattle: Even chances, provides some information
- It's raining in the Sahara: Very unlikely, high information content

So we look for a function that decreases as the probability p of the event increases. 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* is

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

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

$$\begin{aligned} H(\text{Fair coin}) &= H_b(\frac{1}{2}) = \frac{1}{2}\log 2 + \frac{1}{2}\log 2 = 1, \\ H(\text{Sun coming up or not}) &= H_b(2^{-64}) = 2^{-64}\log 2^{64} + (1-2^{-64})\log \frac{1}{1-2^{-64}} \\ &\simeq 65 \times 2^{-64} \simeq 2^{-58} \end{aligned}$$

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.

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).

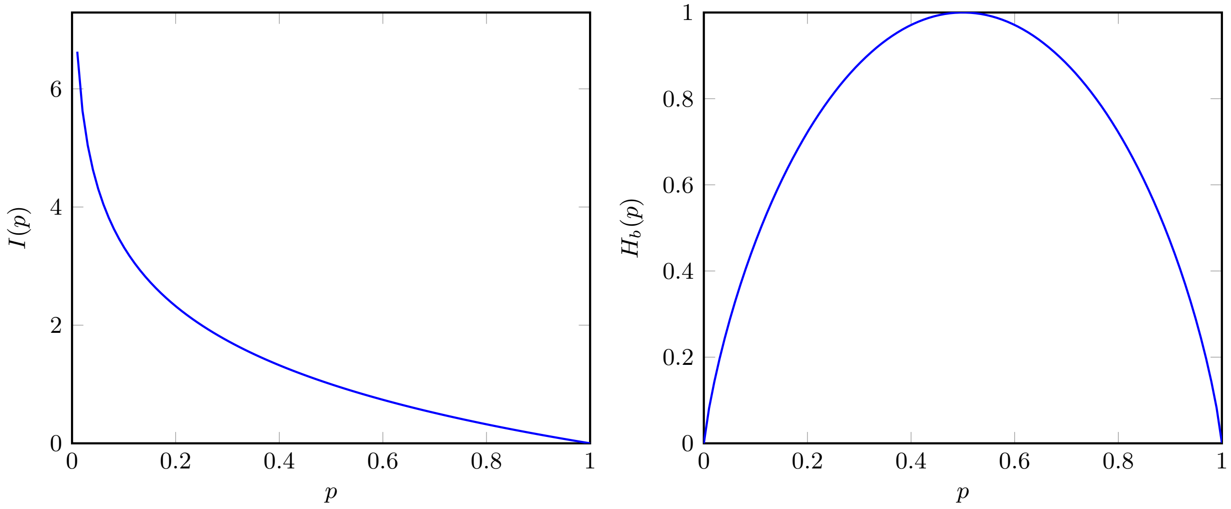


Figure 1.1: Self-information (left) for an event with probability p and binary entropy (right) for a Bernoulli RV with probability of success equal to p.

> “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.

1.6 Conditional entropy*

We can measure the information in multiple random variables also 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

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

Finally, relative entropy between two distributions p and q is defined as

$$KL(p||q) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)},$$

which can be viewed as a measure of difference between distributions.

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 Parameter Estimation

In order to find the distribution of the data, we need to estimate the parameters of the distribution. We have two frameworks for doing so:

- Frequentist methods: frequentists have different methods for estimation including:
    - *Maximum likelihood*
    - least squares
    - moment method
- 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: Introduction and Examples

Suppose data $\mathcal{D}$ is collected and is assumed to be derived from a distribution p with unknown parameter θ. Let the probability of observing $\mathcal{D}$, assuming θ, be denoted by $p(\mathcal{D};\theta)$. **Maximum likelihood** estimation finds θ that maximizes $p(\mathcal{D},\theta)$:

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(\mathcal{D};\theta)$$

The expression $p(\mathcal{D};\theta)$, viewed as a function of θ, is called the **likelihood**; hence the name maximum likelihood estimation. As shorthand, we use $L(\theta) = p(\mathcal{D};\theta)$ and $\ell(\theta) = \ln L(\theta)$, where $\ell(\theta)$ is the

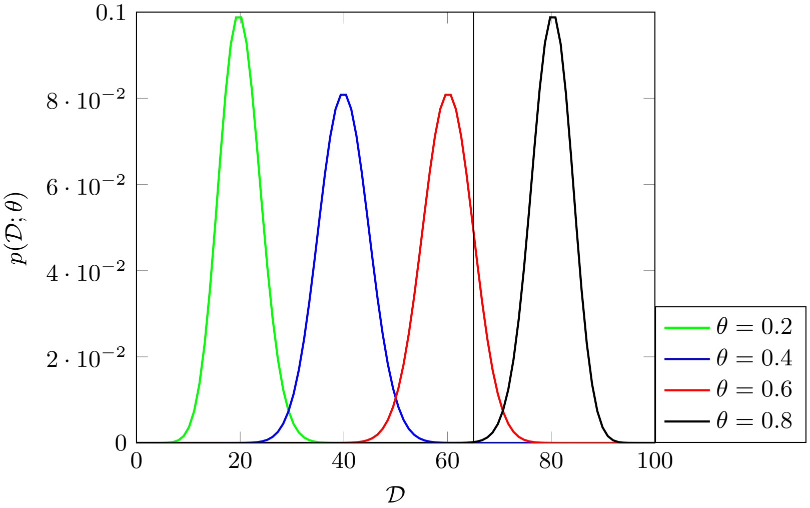
log-likelihood. Clearly, the value of θ that maximizes $L(\theta)$ is the same as the one that maximizes $\ell(\theta)$:

$$\hat{\theta}_{ML} = \arg\max_{\theta} \ell(\theta) = \arg\max_{\theta} \ln p(\mathcal{D};\theta)$$

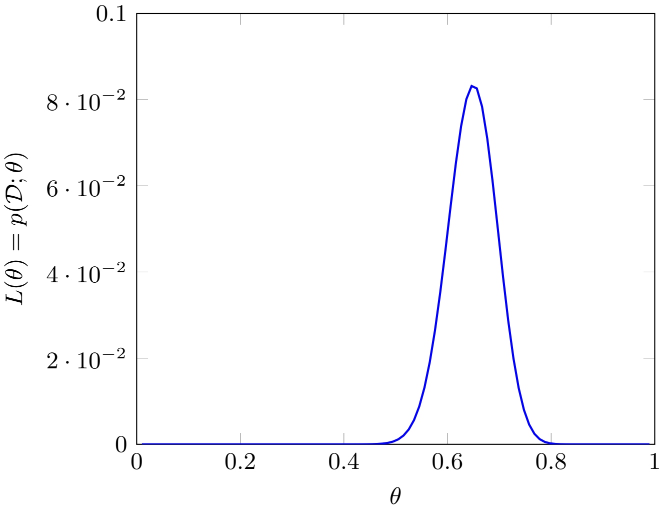
Example 2.1. In this example, we attempt to show the intuition behind maximum likelihood. Let T be a binary random variable such that $T = 1$ if there is traffic and $T = 0$ if there is no traffic. Suppose that data $\mathcal{D}$ collected over 100 days indicates that 65 days had no traffic. We have

$$\begin{aligned}\Pr(T = 0) &= \theta \\ p(\mathcal{D};\theta) &= \binom{100}{65}\theta^{65}(1-\theta)^{35}\end{aligned}$$

Let's try a few different choices for θ and see which one makes more sense. In the figure below, $p(\mathcal{D};\theta)$ is plotted for $\theta \in \{0.2, 0.4, 0.6, 0.8\}$. The vertical line indicates the observation, i.e., 65 days with no traffic. Which is a more appropriate value for θ?



If $\theta = 0.2$, the probability of 65 days with no traffic is very small. So observing $\mathcal{D} = 65$ 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, i.e., the maximum-likelihood estimate. In the figure below, $L(\theta) = p(\mathcal{D}, \theta)$ is plotted as a function of θ. This is the likelihood.



△

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

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

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

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

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 $l(\theta)$.

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

Note that this result is intuitive as it agrees with our observation that 65% of the days had no traffic.

Example 2.2 (**Parameters of the normal distribution**)**.** A device for measuring an unknown quantity μ is used n times producing values $\mathcal{D} = \boldsymbol{y} = (y_1, \dots, y_n)$. 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 $y_i \sim \mathcal{N}(\mu, \sigma^2)$. We consider the problem in two cases: μ is unknown but σ^2 is known; and both μ and σ are unknown.

- Known σ^2, unknown μ: We have

$$
\begin{aligned}
p(y_i;\mu) &= \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2\right)\\
p(\boldsymbol{y};\mu) &= \prod_{i=1}^{n}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2\right)\\
L(\mu) &= \prod_{i=1}^{n}\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2\right)\\
\ell(\mu) &= \sum_{i=1}^{n}\left(-\ln(\sigma\sqrt{2\pi})-\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2\right) \doteq -\frac{1}{2}\sum_{i=1}^{n}\left(\frac{y_i-\mu}{\sigma}\right)^2
\end{aligned}
$$

and so

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

- Unknown σ^2, 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

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

Solving this system of equations for μ and σ yields

$$
\begin{aligned}
\hat{\mu}_{ML} &= \frac{1}{n}\sum_{i=1}^{n}y_i = \bar{y},\\
\hat{\sigma}^2_{ML} &= \frac{1}{n}\sum_{i=1}^{n}(y_i-\bar{y})^2.
\end{aligned}
$$

△

2.3 Properties of Estimators

Maximum likelihood is just one way of estimating parameters. For example, in Example 2.2, we could choose the middle value among $y_1, \ldots, y_n$ as the estimate for μ. Given the fact that there are many estimators, how do we evaluate them and select one? In this section, we will see some of the evaluation criteria.

2.3.1 Estimation error and bias

For an estimator $\hat{\theta}$ of θ, assume $\mathcal{D}$ is collected. Then the error is given as

$$\hat{\theta}(\mathcal{D}) - \theta,$$

where $\hat{\theta}(\mathcal{D})$ is the estimate based on data $\mathcal{D}$.

For a given estimation task that is performed once, since we do not know the true value, we cannot find $\hat{\theta}(\mathcal{D}) - \theta$. Even if we know the true value, the error is the result of only one experiment and does not tell us much about the general behavior of the estimator.

However, we can think of the thought experiment in which estimation is performed many times and consider the behavior of the estimator and its error. For example, we may consider whether the result would be generally an overestimate or an underestimate? The key point in answering such questions is that *the estimate itself is a random value because each time we perform the estimation task, new data samples are obtained and these are random, following a certain distribution.* So for example, we can talk about the expected error. In other words, since $\mathcal{D}$ is random (although its distribution is the same in each experiment), so is $\hat{\theta}(\mathcal{D})$.

So we can consider the expected error, known as **bias**,

$$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}(\mathcal{D}) - \theta] \tag{2.3}$$

The expected value is taken over $\mathcal{D}$. However, the dependence on data is often implicit and we write

$$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta \tag{2.4}$$

Bias of the estimator tells us that 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 2.3. Given n samples $y_1, \ldots, y_n$ from a distribution with mean μ and variance σ^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

$$\mathbb{E}[\hat{\mu}] = \mathbb{E}[\bar{y}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} y_i\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[y_i] = \frac{1}{n}\cdot n \cdot \mathbb{E}[y_1] = \mu$$

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

$$\mathbb{E}\left[\hat{\sigma}^2\right] = \frac{n-1}{n}\sigma^2$$

and the bias of estimating σ^2 is

$$\mathbb{E}\big[\hat{\sigma}^2\big] - \sigma^2 = -\frac{1}{n}\sigma^2.$$

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

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

△

Example 2.4. [1, Example 2.8.2] An urn has N balls, numbered $1, 2, ..., N$. Suppose however that N is unknown to us. We pick one random ball from the urn and the number on the ball is y. We Estimate N using maximum likelihood. First, for $p(y; N)$ we have

$$p(y; N) = \begin{cases} \frac{1}{N} & y \le N, \\ 0 & y > N. \end{cases}$$

and thus

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

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

$$\mathbb{E}[\hat{N}_{ML}] = \mathbb{E}[y] = \sum_{i=1}^{N} i \cdot \frac{1}{N} = \frac{N+1}{2},$$
$$\mathrm{Bias}(\hat{N}_{ML}) = \frac{N+1}{2} - N = -\frac{N-1}{2},$$

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

Example 2.5 (Linear unbiased estimator). Can we design an unbiased estimator for Example 2.4? There are many options but for simplicity we may choose an estimator that is linear in the data, in particular, one of the form

$$\hat{N}_L = ay + b.$$

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

$$\mathbb{E}[\hat{N}_L] = a\,\mathbb{E}\,y + b = a\frac{N+1}{2} + b.$$

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

$$\hat{N}_L = 2y - 1.$$

△

Example 2.6 (**Survival of Humanity (!)**)**.** The human species will eventually die out. We use the two methods to estimate the total number of humans N who will ever live. Let humans be enumerated as $h_1, h_2, ..., h_y, ..., h_N$, where h_1 represents Adam, h_2 represents Eve, h_y represents you, and h_N represents the last human to live. Assuming that your birth order is random, this is similar to the urn in Example 2.4.

Assuming that 100 billion have been born so far, we have $\hat{N}_{ML} = 100$ billion and $\hat{N}_L = 200$ billion. The ML estimates 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. △

Exercise 2.7. Given iid data $\mathcal{D} = (y_1, \ldots, y_n), n \geq 3$, with mean θ, find the bias of each of the following estimators,

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

△

2.3.2 Mean squared error and variance

Example 2.8. 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." △

The mean squared error (MSE) is defined as

$$\mathrm{MSE}(\hat{\theta}) = \mathbb{E}\left[\left(\hat{\theta} - \theta\right)^2\right]$$

Note that

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

and hence

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

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

Example 2.9. Consider data $\mathcal{D} = \{y_1, ..., y_n\}$, where y_i are iid with distribution $\mathcal{N}(\mu, \sigma^2)$. The ML estimator for the mean $\hat{\mu}_{ML} = \bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$ is unbiased. We have

$$\text{MSE}(\hat{\mu}_{ML}) = \text{Var}(\bar{y}) = \frac{\sigma^2}{n}.$$

△

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

Exercise 2.10. For the estimators in Exercise 2.7, find the MSE, assuming the variance is σ^2. △

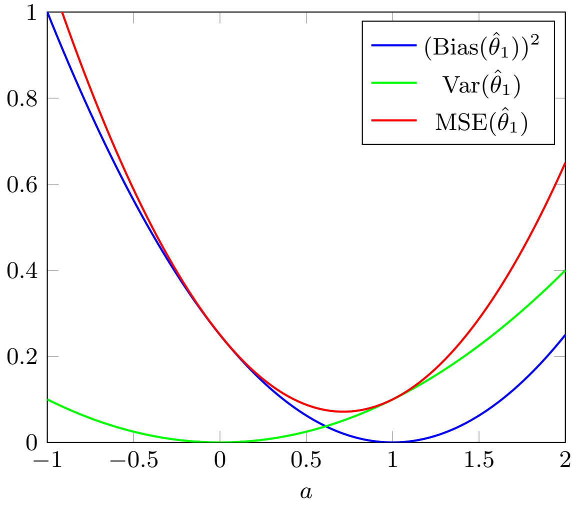
Exercise 2.11 (Bias-variance trade-off)**.** Given iid data $\mathcal{D} = (y_1, \dots, y_n), n \geq 3$, with mean θ and variance σ^2, the MSE of

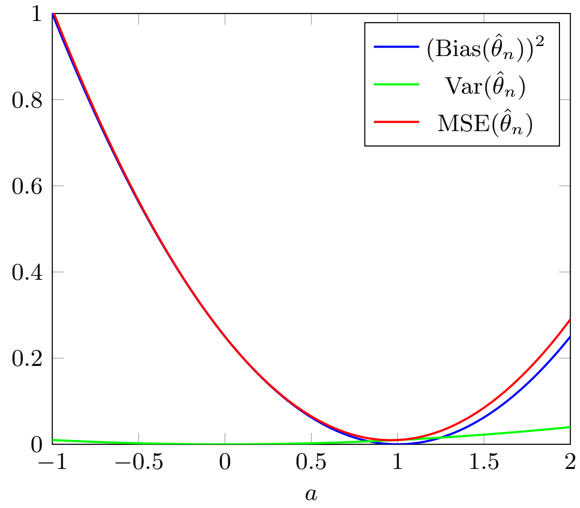
$$\hat{\theta}_1 = a y_1,$$
$$\hat{\theta}_n = a\bar{y} = \frac{a}{n}\sum_{i=1}^n y_i,$$

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

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

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 $n = 10$. 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

An estimator $\hat{\theta}_n$ based on n samples is said to be **consistent** if $\hat{\theta}_n \to \theta$ as $n \to \infty$. More precisely, for all $\epsilon > 0$, we need

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

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

Example 2.12. The ML and linear estimators described in Examples 2.4 and 2.5 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 $\mathcal{D} = (y_1, y_2, \ldots, y_n)$. Define

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

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

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

For the ML estimator, we have (why?)

$$\hat{N}_{ML,n} = \max_i y_i.$$

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

- As $n \to \infty$, by LLN, $\bar{y}$ converges to the mean of the distribution, i.e., $\mathbb{E}\, y_1 = \frac{N+1}{2}$. Hence, $\hat{N}_{L,n} \to 2\frac{N+1}{2} - 1 = N$.
- For the ML estimator, as $n \to \infty$, at some point, we will pick the ball numbered N and so we will eventually have $\hat{N}_{ML} = N$.

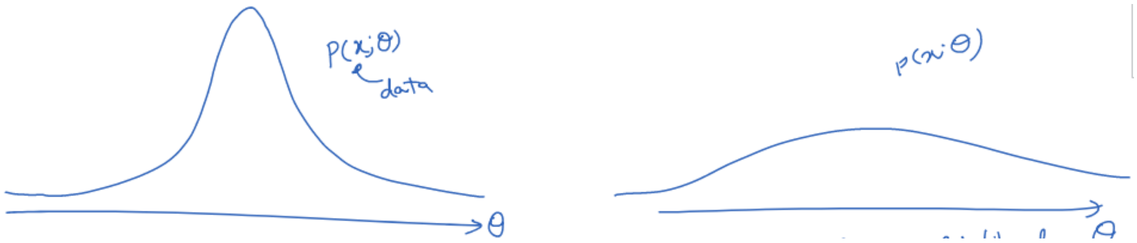


Figure 2.1: The log-likelihood on the left demonstrates strong dependence on θ compared to the one on the right.

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

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 depends on θ. If the dependence is strong, i.e., for values of θ other than the true value, 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 week, 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 x, i.e., $\mathcal{D} = x$. The sharpness of the log-likelihood $\ell(\theta)$ can be quantified as

$$-\frac{\partial^2 \ell(\theta)}{\partial\theta^2} = -\frac{\partial^2 \ln p(x;\theta)}{\partial\theta^2}. \tag{2.5}$$

Given the randomness of data the above quantity is random. So to average over the data, we define

$$I(\theta) = -\mathbb{E}\left[\frac{\partial^2 \ell(\theta)}{\partial\theta^2}\right] = -\int \frac{\partial^2 \ln p(x;\theta)}{\partial\theta^2} p(x;\theta)dx,$$

which is called the **Fisher Information**.

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

Theorem 2.13 (**CRLB**)**.** *Given that the log-likelihood $\ell(\theta)$ satisfies a certain regularity condition*[1]*, the variance of any unbiased estimator $\hat{\theta}$ of θ satisfies*

$$\mathrm{Var}(\hat{\theta}) \geq \frac{1}{I(\theta)}.$$

[1] The regularity condition is $\mathbb{E}\left[\frac{\partial \ell(\theta)}{\partial\theta}\right] = 0$, for all θ.

If an estimator achieves the CRLB, i.e., $\mathrm{Var}(\hat{\theta}) = 1/I(\theta)$, 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

$$\mathrm{Var}(\hat{\theta}_n) \geq \frac{1}{nI(\theta)}. \tag{2.6}$$

Example 2.14. In Example 2.2, where we estimated the mean of a Gaussian distribution with known σ^2 based on n iid samples $y_1, \ldots, y_n$, 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,

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

Then regularity condition is satisfied since

$$\mathbb{E}\left[\frac{\partial \ell(\mu)}{\partial \mu}\right] = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathbb{E}[y_i - \mu] = 0,$$

for all μ. Furthermore,

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

Based on the CRLB, the variance of the estimator satisfies

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

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

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

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

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.

Chapter 3

Bayesian Parameter Estimation

3.1 From Prior to Posterior

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

Specifically, the steps of Bayesian estimation of a parameter θ are:

1. Identifying the prior distribution, $p(\theta)$
2. Collecting data and forming the likelihood: $p(\mathcal{D}|\theta)$
3. Finding the posterior distribution $p(\theta|\mathcal{D})$ as

$$p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})} \tag{3.1}$$

Normalizing distributions. Finding the posterior distribution requires computing the integral $p(\mathcal{D}) = \int_\theta p(\theta)p(\mathcal{D}|\theta)d\theta$. 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, $p(\mathcal{D})$ is one such term. So we often first find a function proportional to $p(\theta|\mathcal{D})$ as

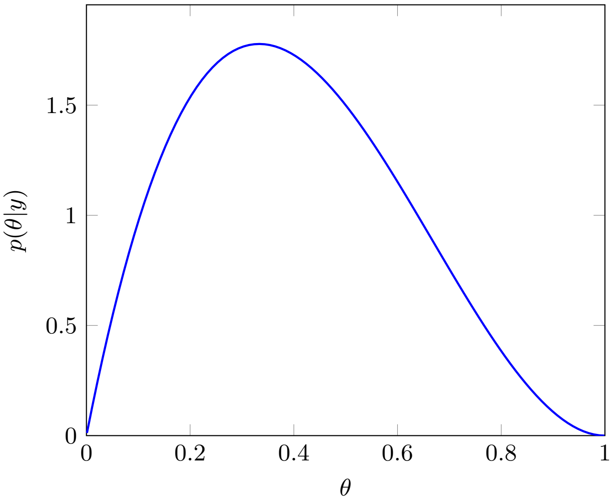
$$p(\theta|\mathcal{D}) \propto p(\theta)p(\mathcal{D}|\theta),$$

where we can also drop constant terms in θ from $p(\theta)$ and $p(\mathcal{D}|\theta)$. 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, 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 $p(\theta|\mathcal{D})$.

Example 3.1. Let θ denote the unknown parameter of a geometric random variable y, where $p(y) = \theta(1-\theta)^{y-1}$. Suppose that we observe y. 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

$$\begin{aligned} p(\theta) &= 1 \\ p(y|\theta) &= \theta(1-\theta)^{y-1} \\ p(\theta|y) &\propto \theta(1-\theta)^{y-1} \end{aligned}$$

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



△

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 y is observed to be $y = 1$. Assuming a uniform prior, find the posterior distribution of θ, i.e., $\theta|y = 1$. △

Example 3.3. The probability of success in a Bernoulli experiment is θ, which we would like to estimate. We show success in the ith 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 $\mathcal{D} = (s, f)$. The likelihood is

$$p(\mathcal{D}|\theta) = \binom{n}{s}\theta^s(1-\theta)^f \tag{3.2}$$

- The posterior distribution:

$$p(\theta|\mathcal{D}) \propto 1 \cdot \theta^s(1-\theta)^f = \theta^s(1-\theta)^f \tag{3.3}$$

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

$$p(\theta|\mathcal{D}) \sim \text{Beta}(s+1, f+1).$$

△

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 example above.

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

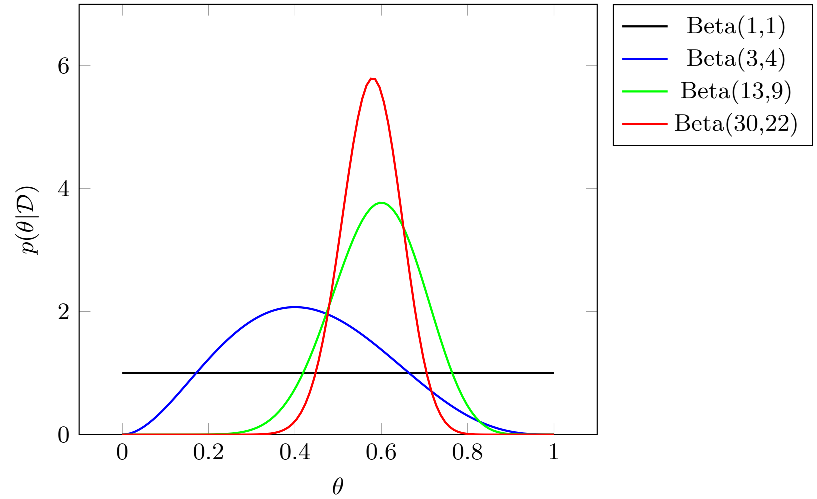
$$\int_{0.4}^{0.6} p(\theta|\mathcal{D}) d\theta \tag{3.4}$$

Example 3.4 (Consecutive Bayesian updating)**.** Continuing the previous example, suppose that we collect more data $\mathcal{D}' = (s', f')$, 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

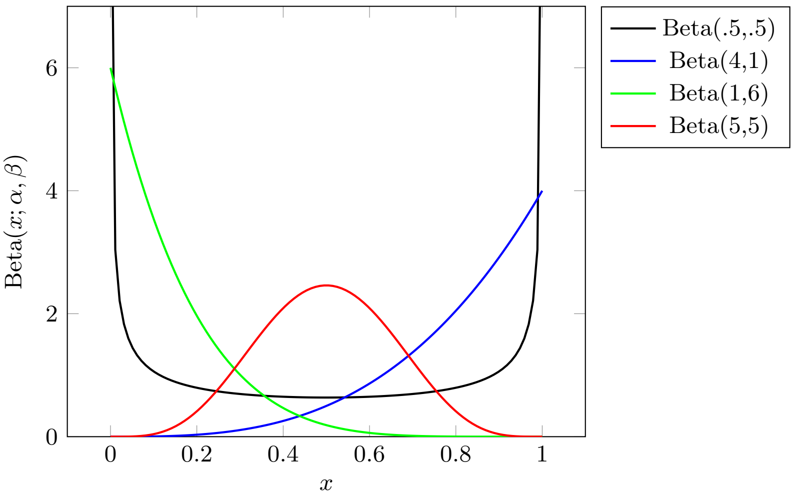
$$\begin{aligned} p(\mathcal{D}'|\theta) &= \binom{s'+f'}{s'}\theta^{s'}(1-\theta)^{f'} \\ p(\theta|\mathcal{D}') &\propto \theta^s(1-\theta)^f\theta^{s'}(1-\theta)^{f'} \\ &= \theta^{s+s'}(1-\theta)^{f+f'} \\ \theta|\mathcal{D}' &\sim \text{Beta}(s+s'+1, f+f'+1). \end{aligned} \tag{3.5}$$

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. △



Example 3.6. Suppose that $y \sim \text{Poi}(\lambda)$ and we intend to estimate λ based on n iid samples $y_1^n = (y_1, \ldots, y_n)$. We assume that the prior for λ is given as $p(\lambda) \sim \text{Gamma}(\lambda; \alpha, \beta) \propto \lambda^{\alpha-1} e^{-\beta\lambda}$. We have

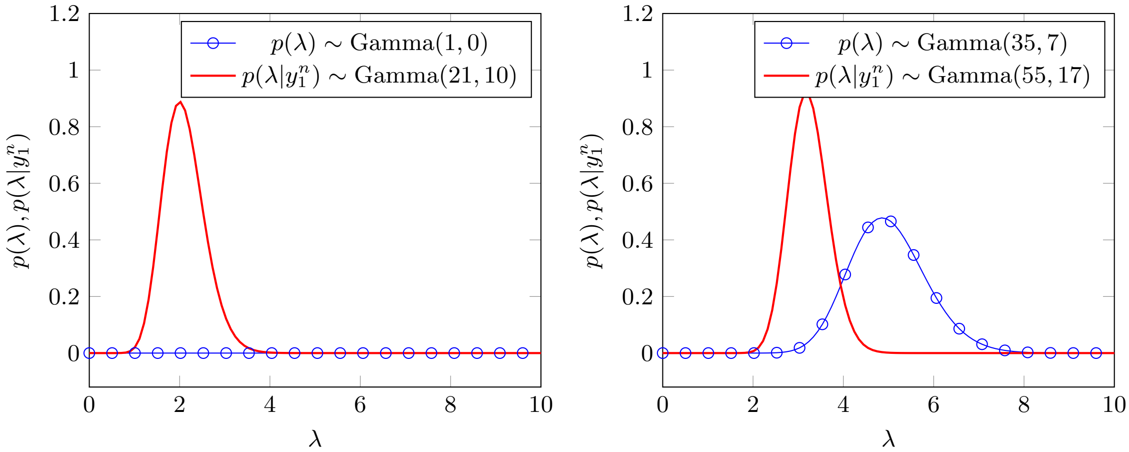
$$
\begin{aligned}
p(\lambda) &\propto \lambda^{\alpha-1} e^{-\beta\lambda} \\
p(y_1^n|\lambda) &= \prod_{i=1}^{n} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} \propto \prod_{i=1}^{n} \lambda^{y_i} e^{-\lambda} = e^{-n\lambda} \lambda^{n\bar{y}},
\end{aligned}
$$

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

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

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 $p(\theta|\mathcal{D})$ is useful since it provides the probability for different values for θ. But sometimes we want to estimate θ with a single value $\hat\theta = \hat\theta(\mathcal{D})$ as a function of data, similar to maximum likelihood. The best choice for $\hat\theta$ then depends on how we characterize the estimation error:

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

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

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

$$\begin{aligned}\mathbb{E}[(\hat\theta-\theta)^2|\mathcal{D}] &= \mathbb{E}[((\hat\theta-\bar\theta)+(\bar\theta-\theta))^2|\mathcal{D}]\\ &= \mathbb{E}[(\hat\theta-\bar\theta)^2+2(\hat\theta-\bar\theta)(\bar\theta-\theta)+(\bar\theta-\theta)^2|\mathcal{D}]\\ &= \mathbb{E}[(\hat\theta-\bar\theta)^2|\mathcal{D}]+\mathbb{E}[(\bar\theta-\theta)^2|\mathcal{D}]\\ &= \mathbb{E}[(\hat\theta-\bar\theta)^2|\mathcal{D}]+\mathrm{Var}(\theta|\mathcal{D})\\ &\geq \mathrm{Var}(\theta|\mathcal{D}),\end{aligned}$$

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

Example 3.7. Generalizing Example 3.3 by assuming $p(\theta) \sim \text{Beta}(\alpha, \beta)$, we obtain $p(\theta|\mathcal{D}) \sim$

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

$$\begin{aligned}\text{Mean } &= \frac{s+\alpha}{s+f+\alpha+\beta},\\ \text{Median } &\simeq \frac{s+\alpha-1/3}{s+f+\alpha+\beta-2/3},\\ \text{Mode } &= \frac{s+\alpha-1}{s+f+\alpha+\beta-2}.\end{aligned}$$

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. △

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, $p(y_{n+1}|y_1^n)$. This distribution is referred to as *predictive posterior*. We have

$$\begin{aligned}p(y_{n+1}|y_1^n) &= \int p(y_{n+1},\theta|y_1^n)d\theta\\ &= \int p(\theta|y_1^n)p(y_{n+1}|\theta,y_1^n)d\theta\\ &= \int p(\theta|y_1^n)p(y_{n+1}|\theta)d\theta,\end{aligned}$$

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

$$p(y_{n+1}=1|y_1^n)=\int \theta p(\theta|y_1^n)=\mathbb{E}[\theta|y_1^n]=\frac{s+1}{s+f+2},$$

where we have used the facts that $p(y_{n+1}=1|\theta)=\theta$ and that the mean of Beta$(s+1,f+1)$ is $\frac{s+1}{s+f+2}$. △

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, given that y_1^n influences y_{n+1} through θ. Recall that

$$\mathbb{E}[\mathbb{E}[Y|X]]=\mathbb{E}[Y], \qquad \mathbb{E}[\mathbb{E}[Y|X,Z]|Z]=\mathbb{E}[Y|Z].$$

Thus,

$$\mathbb{E}[y_{n+1}|y_1^n]=\mathbb{E}[\mathbb{E}[y_{n+1}|\theta,y_1^n]|y_1^n]=\mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n], \tag{3.6}$$

where the last step follows from the fact that $y_{n+1}\perp\!\!\!\perp y_1^n|\theta$.

Exercise 3.9. Find $\mathbb{E}[y_{n+1}|y_1^n]$ in Example 3.6. △

3.4 Gaussian Prior and Likelihood

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

$$p(y_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-\theta)^2}{2\sigma^2}} \tag{3.7}$$

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

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

But since the integral $\int_{-\infty}^{\infty} 1 d\theta = \infty$, 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

$$\begin{aligned}
p(\theta|y_1^n) &\propto p(y_1^n|\theta) \cdot 1 \\
&\propto \exp\left(-\frac{\sum_{i=1}^n (y_i-\theta)^2}{2\sigma^2}\right) \\
&\propto \exp\left(-\frac{\sum_{i=1}^n (\theta^2 - 2y_i\theta + y_i^2)}{2\sigma^2}\right) \\
&\propto \exp\left(-\frac{\theta^2 - 2\bar{y}\theta}{2\sigma^2/n}\right) \\
&\propto \exp\left(-\frac{(\theta-\bar{y})^2}{2\sigma^2/n}\right) \\
\theta|y_1^n &\sim \mathcal{N}(\bar{y}, \sigma^2/n).
\end{aligned}$$

For the expected value of the next sample,we have

$$\mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n] = \mathbb{E}[\mathbb{E}[\theta]|y_1^n] = \bar{y}.$$

We can see more explicitly as well,

$$\begin{aligned}\mathbb{E}[y_{n+1}|y_1^n] &= \int y_{n+1}p(y_{n+1}|y_1^n)dy_{n+1}\\ &= \int y_{n+1}\int p(y_{n+1},\theta|y_1^n)d\theta dy_{n+1}\\ &= \int y_{n+1}\int p(y_{n+1}|\theta)p(\theta|y_1^n)d\theta dy_{n+1}\\ &= \int p(\theta|y_1^n)\int y_{n+1}p(y_{n+1}|\theta)dy_{n+1}d\theta\\ &= \int \theta p(\theta|y_1^n)d\theta\\ &= \mathbb{E}[\theta|y_1^n]\\ &= \bar{y}.\end{aligned}$$

△

Exercise 3.11. Prove that

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

△

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

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

$$\begin{aligned}\mu &\sim \mathcal{N}(\mu_0, \tau_0^2)\\ y_i &\sim \mathcal{N}(\mu, \sigma^2)\end{aligned}$$

where μ_0 and τ_0^2 are the prior mean and variance, respectively, and σ^2 is known. We have

$$\begin{aligned}p(\mu|y_1^n) &\propto p(\mu)p(y_1^n|\mu)\\ &\propto \frac{1}{\sigma\tau_0}\exp\left(-\frac{\sum_{i=1}^n(y_i-\mu)^2}{2\sigma^2} - \frac{(\mu-\mu_0)^2}{2\tau_0^2}\right)\end{aligned}$$

The following claim will be useful.

Claim: If $p(x) \propto e^{-f(x)}$, where $f(x) = ax^2 - bx + c$ with $a > 0$, then $x \sim \mathcal{N}\left(\frac{b}{2a}, \frac{1}{2a}\right)$.

Proof: Since

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

we have

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

proving the claim.

Returning to our problem:

$$a = \frac{n}{2\sigma^2} + \frac{1}{2\tau_0^2}, \quad b = \frac{n\bar{y}}{\sigma^2} + \frac{\mu_0}{\tau_0^2}.$$

Hence

$$\mu | y_1^n \sim \mathcal{N}\left(\frac{\frac{n\bar{y}}{\sigma^2} + \frac{\mu_0}{\tau_0^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}, \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}} \right).$$

△

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: $(y = \sum_{i=1}^n y_i)$

$$\begin{aligned} p(y_i|\theta) &= \theta^{y_i}(1-\theta)^{1-y_i} && \mathrm{Ber}(\theta) \\ p(y_1^n|\theta) &= \theta^{y}(1-\theta)^{n-y} && \\ p(\theta) &\propto \theta^{\alpha-1}(1-\theta)^{\beta-1} && \mathrm{Beta}(\alpha,\beta) \\ p(\theta|y) &\propto \theta^{y+\alpha-1}(1-\theta)^{n-y+\beta-1} && \mathrm{Beta}(y+\alpha, n-y+\beta) \end{aligned}$$

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

$$\begin{aligned} p(y_i|\theta) &= \theta \exp(-\theta y_i) && \mathrm{Exp}(\theta) = \mathrm{Gamma}(1,\theta) \\ p(y_1^n|\theta) &= \theta^n \exp(-\theta y) && \\ p(\theta) &= \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}\exp(-\beta\theta) && \mathrm{Gamma}(\alpha,\beta) \\ p(\theta|y_1^n) &\propto \theta^{n+\alpha-1}\exp(-(y+\beta)\theta) && \mathrm{Gamma}(n+\alpha, y+\beta) \end{aligned}$$

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

$$\begin{aligned} p(y_i|\theta) &\propto \exp\left(\frac{(y_i-\theta)^2}{2\sigma^2}\right) && \mathcal{N}(\theta,\sigma^2) \\ p(y_1^n|\theta) &\propto \exp\left(\frac{\sum_{i=1}^n (y_i-\theta)^2}{2\sigma^2}\right) && \\ p(\theta) &\propto \exp\left(\frac{(\theta-\mu_0)^2}{2\tau_0^2}\right) && \mathcal{N}(\mu_0,\tau_0^2) \\ p(\theta|y_1^n) &\propto \exp\left(\frac{(\theta-\mu_1)^2}{2\tau_1^2}\right) && \mathcal{N}(\mu_1,\tau_1^2), \end{aligned}$$

where

$$\mu_1 = \frac{\frac{1}{\tau_0}\mu_0 + \frac{1}{\sigma^2/n}\bar{y}}{\frac{1}{\tau_0} + \frac{1}{\sigma^2/n}},$$
$$\frac{1}{\tau_1^2} = \frac{1}{\tau_0^2} + \frac{1}{\sigma^2/n}.$$

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 x with parameter θ, $p(x|\theta)$ is said to be from the exponential family if it has the following form

$$p(x|\theta) = \exp\bigl(a(x)^T b(\theta) + f(x) + g(\theta)\bigr),$$

where a, b, x, 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

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

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

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

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

$$\begin{aligned}
p(y_i|\theta) &\propto \exp\bigl(a(y_i)^T b(\theta) + g(\theta)\bigr) \\
p(y_1^n|\theta) &\propto \exp\bigl(t(y_1^n)^T b(\theta) + ng(\theta)\bigr) \\
p(\theta) &\propto \exp\bigl(\nu^T b(\theta) + mg(\theta)\bigr) && \mathit{Dist}(\nu, m) \\
p(\theta|y_1^n) &\propto \exp\bigl((\nu + t(y_1^n))^T b(\theta) + (m+n)g(\theta)\bigr) && \mathit{Dist}(\nu + t(y_1^n), m+n),
\end{aligned}$$

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 3.13. The likelihood for a Bernoulli observation is

$$\begin{aligned}
p(y_i|\theta) &= \theta^{y_i}(1-\theta)^{1-y_i} \\
&= \exp(y_i \ln\theta + (1-y_i)\ln(1-\theta)) \\
&= \exp\left(y_i \ln\frac{\theta}{1-\theta} + \ln(1-\theta)\right).
\end{aligned}$$

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,

$$\begin{aligned}
p(y_1^n|\theta) &= \exp\left(y \ln\frac{\theta}{1-\theta} + n\ln(1-\theta)\right) \\
p(\theta) &= \exp\left(\nu \ln\frac{\theta}{1-\theta} + m\ln(1-\theta)\right) \\
&= \theta^{\nu}(1-\theta)^{m-\nu}, \qquad \text{Beta}(\nu+1, m-\nu+1) \\
p(\theta|y_1^n) &= \exp\left((\nu+y) \ln\frac{\theta}{1-\theta} + (m+n)\ln(1-\theta)\right), \\
&\qquad \text{Beta}(\nu+y+1, m+n-\nu-y+1)
\end{aligned}$$

△

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 Review of Linear Algebra

For two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, the **inner product** $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ of $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^n x_i y_i.$$

where $\boldsymbol{x}^T$ is the transpose of $\boldsymbol{x}$.

The **length** or the ℓ_2 norm of a vector $\boldsymbol{x}$ is $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$ and we have $\|\boldsymbol{x}\|_2^2 = \boldsymbol{x}^T\boldsymbol{x}$. Let α be the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$. Then $\boldsymbol{x}^T\boldsymbol{y} = \|\boldsymbol{x}\|\|\boldsymbol{y}\| \cos\alpha$. If $\boldsymbol{x}^T\boldsymbol{y} = 0$, then the two are called **orthogonal**.

For a collection of vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$, a **linear combination** of these is any vector of the form $a_1\boldsymbol{v}_1 + \cdots + a_m\boldsymbol{v}_m, a_i \in \mathbb{R}$. The set of all linear combinations of $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ is their **span** and denoted as $\mathrm{Span}\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_m\}$. This is a **subspace** (think line, plane, or the whole space). For a matrix A, the span of the columns of A is the **column space** of A.

The vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ 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 $a_1\boldsymbol{v}_1 + \cdots + a_m\boldsymbol{v}_m = 0$ are $a_1, \dots, a_m = 0$. In particular, the columns of a matrix A are linearly independent if and only if the only vector $\boldsymbol{a}$ satisfying $A\boldsymbol{a} = 0$ is $\boldsymbol{a} = 0$.

The **inverse** of a square matrix A is a matrix A^{-1} such that $AA^{-1} = A^{-1}A = I$, where 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 A, the following are equivalent:

- It is invertible.
- For all distinct vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, we have $A\boldsymbol{a} \neq A\boldsymbol{b}$.
- The only solution to $A\boldsymbol{x} = 0$ is $\boldsymbol{x} = 0$.
- Its columns are linearly independent.
- Its determinant $|A|$ is nonzero. (We also have $|A^{-1}| = \frac{1}{|A|}$.)

Given a subspace S (e.g., a plane or the column space of a matrix) and a vector $\boldsymbol{y}$, let $\hat{\boldsymbol{y}}$ be the vector in the subspace that is closest to $\boldsymbol{y}$. That is, we find $\hat{\boldsymbol{y}} \in S$ such that $\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|$ is minimized. Then $\hat{\boldsymbol{y}}$ is called the **projection** of $\boldsymbol{y}$ onto the subspace S.

Lemma 4.1. *Let $\hat{\boldsymbol{y}}$ be the projection of a vector $\boldsymbol{y}$ onto a subspace S. Then $\boldsymbol{y} - \hat{\boldsymbol{y}}$ is orthogonal to every vector in S.*

Proof. Suppose that this is not the case. Then there is a nonzero vector $\boldsymbol{v} \in S$ such that $(\boldsymbol{y}-\hat{\boldsymbol{y}})^T\boldsymbol{v} \neq 0$. We will show that this contradicts the minimality of $\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|$. For any $a \in \mathbb{R}$,

$$\begin{aligned}\|\boldsymbol{y} - \hat{\boldsymbol{y}} - a\boldsymbol{v}\|_2^2 &= (\boldsymbol{y} - \hat{\boldsymbol{y}} - a\boldsymbol{v})^T(\boldsymbol{y} - \hat{\boldsymbol{y}} - a\boldsymbol{v})\\ &= \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 - 2a\boldsymbol{v}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}) + a^2\|\boldsymbol{v}\|_2^2.\end{aligned}$$

This is a convex function in a. So setting the derivative to 0 gives the value of a that minimizes the error:

$$\frac{\partial}{\partial a}\|\boldsymbol{y} - \hat{\boldsymbol{y}} - a\boldsymbol{v}\|_2^2 = -2\boldsymbol{v}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}) + 2a\|\boldsymbol{v}\|_2^2 = 0 \Rightarrow a = \frac{\boldsymbol{v}^T(\boldsymbol{y} - \hat{\boldsymbol{y}})}{\|\boldsymbol{v}\|_2^2} \neq 0.$$

Let

$$\hat{\boldsymbol{y}}' = \hat{\boldsymbol{y}} + \frac{\boldsymbol{v}^T(\boldsymbol{y} - \hat{\boldsymbol{y}})}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v},$$

and note that $\hat{\boldsymbol{y}}'$ is also in S but it is closer to $\boldsymbol{y}$ contradicting the optimality of $\hat{\boldsymbol{y}}$. □

4.2 Random vectors

A **random vector** is a vector of random variables. Consider the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

The **expected value** of $\boldsymbol{x}$ is

$$\mathbb{E}\,\boldsymbol{x} = \begin{pmatrix} \mathbb{E}\,x_1 \\ \vdots \\ \mathbb{E}\,x_m \end{pmatrix}.$$

The **correlation matrix** of $\boldsymbol{x}$ and $\boldsymbol{y}$ is the $m \times n$ matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{y}^T]$, whose i,jth element is $\mathbb{E}[x_i y_j]$. The **cross-covariance matrix** of $\boldsymbol{x}$ and $\boldsymbol{y}$ is $\operatorname{Cov}(\boldsymbol{x}, \boldsymbol{y})$ is the matrix $\mathbb{E}[(\boldsymbol{x} - \mathbb{E}\,\boldsymbol{x})^T(\boldsymbol{y} - \mathbb{E}\,\boldsymbol{y})^T]$, whose i, jth element is $\operatorname{Cov}(x_i, y_j)$. The covariance of a vector $\boldsymbol{x}$ is $\operatorname{Cov}(\boldsymbol{x}) = \operatorname{Cov}(\boldsymbol{x}, \boldsymbol{x})$. The **conditional expectation** $\mathbb{E}[\boldsymbol{x}|\boldsymbol{y}]$ of $\boldsymbol{x}$ given $\boldsymbol{y}$ is a vector whose ith element is $\mathbb{E}[x_i|\boldsymbol{y}]$.

For matrices A, B, deterministic vectors $\boldsymbol{a}, \boldsymbol{b}$, and random vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{z}$, we have [1]

- $\mathbb{E}[A\boldsymbol{x} + \boldsymbol{a}] = A\,\mathbb{E}\,\boldsymbol{x} + \boldsymbol{a}$
- $\operatorname{Cov}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}[\boldsymbol{x}\boldsymbol{y}^T] - \mathbb{E}\,\boldsymbol{x}\,\mathbb{E}\,\boldsymbol{y}^T$
- $\mathbb{E}[(A\boldsymbol{x})(B\boldsymbol{y})^T] = A\,\mathbb{E}[\boldsymbol{x}\boldsymbol{y}^T]B^T$
- $\operatorname{Cov}(A\boldsymbol{x} + \boldsymbol{a}, B\boldsymbol{y} + \boldsymbol{b}) = A\operatorname{Cov}(\boldsymbol{x}, \boldsymbol{y})B^T$
- $\operatorname{Cov}(A\boldsymbol{x} + \boldsymbol{a}) = A\operatorname{Cov}(\boldsymbol{x})A^T$
- $\operatorname{Cov}(\boldsymbol{w} + \boldsymbol{x}, \boldsymbol{y} + \boldsymbol{z}) = \operatorname{Cov}(\boldsymbol{w}, \boldsymbol{y}) + \operatorname{Cov}(\boldsymbol{w}, \boldsymbol{z}) + \operatorname{Cov}(\boldsymbol{x}, \boldsymbol{y}) + \operatorname{Cov}(\boldsymbol{x}, \boldsymbol{z})$

4.3 Gaussian Random Vectors (Joint Gaussian 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

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

Definition 4.2. 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 are jointly Gaussian if the vector obtained by concatenating them is jointly Gaussian.

Example 4.3. For example if $\begin{pmatrix} x \\ y \end{pmatrix}$ is a Gaussian vector, then $z = 2x+3y$ is Gaussian. Furthermore,

$$\begin{aligned}
\mathbb{E}[z] &= 2\,\mathbb{E}[x] + 3\,\mathbb{E}[y], \\
\operatorname{Var}(z) &= \operatorname{Cov}(2x+3y, 2x+3y) = 4\operatorname{Cov}(x,x) + 12\operatorname{Cov}(x,y) + 9\operatorname{Cov}(y,y) \\
&= 4\operatorname{Var}(x) + 12\operatorname{Cov}(x,y) + 9\operatorname{Var}(y),
\end{aligned}$$

which completely characterizes the distribution of z. △

For an m dimensional Gaussian vector $\boldsymbol{x}$, the elements of $\boldsymbol{x}$ are **independent** if and only if the covariance matrix is diagonal.

For an m-dimensional Gaussian random vector $\boldsymbol{x}$, assuming that the covariance matrix $K = \operatorname{Cov}(\boldsymbol{x})$ is invertible, we have

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{m/2}|K|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T K^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right).$$

4.4 Maximum likelihood for Gaussian Random Vectors

Let $\boldsymbol{z}$ be a Gaussian random vector of dimension d with mean $\boldsymbol{\mu}$ and covariance matrix K. If K is invertible, the pdf of $\boldsymbol{z}$ can be written as

$$p(\boldsymbol{z}|\boldsymbol{\mu}, K) = \frac{1}{\sqrt{(2\pi)^d |K|}} \exp\left(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^T K^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right),$$
$$\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{z}], \quad K = \mathbb{E}[(\boldsymbol{z}-\boldsymbol{\mu})(\boldsymbol{z}-\boldsymbol{\mu})^T],$$

where $|K|$ is the determinant of K.

Given a set of n iid samples $\mathcal{D} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n\}$, where each $\boldsymbol{z}_i$ is a d-dimensional vector, how can we estimate $\boldsymbol{\mu}$ and K using maximum likelihood? Estimating these quantities allows us to find the distribution. In particular, if we can view z_d as the output variable and $z_1, \ldots, z_{d-1}$ as input variables, then we can estimate z_d based on $z_1, \ldots, z_{d-1}$ as $\mathbb{E}[z_d|z_1, \ldots, z_{d-1}]$.

To estimate $\boldsymbol{\mu}$ and K, we write

$$\begin{aligned}\ell(\boldsymbol{\mu}, K) = \ln p(\mathcal{D}; \boldsymbol{\mu}, K) &= \sum_{i=1}^n \ln p(\boldsymbol{z}_i; \boldsymbol{\mu}, K)\\ &\doteq \frac{n}{2}\ln|K^{-1}| - \frac{1}{2}\sum_{i=1}^n (\boldsymbol{z}_i-\boldsymbol{\mu})^T K^{-1}(\boldsymbol{z}_i-\boldsymbol{\mu}),\end{aligned}$$

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

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

$$\frac{\partial \ell}{\partial \boldsymbol{\mu}} = -\frac{1}{2}\sum_{i=1}^n 2(\boldsymbol{z}_i-\boldsymbol{\mu})^T K^{-1}(-I) = \sum_{i=1}^n (\boldsymbol{z}_i-\boldsymbol{\mu})^T K^{-1}.$$

Setting this equal to zero yields

$$\hat{\boldsymbol{\mu}}_{ML} = \bar{\boldsymbol{z}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i.$$

Exercise 4.4. Using the facts

$$\frac{\partial}{\partial A}\boldsymbol{x}^T A\boldsymbol{x} = \boldsymbol{x}\boldsymbol{x}^T, \quad \frac{\partial}{\partial A}\ln|A| = A^{-T}$$

prove that

$$\hat{K}_{ML} = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{z}_i-\bar{\boldsymbol{z}})(\boldsymbol{z}_i-\bar{\boldsymbol{z}})^T$$

△

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 $\boldsymbol{x}$. For example, we may be interested in predicting blood pressure given age, sex, weight, exercise, and calorie intake. Applications include prediction as well as understanding the relationship between inputs and output, for example, identifying the most important input components.

Linear regression relies on the assumption that $y \simeq \boldsymbol{x}^T\boldsymbol{\theta}$, where $\boldsymbol{x}$ and $\boldsymbol{\theta}$ are elements of $\mathbb{R}^d$. We formulate the problem as follows: Find

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathbb{E}[L(y, \boldsymbol{x}^T\boldsymbol{\theta})],$$

for a given loss function L. We typically do not have the joint distribution for $\boldsymbol{x}, y$. Thus, given a training set $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$, we aim to find

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} L(y_i, \boldsymbol{x}_i^T\boldsymbol{\theta}). \tag{5.1}$$

The linear form, which assumes $y_i \simeq \sum_{j=1}^{d} \theta_j x_{ij}$, may appear restrictive since it apparently excludes dependence on, for example, x_{ij}^2. This, however, 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 $y_i \simeq \sum_{j=1}^{e} \theta_j g_j(\boldsymbol{x}_i)$, where g_j are any function of $\boldsymbol{x}_i$ such as x_{i1}^2 and $x_{i1}x_{i2}x_{i4}$.

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

$$X = \begin{pmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix}, \qquad \boldsymbol{y} = \begin{pmatrix} y_1^T \\ \vdots \\ y_n^T \end{pmatrix}$$

Furthermore, let $\boldsymbol{\epsilon}$ be such that

$$\boldsymbol{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon},$$

and $\hat{\boldsymbol{y}} = X\boldsymbol{\theta}$. With this notation, our goal is to find $\boldsymbol{\theta}$ such that $\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2 = \|\boldsymbol{\epsilon}\|_2$ is minimized, where, for a vector $\boldsymbol{v} \in \mathbb{R}^d$,

$$\|\boldsymbol{v}\|_2^2 = \boldsymbol{v}^T\boldsymbol{v} = \sum_{j=1}^{d} v_j^2.$$

Example 5.1. Suppose

$$\boldsymbol{x}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{x}_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \qquad \boldsymbol{x}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$y_1 = -1, \qquad y_2 = 1, \qquad y_3 = 0.$$

Then

$$X = \begin{pmatrix} 0 & 1 \\ 2 & 0 \\ 1 & 1 \end{pmatrix}, \qquad \hat{\boldsymbol{y}} = X\theta = \theta_1 \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} + \theta_2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \qquad \boldsymbol{y} - \hat{\boldsymbol{y}} = \begin{pmatrix} -1-\theta_2 \\ 1-2\theta_1 \\ -\theta_1-\theta_2 \end{pmatrix}. \tag{5.2}$$

△

5.2 Least-squares

A common choice for the loss function is

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

This choice is relatively easy to deal with from a computational perspective and also has the same solution as the MLE for a common probabilistic model, thus providing an additional rationale for the resulting approach.

This choice leads to mean squared loss minimization:

$$\mathcal{L}(\boldsymbol{\theta}) = \|\boldsymbol{y} - X\boldsymbol{\theta}\|_2^2, \qquad \hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}).$$

We define

$$\hat{\boldsymbol{y}} = X\hat{\boldsymbol{\theta}}$$

as the predicted value or estimate based on the model.

Projection onto the column space of X. Our first observation is that $\hat{\boldsymbol{y}}$ is in the column space of X, i.e., it is a linear combination of the columns of X. We can thus restate our goal as finding $\hat{\boldsymbol{y}}$ in the column space of X such that $\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|$ is minimized. Hence, $\hat{\boldsymbol{y}}$ is the projection of $\boldsymbol{y}$ onto the column space of X as shown in Figure 5.1. Then, from Lemma 4.1, $\boldsymbol{y} - \hat{\boldsymbol{y}}$ is orthogonal to each column of X.

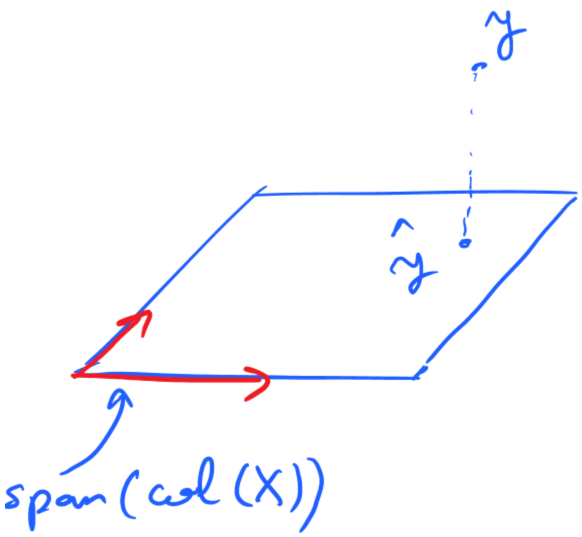


Figure 5.1: Error is minimized by projecting $\boldsymbol{y}$ onto the column space of X, Span(col(X)).

This orthogonality can be written as $X^T(\boldsymbol{y}-\hat{\boldsymbol{y}})=0$. Then

$$\begin{aligned} X^T(\boldsymbol{y}-\hat{\boldsymbol{y}})=0 &\iff X^T(\boldsymbol{y}-X\hat{\boldsymbol{\theta}})=0\\ &\iff X^T\boldsymbol{y}=X^TX\hat{\boldsymbol{\theta}}\\ &\iff \hat{\boldsymbol{\theta}}=(X^TX)^{-1}X^T\boldsymbol{y} \end{aligned}$$

Here we have assumed that X^TX is invertible. This holds if the columns of X are linearly independent. To see this, suppose that X^TX is not invertible. Then there exists a nonzero vector $\boldsymbol{\alpha}$ such that $X^TX\boldsymbol{\alpha}=0$ and hence

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

where the last step follows from the fact that the columns of X are linearly independent. But this contradicts $\boldsymbol{\alpha}\neq\mathbf{0}$. If the columns of X are not linearly independent, then the solution is not unique.

Example 5.2. From Example 5.1, we have

$$X=\begin{pmatrix}0&1\\2&0\\1&1\end{pmatrix},\qquad\qquad \boldsymbol{y}=\begin{pmatrix}-1\\1\\0\end{pmatrix},$$

and so

$$X^TX = \begin{pmatrix} 5 & 1 \\ 1 & 2 \end{pmatrix}, \qquad (X^TX)^{-1} = \frac{1}{9}\begin{pmatrix} 2 & -1 \\ -1 & 5 \end{pmatrix}$$

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

$$\hat{\boldsymbol{y}} = \begin{pmatrix} -7/9 \\ 10/9 \\ -2/9 \end{pmatrix}, \qquad \boldsymbol{y} - \hat{\boldsymbol{y}} = \begin{pmatrix} -2/9 \\ -1/9 \\ 2/9 \end{pmatrix}.$$

△

Gradient descent. Alternatively, we can 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

$$\begin{aligned} \mathcal{L} &= (\boldsymbol{y} - X\boldsymbol{\theta})^T(\boldsymbol{y} - X\boldsymbol{\theta}), \\ \nabla\mathcal{L} &= 2\left[(\boldsymbol{y} - X\boldsymbol{\theta})^T(-X)\right]^T = -2X^T(\boldsymbol{y} - X\boldsymbol{\theta}). \end{aligned}$$

Setting the gradient equal to 0 again gives $\hat{\boldsymbol{\theta}} = (X^TX)^{-1}X^T\boldsymbol{y}$. (Note that the Hessian is X^TX, which is positive-semi-definite.)

Computing $(X^TX)^{-1}$ may be prohibitively expensive computationally. An alternative approach is to start from an arbitrary value $\boldsymbol{\theta}^{(0)}$ and move towards the solution in steps:

$$\begin{aligned} \boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} + \rho X^T(\boldsymbol{y} - X\boldsymbol{\theta}^{(t)}) \\ &= \boldsymbol{\theta}^{(t)} + \rho\sum_{i=1}^{n}\boldsymbol{x}_i(y_i - \boldsymbol{x}_i^T\boldsymbol{\theta}^{(t)}), \end{aligned}$$

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.

5.3 Probabilistic Models for Regression

So far we haven't made any assumptions regarding the statistics of the data. Let us now assume that $\boldsymbol{y} = X\boldsymbol{\theta} + \boldsymbol{\epsilon}$, where $\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}$ and $\mathrm{Cov}(\boldsymbol{\epsilon}) = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \sigma^2 I$. (This is the case if $y_i = \boldsymbol{x}^T\boldsymbol{\theta} + \epsilon_i$, where ϵ_i are iid with mean 0 and variance σ^2.) Then

$$\begin{aligned} \mathbb{E}[\hat{\boldsymbol{\theta}}] &= \mathbb{E}[(X^TX)^{-1}X^T\boldsymbol{y}] \\ &= (X^TX)^{-1}X^T\,\mathbb{E}[\boldsymbol{y}] \\ &= (X^TX)^{-1}X^T\,\mathbb{E}[X\boldsymbol{\theta} + \boldsymbol{\epsilon}] \\ &= \boldsymbol{\theta}, \end{aligned}$$

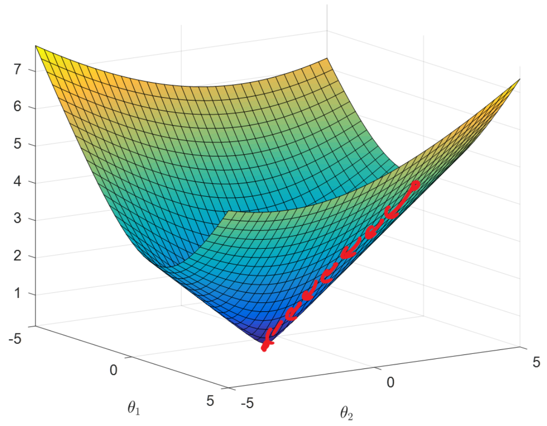


Figure 5.2: Gradient descent for linear regression

using the properties of covariance given in the appendix, so $\hat{\boldsymbol{\theta}}$ is unbiased. The covariance is given by

$$\begin{aligned}\operatorname{Cov}(\hat{\boldsymbol{\theta}}) &= \operatorname{Cov}((X^TX)^{-1}X^T\boldsymbol{y})\\ &= (X^TX)^{-1}X^T\operatorname{Cov}(\boldsymbol{y})X(X^TX)^{-1}\\ &= (X^TX)^{-1}\sigma^2.\end{aligned}$$

The Gauss-Markov theorem. The Gauss-Markov theorem states that under the assumptions that $\mathbb{E}[\boldsymbol{\epsilon}] = 0$ and $\operatorname{Cov}(\boldsymbol{\epsilon}) = \sigma^2 I$, $\hat{\boldsymbol{\theta}}$ is the best *linear* unbiased estimator. More precisely, for any[1] vector $\boldsymbol{u}$, $\boldsymbol{u}^T\hat{\boldsymbol{\theta}}$ is an unbiased estimator of $\boldsymbol{u}^T\boldsymbol{\theta}$ with the smallest possible variance.

Gaussian model

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

$$p(\boldsymbol{y};\boldsymbol{\theta},\sigma^2) \sim \mathcal{N}(X\boldsymbol{\theta},\sigma^2 I).$$

Exercise 5.3. Prove that if $p(\boldsymbol{y};\boldsymbol{\theta},\sigma^2) \sim \mathcal{N}(X\boldsymbol{\theta},\sigma^2 I)$, then for all i, $p(y_i;\boldsymbol{\theta},\sigma^2) \sim \mathcal{N}(\boldsymbol{x}_i^T\boldsymbol{\theta},\sigma^2)$ and the y_i are independent. △

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

[1] This isn't entirely precise!

Maximum Likelihood

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

$$\begin{aligned} p(\boldsymbol{y};\boldsymbol{\theta},\sigma^2) &= \frac{1}{\sqrt{(2\pi\sigma^2)}^n}\exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y}-X\boldsymbol{\theta})^T(\boldsymbol{y}-X\boldsymbol{\theta})\right)\\ &\propto \frac{1}{\sigma^n}\exp\left(-\frac{\|\boldsymbol{y}-X\boldsymbol{\theta}\|_2^2}{2\sigma^2}\right)\\ \ell(\boldsymbol{\theta},\sigma^2) &\doteq -n\ln(\sigma)-\frac{1}{2\sigma^2}\|\boldsymbol{y}-X\boldsymbol{\theta}\|_2^2. \end{aligned}$$

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

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

We can similarly show that

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

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

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

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). To see this, note that, for Fisher information $I(\boldsymbol{\theta})$, we have

$$\begin{aligned} \ell(\boldsymbol{\theta},\sigma^2) &\doteq -n\ln(\sigma)-\frac{1}{2\sigma^2}(\boldsymbol{y}-X\boldsymbol{\theta})^T(\boldsymbol{y}-X\boldsymbol{\theta}).\\ \nabla_{\boldsymbol{\theta}}\ell &= \left(-\frac{1}{\sigma^2}(\boldsymbol{y}-X\boldsymbol{\theta})^T(-X)\right)^T\\ &= \frac{1}{\sigma^2}X^T(\boldsymbol{y}-X\boldsymbol{\theta})\\ \mathsf{H}_{\boldsymbol{\theta}}\ell &= \frac{d\nabla_{\boldsymbol{\theta}}\ell}{d\boldsymbol{\theta}}=-\frac{1}{\sigma^2}X^TX. \end{aligned}$$

and so $I(\boldsymbol{\theta})=\frac{1}{\sigma^2}X^TX$. Hence, $I(\boldsymbol{\theta})^{-1}=\sigma^2(X^TX)^{-1}$, which matches the variance of covariance of $\hat{\boldsymbol{\theta}}$.

Bayesian Linear Regression

In Bayesian linear regression, the Gaussian likelihood

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

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

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

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

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

We start with

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

This 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

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

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

$$\boldsymbol{\theta}^T K^{-1}\boldsymbol{\theta} - 2\boldsymbol{\theta}^T K^{-1}\hat{\boldsymbol{\theta}} \doteq \boldsymbol{\theta}^T X^T X\boldsymbol{\theta} - 2\boldsymbol{\theta}^T X^T\boldsymbol{y},$$

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

$$\begin{aligned}
-2\boldsymbol{\theta}^T K^{-1}\hat{\boldsymbol{\theta}} &= -2\boldsymbol{\theta}^T X^T\boldsymbol{y},\\
-2\boldsymbol{\theta}^T X^T X\hat{\boldsymbol{\theta}} &= -2\boldsymbol{\theta}^T X^T\boldsymbol{y},\\
X^T X\hat{\boldsymbol{\theta}} &= X^T\boldsymbol{y},\\
\hat{\boldsymbol{\theta}} &= (X^T X)^{-1}X^T\boldsymbol{y}.
\end{aligned}$$

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

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

Now we need to find $p(\sigma^2|\boldsymbol{y})$. It turns out this is a distribution called a scaled inverse-χ^2,

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

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

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

While we can continue analytically and find $p(\boldsymbol{\theta}|\boldsymbol{y})$, in practice, we proceed computationally by generating samples from $p(\sigma^2|\boldsymbol{y})$ and then $p(\boldsymbol{\theta}|\boldsymbol{y}, \sigma^2)$. With this sampling approach we can also perform prediction for a given input vector $\boldsymbol{x}_{n+1}$ of by producing samples from $p(y_{n+1}|\boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(\boldsymbol{x}_{n+1}^T\boldsymbol{\theta}, \sigma^2)$.

Standardization

In some texts, an intercept term is also included in the loss function. For ease of notation, we instead assume that X is standardized, meaning that each column $\boldsymbol{v}$ is shifted and scaled such that $\boldsymbol{v}^T\mathbf{1} = 0$ and $\boldsymbol{v}^T\boldsymbol{v} = 1$ and that $\boldsymbol{y}$ is centered so that $\boldsymbol{y}^T\mathbf{1} = 0$. This assumption also holds in the following sections.

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

$$\mathcal{L}(\boldsymbol{\theta}) = \|\boldsymbol{y} - X\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2,$$

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,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - X\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_2^2, \tag{5.3}$$

can be shown to be equivalent to

$$\begin{gathered}\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - X\boldsymbol{\theta}\|_2^2, \\ \text{subject to} : \|\boldsymbol{\theta}\|_2^2 \le t,\end{gathered}$$

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 (5.3),

$$\begin{aligned}\nabla\mathcal{L}(\boldsymbol{\theta}) &= -2X^T(\boldsymbol{y} - X\boldsymbol{\theta}) + 2\lambda\boldsymbol{\theta}, \\ \nabla\mathcal{L}(\hat{\boldsymbol{\theta}}) = 0 &\iff X^T(\boldsymbol{y} - X\hat{\boldsymbol{\theta}}) = \lambda\hat{\boldsymbol{\theta}} \\ &\iff \hat{\boldsymbol{\theta}} = (X^TX + \lambda I)^{-1}X^T\boldsymbol{y}.\end{aligned}$$

Exercise 5.4. Prove that for $\lambda > 0$, $X^TX + \lambda I$ is invertible, even if the columns of X are not linearly independent. △

Bayesian Interpretation

We will now view the regularization penalty from a Bayesian point of view. As before assume the Gaussian likelihood

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

For simplicity, we focus on estimating only $\boldsymbol{\theta}$ and not σ^2. For the prior on $\boldsymbol{\theta}$, let

$$p(\boldsymbol{\theta}|\sigma^2) \sim \mathcal{N}(0, (\sigma^2/\lambda)I) \propto e^{-\frac{\lambda\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\sigma^2}}.$$

Then

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

Based on the previous discussion, it is immediately clear that the mode of the posterior distribution for $\boldsymbol{\theta}$ is $(X^TX + \lambda I)^{-1}X^T\boldsymbol{y}$. 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 $\boldsymbol{\theta}$, which assigns high prior probabilities to smaller length of $\boldsymbol{\theta}$.

Lasso

In lasso, the regularization penalty has the form of the ℓ_1 norm,

$$\|\boldsymbol{\theta}\|_1 = \sum_{i=1}^{m} |\theta_i|,$$

where m is the length of $\boldsymbol{\theta}$. The problem is to find

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - X\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1,$$

or equivalently

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - X\boldsymbol{\theta}\|_2^2,$$
$$\text{subject to} : \|\boldsymbol{\theta}\|_1 \leq t.$$

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 $\boldsymbol{\theta}$ assuming the same model as above but with the double exponential (Laplace) prior

$$p(\boldsymbol{\theta}|\sigma^2) \propto e^{-\frac{\lambda\|\boldsymbol{\theta}\|_1}{2\sigma^2}}.$$

Discussion and generalization

In general we could choose the regularization penalty to be of the form[2]

$$\|\boldsymbol{\theta}\|_q^q = \sum_{i=1}^{m} |\theta_i|^q,$$

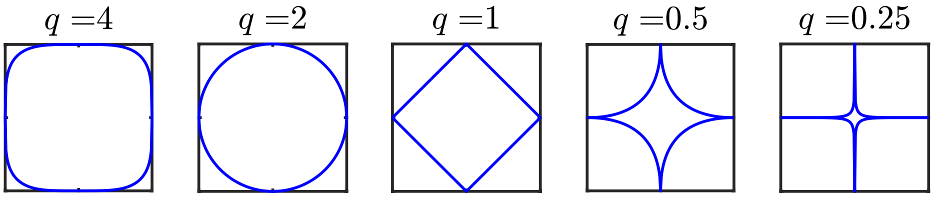
[2] $\|\boldsymbol{\theta}\|_q = \left(\sum_{i=1}^{m} |\theta_i|^q\right)^{1/q}$ is called the ℓ_q-*norm* of $\boldsymbol{\theta}$.

where m is the length of $\boldsymbol{\theta}$. 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

$$\exp\left(-\frac{\lambda}{2\sigma^2}\|\boldsymbol{\theta}\|_q^q\right).$$

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.

5.5 Bias-Variance Trade-off

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?

Our goal is to predict a value y given an input vector $\boldsymbol{x}$. Let the prediction/estimate $\hat{y}$ for y given $\boldsymbol{x}$ be denoted by $\hat{y} = f(\boldsymbol{x})$, where f is the estimator. For linear regression this is of the form $f(\boldsymbol{x}) = \boldsymbol{x}^T\hat{\boldsymbol{\theta}}$, so finding the estimator is the same as finding $\hat{\boldsymbol{\theta}}$. For a *specific* estimator f (e.g., a specific value for $\hat{\boldsymbol{\theta}}$), assuming quadratic loss, we have

$$\mathcal{L}(f) = \mathbb{E}[(y - f(\boldsymbol{x}))^2]. \tag{5.4}$$

Recall from Section 1.4.3 that

$$\mathcal{L}(f) = \mathbb{E}\Big[(y - \bar{y}(\boldsymbol{x}))^2\Big] + \mathbb{E}\Big[(\bar{y}(\boldsymbol{x}) - f(x))^2\Big],$$

where $\bar{y}(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}]$.

An important observation is that the second term in the expected loss for f is a function of f while the first term is not. The first term is an intrinsic noise term which we cannot reduce by choosing a better f or by collecting more data. This term can be viewed as the accumulated effect of all factors that are not included in $\boldsymbol{x}$. Given that this terms is not a function of f, define

$$\bar{\mathcal{L}}(f) = \mathbb{E}\big[(f(\boldsymbol{x}) - \bar{y}(\boldsymbol{x}))^2\big]. \tag{5.5}$$

This compares our estimator to the best possible. So we should choose f to minimize the above quantity.

Let us consider how f is chosen:

1. Determine a set $\mathcal{F}$ from which f can be chosen, e.g., all linear functions.
2. Define an empirical loss function that is related to the expected loss (5.4), but not necessarily identical, e.g., ridge loss.
3. Collect data, $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, 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, by extension, 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). In this context, we write the loss as

$$\bar{\mathcal{L}}(\mathcal{M}) = \mathbb{E}\big[(f_\mathcal{D}(\boldsymbol{x}) - \bar{y}(\boldsymbol{x}))^2\big] = \mathbb{E}\big[(f_\mathcal{D}(\boldsymbol{x}_{n+1}) - \bar{y}(\boldsymbol{x}_{n+1}))^2\big],$$

where we have added the subscript $n+1$ to emphasize that the loss can be viewed as the prediction loss for the next sample. With a similar trick as above, we have

$$\begin{aligned}\bar{\mathcal{L}}(\mathcal{M}) &= \mathbb{E}\big[(f_\mathcal{D}(\boldsymbol{x}) - \bar{y}(\boldsymbol{x}))^2\big] \\ &= \mathbb{E}\big[(f_\mathcal{D}(\boldsymbol{x}) - \mathbb{E}[f_\mathcal{D}(\boldsymbol{x})] + \mathbb{E}[f_\mathcal{D}(\boldsymbol{x})] - \bar{y}(\boldsymbol{x}))^2\big] \\ &= \mathbb{E}\big[(f_\mathcal{D}(\boldsymbol{x}) - \mathbb{E}[f_\mathcal{D}(\boldsymbol{x})])^2\big] + \mathbb{E}\big[(\mathbb{E}[f_\mathcal{D}(\boldsymbol{x})] - \bar{y}(\boldsymbol{x}))^2\big] + \\ &\qquad 2\,\mathbb{E}[(f_\mathcal{D}(\boldsymbol{x}) - \mathbb{E}[f_\mathcal{D}(\boldsymbol{x})])(\mathbb{E}[f_\mathcal{D}(\boldsymbol{x})] - \bar{y}(\boldsymbol{x}))]. \\ &= \mathbb{E}\big[(\mathbb{E}[f_\mathcal{D}(\boldsymbol{x})] - \bar{y}(\boldsymbol{x}))^2\big] + \mathbb{E}\big[(f_\mathcal{D}(\boldsymbol{x}) - \mathbb{E}[f_\mathcal{D}(\boldsymbol{x})])^2\big],\end{aligned}$$

where the last equality follows from conditioning on $\boldsymbol{x}$.

We can understand this loss better by assuming a given value of $\boldsymbol{x}$ (or more precisely, a given value of $\boldsymbol{x}_{n+1}$). The loss then becomes[3]

$$(\mathbb{E}[f_\mathcal{D}(\boldsymbol{x})] - \bar{y}(\boldsymbol{x}))^2 + \mathbb{E}\big[(f_\mathcal{D}(\boldsymbol{x}) - \mathbb{E}[f_\mathcal{D}(\boldsymbol{x})])^2\big] = (\text{bias})^2 + \text{variance}$$

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/flexibility[4] 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.

Example 5.5 (Regularization)**.** 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

[3] Technically the expectation terms need to be conditioned on $\boldsymbol{x}$. But I have dropped those for simplicity.

[4] By flexibility, I mean its responsiveness to changes in the data, i.e., the extent to which the results change when data changes.

be biased. In particular, with $\mathcal{D} = (X, \boldsymbol{y})$, if $X^T X = I$, then

$$\begin{aligned}
\mathbb{E}[\hat{y}_{n+1}] &= \mathbb{E}[\boldsymbol{x}_{n+1}^T \hat{\boldsymbol{\theta}}] \\
&= \boldsymbol{x}_{n+1}^T (X^T X + \lambda I)^{-1} E[X^T \boldsymbol{y}] \\
&= \boldsymbol{x}_{n+1}^T (X^T X + \lambda I)^{-1} E[X^T X \boldsymbol{\theta}] \\
&= \frac{\boldsymbol{x}_{n+1}^T \boldsymbol{\theta}}{1+\lambda} \\
&\neq \boldsymbol{x}_{n+1}^T \boldsymbol{\theta} \\
&= \mathbb{E}[y_{n+1}].
\end{aligned}$$

But it can be shown to have lower variance. If the choice of λ is appropriate, it will have a smaller total loss. △

Example 5.6 (Overfitting and Underfitting)**.** Suppose the true relationship between two scalar variables x and y is

$$y = ax + w, \quad w \sim \mathcal{N}(0, \sigma^2).$$

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

$$\mathcal{D} = \{(x_1 = 1, y_1), (x_2 = 2, y_2)\}.$$

We consider three predictors of the forms

- $\hat{y}(x) = 0$,
- $\hat{y}(x) = \theta x$,
- $\hat{y}(x) = \theta_1 x + \theta_2 x^2$,

and find $\theta, \theta_1, \theta_2$ to minimize the square loss for our data,

$$\frac{1}{2}\left[(y_1 - \hat{y}(x_1))^2 + (y_2 - \hat{y}(x_2))^2\right].$$

We then find the error for 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 $x_3 = 3$			
		Irred.	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	$18\sigma^2$	$19\sigma^2$

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.

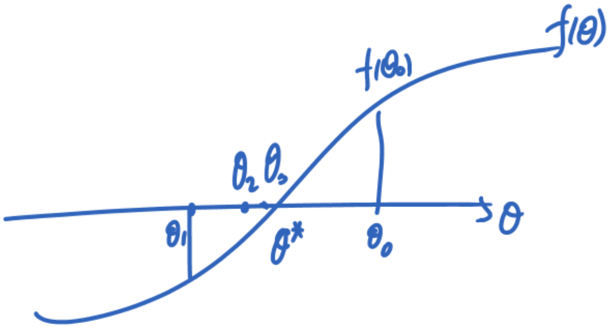
Given the assumption that σ is small relative to a, the smallest total error is obtained by the middle predictor. The zero predictor is not complex enough to be able to fit even the training data well. This situation is referred to as **underfitting**. The quadratic predictor is so complex that it can 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.

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. △

5.6 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

$$\theta^{(t+1)} = \theta^{(t)} - a_t f(\theta^{(t)}),$$

where a_t satisfies

$$\sum_{t=1}^{\infty} a_t = \infty, \qquad \sum_{t=1}^{\infty} a_t^2 < \infty.$$

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

$$\theta^{(t+1)} = \theta^{(t)} - a_t F(\theta^{(t)}),$$

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)

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}[(y - \boldsymbol{x}^T\boldsymbol{\theta})^2],$$

where we are also assuming that $\boldsymbol{x}$ is random with some distribution. To minimize this loss, we compute the gradient:

$$\nabla\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}[-2(y - \boldsymbol{x}^T\boldsymbol{\theta})\boldsymbol{x}]$$

We would like to find $\boldsymbol{\theta}$ such that the gradient above is zero.

Let

$$\begin{aligned} f(\boldsymbol{\theta}) &= \mathbb{E}[-2(y - \boldsymbol{x}^T\boldsymbol{\theta})\boldsymbol{x}] \\ F(\boldsymbol{\theta}) &= -2(y - \boldsymbol{x}^T\boldsymbol{\theta})\boldsymbol{x}, \end{aligned}$$

so that $f(\boldsymbol{\theta}) = \mathbb{E}[F(\boldsymbol{\theta})]$. Now the elements of the data set $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$ can be used to produce samples for $F(\boldsymbol{\theta})$. So we let

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + a_t(y_i - \boldsymbol{x}_i^T\boldsymbol{\theta}^{(t)})\boldsymbol{x}_i,$$

which is the stochastic gradient descent algorithm for linear regression.

Chapter 6

Linear Classification

In a classification problem, we have an input vector $\boldsymbol{x}$ together with a corresponding label y. Based on a data set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, our goal is to predict y given a new value for $\boldsymbol{x}$. 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, $\boldsymbol{x}$ is a vector providing the values of pixels of the image and $y \in \{0, 1, \ldots, 9\}$ 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 $p(y|\boldsymbol{x})$, which for any given $\boldsymbol{x}$ provides the probability of belonging to different classes. We can identify the class for a given $\boldsymbol{x}$ as the class that has the maximum probability,

$$\hat{y}(\boldsymbol{x}) = \arg\max_j p(y = j|\boldsymbol{x}).$$

This choice *minimizes the probability of predicting the wrong class*

$$\mathcal{L} = \Pr(\hat{y}(\boldsymbol{x}) \neq y) = \mathbb{E}[I(y \neq \hat{y}(\boldsymbol{x}))].$$

To find the distribution $p(y|\boldsymbol{x})$, our first step is developing a model that relates $\boldsymbol{x}$ 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 $\boldsymbol{x}$. A generative model has two components, both of which must be learned from data:

- Prior class probabilities: $p(y)$
- Class-conditional probabilities: $p(\boldsymbol{x}|y)$

From these, using Bayes' theorem we can find $p(y|\boldsymbol{x})$ as

$$p(y|\boldsymbol{x}) = \frac{p(y)p(\boldsymbol{x}|y)}{p(\boldsymbol{x})} \propto p(y)p(\boldsymbol{x}|y).$$

We can often estimate $p(y = j)$ simply by computing the fraction of class j in our training data. For $p(\boldsymbol{x}|y)$, 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, $\boldsymbol{x}$ 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 $p(y|\boldsymbol{x})$ since this is the distribution that we need to decide which class $\boldsymbol{x}$ belongs to.

6.2 Generative Probabilistic Models

6.2.1 Gaussian Class-Conditionals

Let us denote

$$p(y = j) = \pi_j.$$

We further assume $p(\boldsymbol{x}|y = j)$ is Gaussian with mean $\boldsymbol{\mu}_j$ and covariance matrix Σ_j. For our purpose, it suffices to consider $\ln p(\boldsymbol{x}|y = j)$,

$$\ln p(\boldsymbol{x}|y = j) \doteq -\frac{1}{2}\ln|\Sigma_j| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_j).$$

From these, we can find $\ln p(\boldsymbol{x}|y = j)$ and then decide $\hat{y}(\boldsymbol{x})$ as

$$\hat{y}(\boldsymbol{x}) = \arg\max_j \ln p(y = j|\boldsymbol{x}).$$

More specifically,

$$\begin{aligned}\ln p(y = j|\boldsymbol{x}) &\doteq \ln\pi_j - \frac{1}{2}\ln|\Sigma_j| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_j)\\ &\doteq \boldsymbol{x}^T\left(-\frac{1}{2}\Sigma_j^{-1}\right)\boldsymbol{x} + (\boldsymbol{\mu}_j^T\Sigma_j^{-1})\boldsymbol{x} + \left(-\frac{1}{2}\boldsymbol{\mu}_j^T\Sigma_j^{-1}\boldsymbol{\mu}_j - \frac{1}{2}\ln|\Sigma_j| + \ln\pi_j\right) \qquad (6.1)\\ &\doteq \boldsymbol{x}^T A_j\boldsymbol{x} + \boldsymbol{\beta}_j^T\boldsymbol{x} + \gamma_j,\end{aligned}$$

For an appropriately defined symmetric matrix A_j, vector $\boldsymbol{\beta}_j$, and scalar γ_j.

6.2.2 Linear Discriminant Analysis

First, let us suppose all classes have the same covariance matrix $\Sigma_j = \Sigma$. Then, the terms $\boldsymbol{x}^T\left(-\frac{1}{2}\Sigma^{-1}\right)\boldsymbol{x}$ and $-\frac{1}{2}\ln|\Sigma_j|$ in (6.1) become independent of the class and we thus have

$$\ln p(y = j|\boldsymbol{x}) \doteq \boldsymbol{\beta}_j^T\boldsymbol{x} + \gamma_j, \qquad (6.2)$$

where

$$\boldsymbol{\beta}_j^T = \boldsymbol{\mu}_j^T\Sigma^{-1}, \qquad \gamma_j = -\frac{1}{2}\boldsymbol{\mu}_j^T\Sigma^{-1}\boldsymbol{\mu}_j + \ln\pi_j,$$

Suppose we have only two classes, $y = 0$ and $y = 1$, with $p(y = 1) = \pi = 1 - p(y = 0)$. We can then divide the space into two regions,

$$\ln p(y = 1|\boldsymbol{x}) \underset{\hat{y}=0}{\overset{\hat{y}=1}{\gtrless}} \ln p(y = 0|\boldsymbol{x}).$$

What is the decision boundary between them? We can find it by solving $\ln p(y = 1|\boldsymbol{x}) = \ln p(y = 0|\boldsymbol{x})$,

$$\boldsymbol{\beta}_1^T\boldsymbol{x} + \gamma_1 = \boldsymbol{\beta}_0^T\boldsymbol{x} + \gamma_0 \iff (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0)^T\boldsymbol{x} + \gamma_1 - \gamma_0 = 0 \iff \boldsymbol{\beta}^T\boldsymbol{x} + \gamma = 0,$$

where

$$\boldsymbol{\beta}^T = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\Sigma^{-1}, \qquad \gamma = \ln\frac{\pi}{1-\pi} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\Sigma^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0). \tag{6.3}$$

Hence, the decision boundary is the hyperplane $\boldsymbol{\beta}^T\boldsymbol{x} + \gamma = 0$. On one side of this plane, we predict class 1 (we let $\hat{y}(\boldsymbol{x}) = 1$) and on the other side, we declare class 0:

$$\hat{y}(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{\beta}^T\boldsymbol{x} + \gamma > 0 \\ 0, & \boldsymbol{\beta}^T\boldsymbol{x} + \gamma < 0 \end{cases}$$

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

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\left(\boldsymbol{x} - \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0}{2}\right) = 0,$$

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 $p(y|\boldsymbol{x})$ of each class for a given $\boldsymbol{x}$, which can tell us about the certainty of belonging to each class? From (6.2), we have $p(y = j|\boldsymbol{x}) \propto e^{\boldsymbol{\beta}_j^T\boldsymbol{x}+\gamma_j}$ and so for two classes

$$p(y = 1|\boldsymbol{x}) = \frac{e^{\boldsymbol{\beta}_1^T\boldsymbol{x}+\gamma_1}}{e^{\boldsymbol{\beta}_1^T\boldsymbol{x}+\gamma_1} + e^{\boldsymbol{\beta}_0^T\boldsymbol{x}+\gamma_0}} = \frac{1}{1 + e^{-(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma)}} = \sigma(\boldsymbol{\beta}^T\boldsymbol{x} + \gamma),$$

where $\boldsymbol{\beta}$ 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

$$p(y = j|\boldsymbol{x}) = \frac{e^{\boldsymbol{\beta}_j^T\boldsymbol{x}+\gamma_j}}{\sum_{k=1}^c e^{\boldsymbol{\beta}_k^T\boldsymbol{x}+\gamma_k}} = \sigma_j(\boldsymbol{\beta}_1^T\boldsymbol{x} + \gamma_1, \ldots, \boldsymbol{\beta}_c^T\boldsymbol{x} + \gamma_c),$$

where $\sigma_j(\boldsymbol{v}) = \frac{e^{v_j}}{\sum_k e^{v_k}}$ is the softmax function.

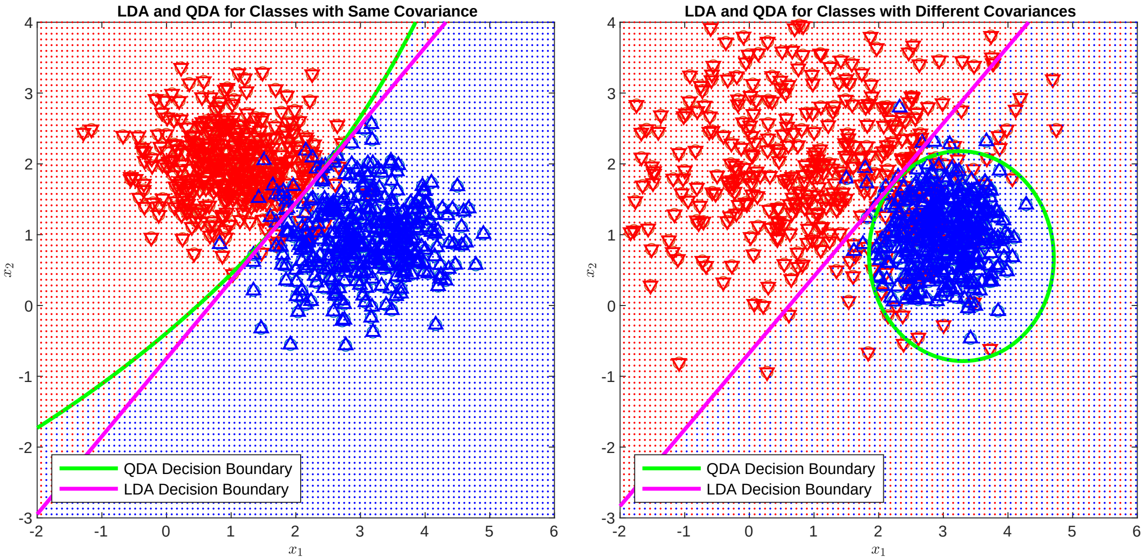


Figure 6.1: LDA vs QDA for when $\Sigma_1 = \Sigma_2$ (left) and when $\Sigma_1 \neq \Sigma_2$ (right).

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 $\ln p(y = 1|\boldsymbol{x}) = \ln p(y = 0|\boldsymbol{x})$. This will lead to a quadratic equation of the form $\boldsymbol{x}^T A \boldsymbol{x} + \boldsymbol{\beta}^T \boldsymbol{x} + \gamma = 0$, 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 $p(\boldsymbol{x}|y = j)$, 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 $\boldsymbol{x}$ along with their corresponding class labels. Let the n independent samples be denoted by $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^m$ and $y_i \in \{0, 1\}$ for all i.

Model:

$$p(y=j)=\begin{cases}\pi, & j=1\\ 1-\pi, & j=0\end{cases}$$

$$\begin{aligned}p(\boldsymbol{x}|y=0)&\sim\mathcal{N}(\boldsymbol{\mu}_0,\Sigma),\\ p(\boldsymbol{x}|y=1)&\sim\mathcal{N}(\boldsymbol{\mu}_1,\Sigma),\end{aligned}$$

for some $\pi,\boldsymbol{\mu}_0,\boldsymbol{\mu}_1$ and diagonal matrix $\Sigma=\operatorname{diag}(\sigma_1^2,\sigma_2^2,\dots,\sigma_m^2)$. 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:

$$\begin{aligned}p(\mathcal{D}|\pi,\boldsymbol{\mu}_0,\boldsymbol{\mu}_1,\Sigma)&=\prod_{i=1}^{n}p(y_i)p(\boldsymbol{x}_i|y_i)=\left(\prod_{i:y_i=0}p(y_i)p(\boldsymbol{x}_i|y_i)\right)\left(\prod_{i:y_i=1}p(y_i)p(\boldsymbol{x}_i|y_i)\right)\\ &=\prod_{i:y_i=0}\left((1-\pi)\prod_{j=1}^{m}\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp(-(x_{i,j}-\mu_{0,j})^2/2\sigma_j^2)\right)\times\\ &\quad\prod_{i:y_i=1}\left(\pi\prod_{j=1}^{m}\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp(-(x_{i,j}-\mu_{1,j})^2/2\sigma_j^2)\right),\end{aligned}$$

where $x_{i,j}$ is the j^{th} component of $\boldsymbol{x}_i$ 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 $(\boldsymbol{x}_1,y_1),\dots,(\boldsymbol{x}_n,y_n)$, where each $\boldsymbol{x}_i$ is composed of vectors, that is[1]

$$\boldsymbol{x}_i=(\boldsymbol{x}_{i1},\dots,\boldsymbol{x}_{im}),$$

[1] All vectors in this section are column vectors and all concatenations are also along the vertical dimension. However, for simplicity of notation, we write $\boldsymbol{x}_i=(\boldsymbol{x}_{i1},\dots,\boldsymbol{x}_{im})$ instead of $\boldsymbol{x}_i^T=(\boldsymbol{x}_{i1}^T,\dots,\boldsymbol{x}_{im}^T)^T$

and each $\boldsymbol{x}_{ij} = (x_{ij1}, \ldots, x_{ijl}, \ldots)$ is a binary vector of finite length which represents a one-hot encoding of a feature.

Example 6.1 (One-hot encoding)**.** Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ provide information about a set of movies, where $\boldsymbol{x}_1 = (\boldsymbol{x}_{11}, \ldots, \boldsymbol{x}_{1m}), \boldsymbol{x}_2 = (\boldsymbol{x}_{21}, \ldots, \boldsymbol{x}_{2m}), \ldots$, with $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ 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 $\boldsymbol{x}_{11} = (0,0,1,0,0), \boldsymbol{x}_{12} = (0,0,0,1,0)$ means movie 1 is a drama directed by director D, and $\boldsymbol{x}_{21} = (1,0,0,0,0), \boldsymbol{x}_{22} = (1,0,0,0,0)$ means that movie 2 is a comedy directed by director A. △

Model: We model this classification problem in the following way:

$$p(x_{ijl} = 1 | y_i = k) = \eta_{kjl}, \qquad \sum_l \eta_{kjl} = 1,$$

and all x_{ijl} are independent from one another. For two vectors $\boldsymbol{a}, \boldsymbol{b}$ with the same length, we define $\boldsymbol{a}^{\boldsymbol{b}} = \prod_{i=1}^{|\boldsymbol{a}|} a_i^{b_i}$. Let $\boldsymbol{\eta}_{kj} = (\eta_{kj1}, \ldots)$. We have

$$p(\boldsymbol{x}_{ij} | y_i = k) = \boldsymbol{\eta}_{kj}^{\boldsymbol{x}_{ij}},$$

and then

$$p(\boldsymbol{x}_i | y_i = k) = \prod_{j=1}^m p(\boldsymbol{x}_{ij} | y_i = k) = \prod_{j=1}^m \boldsymbol{\eta}_{kj}^{\boldsymbol{x}_{ij}} = \boldsymbol{\eta}_k^{\boldsymbol{x}_i},$$

where $\boldsymbol{\eta}_k = (\boldsymbol{\eta}_{k1}, \ldots, \boldsymbol{\eta}_{km})$. It follows that

$$p(y_i = k | \boldsymbol{x}_i) \propto p(\boldsymbol{x}_i | y_i = k) p(y_i = k) = \pi_k \boldsymbol{\eta}_k^{\boldsymbol{x}_i} \propto \exp(\ln \pi_k + \boldsymbol{x}_i^T \ln \boldsymbol{\eta}_k).$$

For a new data point $(\boldsymbol{x}, y)$, we similarly have

$$\ln p(y = k | \boldsymbol{x}) \doteq \boldsymbol{\beta}_k^T \boldsymbol{x} + \gamma_k, \tag{6.4}$$

where $\boldsymbol{\beta}_k = \ln \boldsymbol{\eta}_k$ and $\gamma_k = \ln \pi_k$. The log-probabilities are again linear in $\boldsymbol{x}$, 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

$$p(\boldsymbol{x} | \boldsymbol{\theta}) = \exp[\boldsymbol{b}(\boldsymbol{\theta})^T \boldsymbol{a}(\boldsymbol{x}) + f(\boldsymbol{x}) + g(\boldsymbol{\theta})].$$

Let us consider the case in which $\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{x}$, and parameters are functions of class y. So instead of $\boldsymbol{\theta}$ we write $\boldsymbol{\theta}_j$, when considering the jth class. Then the class-conditional distribution will become

$$p(\boldsymbol{x} | y = j) = \exp[\boldsymbol{b}(\boldsymbol{\theta}_j)^T \boldsymbol{x} + f(\boldsymbol{x}) + g(\boldsymbol{\theta}_j)].$$

Furthermore, let $p(y = j) = \pi_j$. Given $\boldsymbol{x}$, the log-probability of each class is given as

$$\ln p(y = j | \boldsymbol{x}) \doteq \ln \pi_j + \ln p(\boldsymbol{x} | y = j) \doteq \ln \pi_j + \boldsymbol{b}(\boldsymbol{\theta}_j)^T \boldsymbol{x} + g(\boldsymbol{\theta}_j) \doteq \boldsymbol{\beta}_j^T \boldsymbol{x} + \gamma_j, \tag{6.5}$$

where $\boldsymbol{\beta}_j = \boldsymbol{b}(\boldsymbol{\theta}_j)$ 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 $\boldsymbol{x}$ is linear in $\boldsymbol{x}$.

6.3 Discriminative Models and Logistic Regression

In the discriminative approach, we model $p(y = j|\boldsymbol{x})$ directly. But what is a good model? As we have seen in (6.2), (6.4) and (6.5), in many generative cases, the log-probabilities of classes given data is linear in $\boldsymbol{x}$,

$$\ln p(y = j|\boldsymbol{x}) \doteq \boldsymbol{\beta}_j^T \boldsymbol{x} + \gamma_j.$$

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,

$$p(y = 1|\boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{\beta}^T \boldsymbol{x} + \gamma)}}, \qquad p(y = 0|\boldsymbol{x}) = \frac{e^{-(\boldsymbol{\beta}^T \boldsymbol{x} + \gamma)}}{1 + e^{-(\boldsymbol{\beta}^T \boldsymbol{x} + \gamma)}},$$

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 $p(y|\boldsymbol{x})$ is of the logistic form and learn this distribution instead of learning first $p(\boldsymbol{x}|y)$ and $p(y)$?" Doing so leads to a *discriminative model* resulting in *logistic regression*.

Let $h(\boldsymbol{x}) = p(y = 1|\boldsymbol{x})$ and assume that the data consists of n iid samples, $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$. We have

$$p(\mathcal{D}; \boldsymbol{\beta}, \gamma) = \prod_{i=1}^n h(\boldsymbol{x}_i)^{y_i} (1 - h(\boldsymbol{x}_i))^{1-y_i}$$

and the negative log-likelihood loss is given by

$$\mathcal{L}(\boldsymbol{\beta}, \gamma) = \sum_{i=1}^n \left(y_i \ln \frac{1}{h(\boldsymbol{x}_i)} + (1 - y_i) \ln \frac{1}{1 - h(\boldsymbol{x}_i)} \right). \tag{6.6}$$

We can use gradient descent to minimize this loss (maximize the likelihood). For simplicity, let $\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\beta} \\ \gamma \end{pmatrix}$, $\tilde{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix}$, and $h_{\boldsymbol{\theta}} = p(y = 1|\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T \tilde{\boldsymbol{x}}}}$. Then

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \rho_t \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}),$$

where

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^n (y_i - h_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i)) \tilde{\boldsymbol{x}}_i.$$

When we find $\boldsymbol{\theta}$ and thus $\boldsymbol{\beta}, \gamma$ we have the decision boundary as $\boldsymbol{\beta}^T \boldsymbol{x} + \gamma = 0$. Points $\boldsymbol{x}$ for which $\boldsymbol{\beta}^T \boldsymbol{x} + \gamma > 0$ 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,

$$\frac{1}{n} \sum_{i=1}^n L(y_i, \boldsymbol{x}_i, \hat{y}(\boldsymbol{x}_i)),$$

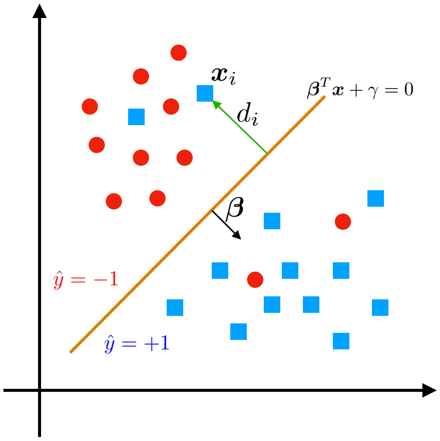


Figure 6.2: A linear classifier defined by the vector $\boldsymbol{\beta}$ and scalar γ. Squares represents points with $y=+1$ and circles $y=-1$. For a point $\boldsymbol{x}_i$, many loss functions can be viewed as a function of the signed distance d_i of $\boldsymbol{x}_i$ to the decision hyperplane.

where $\hat{y}(\boldsymbol{x})$ is the predictor of the class for input vector $\boldsymbol{x}$. 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 $\boldsymbol{\beta}$ and a constant γ, which define the hyperplane $\boldsymbol{\beta}^T\boldsymbol{x}+\gamma=0$. On one side of the hyperplane, we decide class 1 and the other side class -1,

$$\hat{y}(\boldsymbol{x})=\operatorname{sign}(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma)=\begin{cases}1, & \text{if } \boldsymbol{\beta}^T\boldsymbol{x}+\gamma>0,\\ -1, & \text{if } \boldsymbol{\beta}^T\boldsymbol{x}+\gamma<0,\end{cases}$$

where the dependence of $\hat{y}$ on $\boldsymbol{\beta},\gamma$ is implicit.

One such linear classifier is shown in Figure 6.2. Below, we will use the fact that for any point $\boldsymbol{x}_i$ with label y_i and prediction $\hat{y}(\boldsymbol{x}_i)$, 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 $\boldsymbol{x}_i=\boldsymbol{x}_0+d_i\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ for some $\boldsymbol{x}_0$ on the decision boundary. If d_i is positive, then this point is classified correctly, since $\boldsymbol{\beta}^T\boldsymbol{x}_i+\gamma>0$. The distance between $\boldsymbol{x}_i$ and the decision boundary equals $|d_i|$.

6.4.0.1 Zero-one loss

The most natural loss function for classification is the **0-1 loss**,

$$L_{01}(y,\hat{y}(\boldsymbol{x}))=\left\{\begin{matrix}1, & \text{if } y\neq\hat{y}(\boldsymbol{x})\\ 0, & \text{if } y=\hat{y}(\boldsymbol{x})\end{matrix}\right\}=\begin{cases}1, & \text{if } y(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma)<0,\\ 0, & \text{if } y(\boldsymbol{\beta}^T\boldsymbol{x}+\gamma)>0.\end{cases}$$

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.

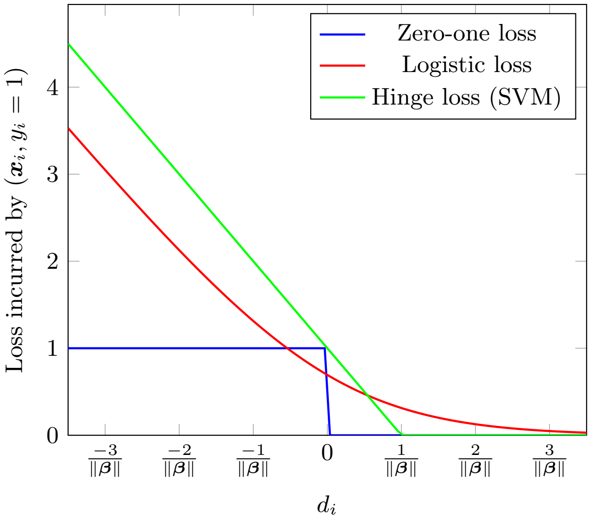


Figure 6.3: Loss functions for a data point $(\boldsymbol{x}_i, y_i = 1)$ as a function of the distance of $\boldsymbol{x}_i$ from the boundary (in terms of the length of $\boldsymbol{\beta}$).

Unfortunately, minimizing this loss function is computationally difficult (NP-hard) [2]. So in practice, we use differentiable loss-functions for which efficient algorithms exist. Here we will consider two such loss functions. 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.

6.4.0.2 Logistic regression

Let us re-examine the logistic regression loss function (6.6). The loss incurred by a data point $\boldsymbol{x}_i$ at signed distance d_i from the decision hyperplane (i.e., $\boldsymbol{x}_i = \boldsymbol{x}_0 + d_i\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$) is

$$\ln(1 + e^{-(\boldsymbol{\beta}^T\boldsymbol{x}_i+\gamma)}) = \ln(1 + e^{-d_i\|\boldsymbol{\beta}\|}).$$

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.

6.4.0.3 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

$$\max(0, 1 - y_i(\boldsymbol{\beta}^T\boldsymbol{x}_i + \gamma)).$$

Letting $y_i = 1$ and $\boldsymbol{x}_i = \boldsymbol{x}_0 + d_i\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ as before, results in

$$\max(0, 1 - d_i\|\boldsymbol{\beta}\|).$$

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 $1/\|\boldsymbol{\beta}\|$ of the decision boundary.

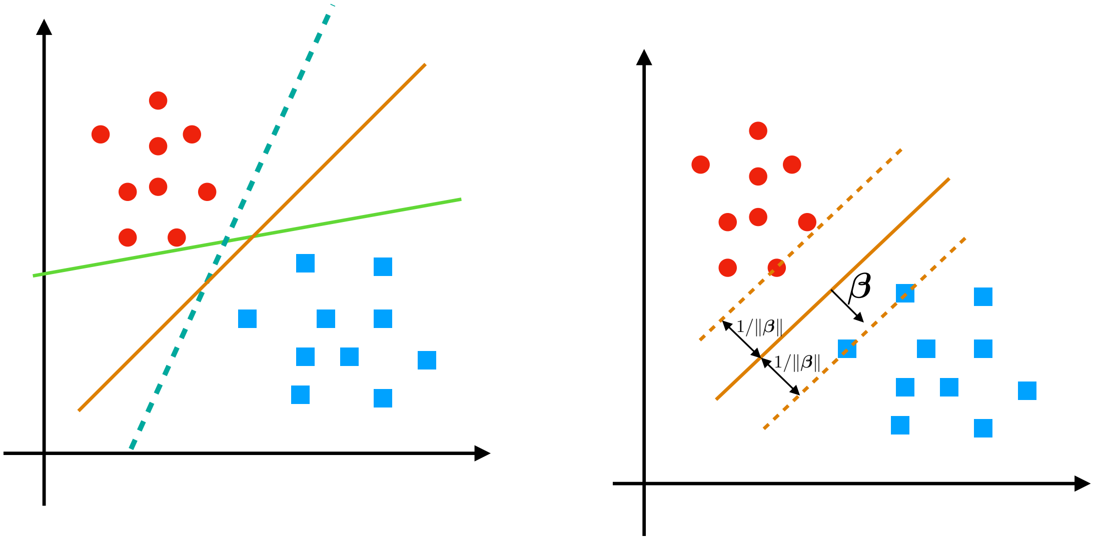
In addition to penalizing points within the margin, we would like to ensure that the margin is not very small. This can be done by ensuring $1/\|\boldsymbol{\beta}\|$ is large or equivalently $\|\boldsymbol{\beta}\|^2$ is small. Both of these goals can be achieved with the loss

$$\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\boldsymbol{\beta}^T\boldsymbol{x}_i + \gamma)) + \lambda\|\boldsymbol{\beta}\|^2, \tag{6.7}$$

where λ is a constant that balances the two components of the loss. This results in the so called **support 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 (6.7) 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.



(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. For example, for classification, the complete data consists of the features $\boldsymbol{x}$ 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

$$\log p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}),$$

where $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ and $\boldsymbol{y} = (y_1, \dots, y_n)$ 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.

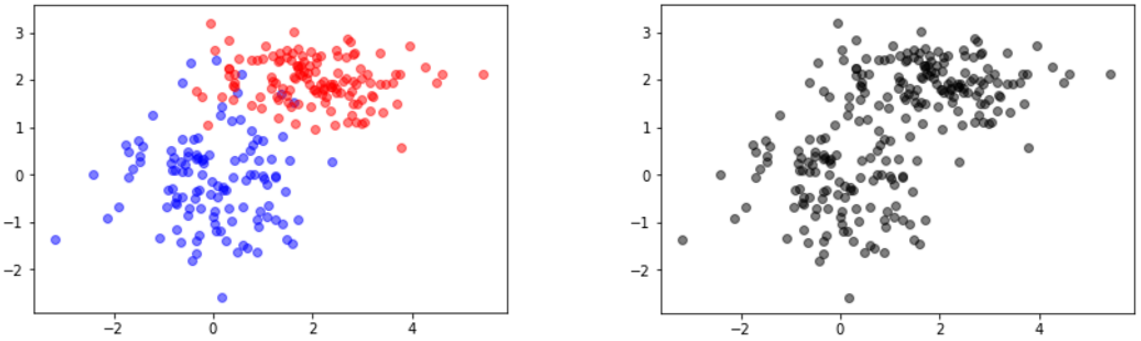


Figure 7.1: Data from two classes, with labels given as colors (left) and not given (right).

We thus formulate this problem as finding $\boldsymbol{\theta}$ that maximizes

$$\log p(\boldsymbol{x};\boldsymbol{\theta}) = \log \sum_{\boldsymbol{y}} p(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta})$$

In this case, $(\boldsymbol{x},\boldsymbol{y})$ is the complete data, for which computing the likelihood is easy, but a component of this data, namely $\boldsymbol{y}$, 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 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 $\log p(x;\theta)$ that coincides with it at $\theta=\theta'$, i.e.,

$$\begin{aligned}\log p(x;\theta) &\ge B(\theta,\theta'), &&\text{for all } \theta,\\ \log p(x;\theta) &= B(\theta,\theta'), &&\text{for } \theta=\theta'.\end{aligned} \tag{7.1}$$

Now let our new estimate be

$$\theta'' = \arg\max_{\theta} B(\theta,\theta').$$

Note that we have not used $\log p(x;\theta)$ to find θ''. It follows that

$$\log p(x;\theta'') \ge \log p(x;\theta').$$

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 $\log p(x;\theta)$ is bounded, since the sequence $\log p(x;\theta')$ is non-decreasing, it will converge. Under appropriate conditions, this means that θ' also converges to a stationary point of $p(x;\theta)$. See [1] for details.

It remains to find a lower bound that satisfies (7.1). For the likelihood of the observation and for any y such that $p(y|x;\theta)>0$,

$$\ell(\theta) = \ln p(x;\theta) = \ln \frac{p(x,y;\theta)}{p(y|x;\theta)}.$$

Then, for any distribution q for the missing data y,

$$\begin{aligned}\ell(\theta) &= \sum_y q(y) \ln \frac{p(x,y;\theta)}{p(y|x;\theta)}\\ &\ge \sum_y q(y) \ln \frac{p(x,y;\theta)}{p(y|x;\theta)} - D(q(y)||p(y|x;\theta))\\ &= \sum_y q(y) \ln \frac{p(x,y;\theta)}{p(y|x;\theta)} - \sum_y q(y) \ln \frac{q(y)}{p(y|x;\theta)}\\ &= \sum_y q(y) \ln p(x,y;\theta) - \sum_y q(y) \ln q(y),\end{aligned}$$

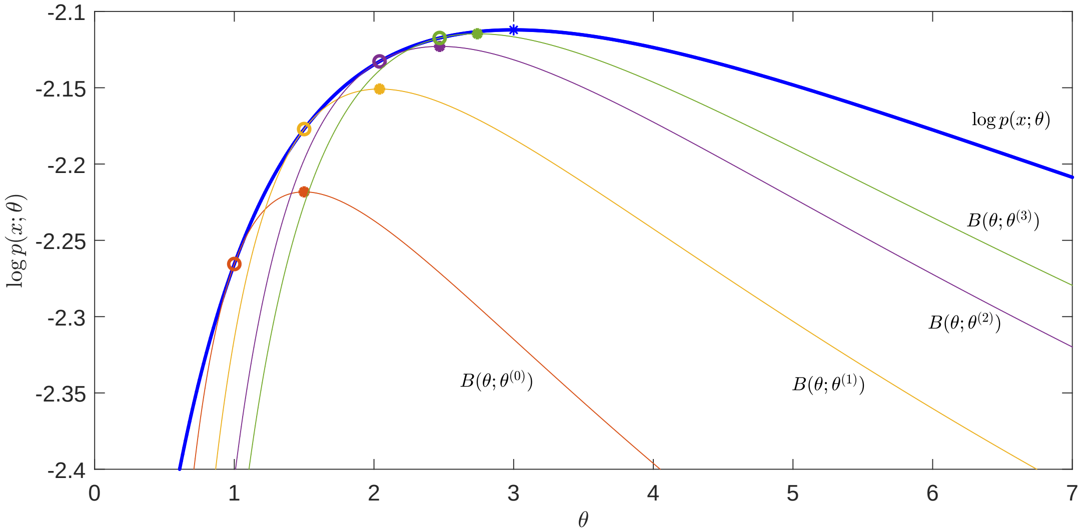


Figure 7.2: The log-likelihood of the observation and consecutive EM lower bounds and estimates. In each iteration, the current value of θ is denoted by $\circ$ and the new value by $*$. Here, $\theta^{(0)} = 1, \theta^{(1)} = 1.5, \theta^{(2)} = 2.04, \theta^{(3)} = 2.472$. Continuing in the same manner, we would obtain estimates $2.740, 2.880, 2.946, 2.976, \ldots$, where 3 is the true maximum.

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 $\ell(\theta)$. Suppose our current guess for θ is $\theta^{(t)}$. We would like this lower bound to be equal to $\ell(\theta)$ at $\theta = \theta^{(t)}$. For this to occur, we need

$$D(q(y)||p(y|x;\theta^{(t)})) = 0 \iff q(y) = p(y|x;\theta^{(t)}),$$

resulting in

$$\ell(\theta) \geq \sum_y p(y|x;\theta^{(t)}) \ln p(x,y;\theta) - \sum_y p(y|x;\theta^{(t)}) \ln p(y|x;\theta^{(t)}) = B(\theta, \theta^{(t)}).$$

Now instead of maximizing ℓ, we can maximize B. We note however that the second term in B is not a function of θ. So we instead define the following expectation

$$Q(\theta, \theta^{(t)}) = \sum_y p(y|x;\theta^{(t)}) \ln p(x,y;\theta),$$

and find

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta, \theta^{(t)}).$$

For simplicity of notation, I often use θ' to denote $\theta^{(t)}$ and θ'' to denote $\theta^{(t+1)}$. Also, let $\mathbb{E}'$ be expected value assuming the value of θ'. We can then describe the EM algorithm as

- The E-step:

$$Q(\theta;\theta') = \sum_{y} \ln p(x,y;\theta)p(y|x;\theta') = \mathbb{E}[\ln p(x,y;\theta)|x;\theta'] = \mathbb{E}'[\ln p(x,y;\theta)|x]$$

- The M-step:

$$\theta'' = \arg\max_{\theta} Q(\theta;\theta').$$

Update $\theta' \leftarrow \theta''$ 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 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. When the labels y_i are missing, the problem becomes *clustering.*

We assume Gaussian class-conditionals:

$$\begin{aligned} p(y_i = 1) &= \pi, & \boldsymbol{x}_i|y_i = 1 &\sim \mathcal{N}(\mu_1, K_1) \\ p(y_i = 0) &= 1 - \pi, & \boldsymbol{x}_i|y_i = 0 &\sim \mathcal{N}(\mu_0, K_0) \end{aligned}$$

Let $\boldsymbol{\theta} = (\pi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, K_0, K_1)$. Ideally, we would want to maximize the likelihood for the observed data $\{(\boldsymbol{x}_i)\}_{i=1}^n$,

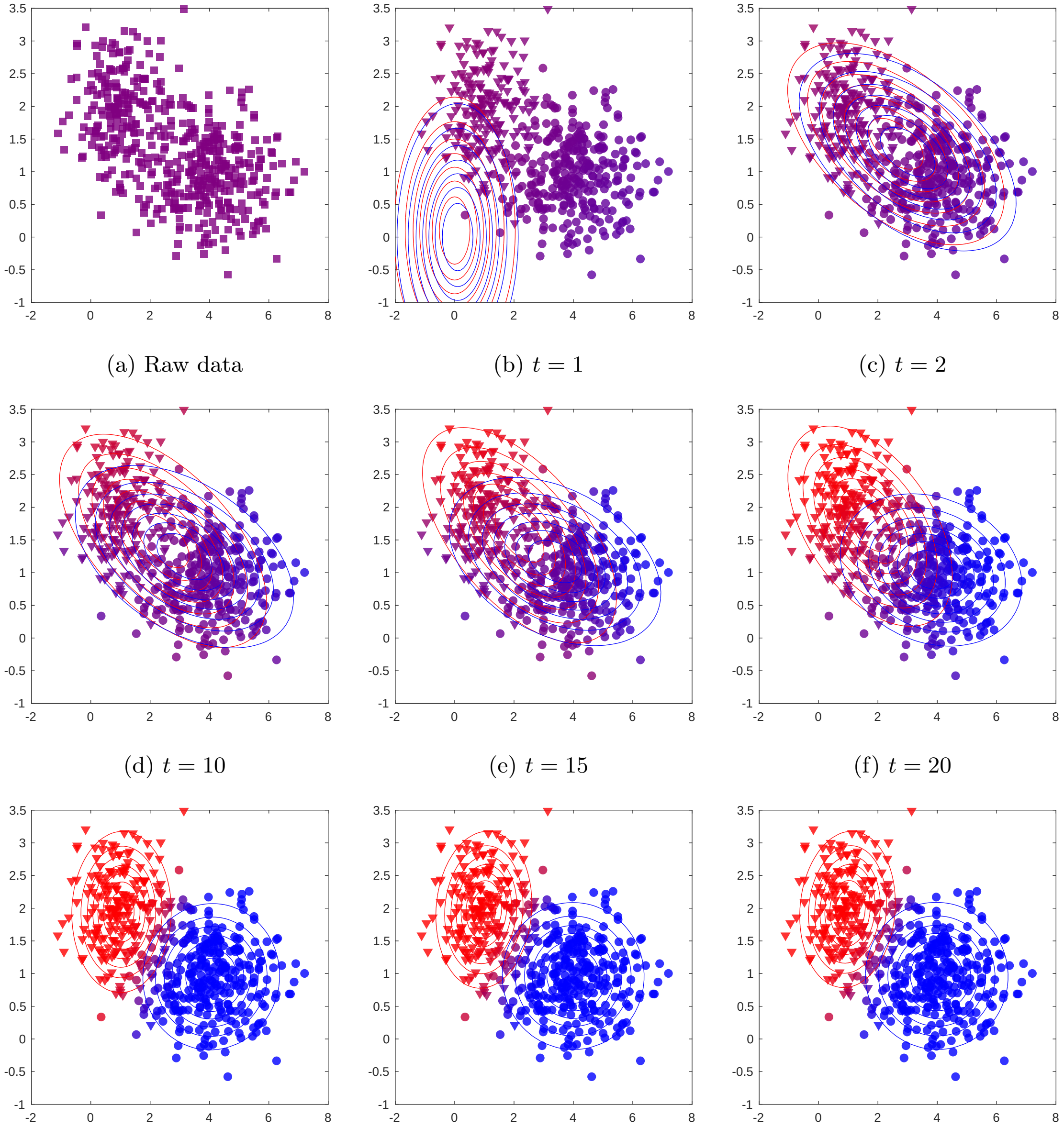
$$\ell(\boldsymbol{\theta}) = \ln p(\boldsymbol{x}_1^n|\boldsymbol{\theta}) = \ln \sum_{y_1^n} p(\boldsymbol{x}_1^n, y_1^n|\boldsymbol{\theta}).$$

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

$$\ln p(\boldsymbol{x}_1^n, y_1^n; \theta) = \sum_{i=1}^{n} \ln p(\boldsymbol{x}_i, y_i; \theta),$$



(a) Raw data

(b) $t = 1$

(c) $t = 2$

(d) $t = 10$

(e) $t = 15$

(f) $t = 20$

(g) $t = 30$

(h) $t = 40$

(i) $t = 50$

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 ($\gamma_i' = p(y_i = 1|\boldsymbol{x}_i; \theta')$). A video of the clustering can be found here.

and for each term in this sum,

$$\begin{aligned}\ln p(\boldsymbol{x}_i, y_i; \theta) &= \ln\Big((\pi p(\boldsymbol{x}_i|y_i = 1; \theta))^{y_i}((1-\pi)p(\boldsymbol{x}_i|y_i = 0; \theta))^{1-y_i}\Big)\\ &= y_i \ln(\pi p(\boldsymbol{x}_i|y_i = 1; \theta)) + (1-y_i)\ln((1-\pi)p(\boldsymbol{x}_i|y_i = 0; \theta)).\end{aligned}$$

- **The E-step:** Let θ' be the current estimate for θ and let $\mathbb{E}'$ denote expected value operator with respect to the distribution $p(y|x;\theta')$. We have

$$\begin{aligned}Q(\theta;\theta') &= \mathbb{E}'[\ln p(\boldsymbol{x}_1^n, y_1^n; \theta)|\boldsymbol{x}_1^n]\\ &= \mathbb{E}'\left[\sum_{i=1}^n \ln p(\boldsymbol{x}_i, y_i; \theta)|\boldsymbol{x}_1^n\right]\\ &= \sum_{i=1}^n \mathbb{E}'[\ln p(\boldsymbol{x}_i, y_i; \theta)|\boldsymbol{x}_i]\end{aligned}$$

And for each term in the sum,

$$\begin{aligned}\mathbb{E}'[\ln p(\boldsymbol{x}_i, y_i; \theta)|\boldsymbol{x}_i] &= \mathbb{E}'[y_i \ln(\pi p(\boldsymbol{x}_i|y_i = 1; \theta)) + (1-y_i)\ln((1-\pi)p(\boldsymbol{x}_i|y_i = 0; \theta))|\boldsymbol{x}_i]\\ &= \mathbb{E}'[y_i|\boldsymbol{x}_i]\ln(\pi p(\boldsymbol{x}_i|y_i = 1; \theta)) + \mathbb{E}'[1-y_i|\boldsymbol{x}_i]\ln((1-\pi)p(\boldsymbol{x}_i|y_i = 0; \theta))\\ &= \gamma_i' \ln\pi + (1-\gamma_i')\ln(1-\pi) + \gamma_i' \ln p(\boldsymbol{x}_i|y_i = 1; \theta)\\ &\qquad + (1-\gamma_i')\ln p(\boldsymbol{x}_i|y_i = 0; \theta),\end{aligned}$$

where

$$\begin{aligned}\gamma_i' &= \mathbb{E}'[y_i|\boldsymbol{x}_i]\\ &= p(y_i = 1|\boldsymbol{x}_i; \theta')\\ &= \frac{p(x_i, y_i = 1; \theta')}{p(x_i, y_i = 1; \theta') + p(x_i, y_i = 0; \theta')}\\ &= \frac{\pi'\mathcal{N}(x_i; \mu_1', K_1')}{\pi'\mathcal{N}(x_i; \mu_1', K_1') + (1-\pi')\mathcal{N}(x_i; \mu_0', K_0')}.\end{aligned}$$

Here, γ_i' has a significant meaning. It represents the probability that a given point $\boldsymbol{x}_i$ 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 $\boldsymbol{x}_i$ 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'' :

$$\begin{aligned}\frac{\partial Q}{\partial \mu_1} &= \frac{\partial}{\partial \mu_1}\sum_{i=1}^n \gamma_i' \ln p(\boldsymbol{x}_i|y_i=1;\theta)\\ &= \frac{\partial}{\partial \mu_1}\sum_{i=1}^n \gamma_i'\Big(-\frac{1}{2}(\boldsymbol{x}_i-\mu_1)^T K_1^{-1}(\boldsymbol{x}_i-\mu_1)\Big)\\ &= \sum_{i=1}^n \gamma_i' K_1^{-1}(\boldsymbol{x}_i-\mu_1) = 0 \Rightarrow \mu_1'' = \frac{\sum_{i=1}^n \gamma_i'\boldsymbol{x}_i}{\sum_{i=1}^n \gamma_i'}.\end{aligned}$$

To find K_1'':

$$\begin{aligned}\frac{\partial Q}{\partial K_1^{-1}} &= \frac{\partial}{\partial K_1^{-1}}\sum_{i=1}^n \gamma_i'\Big(\frac{1}{2}\ln\big|K_1^{-1}\big| - \frac{1}{2}(\boldsymbol{x}_i-\mu_1)^T K_1^{-1}(\boldsymbol{x}_i-\mu_1)\Big)\\ &= \frac{1}{2}K_1\sum_{i=1}^n \gamma_i' - \frac{1}{2}\sum_{i=1}^n \gamma_i'(\boldsymbol{x}_i-\mu_1)(\boldsymbol{x}_i-\mu_1)^T = 0\\ &\Rightarrow K_1'' = \frac{\sum_{i=1}^n \gamma_i'(\boldsymbol{x}_i-\mu_1)(\boldsymbol{x}_i-\mu_1)^T}{\sum_{i=1}^n \gamma_i'}.\end{aligned}$$

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

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}') = \sum_y p(y|x;\boldsymbol{\theta}')\ln p(x,y;\boldsymbol{\theta})$$

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

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}') = \sum_z p(z|x;\boldsymbol{\theta}')\ln p(z;\boldsymbol{\theta})$$

Example 7.1 ([1]). Let

$$\begin{aligned}&x = s+\epsilon,\\ &s \sim \mathcal{N}(0,\theta), \qquad \theta \ge 0\\ &\epsilon \sim \mathcal{N}(0,\sigma^2) \qquad \sigma^2 > 0,\end{aligned}$$

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 $\boldsymbol{z} = (s, \epsilon)$ 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

$$\mathrm{Var}(x) = \mathrm{Cov}(s + \epsilon, s + \epsilon) = \sigma^2 + \theta.$$

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:

$$\begin{aligned} Q(\theta; \theta') &= \mathbb{E}'[\ln p(\boldsymbol{z}; \theta)|x] \\ &= \mathbb{E}'[\ln p(s; \theta) + \ln p(\epsilon; \theta)|x] \\ &\doteq \mathbb{E}'[\ln p(s; \theta)|x] \\ &\doteq \mathbb{E}'\left[-\frac{\ln \theta}{2} - \frac{s^2}{2\theta}|x\right] \\ &= -\frac{\ln \theta}{2} - \frac{\mathbb{E}'\left[s^2|x\right]}{2\theta} \end{aligned}$$

- The M-step:

$$\frac{\partial Q}{\partial \theta} = -\frac{1}{2\theta} + \frac{\mathbb{E}'\left[s^2|x\right]}{2\theta^2} = 0 \Rightarrow \theta'' = \mathbb{E}'[s^2|x].$$

  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 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 $\mathcal{D} = \{w_{12}, w_{13}, \ldots, w_{n-1,n}\}$, 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 $\boldsymbol{s} = (s_1, \ldots, s_n)$.

This leads to the log-likelihood

$$\mathcal{L}(\boldsymbol{s}) = \sum_{i,j} w_{ij}(s_i - \ln(e^{s_i} + e^{s_j}))$$

As an aside, note that for a differentiable convex function $f(x)$, we have

$$\begin{aligned} f(x) &\geq f(x') + f'(x')(x - x'), \qquad && \text{for all } x', \\ f(x) &= f(x') + f'(x')(x - x'), \qquad && \text{for } x' = x. \end{aligned}$$

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

$$Q(\boldsymbol{s}, \boldsymbol{s}') = \sum_{i,j} w_{ij} \left(s_i - \ln\left(e^{s'_i} + e^{s'_j}\right) - \frac{e^{s_i} + e^{s_j}}{e^{s'_i} + e^{s'_j}} + 1 \right),$$

then,

$$\begin{aligned} \mathcal{L}(\boldsymbol{s}) &\geq Q(\boldsymbol{s}, \boldsymbol{s}'), \qquad && \text{for all } \boldsymbol{s}', \\ \mathcal{L}(\boldsymbol{s}) &= Q(\boldsymbol{s}, \boldsymbol{s}'), \qquad && \text{for } \boldsymbol{s} = \boldsymbol{s}'. \end{aligned}$$

We can simplify Q by ignoring terms that do not involve $\boldsymbol{s}$, and then separating the parameters (the latter was not possible for $\mathcal{L}$)

$$\begin{aligned}
Q(\boldsymbol{s},\boldsymbol{s}') &= \sum_{i,j} w_{ij}\left(s_i - \frac{e^{s_i}+e^{s_j}}{e^{s'_i}+e^{s'_j}}\right)\\
&= \sum_i s_i \sum_j w_{ij} - \sum_i e^{s_i} \sum_j \frac{w_{ij}}{e^{s'_i}+e^{s'_j}} - \sum_i\sum_j w_{ij}\frac{e^{s_j}}{e^{s'_i}+e^{s'_j}}\\
&= \sum_i s_i \sum_j w_{ij} - \sum_i e^{s_i} \sum_j \frac{w_{ij}}{e^{s'_i}+e^{s'_j}} - \sum_i\sum_j w_{ji}\frac{e^{s_i}}{e^{s'_i}+e^{s'_j}}\\
&= \sum_i s_i \sum_j w_{ij} - \sum_i e^{s_i} \sum_j \frac{w_{ij}+w_{ji}}{e^{s'_i}+e^{s'_j}}.
\end{aligned}$$

Given the current estimate $\boldsymbol{s}'$, we can now find the next estimate $\boldsymbol{s}''$ 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.

Bibliography

[1] B. Hajek, *Random Processes for Engineers.* 2014.

[2] T. T. Nguyen and S. 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), pp. III–1085–III–1093, JMLR.org, June 2013.

[3] M. I. Jordan, *An Introduction to Probabilistic Graphical Models (Preprints and Course Notes).* 2003.

[4] D. J. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge university press, 2003.

[5] C. M. Bishop, *Pattern Recognition And Machine Learning.* New York: Springer, 2006.

[6] C. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[7] A. Furman, “WHAT IS . . . a Stationary Measure?,” *Notices of the AMS*, vol. 58, no. 9.

Chapter 8

Basics of Graphical Models

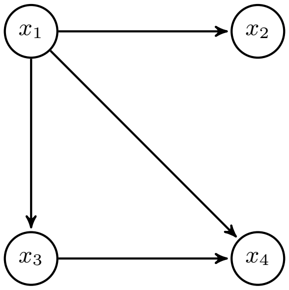
8.1 Introduction

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 reperesented 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 $x_1, \ldots, x_m$ represent variables or quantities (can be scalar or vector)
- Edges represent causal relationships
- The probability distribution over $x_1^m = x_1, \ldots, x_m$ can be expressed as:

$$p(x_1^m) = \prod_{i=1}^{m} p(x_i | \operatorname{pa}(x_i))$$

  where $\operatorname{pa}(x_i)$ 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

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_1, x_3). \tag{8.1}$$

What does (8.1) tell us about the distribution? Recall that based on the chain rule of probability, we always have

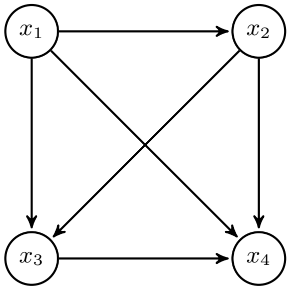
$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3).$$

It is straightforward to show that (8.1) is equivalent to

$$\begin{aligned} p(x_3|x_1, x_2) &= p(x_3|x_1), \\ p(x_4|x_1, x_2, x_3) &= p(x_4|x_1, x_3). \end{aligned} \tag{8.2}$$

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 (8.2). This is in general true. The set of missing incoming edges for each node in the graph represents a conditional independence assumption.

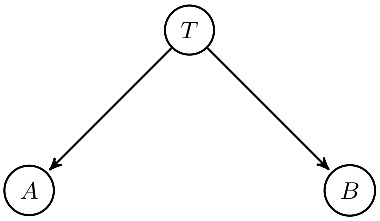
The complete graph, shown for four nodes below, represents the factorization given (8.2), which holds for any distribution and thus the graph can represent any distribution. But such a graph is not particularly useful since the power of graphical models results from the independence assumptions that they encode.



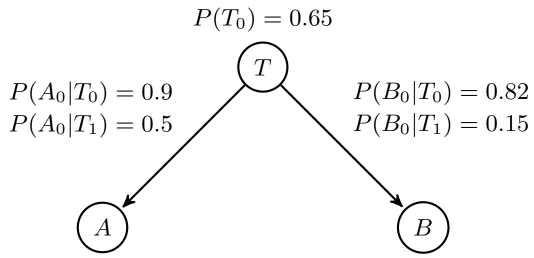
Note that the complete graph is acyclic as it imposes an ordering over the nodes (in this case, x_1, x_2, x_3, x_4). We can view any Bayesian network as being obtained from such a complete graph by removing edges. So every Bayesian network is also acyclic.

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.

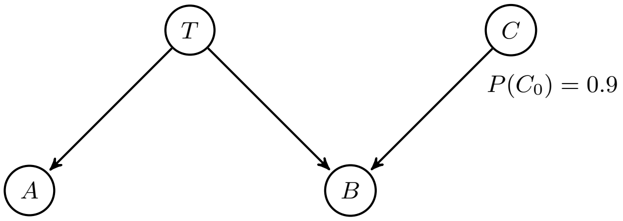
The Bayesian Network that models the probability distribution is shown below.



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,

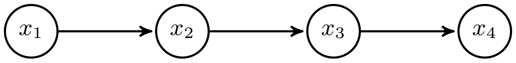
$$p(T_1|B_1) = \frac{p(B_1|T_1)p(T_1)}{p(B_1)} = 0.7177 > p(T_1) = 0.35,$$

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**. △

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 the forecast: given that we know today's weather, days before that are irrelevant for tomorrow's forecast. As an example, consider

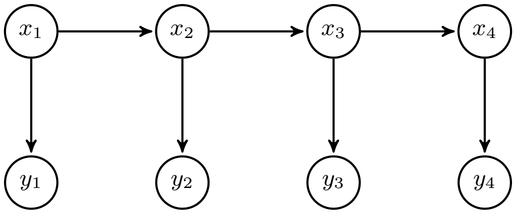


which corresponds to

$$p(x_1^4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3).$$

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?

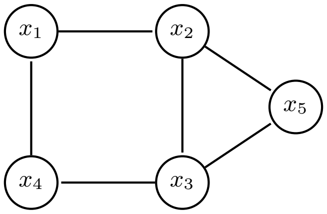
Graphical models, such as Bayesian networks are useful for several reasons.

- 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 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 friends that 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

$$\psi_{ij}(x_i, x_j) = \begin{cases} 1, & x_i = x_j \\ 1/2, & x_i \neq x_j \end{cases} \tag{8.3}$$

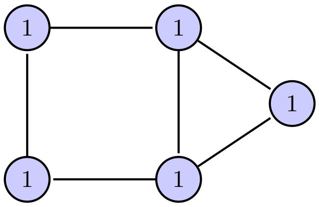
and

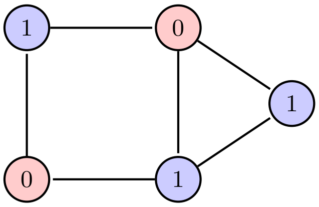
$$\psi_{ijk}(x_i, x_j, x_k) = \begin{cases} 1, & x_i = x_j = x_k \\ 1/2, & \text{if two of the three are equal} \end{cases} \tag{8.4}$$

So agreements are assigned a higher value. Now we can define a probability distribution as

$$p(x_1, \ldots, x_5) \propto \psi_{12}(x_1, x_2)\psi_{14}(x_1, x_4)\psi_{34}(x_3, x_4)\psi_{235}(x_2, x_3, x_5), \tag{8.5}$$

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 (8.5) 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

$$p(x_1, \ldots, x_5) = \frac{1}{Z}\psi_{12}(x_1, x_2)\psi_{14}(x_1, x_4)\psi_{34}(x_3, x_4)\psi_{235}(x_2, x_3, x_5).$$

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 chose different different functions for ψ_{12} and $\psi_{3,4}$.

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.,

$$p(x_1^m) = \frac{1}{Z} \prod_{C \text{ is a clique in } G} \psi_C(x_C), \tag{8.6}$$

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 $\boldsymbol{x} = x_1^m$, the probability $p(\boldsymbol{x})$ is positive, it is helpful to represent the distribution as

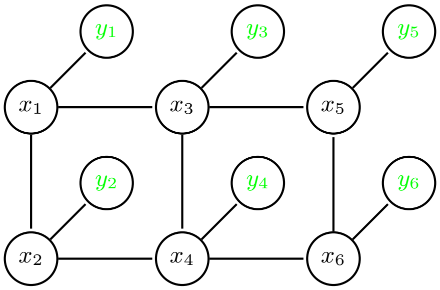
$$p(\boldsymbol{x}) \propto e^{-E(\boldsymbol{x})},$$

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,

$$E(\boldsymbol{x}) = \sum_{C \text{ is a clique in } G} -\phi_C(\boldsymbol{x}_C) \quad \Rightarrow \quad p(\boldsymbol{x}) \propto \prod e^{\phi_C(\boldsymbol{x}_C)}$$

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 $x_1, x_2, \cdots, x_6$ represent the true B/W status of the pixels and $y_1, y_2, \cdots, y_6$ 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

$$E(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{i}^{m} \alpha_i x_i - \sum_{(i,j)\in\mathcal{E}(G)} \beta_{i,j} x_i x_j - \sum_{i}^{m} \zeta_i x_i y_i,$$

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 β_{ij}; 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 $\boldsymbol{y}$ and our goal is to recover $\boldsymbol{x}$. A reasonable solution is

$$\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}).$$

If we can output fractional values (if the denoised image can be grayscale), another possible solution is

$$\mathbb{E}[\boldsymbol{x}|\boldsymbol{y}].$$

△

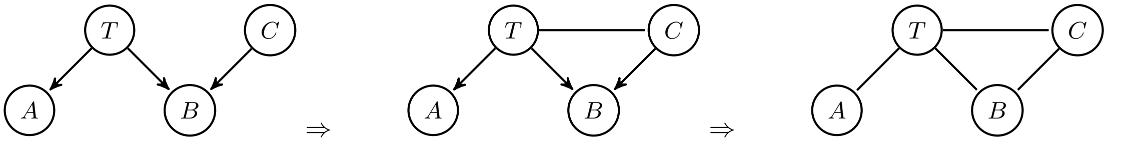
8.4 Moralization: Converting BNs to MRFs

In a BN, there is a term for each node x_i of the form

$$p(x_i|\,\mathrm{pa}(x_i)).$$

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

$$p(A,B,T,C) = p(T)p(C)p(A|T)p(B|T,C) \qquad \Rightarrow \qquad p(A,B,T,C) = \psi(T)\psi(C)\psi(A,T)\psi(B,T,C),$$

where, for example, $\psi(B,T,C) = p(B|T,C)$.

8.5 Latent Dirichlet Allocation**

(**)

Latent Dirichlet Allocation is commonly used for topic modeling - e.g. classifying documents based on their content or topic.

Suppose there are two topics, cats and dogs. The words that appear under any document are represented as probabilities in a matrix β:

Cats: "cat": 50%, "kitten": 20%, "litter": 20%, "paw": 10%
Dogs: "dog": 40%, "puppy": 20%, "bark": 20%, "chew": 10%, "paw": 10%

Each document is a mixture of topics:

document1: 80% cats, 20% dogs
document2: 100% cats
document3: 50% cats, 50% dogs

θ represents a topic mixture for a document and is generated from some distribution with parameter α.

Each word in the document has a topic, and the probability of that topic is given by θ.

Let Z be the topics for each word in document1: cats, cats, dogs, cats, dogs, cats.

We can choose each word in the document based on the word distribution for its topic:

document1: "cat litter dog kitten bark cat"

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

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

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 $x \sim \mathrm{Ber}(1/2), y \sim \mathrm{Ber}(1/2)$ 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,

$$p(A, B \mid C) \propto f(A, C)g(B, C),$$

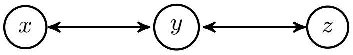
where the constant of proportionality may depend on C, then $A \perp\!\!\!\perp B \mid C$. △

9.2 Independence in Bayesian Networks

In the last chapter that 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.

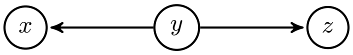
9.2.1 Simple Bayesian networks

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 $x \perp\!\!\!\perp z$?

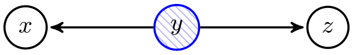


The answer: not in general. The only thing we know from the GM is $p(x, y, z) = p(y)p(x|y)p(z|y)$. We thus have

$$p(x, z) = \sum_y p(x, y, z) = \sum_y p(y)p(x|y)p(z|y)$$

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 $x \not\perp\!\!\!\perp z$. △

Example 9.3. Is $x \perp\!\!\!\perp z \mid y$ in the graph below?

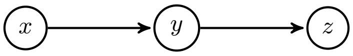


The answer: yes. We need to show $p(x,z \mid y) = p(x|y)p(z|y)$,

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

△

Example 9.4. Is $x \perp\!\!\!\perp z$ in the graph below?

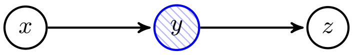


The answer: not in general since

$$p(x,z) = \sum_y p(x)p(y|x)p(z|y) = p(x)\sum_y p(y|x)p(z|y)$$

is not necessarily equal to $p(x)p(z)$. *Exercise:* Provide a counter example for $x \perp\!\!\!\perp z$. △

Example 9.5. Is $x \perp\!\!\!\perp z \mid y$ in the graph below?

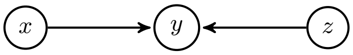


The answer: yes. We have

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

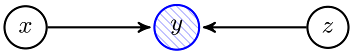
△

Example 9.6. Is $x \perp\!\!\!\perp z$ in the graph below?



Yes: $p(x,z) = \sum_y p(x,y,z) = \sum_y p(x)p(z)p(y \mid x,z) = p(x)p(z)\sum_y p(y \mid x,z) = p(x)p(z)$. △

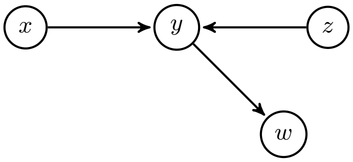
Example 9.7. Is $x \perp\!\!\!\perp z \mid y$ in the graph below?



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 $x \not\perp\!\!\!\perp z \mid y$. △

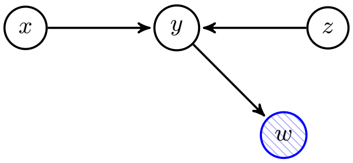
In graphs of Examples 9.8 and 9.9, if y has a descendant, that will also affect the independence relationship between x and z. These cases are considered next.

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 $x \not\perp\!\!\!\perp z \mid w$. △

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.

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 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. A path is blocked if it has a node v such that the nodes incident to v are:

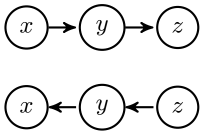
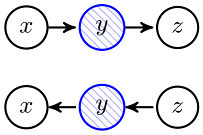
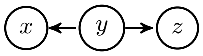
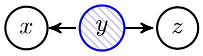
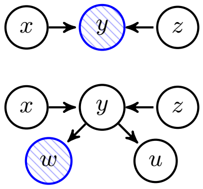
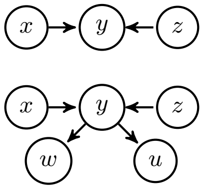
- HT,TH, or TT and $v \in C$;
- HH, and neither v nor its descendants are in C.

Theorem 9.11. *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

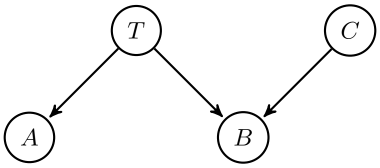
Table 9.1: Flow of information between x and z. Nodes with style y are knwon.

	Passing through	Blocked
HT/TH		
TT		
HH		

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

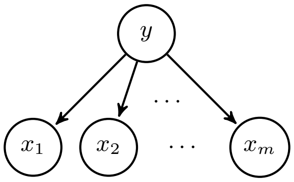
Example 9.12. In the traffic graphic from last chapter, shown below, we want to find all independences of the form $x \perp\!\!\!\perp y$ and $x \perp\!\!\!\perp y \mid z$ for vertices x, y, z. For those that do not follow form d-separation, we write $x \not\perp\!\!\!\perp y$ and $x \not\perp\!\!\!\perp y \mid z$. 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: $T \not\perp\!\!\!\perp A \mid B$, $T \not\perp\!\!\!\perp C \mid B$, $A \not\perp\!\!\!\perp C \mid B$.



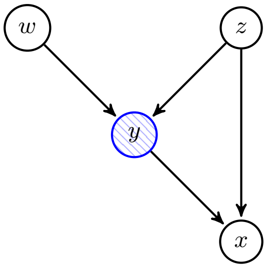
In addition, we have $A \perp\!\!\!\perp \{B, C\} \mid T$ but $\{T, A\} \not\perp\!\!\!\perp C \mid B$. △

Example 9.13 (The Naive Bayes model). The graph for the naive Bayes classification model is



where y denotes the class and $x_1, \ldots, x_m$ 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, $x_i \not\perp\!\!\!\perp x_j$. △

Example 9.14. 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.



△

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.

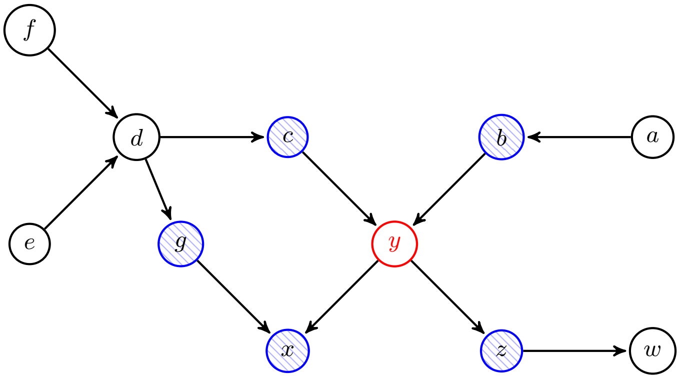
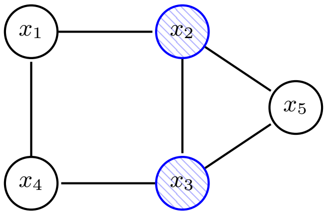


Figure 9.1: The Markov blanket of node y are the set of nodes colored red.



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

$$\begin{aligned} p(x_A, x_B \mid x_C) &= P(x_A \mid x_C)P(x_B \mid x_C) \\ &= \frac{p(x_A, x_B, x_C)}{p(x_C)} \propto p(x_A, x_B, x_C) \\ &\propto \prod_{Q \text{ is a clique in } G} \psi_Q(x_Q) = \prod_{Q \in x_A \cup x_C} \psi(x_Q) \prod_{Q \in x_B \cup x_C} \psi(x_Q). \end{aligned}$$

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.

9.4 In-class activity

Given the graph in Figure 9.5, find the largest set A such that $x_1 \perp\!\!\!\perp x_A \mid x_2, x_3$.

Solution: $A = \{x_4, x_5, x_{10}\}$.

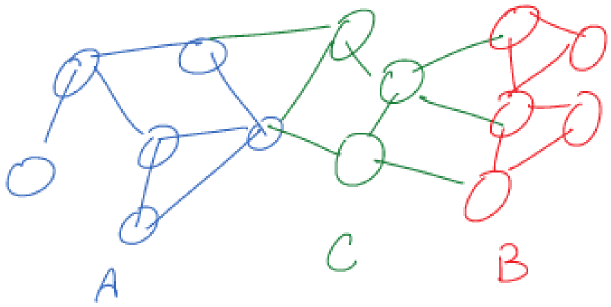


Figure 9.2: MRF Theorem

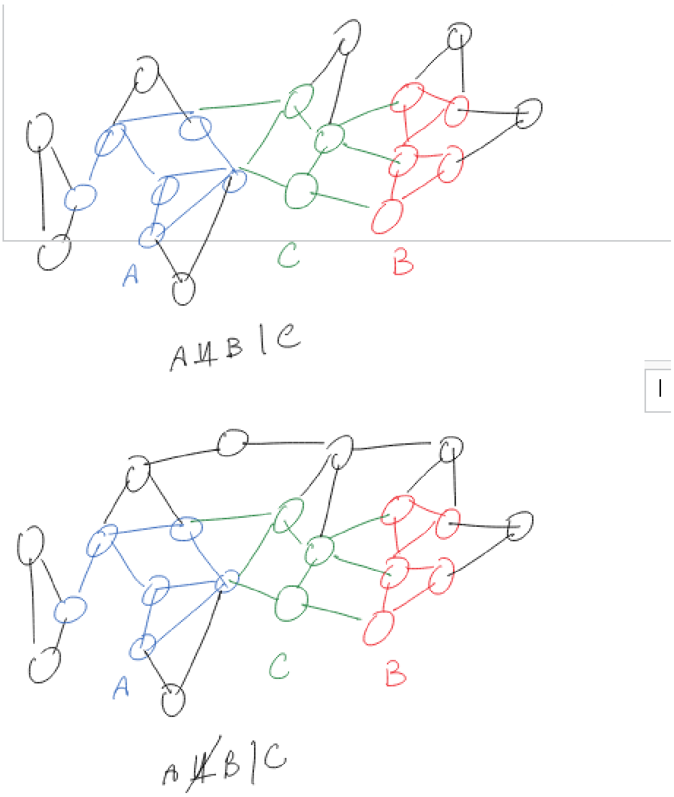


Figure 9.3: Two examples of MRFs

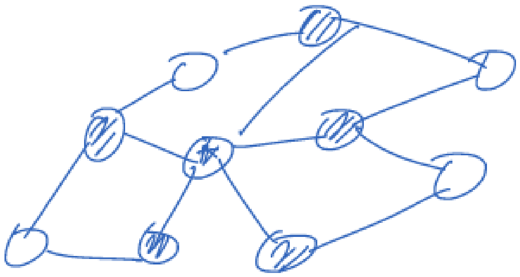


Figure 9.4: As example of a Markov Blanket

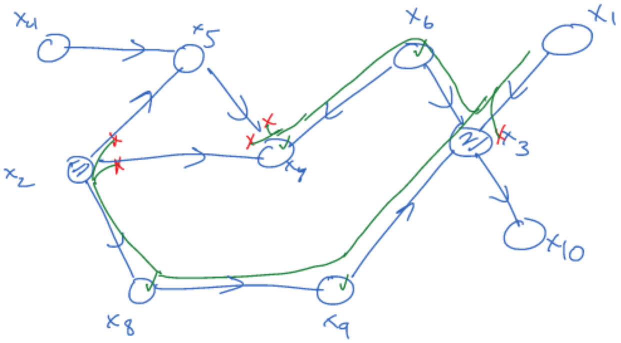


Figure 9.5: In-class activity

Chapter 10

Parameter Estimation in Graphical Models

10.1 Introduction

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:

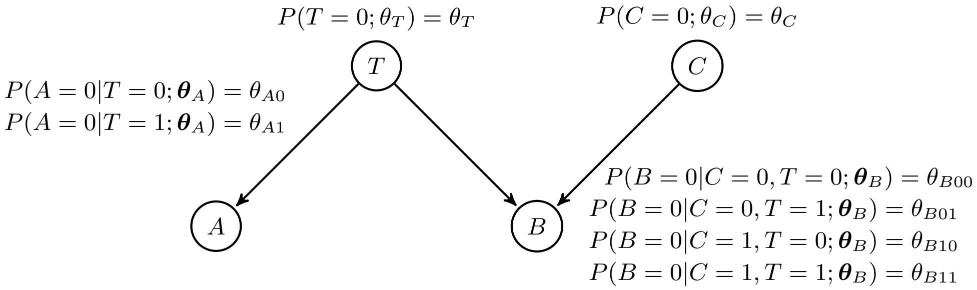
- We will consider the problem of estimating the parameters in graphical models. The problem is simpler in the case of Bayesian networks and for simplicity, that is were our attention will be focused.
- However, we will not consider the more challenging problem of learning the structure of a network. The best case scenario is that you have good reason to design a graph in a certain way, e.g., based on causality.

10.2 Maximum Likelihood Estimation in Bayesian Networks

Consider a BN with known graph with m nodes $x_1, \ldots, x_m$ in which the parameters of the conditional distribution are unknown. There are m conditional probability distributions (**CPD**s)[1], one for each node, and each of these has an unknown parameter vector. We denote the concatenated vector of all parameters as $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$. To determine the parameters, we collect a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ of n iid samples, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})$.

Example 10.1. As an example, we may consider the network from previous chapters with the vector of parameters $\boldsymbol{\theta} = (\theta_T, \theta_C, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B)$.

[1] Some 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.


$P(T = 0; \theta_T) = \theta_T$
$P(C = 0; \theta_C) = \theta_C$
T
C
$P(A = 0|T = 0; \boldsymbol{\theta}_A) = \theta_{A0}$
$P(A = 0|T = 1; \boldsymbol{\theta}_A) = \theta_{A1}$
A
B
$P(B = 0|C = 0, T = 0; \boldsymbol{\theta}_B) = \theta_{B00}$
$P(B = 0|C = 0, T = 1; \boldsymbol{\theta}_B) = \theta_{B01}$
$P(B = 0|C = 1, T = 0; \boldsymbol{\theta}_B) = \theta_{B10}$
$P(B = 0|C = 1, T = 1; \boldsymbol{\theta}_B) = \theta_{B11}$

Our goal is to determine $\boldsymbol{\theta}$ by collecting data and determine the conditional probability distributions, thereby determining the network. To collect data, on n days, we record whether there is heavy traffic and whether Alice, Bob, and/or Charlie are late. △ △

We can find the parameters through maximum likelihood. Given that our network can have many nodes, the size of the parameter vector may be very large. 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 show. To see why this is helpful, suppose that we optimize by grid search, i.e., trying a set of values at regular intervals. If we try K points for one dimension, for m dimensions we need to try K^m points to get the same precision. However, if we optimize m parameters separately, then we only need to try mK points, typically a significantly smaller number.

Decomposability of likelihood. For the ith data sample, the likelihood function is

$$p(\boldsymbol{x}_i; \boldsymbol{\theta}) = \prod_{j=1}^{m} p(x_{ij} | \operatorname{pa}(x_{ij}); \boldsymbol{\theta}_j)$$

and thus the log-likelihood of the whole dataset is

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p(\boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \ln p(x_{ij} | \operatorname{pa}(x_{ij}); \boldsymbol{\theta}_j) = \sum_{j=1}^{m}\sum_{i=1}^{n} \ln p(x_{ij} | \operatorname{pa}(x_{ij}); \boldsymbol{\theta}_j).$$

Thus for a given j, $\boldsymbol{\theta}_j$ only appears in the term $\sum_{i=1}^{n} \ln p(x_{ij} | \operatorname{pa}(x_{ij}); \boldsymbol{\theta}_j)$ 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.

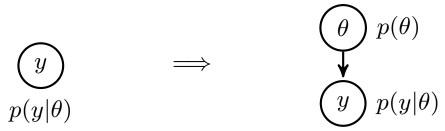
Exercise 10.2. For the TABC network above, what would our data look like? What is the ML estimate for each parameter based on this data? △

10.3 Bayesian Parameter Estimation in Bayesian Networks

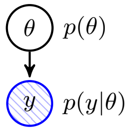
Suppose that we want to estimate the parameters of the conditional probability distributions of a Bayesian network using Bayesian inference. Since in the Bayesian view, parameters are considered random, we can augment the Bayesian network by adding the parameters as nodes. In particular, we can recast Bayesian estimation problems that we have seen before as Bayesian networks.

10.3.1 Bayesian Estimation formulated as Bayesian Networks

Example 10.3. As a simple example, consider the Bayesian network consisting of a singe node y whose distribution has an unknown parameter θ. We can transform this to a network with node θ and y, in which the conditional distributions are the prior $p(\theta)$ and the likelihood $p(y|\theta)$.



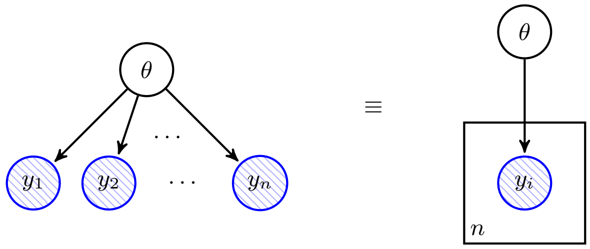
The joint distribution resulting from the network is $p(\theta, y) = p(\theta)p(y|\theta)$, which indeed factorizes with respect to the network on the right. Now, if y is given (which we show by a hatched pattern), we can find $p(\theta|y)$ using Bayes rule,



$$p(\theta|y) = \frac{p(y,\theta)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)}$$

△

Example 10.4. The problem in Example 10.3 becomes more interesting when we have n independent samples, $\mathcal{D} = \{y_1, y_2, \ldots, y_n\}$, from the distribution. We can simplify the network with the plate notation, by representing nodes that have the same conditional probability distribution (and are independent) using *plates*, as shown below.



The joint distribution of θ and y_1^n can be written as

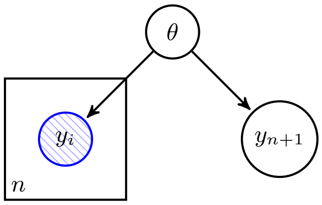
$$p(y_1^n, \theta) = p(\theta) \prod_{i=1}^{n} p(y_i|\theta),$$

and the posterior distribution for θ as

$$p(\theta|y_1^n) \propto p(y_1^n, \theta) = p(\theta) \prod_{i=1}^{n} p(y_i|\theta).$$

△

Example 10.5. Following Example 10.4, suppose we have n independent samples $\mathcal{D} = \{y_1, y_2, \ldots, y_n\}$ from the distribution. We want to predict the distribution of the next sample $p(y_{n+1}|y_1^n)$. The graph is shown below.



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,

$$\mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta, y_1^n]|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n]. \tag{10.1}$$

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 $p(\theta) \propto 1$, $y_i|\theta \sim \mathrm{Ber}(\theta)$, and that out of the n samples y_i, there s 1s and f 0s. Then

$$p(y_{n+1} = 1|y_1^n) = \mathbb{E}[y_{n+1}|y_1^n] = \mathbb{E}[\mathbb{E}[y_{n+1}|\theta]|y_1^n] = \mathbb{E}[\theta|y_1^n] = \frac{s+1}{s+f+2}.$$

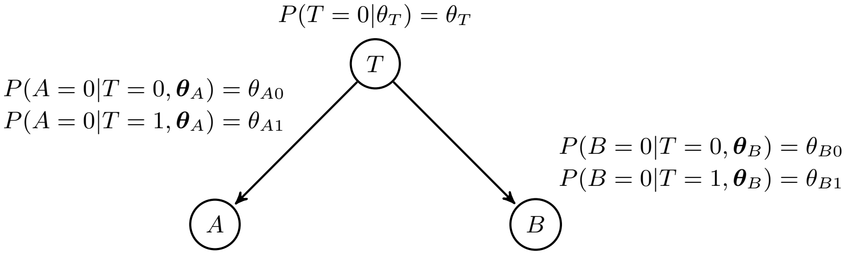
△

10.3.2 Estimating Parameters of CPDs in Bayesian Networks

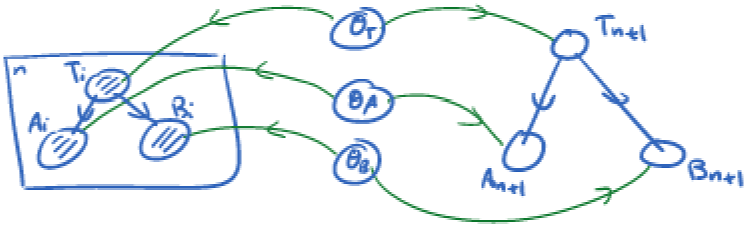
So far for the most part, we have cast Bayesian inference problems that we had seen before as Bayesian networks. In the next example, we consider the problem of estimating the parameters of the conditional probability distributions (**CPD**s) of Bayesian network.

Similar to Section 10.2, consider a BN with m nodes $x_1, \ldots, x_m$ in which the parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$ of the CPDs are unknown. Our dataset is $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ consisting of n iid samples, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im})$. We are interested in determining $p(\boldsymbol{\theta}|\mathcal{D})$ and $p(\boldsymbol{x}_{n+1}|\mathcal{D})$.

Example 10.6. Let us consider a simpler version of the network given in Example 10.1, with unknown parameter vector $\boldsymbol{\theta} = (\theta_T, \boldsymbol{\theta}_A, \boldsymbol{\theta}_B)$,

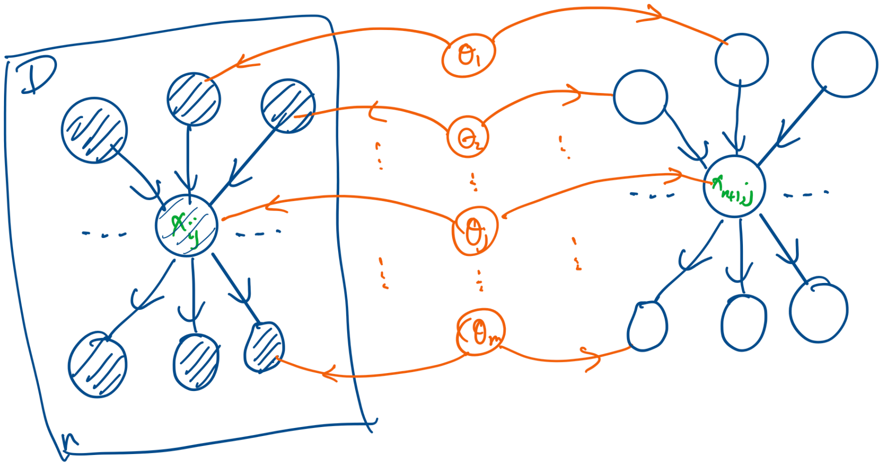

$P(T = 0|\theta_T) = \theta_T$
T
$P(A = 0|T = 0, \boldsymbol{\theta}_A) = \theta_{A0}$
$P(A = 0|T = 1, \boldsymbol{\theta}_A) = \theta_{A1}$
$P(B = 0|T = 0, \boldsymbol{\theta}_B) = \theta_{B0}$
$P(B = 0|T = 1, \boldsymbol{\theta}_B) = \theta_{B1}$
A
B

Given n samples $\mathcal{D} = \{(T_1, A_1, B_1), \dots, (T_n, A_n, B_n)\}$, and our goal is to estimate the posterior we augment the network as



so that we can learn about $p(\boldsymbol{\theta}_A, \boldsymbol{\theta}_B, \theta_T|\mathcal{D})$ and $p(T_{n+1}, A_{n+1}, B_{n+1}|\mathcal{D})$. △ △

Decomposability of posterior and predictive posterior. Consider a Bayesian network with $n \times m$ nodes for the data $\mathcal{D} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i = (x_{i1}, \dots, x_{im})$; m nodes for $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m$; and m nodes for the future observation $x_{n+1,1}, \dots, x_{n+1,m}$ as shown below (see also the second graph in Example 10.6 for a concrete example)



Let us start by trying to decompose $p(\boldsymbol{\theta}|\mathcal{D})$. First, note that by d-separation

$$p(\boldsymbol{\theta}|\mathcal{D}) = \prod_{j=1}^{m} p(\boldsymbol{\theta}_j|\mathcal{D}).$$

Next, define

$$N_j = \{x_{1j}, \dots, x_{nj}, \mathrm{pa}(x_{1j}), \dots, \mathrm{pa}(x_{nj})\}, \tag{10.2}$$

i.e., the set of children and parents of children of $\boldsymbol{\theta}_j$ among the nodes of $\mathcal{D}$. Similar to Markov blankets, we see that $\boldsymbol{\theta}_j \perp\!\!\!\perp \mathcal{D} \setminus N_j \mid N_j$. That is, given N_j, $\boldsymbol{\theta}_j$ is independent of all other nodes in

$\mathcal{D}$, and so $p(\boldsymbol{\theta}_j|\mathcal{D}) = p(\boldsymbol{\theta}_j|N_j)$. Hence,

$$p(\boldsymbol{\theta}|\mathcal{D}) = \prod_{j=1}^{m} p(\boldsymbol{\theta}_j|\mathcal{D}) = \prod_{j=1}^{m} p(\boldsymbol{\theta}_j|N_j). \tag{10.3}$$

This is good news, because it means we can find the posterior for the parameters of each CPD can be computed separately.

Exercise 10.7. Using the Bayesian network above, prove that the last two equalities in the expression below hold:

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{x}_{n+1} \mid \mathcal{D}) = p(\boldsymbol{\theta}|\mathcal{D})p(\boldsymbol{x}_{n+1} \mid \mathcal{D}, \boldsymbol{\theta}) &= p(\boldsymbol{\theta}|\mathcal{D})p(\boldsymbol{x}_{n+1}|\boldsymbol{\theta}) && (10.4)\\ &= p(\boldsymbol{\theta}|\mathcal{D}) \prod_{j=1}^{m} p(x_{n+1,j}|\boldsymbol{\theta}_j, \mathrm{pa}(x_{n+1,j})). && (10.5) \end{aligned}$$

We can find the posterior predictive $p(\boldsymbol{x}_{n+1}|\mathcal{D})$ by integrating the above expression with respect to $\boldsymbol{\theta}$. △ △

Example 10.8. Getting back to Example 10.6, let us find $p(\boldsymbol{\theta}_A|\mathcal{D})$ and $p(A_{n+1}, B_{n+1}|\mathcal{D})$. As in (10.2), the set of children and parents of children of $\boldsymbol{\theta}_A$ among data nodes are $N_A = \{A_1, \ldots, A_n, T_1, \ldots, T_n\}$ and

$$p(\boldsymbol{\theta}_A|\mathcal{D}) = p(\boldsymbol{\theta}_A|A_1^n, T_1^n).$$

This makes intuitive sense: to estimate the probability of Alice being late as a function of traffic, only the part of data that deals with Alice's arrival time and traffic is relevant.

Assuming that the prior satisfies $p(\boldsymbol{\theta}_A) = p(\theta_{A0})p(\theta_{A1})$,

$$\begin{aligned} p(\boldsymbol{\theta}_A|A_1^n, T_1^n) &\propto p(\boldsymbol{\theta}_A)p(T_1^n|\boldsymbol{\theta}_A)p(A_1^n|T_1^n, \boldsymbol{\theta}_A)\\ &= p(\boldsymbol{\theta}_A)p(T_1^n)p(A_1^n|T_1^n, \boldsymbol{\theta}_A)\\ &\propto p(\boldsymbol{\theta}_A)p(A_1^n|T_1^n, \boldsymbol{\theta}_A)\\ &= p(\boldsymbol{\theta}_A)\prod_{i=1}^{n} p(A_i|T_1^n, \boldsymbol{\theta}_A)\\ &= p(\boldsymbol{\theta}_A)\prod_{i=1}^{n} p(A_i|T_i, \boldsymbol{\theta}_A)\\ &= \left(p(\theta_{A0}) \prod_{i:T_i=0} p(A_i|T_i = 0, \theta_{A0})\right)\left(p(\theta_{A1}) \prod_{i:T_i=1} p(A_i|T_i = 1, \theta_{A1})\right). \end{aligned}$$

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_A^0 and θ_A^1 are

$$\begin{aligned} p(\theta_{A0}|\mathcal{D}) &\propto p(\theta_{A0}) \prod_{i:T_i=0} p(A_i|T_i = 0, \theta_{A0}),\\ p(\theta_{A1}|\mathcal{D}) &\propto p(\theta_{A1}) \prod_{i:T_i=1} p(A_i|T_i = 1, \theta_{A1}). \end{aligned}$$

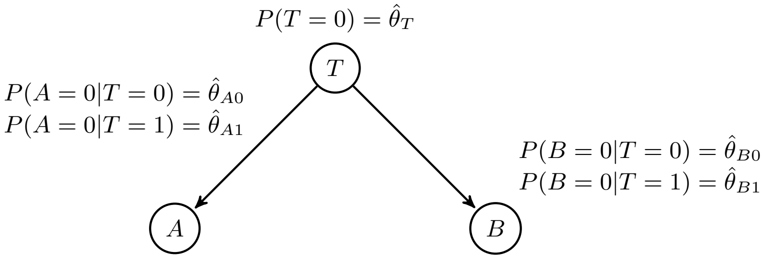
Suppose $p(\theta_A^0) \sim \text{Beta}(1,1)$ and out of 100 days with no traffic, in 40 days Alice was on time. Hence,

$$\theta_{A0}|\mathcal{D} \sim \text{Beta}(41, 61).$$

Furthermore, the posterior probability of the next sample (A_{n+1}, B_{n+1}) is

$$\begin{aligned} p(A_{n+1}, B_{n+1}|\mathcal{D}) &= \int_{\boldsymbol{\theta}} p(A_{n+1}, B_{n+1}, \boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{D}) p(A_{n+1}, B_{n+1}|\boldsymbol{\theta}) d\boldsymbol{\theta}. \end{aligned}$$

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.


$P(T=0) = \hat{\theta}_T$
T
$P(A=0|T=0) = \hat{\theta}_{A0}$
$P(A=0|T=1) = \hat{\theta}_{A1}$
$P(B=0|T=0) = \hat{\theta}_{B0}$
$P(B=0|T=1) = \hat{\theta}_{B1}$
A
B

△

10.4 Parameter Estimation in MRFs

Recall that for an MRF G, the probability distribution is given as

$$p(\boldsymbol{x}; \boldsymbol{\theta}) = \prod_{c \text{ is a clique in } G} \psi_{\boldsymbol{\theta}}(\boldsymbol{x}_c)/Z(\boldsymbol{\theta}),$$

where $Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_c \psi_{\boldsymbol{\theta}}(\boldsymbol{x}_c)$ is the partition function. Let us consider the frequentist estimation of $\boldsymbol{\theta}$, e.g., maximum likelihood. Unfortunately, the log-likelihood function does not decompose into terms each depending on one component of $\boldsymbol{\theta}$. This is due to the presence of the partition function, which generally depends on all the components of $\boldsymbol{\theta}$, 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.

Bibliography

[1] B. Hajek, *Random Processes for Engineers.* 2014.

[2] T. T. Nguyen and S. 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), pp. III–1085–III–1093, JMLR.org, June 2013.

[3] M. I. Jordan, *An Introduction to Probabilistic Graphical Models (Preprints and Course Notes).* 2003.

[4] D. J. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge university press, 2003.

[5] C. M. Bishop, *Pattern Recognition And Machine Learning.* New York: Springer, 2006.

[6] C. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[7] A. Furman, "WHAT IS . . . a Stationary Measure?," *Notices of the AMS*, vol. 58, no. 9.

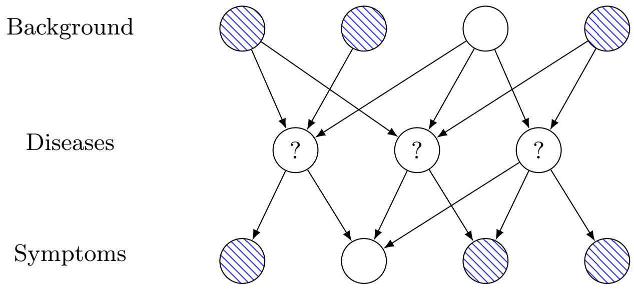
Chapter 11

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 $x_1 \to x_2 \to x_3 \to x_4 \to x_5$, 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).

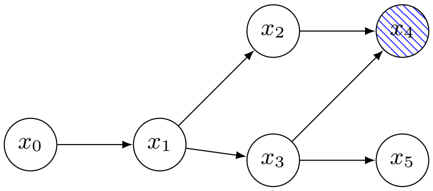
We could also choose the following ordering, which would lead to a different complexity:

$$\begin{aligned}
p(x_4) &= \sum_{x_1}\sum_{x_3}\sum_{x_2} p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3)\\
&= \sum_{x_1}\sum_{x_3} p(x_1)p(x_4|x_3)\sum_{x_2} p(x_2|x_1)p(x_3|x_2)\\
&= \sum_{x_1}\sum_{x_3} p(x_1)p(x_4|x_3)T_1(x_1,x_3)\\
&= \sum_{x_1} p(x_1)\sum_{x_3} p(x_4|x_3)T_1(x_1,x_3)\\
&= \sum_{x_1} p(x_1)T_2(x_1,x_4)\\
&= p(x_4).
\end{aligned}$$

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.

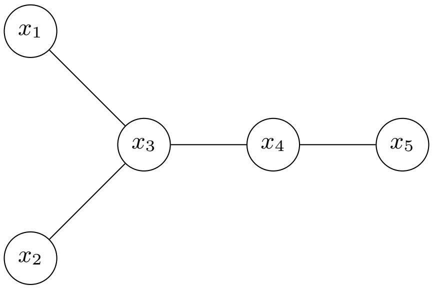
11.3 The Sum-Product Algorithm

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.

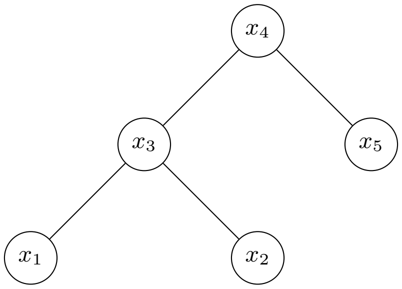
We need to clarify what we mean by trees. For Markov random fields, the algorithm works on trees, in the graph-theoretic sense. But for Bayesian networks, it works for graphs whose equivalent MRF (the moralized graph) is a tree. In particular, no node can have more than one parent. Given the straightforward 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

$$p(x_1^4) \propto \psi(x_1,x_3)\psi(x_2,x_3)\psi(x_2)\psi(x_3,x_4)\psi(x_4)\psi(x_4,x_5)$$



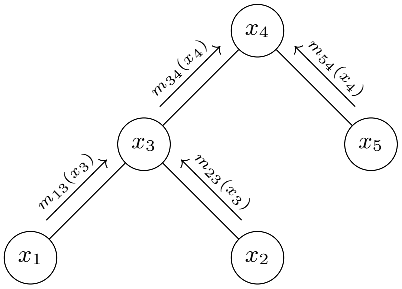
Let's look at this graph as a rooted tree,



and perform elimination starting from the leaves to the roots:

$$\begin{aligned}
p(x_4) &\propto \sum_{x_1,x_2,x_3,x_5} \psi(x_1,x_3)\psi(x_2,x_3)\psi(x_2)\psi(x_3,x_4)\psi(x_4)\psi(x_4,x_5) \\
&= \sum_{x_3} \psi(x_3,x_4)\psi(x_4)\left(\sum_{x_1}\psi(x_1,x_3)\right)\left(\sum_{x_2}\psi(x_2,x_3)\psi(x_2)\right)\left(\sum_{x_5}\psi(x_4,x_5)\right) \\
&= \sum_{x_3} \psi(x_3,x_4)\psi(x_4)\ m_{13}(x_3)\ m_{23}(x_3)\ m_{54}(x_4) \\
&= \psi(x_4)m_{54}(x_4)\sum_{x_3}\psi(x_3,x_4)\ m_{13}(x_3)\ m_{23}(x_3) \\
&= \psi(x_4)m_{54}(x_4)m_{34}(x_4)
\end{aligned} \tag{11.1}$$

We can view this computation as being done on each node and then messages being passed to neighbors:



where

$$\begin{aligned} m_{13}(x_3) &= \sum_{x_1} \psi(x_1, x_3), \\ m_{23}(x_4) &= \sum_{x_2} \psi(x_2, x_3)\psi(x_2), \\ m_{54}(x_4) &= \sum_{x_5} \psi(x_4, x_5), \\ m_{34}(x_4) &= \sum_{x_3} \psi(x_3, x_4)\ m_{13}(x_3)\ m_{23}(x_3). \end{aligned}$$

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.

- Sum step: Sum over all possible values of x_j to produce the message

  $$m_{jk}(x_k) = \sum_{x_j} \psi(x_j)\psi(x_j, x_k) \prod_i m_{ij}(x_j), \tag{11.2}$$

  and send to x_k.

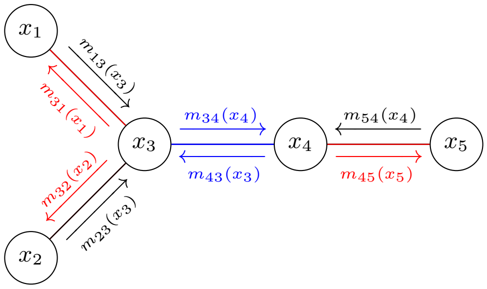
A critical point in the correctness of the sum-product algorithm is that the messages received by each node are functions of the value of that node. This is easy to see by induction. After the product step, we get a function of both the current node x_j and its parent x_k. The sum eliminates the current 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 $\{1, 2, \ldots, K\}$. So the sum in (11.2) contains K terms. Furthermore, $m_{jk}(x_k)$ needs to be computed for $x_k = 1, 2, \ldots, K$. We can imagine a vector

$$\boldsymbol{m}_{jk} = \begin{pmatrix} m_{jk}(1) \\ m_{jk}(2) \\ \vdots \\ m_{jk}(K) \end{pmatrix}$$

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:



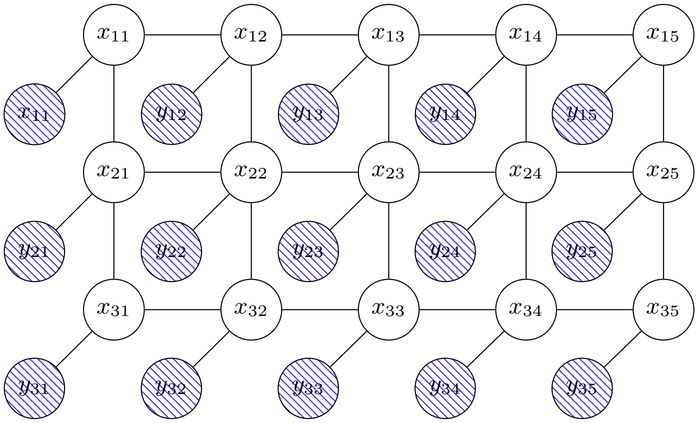
Here the order of messages is color-coded: 1, 2, 3. We can now find the marginal at each node. For example,

$$\begin{aligned} p(x_2) &\propto m_{32}(x_2)\psi(x_2), \\ p(x_3) &\propto m_{13}(x_3)m_{23}(x_3)m_{43}(x_3). \end{aligned}$$

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

$$\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}).$$

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

$$\arg\max_{x_1,x_2} p(x_1, x_2) = (1, 0)$$
$$\arg\max_{x_1} p(x_1) = \arg\max_{x_1}(p(x_1, x_2 = 0) + p(x_1, x_2 = 1)) = 0.$$

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:

$$\begin{aligned}m_{13}(x_3) &= \max_{x_1}\psi(x_1,x_3),\\ m_{23}(x_4) &= \max_{x_2}\psi(x_2,x_3)\psi(x_2),\\ m_{54}(x_4) &= \max_{x_5}\psi(x_4,x_5),\\ m_{34}(x_4) &= \max_{x_3}\psi(x_3,x_4)\ m_{13}(x_3)\ m_{23}(x_3).\end{aligned}$$

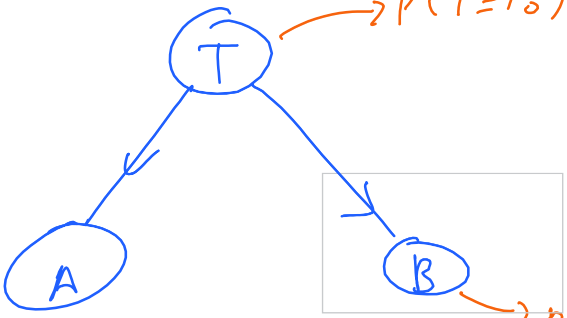
If we have Z, we can find the maximum probability. But we are interested in the values x^* of x that achieve this maximum probability (also we don't have Z). To find x^*, we simply need to keep track of which values of x_i maximize the message. Specifically, for a message $m_{ij}(x_j)$, we should know 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_is.

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:



$P(T=T_0) = 0.65 = 1-P(T=T_1)$

$P(B=B_0 \mid T=T_0) = 0.82$

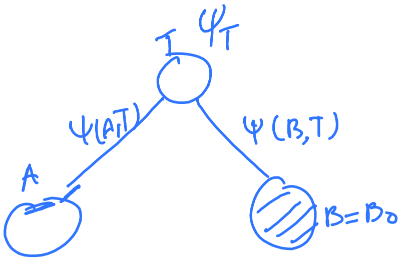
$P(B=B_0 \mid T=T_1) = 0.15$

$P(A=A_0 \mid T=T_0) = 0.9$

$P(A=A_0 \mid T=T_1) = 0.5$

$P(ABT) = P(T)\,P(A \mid T)\,P(B \mid T)$

We first convert this to an MRF



$\psi_T(T) = P(T) =$

T=0	T=1
0.65	0.35

$\psi_{BT}(B,T) = P(B \mid T) =$

	B=0	B=1
T=0	0.82	0.18
T=1	0.15	0.85

$\psi_{AT}(A,T) = P(A \mid T) =$

	A=0	A=1
T=0	0.9	0.1
T=1	0.5	0.5

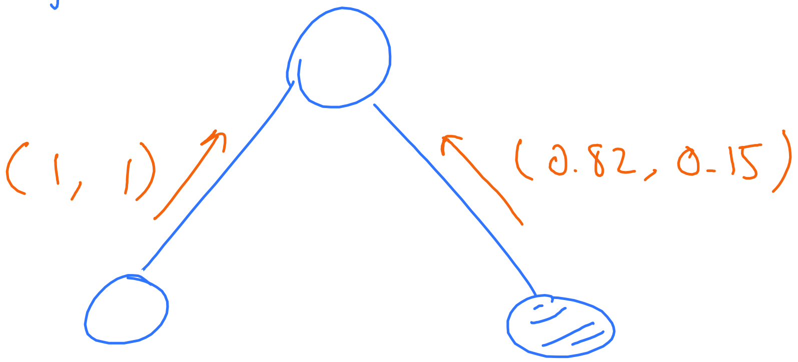
$$\mu_{BT}(T=0) = \sum_{B\in\{0\}} \psi(B, T=0) = 0.82$$

$$\mu_{BT}(T=1) = \sum_{B\in\{0\}} \psi(B, T=1) = 0.15$$

$$\psi(A, T=0) = 0.9 + 0.1 = 1$$

$$\mu_{AT}(T=0) = \sum_{A\in\{0,1\}}$$

$$\mu_{AT}(T=1) = \sum_{A\in\{0,1\}} \psi(A, T=1) = 0.5 + 0.5 = 1$$



$$P(T \mid B=B_0) \propto \mu_{AT}(T)\, \mu_{BT}(T)\, \psi_T(T)$$

$T=0$: $1 \times 0.82 \times 0.65$

$T=1$: $1 \times 0.15 \times 0.35$

Normalization :

$$P(T=0 \mid B=0) = \frac{0.82 \times 0.65}{0.82 \times 0.65 + 0.15 \times 0.35} = 0.91$$

$$\mu_{TA}(A=0) = \sum_{T\in\{0,1\}} \mu_{BT}(T)\,\psi(T, A=0)\,\psi(T)$$

$$= 0.65 \times 0.82 \times 0.9 + 0.35 \times 0.15 \times 0.5$$

$$= 0.506$$

$$\mu_{TA}(A=1) = \sum_{T\in\{0,1\}} \mu_{BT}(T)\,\psi(T, A=1)\,\psi(T)$$

$$= 0.65 \times 0.82 \times 0.1 + 0.35 \times 0.15 \times 0.5$$

$$= 0.08$$

$$P(A \mid B=0) \propto \mu_{TA}(A)$$

$A = A_0$: 0.506

$A = A_1$: 0.08

Normalization

$$P(A=0 \mid B=0) = \frac{0.506}{0.506 + 0.08} = \frac{0.506}{0.596} = 0.863$$

$$\mu_{TB}(B=0) = \sum_{T\in\{0,1\}} \mu_{AT}(T)\,\psi(T, B=0)$$

$$= 1 \times 0.82 + 1 \times 0.15$$

$$= 0.97$$

But the only case is $B=0$, so after normalization regardless of the value of $\mu_{TB}(B=0)$, we have

$$p(B=0 \mid B=0) = 1$$

Bibliography

[1] B. Hajek, *Random Processes for Engineers.* 2014.

[2] T. T. Nguyen and S. 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), pp. III–1085–III–1093, JMLR.org, June 2013.

[3] M. I. Jordan, *An Introduction to Probabilistic Graphical Models (Preprints and Course Notes).* 2003.

[4] D. J. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge university press, 2003.

[5] C. M. Bishop, *Pattern Recognition And Machine Learning.* New York: Springer, 2006.

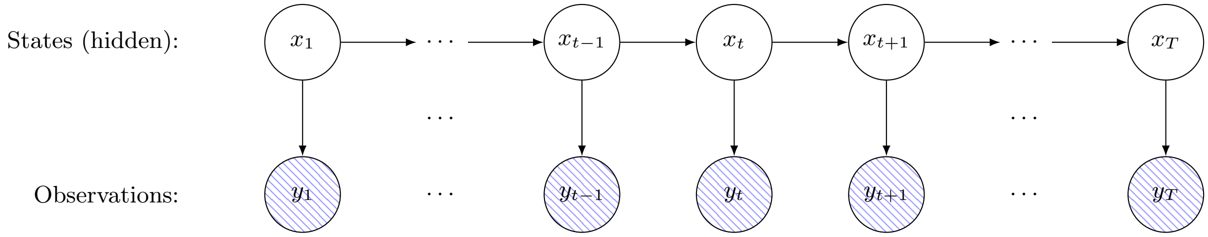
[6] C. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[7] A. Furman, “WHAT IS . . . a Stationary Measure?,” *Notices of the AMS*, vol. 58, no. 9.

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,

$$B_{ij} = p(y_t = j|x_t = i).$$

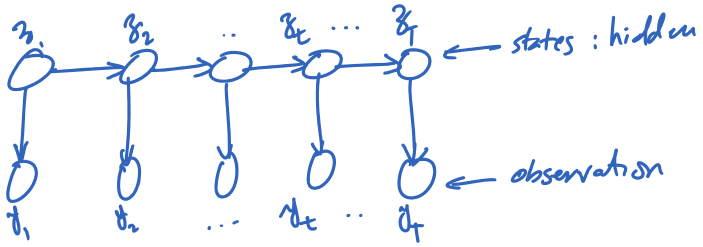
Below are three common inference problems associated with HMMs and the methods for solving them. We will not derive the solutions but they can be found in [1].

- Evaluation: $p(x_t|y_1^T;\theta) \to$ *forward-backward algorithm* (sum-product).
- Decoding: $\arg\max_{x_1^T} p(x_1^T|y_1^T;\theta) \to$ *Viterbi algorithm* (max-product).
- Learning: $\arg\max_\theta p(y_1^T;\theta) \to$ *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.

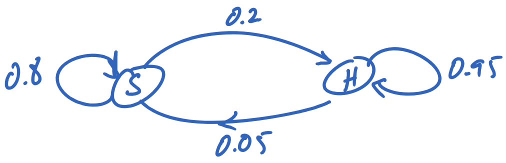
HMM

Tuesday, November 07, 2017 5:06 PM



* A person can be either sick or healthy : hidden state

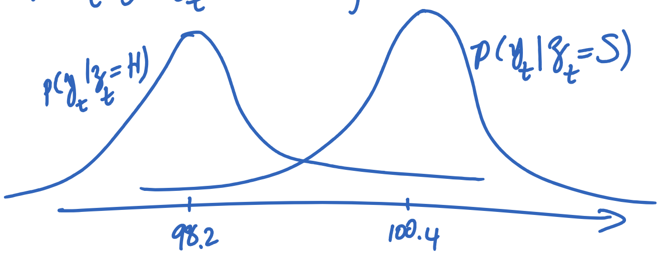
temprature : observation



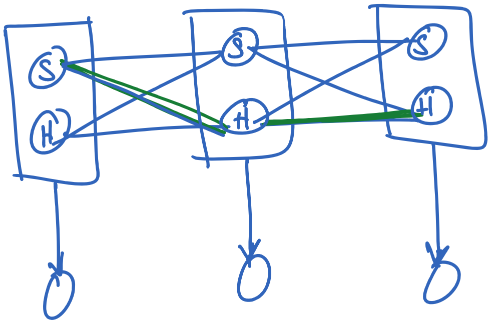
$p(z_{t+1}=j \mid z_t=i) = A_{ij}$: transition probs.

prob. distn over initial state $p(z_1=i)=\pi_i$

$p(y_t=j \mid z_t=i) = B_{ij}$: emission probs.



Trellis :



paths : configurations of hidden states

$$p(z_1^T, y_1^T \mid \theta) = p(z_1)\left(\prod_{t=2}^{T} p(z_t \mid z_{t-1})\right)\left(\prod_{t=1}^{T} p(y_t \mid z_t)\right)$$

$\theta = (\pi, A, B)$

$\pi_i = p(z_1 = i \mid \theta)$

$A_{ij} = p(z_{t+1} = j \mid z_t = i, \theta)$

$B_{ij} = p(y_t = j \mid z_t = i, \theta)$

Three HMM problems:

$*$ Evaluation : $p(z_t \mid y_1^T, \theta)$

– Forward – Backward (Sum – product)

$*$ Decoding : $\arg\max_{z_1^T} p(z_1^T \mid y_1^T, \theta)$

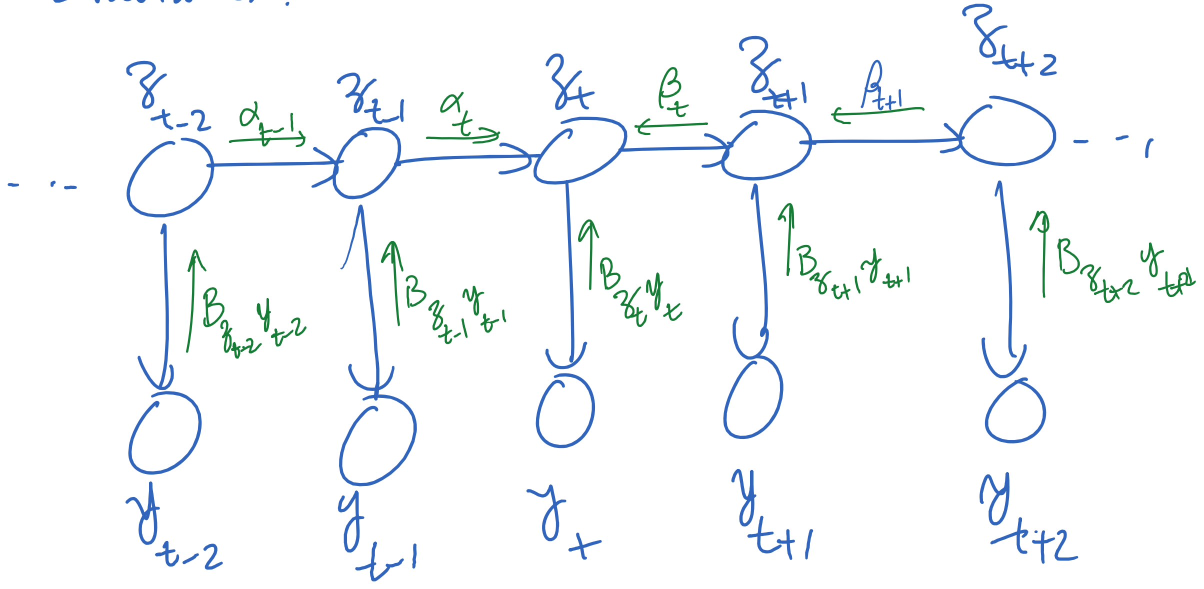
– Viterbi Alg (Max – product)

Cofounder of Qualcomm

$*$ Learning : $\arg\max_{\theta} p(y_1^T \mid \theta)$

– Baum – Welch alg (EM)

Evaluation:



Define

$\dots \; \mu \; (z = i) \quad t \geq 2 \;\Big|\; \beta(i) = \mu \; (z_t = i) \quad t \leq T-1$

Define

$$\alpha_t(i) = \mu_{z_{t-1} z_t}(z_t = i) \quad t \geq 2 \qquad \Big| \qquad \beta_t(i) = \mu_{z_{t+1} z_t}(z_t = i) \quad t \leq T-1$$

$$\alpha_1(i) = \pi_i \qquad \Big| \qquad \beta_T(i) = 1$$

It can be shown (by induction) that:

$$\alpha_t(i) = p(z_t = i, y_1^{t-1} \mid \theta), \qquad \beta_t(i) = p(y_{t+1}^T \mid z_t = i, \theta)$$

$$\alpha_t(i) = \sum_j \overset{\downarrow z_{t-1}=j}{\alpha_{t-1}(j)} B_{jy_{t-1}} A_{ji} \qquad \beta_t(i) = \sum_j \beta_{t+1}(j) B_{jy_{t+1}} A_{ij}$$

Marginals:

$$p(z_t = i \mid y_1^T, \theta) = \gamma_t(i) \propto \alpha_t(i) \beta_t(i) B_{iy_t}$$

$$p(z_{t-1} = i, z_t = j \mid y_1^T, \theta) = \xi_t(i,j) \propto p(y_1^T, z_{t-1} = i, z_t = j \mid \theta)$$

$$= p(y_1^{t-2}, z_{t-1} = i) \underbrace{p(y_{t-1} \mid z_{t-1} = i)}_{\downarrow = \; p(y_{t-1} \mid y_1^{t-2}, z_{t-1} = i)} p(z_t = j \mid z_{t-1} = i) \, p(y_t \mid z_t = j) \, p(y_{t+1}^T \mid z_t = j)$$

$$= \alpha_{t-1}(i) B_{iy_{t-1}} A_{ij} B_{jy_t} \beta_t(j)$$

In traditional form of Forward-Backward, forward msgs included B_{iy_t}.

$$\bar{\alpha}_t(i) = p(z_t = i, y_1^t \mid \theta) = \alpha_t(i) B_{iy_t}$$

$$\bar{\alpha}_t(i) = \sum_j \bar{\alpha}_{t-1}(j) A_{ji} B_{iy_t}$$

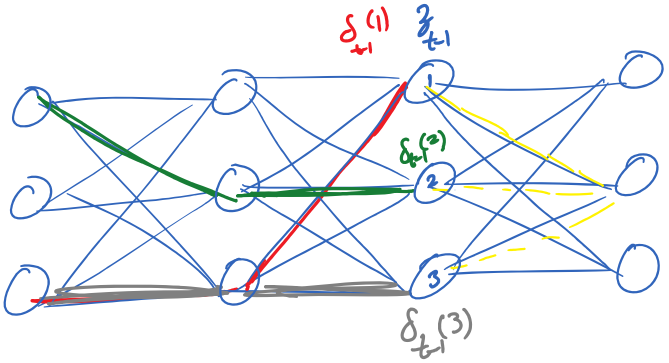
$$\gamma_t(i) \propto \bar{\alpha}_t(i) \beta_t(i)$$

Max-product: choose z_T as root.

Define $\delta_t(i) = \mu_{z_{t-1} z_t}(z_t = i)$, $t \geq 2$ $\qquad \delta_1(i) = \pi_i$

Can be shown that

$$\delta_t(i) = \max_{z_1^{t-1}} p(z_1^{t-1}, z_t = i, y_1^{t-1} | \theta)$$



$$\delta_t(i) = \max_j \delta_{t-1}(j) B_{jy_{t-1}} A_{ji}$$

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 $\theta = (\pi, A, B)$

$$\hat{\pi}_i = \begin{cases} 1 & z_1 = i \\ 0 & \text{else} \end{cases}$$

$$\hat{A}_{ij} = \frac{\sum_{t=1}^{T-1} I(z_t = i, z_{t+1} = j)}{\sum I(z_t = i)}$$

$$\hat{A}_{ij} = \frac{\sum_{t=1} I(z_t \quad {}_{t+1}}{\sum_t I(z_t=i)}$$

$$\hat{B}_{ij} = \frac{\sum_{t=1}^{T} I(z_t=i, y_t=j)}{\sum_{t=1}^{T} I(z_t=i)}$$

log likelihood of the complete data

$$\ln p(z_1^T, y_1^T \mid \theta) = \ln \pi_{z_1} + \sum_{t=2}^{T} \ln A_{z_{t-1} z_t} + \sum_{t=1}^{T} \ln B_{z_t y_t}$$

E-step

$$Q(\theta \mid \theta') = E\left[\ln p(z_1^T, y_1^T \mid \theta) \mid y_1^T, \theta'\right]$$

$$E\left[\ln \pi_{z_1} \mid y_1^T, \theta'\right] = \sum_i (\ln \pi_i) \underbrace{p(z_1 = i \mid y_1^T, \theta')}_{\gamma_1'(i)} = \sum_i \gamma_1'(i) \ln \pi_i$$

$$E\left[\sum_{t=2}^{T} \ln A_{z_{t-1} z_t} \mid y_1^T \theta'\right] = \sum_{t=2}^{T} \sum_i \sum_j (\ln A_{ij}) \underbrace{p(z_t = j, z_{t-1} = i \mid y_1^T, \theta')}_{\xi_t'(i,j)}$$

$$= \sum_i \sum_j \left(\sum_{t=2}^{T} \xi_t'(i,j)\right) \ln A_{ij}$$

$$E\left[\sum_{t=1}^{T} \ln B_{z_t y_t} \mid y_1^T, \theta'\right] = \sum_i \sum_{t=1}^{T} \underbrace{p(z_t = i \mid y_1^T, \theta')}_{\gamma_t'(i)} \ln B_{i y_t}$$

ML for p if the LL $= \sum_j n_j \ln p_j \Rightarrow p_j \propto n_j$

$$\pi_i'' \propto \gamma_1'(i) \qquad A_{ij}'' \propto \sum_{t=2}^{T} \xi_t'(i,j)$$

$$\rightarrow = \sum_i \sum_j \left(\sum_{t=1}^{T} \gamma_t'(i)\, I(y_t = j)\right) \ln B_{ij} \Rightarrow B_{ij} \propto \sum_{t=1}^{T} \gamma_t'(i)\, I(y_t = j)$$

Bibliography

[1] B. Hajek, *Random Processes for Engineers.* 2014.

[2] T. T. Nguyen and S. 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), pp. III–1085–III–1093, JMLR.org, June 2013.

[3] M. I. Jordan, *An Introduction to Probabilistic Graphical Models (Preprints and Course Notes).* 2003.

[4] D. J. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge university press, 2003.

[5] C. M. Bishop, *Pattern Recognition And Machine Learning.* New York: Springer, 2006.

[6] C. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[7] A. Furman, “WHAT IS . . . a Stationary Measure?,” *Notices of the AMS*, vol. 58, no. 9.

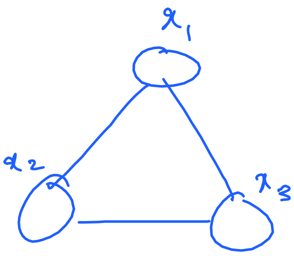
Chapter 13

Factor Graphs and Sum/Max-product Algorithms **

Factor graphs

Sunday, August 27, 2017 4:31 PM

This MRF implies the factorization



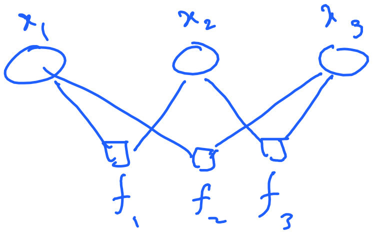
$$p(x_1, x_2, x_3) \propto \psi(x_1, x_2, x_3)$$

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

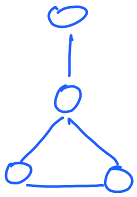
$$p(x_1^m) = \prod_{f_i \in F} f_i(x_{f_i})$$

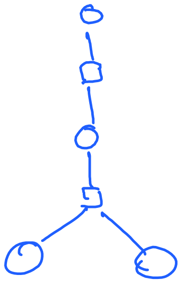
variable nodes adjacent to f_i

MRF → FG

nodes → variable nodes

maximal cliques → factor nodes

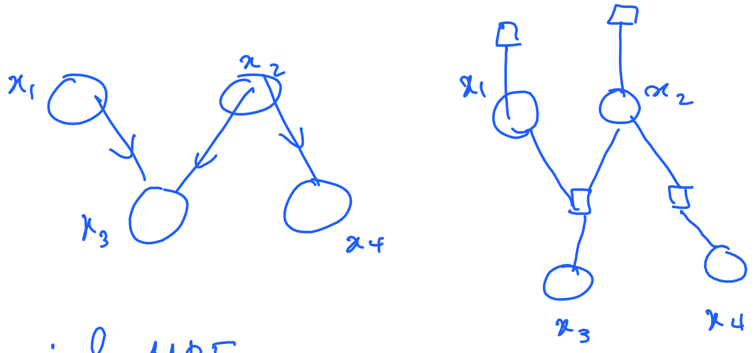




BN → FG

nodes → variable nodes

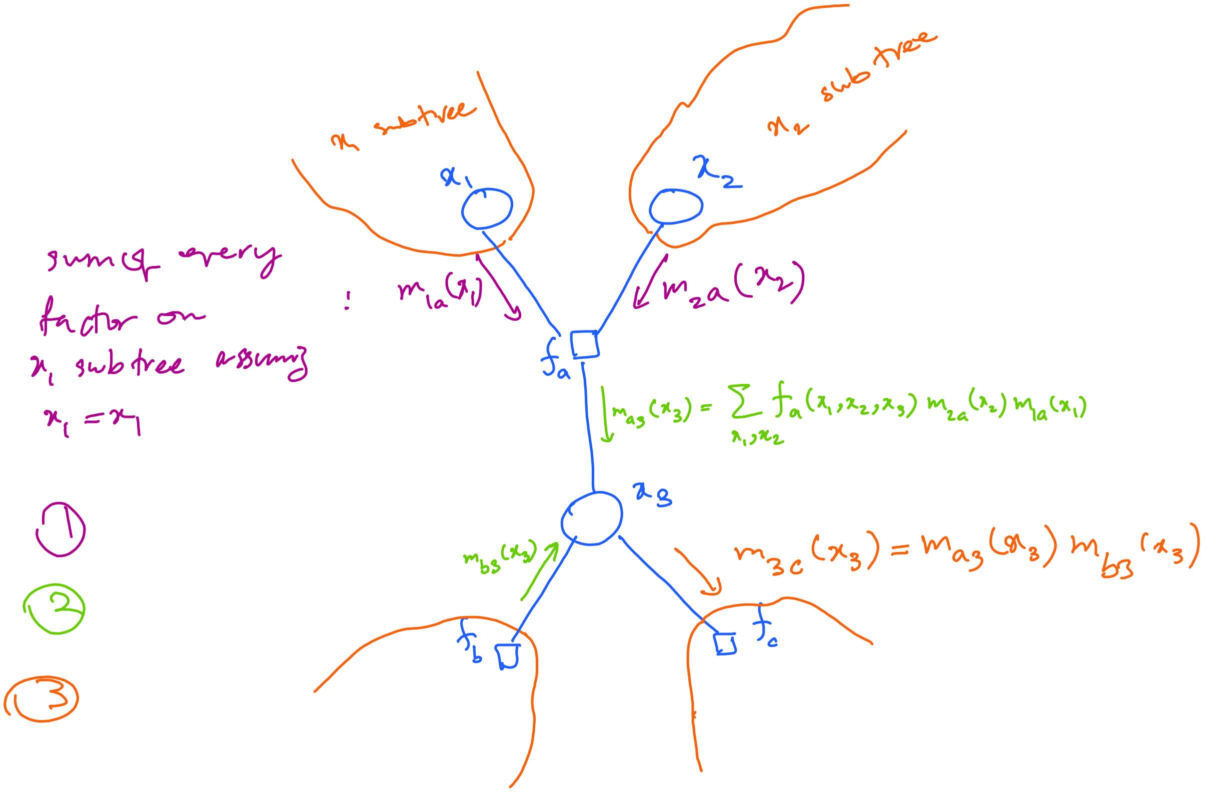
(CPDs at) nodes → factor nodes



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 many still be a tree: that is discarding the types of the nodes in FG leads to a tree.

This is good because ↴

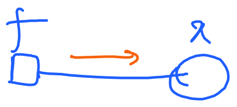
Sum-product for factor tree graphs:



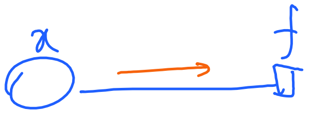
$$m_{a3}(x_3) = \sum_{x_1, x_2} f_a(x_1, x_2, x_3)\, m_{2a}(x_2)\, m_{1a}(x_1)$$

$$m_{3c}(x_3) = m_{a3}(x_3)\, m_{b3}(x_3)$$

Starting steps at leaves:



$m_{fx}(x) = f(x)$

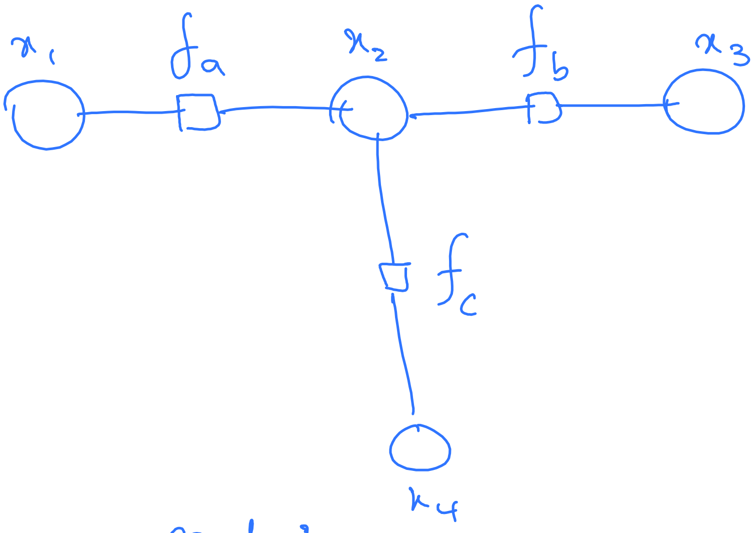


$m_{xf} = 1$

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

$\mu_{1a}(x_1) = 1$

$\mu_{4c}(x_4) = 1$

$\mu_{3b}(x_3) = 1$

round 2

$\mu_{a2}(x_2) = \sum_{x_1} f_a(x_1, x_2)\, \mu_{1a}(x_1)$

$\mu_{b2}(x_2) = \sum_{x_3} f_b(x_3, x_2)\, \mu_{3b}(x_3)$

$\mu_{c2}(x_2) = \sum_{x_4} f_c(x_2, x_4)\, \mu_{4c}(x_4)$

Max-product

Sunday, August 27, 2017 6:53 PM

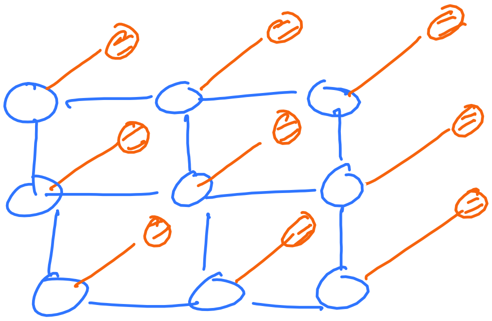
Problem: Identify the most likely configuration:

Find a set of values $x_1^*, \dots, x_m^*$ s.t.

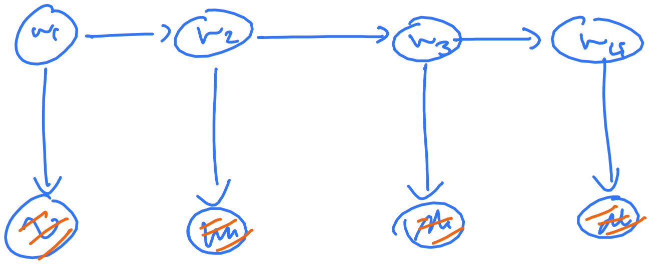
$$P(x_1^*, \dots, x_m^*) \geq P(x_1^m) \text{ for all } x_1^m$$

Applications:

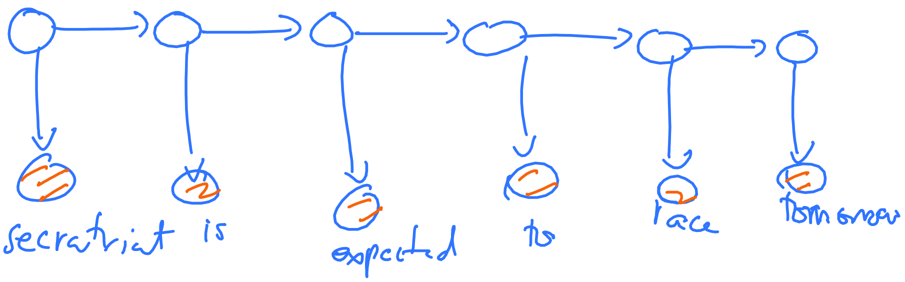
- Image denoising (Lab 1)



- 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

$$x_{max} = \arg\max \; p(x)$$

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)

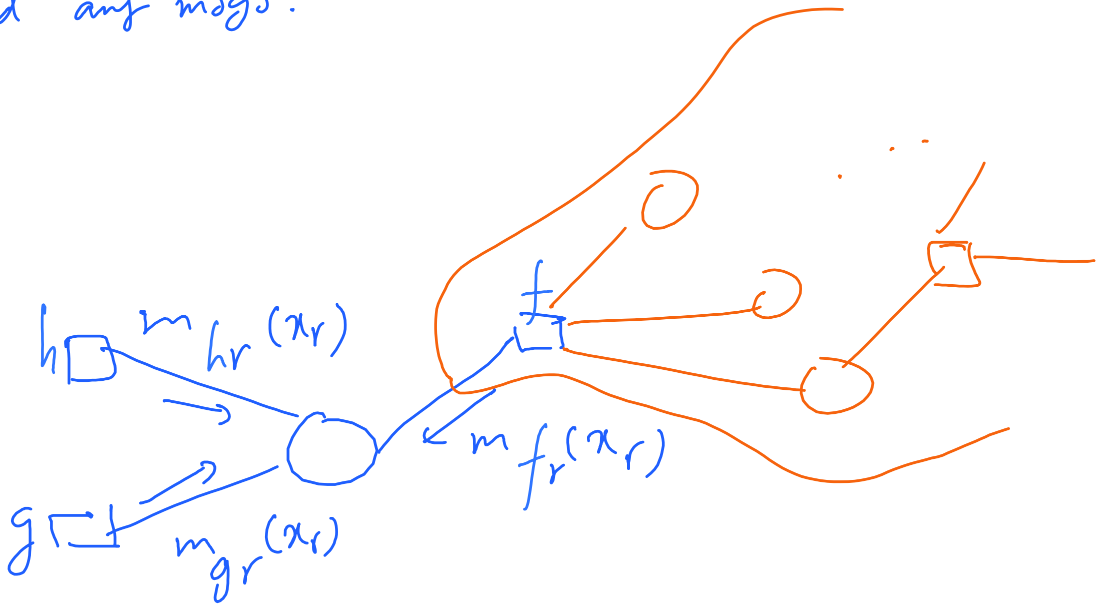
Let's instead try to find $\max_{x_1^n} p(x_1^n)$

$$\max_{x_1^n} p(x_1^n) = \max_{x_1} \max_{x_2} \cdots \max_{x_n} p(x_1^n)$$

* 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 $p(x)$

* 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

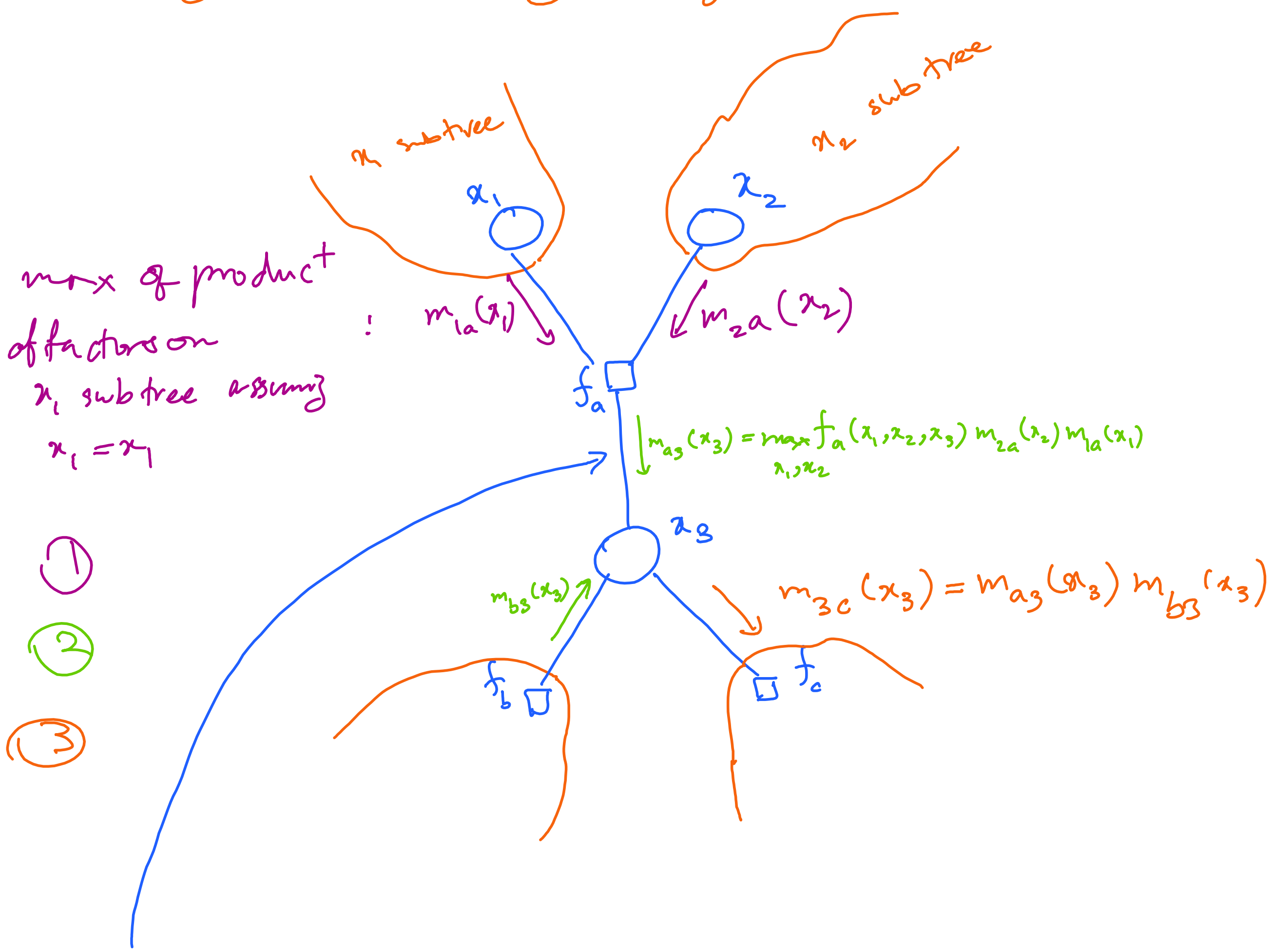
- What is the most likely configuration for the subtree of f

- What is the "probability" of this configuration

$$\max p(x_i^m) = \max_{x_r} m_{fr}(x_r)\, m_{gr}(x_r)\, m_{hr}(x_r)$$

from the value of x_r that maximizes this sum and messages we find the most likely configuration

* Finding the maximizing configuration:


x_1 subtree
x_1
x_2 subtree
x_2
max of product of factors on x_1 subtree assuming $x_1 = x_1$
①
②
③
$m_{1a}(x_1)$
$m_{2a}(x_2)$
f_a
$m_{a3}(x_3) = \max_{x_1, x_2} f_a(x_1, x_2, x_3)\, m_{2a}(x_2)\, m_{1a}(x_1)$
x_3
$m_{b3}(x_3)$
$m_{3c}(x_3) = m_{a3}(x_3)\, m_{b3}(x_3)$
f_b
f_c

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 backtrack and find maximizing values for all nodes.

* It may be more convenient to maximize

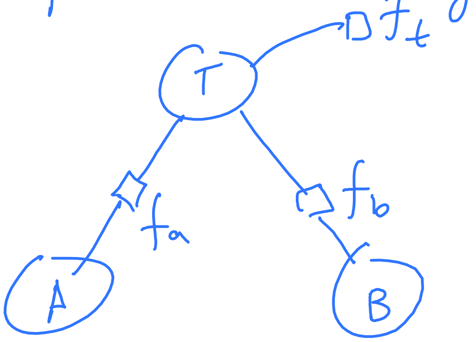
$\ln p(x) \implies$ max-sum algorithm

$\sum f_i(x_{f_i})$

$\max \; y \sim p(x) = \max_{x_1} \quad x_n \quad i \quad .$

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



$f_t(T=0) = 0.65$

$f_t(T=1) = 0.35$

$f_a(A=0, T=0) = 0.9$

$f_a(A=0, T=1) = 0.5$

$f_a(A=1, T=0) = 0.1$

$f_a(A=1, T=1) = 0.5$

$f_b(B=0, T=0) = 0.82$

$f_b(B=0, T=1) = 0.15$

$f_b(B=1, T=0) = 0.18$

$f_a(B=1, T=0) = 0.12$

μ_{Aa}: max 1

$A=0 \longrightarrow 1$

$A=1 \longrightarrow 1$

μ_{aT} : $\max_{A} f_a(A,T)$

$T=0 \longrightarrow 0.9$ for $A=0$

$T=1 \longrightarrow 0.5$ for $A=0$ & $A=1$

μ_{tT} : $\max f_t(T)$

$T=0 \longrightarrow 0.65$

$T=1 \longrightarrow 0.35$

μ_{Tb} : $\max_{T} \mu_{tT}(T)\, \mu_{aT}(T)\, f_b(B,T)$

...

Bibliography

[1] B. Hajek, *Random Processes for Engineers.* 2014.

[2] T. T. Nguyen and S. 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), pp. III–1085–III–1093, JMLR.org, June 2013.

[3] M. I. Jordan, *An Introduction to Probabilistic Graphical Models (Preprints and Course Notes).* 2003.

[4] D. J. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge university press, 2003.

[5] C. M. Bishop, *Pattern Recognition And Machine Learning.* New York: Springer, 2006.

[6] C. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

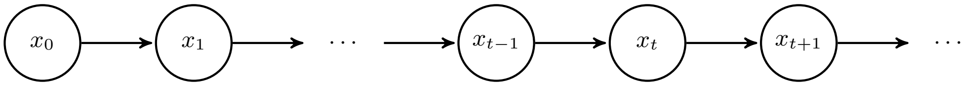
[7] A. Furman, “WHAT IS . . . a Stationary Measure?,” *Notices of the AMS*, vol. 58, no. 9.

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 $\{0, 1, 2, \ldots\}$.

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

$$P_{ij} = p(x_{t+1} = j|x_t = i),$$

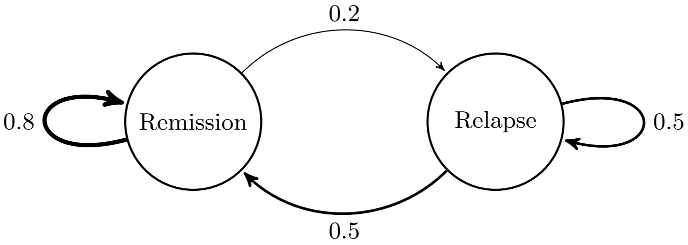
which is particularly useful if the state space is a finite set.

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

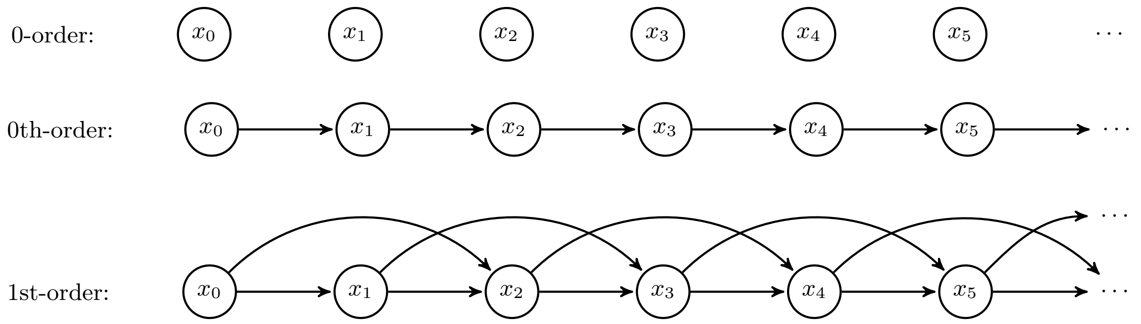


Here are some other examples of common MCs:

- **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 ([6]). 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 QPAAMKBZAACIBZLHJQD

2. Zero-order approximation (symbols independent but their probability is the same as English text).

   > OCRO IlLI RGWR NMIELWIS EU LL NBNESEBYA TH EEl ALHENHTTPA OOBTTVA NAH BRL

3. First-order approximation (digram structure; the conditional probability of each symbol given the previous is like English).

   > ON IE ANTSOUTINYS 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 DEMONSTURES 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 LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED

△

14.2 State distribution as a function of time

Consider a MC with m states. Let $\boldsymbol{\pi}_t = (\pi_{t1}, \pi_{t2}, \ldots, \pi_{tm})$ denote the probability distribution over the states at time t, where $\pi_{tj} = p(x_t = j)$. Usually, $\boldsymbol{\pi}_0$, 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

$$\boldsymbol{\pi}_t = \boldsymbol{\pi}_{t-1} P \qquad \text{and} \qquad \boldsymbol{\pi}_t = \boldsymbol{\pi}_0 P^t.$$

Furthermore, the ijth 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,

$$\boldsymbol{\pi}_1 = (1, 0) \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix} = (0.8, 0.2), \qquad \boldsymbol{\pi}_2 = \boldsymbol{\pi}_1 \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix} = (0.74, 0.26)$$

$$\boldsymbol{\pi}_5 = \boldsymbol{\pi}_0 P^5 = (0.71498, 0.28502), \qquad \boldsymbol{\pi}_{10} = \boldsymbol{\pi}_0 P^{10} = (0.71429, 0.28571)$$

So after 10 days, the probability of being in remission is about 71%.

Now suppose the patient starts in relapse. Then

$$\boldsymbol{\pi}_1 = (0.5, 0.5), \qquad \boldsymbol{\pi}_2 = (0.65, 0.35)$$

$$\boldsymbol{\pi}_5 = (0.71255, 0.28745), \qquad \boldsymbol{\pi}_{10} = (0.71428, 0.28572)$$

We can see that, interestingly, $\boldsymbol{\pi}_5$ and $\boldsymbol{\pi}_{10}$ 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 $\boldsymbol{\sigma}$ that satisfies

$$\boldsymbol{\sigma} = \boldsymbol{\sigma} P.$$

Any finite-state Markov chain has at least one stationary distribution [7]. The limiting distribution, if it exists, must be a stationary distribution.

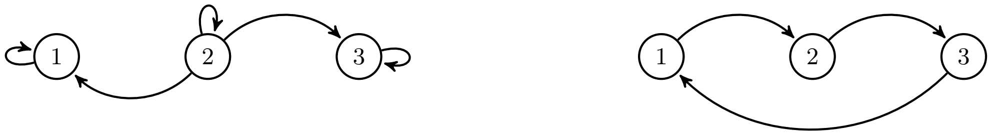
Example 14.4 (Example 14.3 continued)**.** The stationary distribution $\boldsymbol{\sigma} = (\sigma_1, \sigma_2)$ is obtained by solving $(\sigma_1, \sigma_2) = (\sigma_1, \sigma_2)P$ and $\sigma_1 + \sigma_2 = 1$. It can be shown that the unique solution to these equations is

$$\boldsymbol{\sigma} = (5/7, 2/7) = (0.71429, 0.28571),$$

which indeed appears to be the limiting distribution regardless of the initial distribution. △

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 $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma} = \boldsymbol{\sigma}P$ and for any $\boldsymbol{\pi}_0$, we have $\boldsymbol{\pi}_t = \boldsymbol{\pi}_0 P^t \to \boldsymbol{\sigma}$ as $t \to \infty$.*

The above theorem guarantees that regular MCs converge to their unique stationary distributions. Furthermore, since we can choose $\boldsymbol{\pi}_0$ to have a 1 in any position, the theorem also implies that each row of P^t converges to $\boldsymbol{\sigma}$.

Example 14.6 (Example 14.4 continued)**.** Indeed, $\boldsymbol{\sigma} = (5/7, 2/7) = (0.71429, 0.28571)$ is the stationary distribution of

$$P = \begin{pmatrix} 0.8 & 0.2 \\ 0.5 & 0.5 \end{pmatrix}$$

and $\boldsymbol{\pi}_t \to \boldsymbol{\sigma}$ regardless of $\boldsymbol{\pi}_0$ as we saw in Example 14.1. Furthermore,

$$P^2 = \begin{pmatrix} 0.74 & 0.26 \\ 0.65 & 0.35 \end{pmatrix}, \qquad P^5 = \begin{pmatrix} 0.71498 & 0.28502 \\ 0.71255 & 0.28745 \end{pmatrix}, \qquad P^{10} = \begin{pmatrix} 0.71429 & 0.28571 \\ 0.71428 & 0.28572 \end{pmatrix}$$

△

14.3.1 How often does the Markov Chain visit each state?

For a regular MC with stationary distribution $\boldsymbol{\sigma}$, 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 Global Balance

A distribution $\boldsymbol{\pi}$ over the states of the MC satisfies the **Global Balance Property (GBP)** if for any partition[1] $\{R, L\}$ 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 $\boldsymbol{\sigma}$ 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 $\boldsymbol{\pi}_0 = \boldsymbol{\sigma}$, i.e., Alice chooses her initial position according to $\boldsymbol{\sigma}$. It follows that $\boldsymbol{\pi}_t = \boldsymbol{\sigma}$. 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 $\sum_{j\in R}\pi_j \sum_{i\in L} P_{ji}$. Sinece Alice cannot disapparate, 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 14.7 (Example 14.6 continued)**.** For this chain we can set $L =$\{Remission\} and $R =$\{Relapse\}. Then the GBP says

$$\sigma_1 \times 0.2 = (1-\sigma_1)\times 0.5 \Rightarrow 7\sigma_1 = 5 \Rightarrow \sigma_1 = 5/7, \sigma_2 = 2/7 \Rightarrow \boldsymbol{\sigma} = (0.71429, 0.28571),$$

[1] A partition of a set S is a collection of disjoint sets whose union is equal to S.

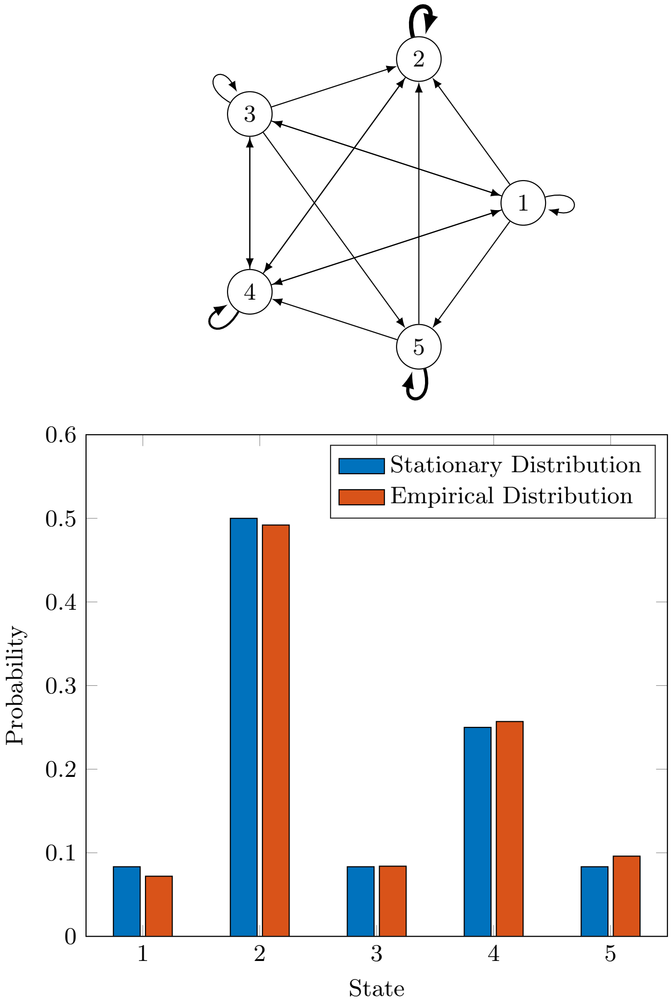


Figure 14.1: In the Markov chain (left), edges between different nodes have probability 1/5 and the probability of self loops is such that the outgoing probabilities sum to 1. The stationary distribution and an empirical (time-averaged) distribution are given on the right.

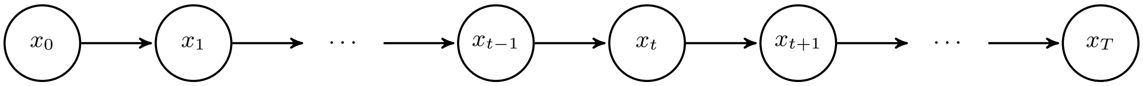
which is indeed the stationary distribution. △

14.4.2 Detailed Balance

A distribution $\boldsymbol{\pi}$ satisfies the **Detailed Balance Property (DBP)** if

$$\pi_i P_{ij} = \pi_j P_{ji}.$$

Time-reversibility and DBP. Consider the Markov chain



and assume that $\boldsymbol{\pi}_t = \boldsymbol{\sigma}$, where $\boldsymbol{\sigma}$ 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

$$p(x_t|x_{t+1}, \ldots, x_T) = p(x_t|x_{t+1})$$

So what are the transition probabilities P^- for the reversed MC? We have

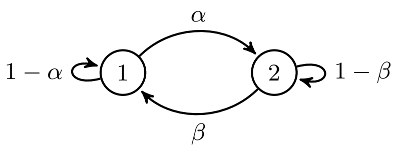
$$P^-_{ij} = p(x_t = j|x_{t+1} = i) = \frac{p(x_t = j, x_{t+1} = i)}{p(x_{t+1} = i)} = \frac{\pi_j P_{ji}}{\pi_i}.$$

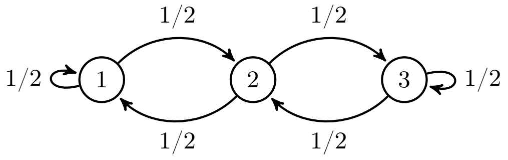
The MC is called **time-reversible** if $P^- = P$, which is equivalent to $\pi_i P_{ij} = \pi_j P_{ji}$ for all i, j, which are the detailed balance equations.

Based on the following theorem, it is easy to find the stationary distribution for Time reversible MCs, and for this reason, they are commonly used in Markov Chain Monte Carlo (MCMC) methods which we discuss later.

Theorem 14.8. *For a regular MC, if a vector $\boldsymbol{\pi}$ satisfies the detailed balance property, then π is the unique stationary distribution ($\boldsymbol{\pi} = \boldsymbol{\sigma}$) and the MC is time-reversible.*

Exercise 14.9. Using DBP, find the stationary distribution for the following MCs.





△

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,

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{Bayes} &= \mathbb{E}[\boldsymbol{\theta}|\mathcal{D}],\\ p(\boldsymbol{\theta} \in A|\mathcal{D}) &= \mathbb{E}[\mathbf{1}(\boldsymbol{\theta} \in A)|\mathcal{D}],\end{aligned}$$

where A is an event and $\mathbf{1}$(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 challenging.

In such cases, sampling from this distribution will be useful because sampling allows us to find expected values. For example, for a function h,

$$\mathbb{E}[h(x)] \simeq \sum_{i=1}^{N} h(x_i),$$

by the law of large numbers, where x_i are independent samples drawn from the distribution with respect to which the expected value is to be computed.

For example, recall that in Bayesian linear regression, a common likelihood is

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

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 $p(\sigma^2|\boldsymbol{y})$ and then $p(\boldsymbol{\theta}|\boldsymbol{y}, \sigma^2)$. With this sampling approach we can also perform prediction for a given input vector $\boldsymbol{x}_{n+1}$ of by producing samples from $p(y_{n+1}|\boldsymbol{\theta}, \sigma^2) \sim \mathcal{N}(\boldsymbol{x}_{n+1}^T\boldsymbol{\theta}, \sigma^2)$, 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 $i = 1, \ldots, N$,

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:

$$p(\theta \le c) = p(F^{-1}(u) \le c) = p(U \le F(c)) = F(c).$$

The algorithm is slightly modified if F has discontinuities or is not invertible. Specifically, we define $F^{-1}(u) = \min\{x : F(x) \ge u\}$.

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

$$\begin{aligned} &g(\boldsymbol{\theta}) > 0, && \text{if } q(\boldsymbol{\theta}) > 0, \\ &q(\boldsymbol{\theta}) \le M g(\boldsymbol{\theta}) && \text{for some known } M \text{ and for all } \boldsymbol{\theta}. \end{aligned}$$

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. $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}'$ if $U \le \frac{q(\boldsymbol{\theta}')}{Mg(\boldsymbol{\theta}')}$. (Accept $\boldsymbol{\theta}'$ as a new sample if $U \le \frac{q(\boldsymbol{\theta}')}{Mg(\boldsymbol{\theta}')}$; else reject the sample.)

We define the normalizing constants for the distribution,

$$Z_q = \int q(\boldsymbol{\theta}) d\boldsymbol{\theta}, \qquad Z_g = \int g(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

Note that the probability of a sample being accepted is

$$\begin{aligned} p(accepted) &= \int p(\boldsymbol{\theta}', accepted) d\boldsymbol{\theta}' = \int p(\boldsymbol{\theta}') p(accepted|\boldsymbol{\theta}') d\boldsymbol{\theta}' \\ &= \int \frac{g(\boldsymbol{\theta}')}{Z_g} \cdot \frac{q(\boldsymbol{\theta}')}{Mg(\boldsymbol{\theta}')} d\boldsymbol{\theta}' = \frac{Z_q}{MZ_g}. \end{aligned}$$

Let us now find the distribution for an accepted sample,

$$\begin{aligned} p(\boldsymbol{\theta}) &= p(\boldsymbol{\theta}'|accepted) \\ &= \frac{p(\boldsymbol{\theta}') p(accepted|\boldsymbol{\theta}')}{p(accepted)} \\ &= \frac{\frac{g(\boldsymbol{\theta}')}{Z_g} \cdot \frac{q(\boldsymbol{\theta}')}{Mg(\boldsymbol{\theta}')}}{\frac{Z_q}{MZ_g}} \\ &= \frac{q(\boldsymbol{\theta}')}{Z_q}, \end{aligned}$$

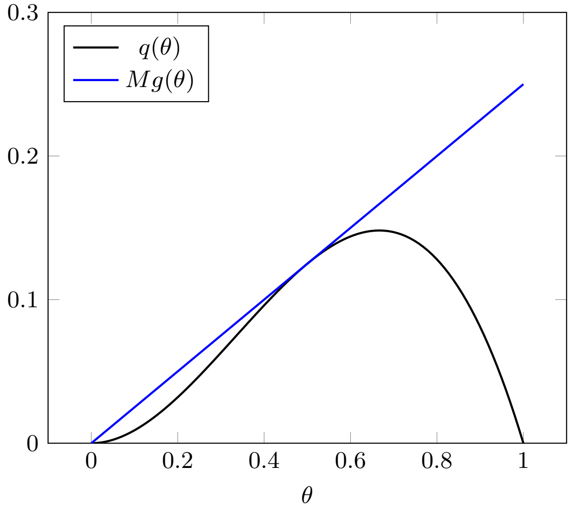
which is the desired distribution.

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.

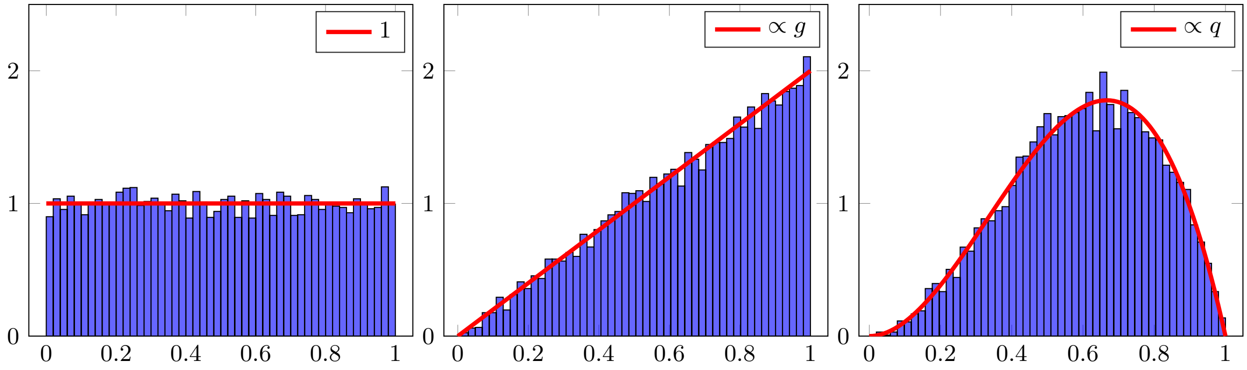
Example 15.1. Suppose we need to sample from Beta$(3, 2)$ so we let $q(\theta) = \theta^2(1-\theta)$. We would like to do this by sampling from $g(\theta) = \theta$, which we can do using inverse-CDF sampling. First, let us find the required value for M. Observe that

$$Mg(\theta) \geq q(\theta) \iff M\theta \geq \theta^2(1-\theta) \iff M \geq \theta(1-\theta).$$

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 Uni$(0, 1)$, obtaining $S_1 = \{x_1, \ldots, x_N\}$. 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 $S_2 = \{\theta'_1, \ldots, \theta'_N\}$, 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 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

$$\mathbb{E}_q[h(\boldsymbol{\theta})],$$

where $\mathbb{E}_q$ denotes expectation with respect to distribution q. Now if we 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 $\boldsymbol{\theta}$ such that $h(\boldsymbol{\theta})q(\boldsymbol{\theta})$ are large contribute to the expectation significantly. But $h(\boldsymbol{\theta})q(\boldsymbol{\theta})$ may be large in places where $q(\boldsymbol{\theta})$ is small. So unless we generate a lot of samples, we may not produce one for which $h(\boldsymbol{\theta})q(\boldsymbol{\theta})$ is large, and thus miss significant contributions to the expectation from such points.

Suppose we have a second (possibly unnormalized) distribution $g(\boldsymbol{\theta})$, 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

$$\mathbb{E}_q[h(\boldsymbol{\theta})] = \frac{\mathbb{E}_g[h(\boldsymbol{\theta})w(\boldsymbol{\theta})]}{\mathbb{E}_g[w(\boldsymbol{\theta})]} \simeq \frac{\frac{1}{N}\sum_{i=1}^N h(\boldsymbol{\theta}_i)w(\boldsymbol{\theta}_i)}{\frac{1}{N}\sum_{i=1}^N w(\boldsymbol{\theta}_i)}, \qquad \text{with } \boldsymbol{\theta}_i \sim g(\boldsymbol{\theta}),$$

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

$$\mathbb{E}_q[h(x)] = \int_0^1 (1-x)(2x)dx = \left(x^2 - 2x^3/3\right)_0^1 = 1/3.$$

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 $0.33\cdots$. Of course, for such a simple q we wouldn't resort to importance sampling.
△

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 $t = 1, 2, 3, \cdots$, do
   (a) Generate a proposal θ' based on the jump distribution $J(\theta'|\theta^{t-1})$.
   (b) Calculate
   $$r = \frac{p(\theta')}{p(\theta^{t-1})} = \frac{q(\theta')}{q(\theta^{t-1})},$$
   where the $q(\theta)$ is known.
   (c) Generate $u \sim \text{Uni}[0, 1]$.

(d) The next state of the MC, θ^t, is given by

$$\theta^t = \begin{cases} \theta', & u \le r \\ \theta^{t-1}, & u > r \end{cases} \tag{15.1}$$

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 $r = \frac{q(\theta')}{q(\theta^{t-1})}$. 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 $\theta_a \to \theta_b$ in the MC is denoted by $\Pr(\theta_a \to \theta_b)$, we have

$$\Pr(\theta_a \to \theta_b) = J(\theta_b|\theta_a) \min\left(1, \frac{p(\theta_b)}{p(\theta_a)}\right).$$

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

$$J(\theta'|\theta) > 0, \qquad \forall \theta, \theta' \in D. \tag{15.2}$$

2. *Symmetry.* For $\theta_a, \theta_b \in D$, the detailed balance property with distribution $\pi(\theta) = p(\theta)$ can be written as

$$p(\theta_a) \Pr(\theta_a \to \theta_b) = p(\theta_b) \Pr(\theta_b \to \theta_a).$$

   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.,

$$J(\theta'|\theta) = J(\theta|\theta'), \qquad \forall \theta, \theta' \in D. \tag{15.3}$$

If the jump distribution satisfies (15.2) and (15.3), then the MC is regular and $\boldsymbol{\sigma}(\theta) = p(\theta)$ 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 $x_i = 0, 0.1, 0.2, \ldots, 5$ as well as the line $y = 2x$.

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

$$\ln p(\theta, \sigma|\mathcal{D}) \propto -(2+N)\ln\sigma - \frac{1}{2\sigma^2}(\boldsymbol{y} - \theta\boldsymbol{x})'(\boldsymbol{y} - \theta\boldsymbol{x}).$$

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[1]:

$$\begin{aligned} J(\theta', \sigma'|\theta, \sigma) &= J(\theta'|\theta)J(\sigma'|\sigma), \\ J(\theta'|\theta) &\sim \mathcal{N}(\theta, 0.01), \\ J(\sigma'|\sigma) &\sim \mathcal{N}(\sigma, 0.01). \end{aligned}$$

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. △

Metropolis-Hastings algorithm. We can eliminate the symmetry property of the jump distribution if we modify r in the Metropolis algorithm as

$$r = \frac{p(\theta')/J(\theta'|\theta^{t-1})}{p(\theta^{t-1})/J(\theta^{t-1}|\theta')}.$$

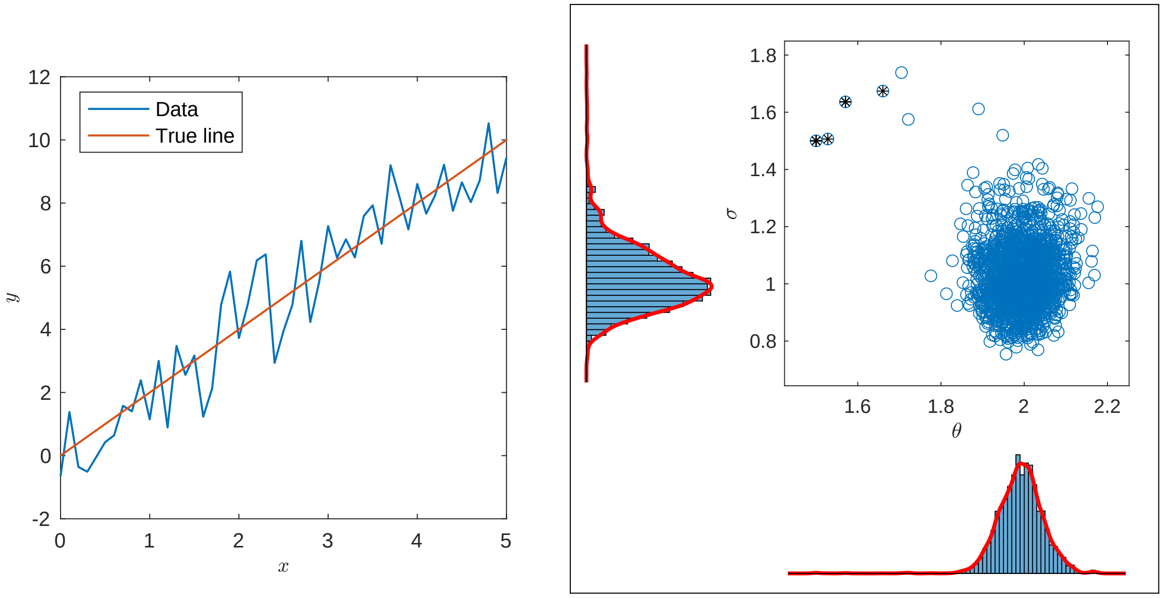
Exercise 15.4. Prove that with this definition for r, DBP holds even if J is not symmetric. △

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.

[1] Technically, we should not choose $J(\sigma'|\sigma)$ as we did because there is a possibility of producing $\sigma' < 0$. But given that the mass of probability for σ is far from 0, in this problem, this isn't a big issue since negative σ' is unlikely. A more sound solution is to use a truncated Gaussian, but that would not be a symmetric proposal, so we will have to use Metropolis-Hastings discussed next.



(a) Data as well as the true (noiseless) line $y = \theta x$, with $\theta = 2$.

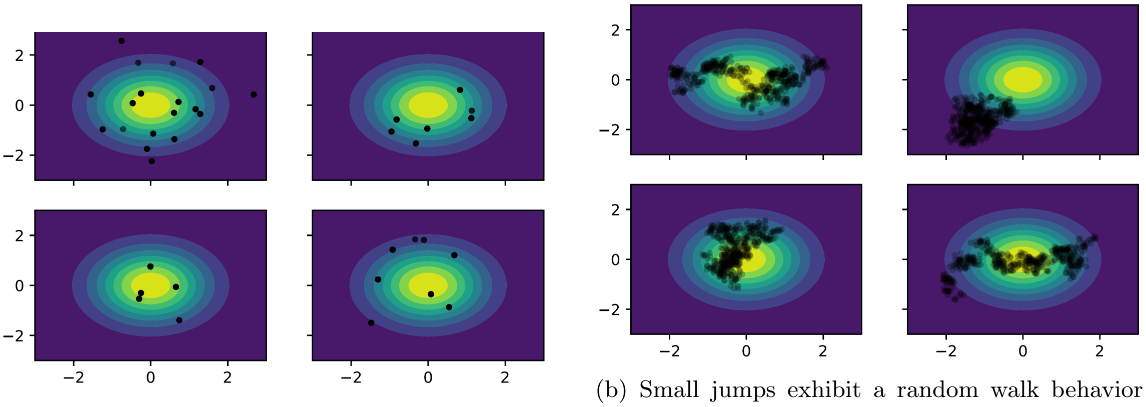
(b) Metropolis sampling for θ and σ. The first 10 samples are marked with $*$.

Figure 15.1: Metropolis sampling for 1-D Bayesian linear regression.

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



(a) Large jumps cause many proposals to be rejected, making the chain stay in its current state.

(b) Small jumps exhibit a random walk behavior which does not necessarily explore the space efficiently.

Figure 15.2: Metropolis sampling with poorly designed jump distributions (4 runs for each case).

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 $L = \mathbb{E}\Big[|\sum_{i=1}^{T} X_i|\Big]$ is difficult to find. But we can approximate the distance as

$$L^2 \simeq \mathbb{E}\left[(\sum_{i=1}^{T} X_i)^2\right] = T\varepsilon^2 \text{ (exercise)}.$$

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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_d)$, at each iteration, components are updated one-by-one as

$$\theta_j^t \sim p(\theta_j | \theta_1^t, \cdots, \theta_{(j-1)}^t, \theta_{(j+1)}^{(t-1)}, \cdots, \theta_d^{t-1}), \qquad \text{for } j = 1, \ldots, d.$$

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 $\boldsymbol{y} = (y_1, y_2)$ 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 $p(\boldsymbol{\theta}|\boldsymbol{y})$ 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 $\boldsymbol{\theta}|\boldsymbol{y}$. 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

$$\theta_1^t | \theta_2^{t-1} \sim \mathcal{N}\big(y_1 + \rho(\theta_2^{t-1} - y_2), (1-\rho^2) \big).$$

Similarly, the θ_2^t can be updated by

$$\theta_2^t | \theta_1^t \sim \mathcal{N}\big(y_2 + \rho(\theta_1^t - y_1), (1-\rho^2) \big).$$

So we produce a new sample using 1-D distributions. △

Stationary distribution. We prove that Gibbs sampling satisfies the DBP with distribution $p(\boldsymbol{\theta})$.

Suppose we are in state $\boldsymbol{\theta}$ and we update the jth 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 $\boldsymbol{\theta}'_{-j} = \boldsymbol{\theta}_{-j}$.

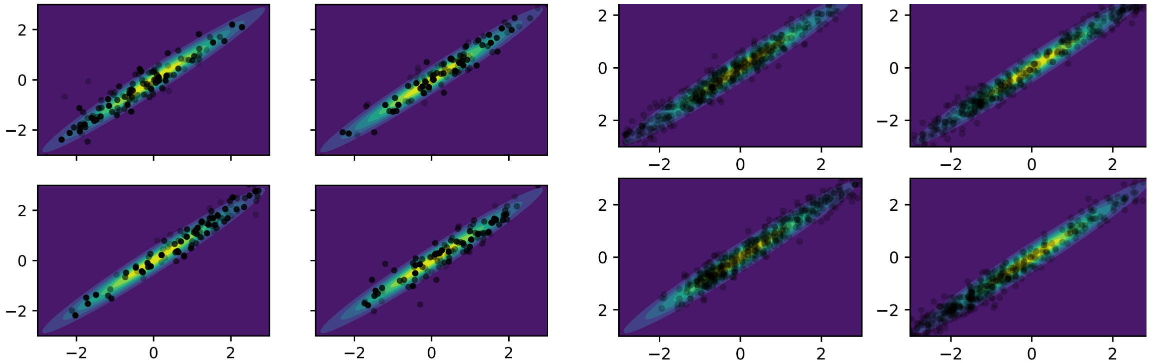
To prove DBP for this step, we need to prove $p(\boldsymbol{\theta}) \Pr(\boldsymbol{\theta} \to \boldsymbol{\theta}') = p(\boldsymbol{\theta}') \Pr(\boldsymbol{\theta}' \to \boldsymbol{\theta})$, which holds since

$$\begin{aligned} p(\boldsymbol{\theta}) \Pr(\boldsymbol{\theta} \to \boldsymbol{\theta}') &= p(\boldsymbol{\theta}) p(\theta_j' | \boldsymbol{\theta}_{-j}) = p(\boldsymbol{\theta}_{-j}) p(\theta_j | \boldsymbol{\theta}_{-j}) p(\theta_j' | \boldsymbol{\theta}_{-j}), \\ p(\boldsymbol{\theta}') \Pr(\boldsymbol{\theta}' \to \boldsymbol{\theta}) &= p(\boldsymbol{\theta}') p(\theta_j | \boldsymbol{\theta}'_{-j}) = p(\boldsymbol{\theta}_{-j}) p(\theta_j' | \boldsymbol{\theta}_{-j}) p(\theta_j | \boldsymbol{\theta}_{-j}). \end{aligned}$$

Since the DBP holds for each sub-iteration, it holds for each iteration.

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, dose not suffer from this.

Note that θ_j may be independent from some dimensions of $\boldsymbol{\theta}_{-j}$ given others. In particular, if $\boldsymbol{\theta}$ denotes the nodes in a graphical model, given its Markov blanket, θ_j is independent of other elements of $\boldsymbol{\theta}_{-j}$.



(a) Four runs of the Metropolis algorithm.

(b) Four runs of the Gibbs algorithm.

Figure 15.3: Metropolis and Gibbs sampling for highly correlated dimensions. Many proposals for Metropolis are rejected.

15.5 Hamiltonian Monte Carlo **

One problem with the Metropolis algorithm is that, in certain situations, the proposed $\boldsymbol{\theta}'$ by the jump distribution may be rejected too often because $p(\boldsymbol{\theta}')$ is much smaller than $p(\boldsymbol{\theta}^{t-1})$, in which case we will let $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1}$. While the stationary distribution is still $p(\boldsymbol{\theta})$, 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 $p(\boldsymbol{\theta})$ as

$$p(\boldsymbol{\theta}) \propto e^{-E(\boldsymbol{\theta})}$$

and suppose that we can also compute $\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta})$. Note that as $E(\boldsymbol{\theta})$ 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 $\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} - \epsilon \nabla E(\boldsymbol{\theta})$, 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 $\boldsymbol{\theta}$ as location and of $E(\boldsymbol{\theta})$ as potential energy. Note that lower potential has a higher probability (a river flows down the valley). Now let us also include momentum (speed) $\boldsymbol{\phi}$, which has the same number of dimensions as $\boldsymbol{\theta}$, in our formulation and define the total energy as

$$H(\boldsymbol{\theta}, \boldsymbol{\phi}) = E(\boldsymbol{\theta}) + K(\boldsymbol{\phi}),$$

where $K(\boldsymbol{\phi})$ is the Kinetic energy

$$K(\boldsymbol{\phi}) = \frac{1}{2}\boldsymbol{\phi}^T \boldsymbol{\phi}.$$

With this physical viewpoint, Hamilton's equations describing the motion of an object with position $\boldsymbol{\theta}$ and momentum $\boldsymbol{\phi}$ are

$$\begin{aligned}\dot{\boldsymbol{\theta}} &= \frac{\partial \boldsymbol{\theta}}{\partial t} = \boldsymbol{\phi}\\ \dot{\boldsymbol{\phi}} &= \frac{\partial \boldsymbol{\phi}}{\partial t} = -\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta})\end{aligned}$$

(A more familiar form of these equations are obtained by representing position with $\boldsymbol{x}$ and speed with $\boldsymbol{v}$. Then, $\dot{\boldsymbol{x}} = \boldsymbol{v}, \dot{\boldsymbol{v}} = -\nabla_{\boldsymbol{x}} E(\boldsymbol{x})$.) It can then be shown that H, the total energy, stays constant in time.

Back to Sampling. Instead of sampling from $p(\boldsymbol{\theta})$, let us define and sample from

$$p(\boldsymbol{\theta}, \boldsymbol{\phi}) \propto e^{-H(\boldsymbol{\theta},\boldsymbol{\phi})} = e^{-E(\boldsymbol{\theta})} e^{-K(\boldsymbol{\phi})},$$

where $K(\boldsymbol{\phi}) = \frac{1}{2}\boldsymbol{\phi}^T\boldsymbol{\phi}$. We will then discard the $\boldsymbol{\phi}$ component of the samples.

The Hamiltonian Monte Carlo Algorithm is as follows:

1. Randomly choose $\boldsymbol{\theta}^0$ from the domain and choose $\boldsymbol{\phi}^0$ arbitrarily.
2. For $t = 1, 2, ...,$ do
   (a) Pick a random momentum $\boldsymbol{\phi}'$ according to the distribution $p(\boldsymbol{\phi}) \propto e^{-K(\boldsymbol{\phi})}$.
   (b) Starting from $(\boldsymbol{\theta}^{t-1}, \boldsymbol{\phi}')$, simulate the dynamic system for a certain amount of time according to

$$\begin{aligned}\dot{\boldsymbol{\theta}} &= \boldsymbol{\phi},\\ \dot{\boldsymbol{\phi}} &= -\nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}).\end{aligned}$$

   The final values of $(\boldsymbol{\theta}, \boldsymbol{\phi})$ are the new sample, $(\boldsymbol{\theta}^t, \boldsymbol{\phi}^t)$.

It can be shown this process leads to a Markov chain whose stationary distribution is $p(\boldsymbol{\theta}, \boldsymbol{\phi})$. 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 $i = 1, 2, \ldots, L$, perform the following steps, called leapfrog updates:

$$\begin{aligned}\boldsymbol{\phi} &\leftarrow \boldsymbol{\phi} - \frac{1}{2}\epsilon \nabla E(\boldsymbol{\theta})\\ \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} + \epsilon \boldsymbol{\phi}\\ \boldsymbol{\phi} &\leftarrow \boldsymbol{\phi} - \frac{1}{2}\epsilon \nabla E(\boldsymbol{\theta})\end{aligned}$$

Let the final $(\boldsymbol{\theta}, \boldsymbol{\phi})$ be denoted by $(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$. 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

$$r = \frac{e^{-H(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)}}{e^{-H(\boldsymbol{\theta}^{t-1}, \boldsymbol{\phi}^{t-1})}}.$$

If $r \geq 1$, we let $(\boldsymbol{\theta}^t, \boldsymbol{\phi}^t) = (\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$. If $r \leq 1$, then we let $(\boldsymbol{\theta}^t, \boldsymbol{\phi}^t) = (\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$ with probability r and with probability $1 - r$, we let $(\boldsymbol{\theta}^t, \boldsymbol{\phi}^t) = (\boldsymbol{\theta}^{t-1}, \boldsymbol{\phi}^{t-1})$.

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

Appendix

16.1 Vector and matrix differentiation

Definition 16.1 (The three derivatives)**.** For a matrix A, scalar z, and two vectors $\boldsymbol{x}, \boldsymbol{y}$ (possibly one-dimensional), let

$$\frac{dA}{dz} = \begin{pmatrix} \frac{\partial A_{11}}{\partial z} & \cdots & \frac{\partial A_{1n}}{\partial z} \\ \vdots & \ddots & \vdots \\ \frac{\partial A_{m1}}{\partial z} & \cdots & \frac{\partial A_{mn}}{\partial z} \end{pmatrix}, \qquad \frac{dz}{dA} = \begin{pmatrix} \frac{\partial z}{\partial A_{11}} & \cdots & \frac{\partial z}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z}{\partial A_{m1}} & \cdots & \frac{\partial z}{\partial A_{mn}} \end{pmatrix}, \qquad \frac{d\boldsymbol{y}}{d\boldsymbol{x}} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_m} \end{pmatrix}$$

Lemma 16.2. *For a scalar a, vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{v}$, and constant matrices A and S,*

$$\begin{aligned} \frac{d\boldsymbol{y}}{d\boldsymbol{v}} &= \frac{d\boldsymbol{y}}{d\boldsymbol{x}}\frac{d\boldsymbol{x}}{d\boldsymbol{v}}, \\ \frac{d}{d\boldsymbol{v}}(a\boldsymbol{x}) &= a\frac{d\boldsymbol{x}}{d\boldsymbol{v}} + \boldsymbol{x}\frac{da}{d\boldsymbol{v}}, \\ \frac{d}{d\boldsymbol{v}}(\boldsymbol{y}^T A\boldsymbol{x}) &= \boldsymbol{y}^T A\frac{d\boldsymbol{x}}{d\boldsymbol{v}} + \boldsymbol{x}^T A^T\frac{d\boldsymbol{y}}{d\boldsymbol{v}}, \\ \frac{d}{d\boldsymbol{v}}(\boldsymbol{y}^T S\boldsymbol{y}) &= 2\boldsymbol{y}^T S\frac{d\boldsymbol{y}}{d\boldsymbol{v}}, \qquad (S \text{ is symmetric}) \\ \frac{d}{d\boldsymbol{v}}(A\boldsymbol{x}) &= A\frac{d\boldsymbol{x}}{d\boldsymbol{v}}. \end{aligned}$$

Lemma 16.3. *For matrix A and constant vector $\boldsymbol{x}$,*

$$\begin{aligned} \frac{d}{dA}(\boldsymbol{x}^T A\boldsymbol{x}) &= \boldsymbol{x}\boldsymbol{x}^T \\ \frac{d}{dA}\ln|A| &= A^{-T} \end{aligned}$$

Definition 16.4. Let $f : \mathbb{R}^m \to \mathbb{R}$. The gradient of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is defined as

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \left(\frac{df(\boldsymbol{x})}{d\boldsymbol{x}}\right)^T = \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_m} \end{pmatrix}$$

and the Hessian of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is defined as

$$\mathsf{H}_{\boldsymbol{x}}(f(\boldsymbol{x})) = \frac{d\nabla_{\boldsymbol{x}} f(\boldsymbol{x})}{d\boldsymbol{x}} = \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial f(\boldsymbol{x})}{\partial x_m \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(\boldsymbol{x})}{\partial x_1 \partial x_m} & \cdots & \frac{\partial f(\boldsymbol{x})}{\partial x_m \partial x_m} \end{pmatrix}$$

Chain rule. Consider $h : \mathbb{R}^m \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$, and $f(\boldsymbol{x}) = g(h(\boldsymbol{x}))$. From Lemma 16.2,

$$\begin{aligned} \nabla f(\boldsymbol{x}) &= g'(h(\boldsymbol{x}))\nabla h(\boldsymbol{x}), \\ \mathsf{H} f(\boldsymbol{x}) &= g'(h(\boldsymbol{x}))\mathsf{H} h(\boldsymbol{x}) + g''(h(\boldsymbol{x}))\nabla h(\boldsymbol{x})\nabla^T h(\boldsymbol{x}) \end{aligned}$$

since

$$\begin{aligned} \mathsf{H} f(\boldsymbol{x}) &= \frac{d\nabla f}{d\boldsymbol{x}} \\ &= \frac{d(g'(h(\boldsymbol{x}))\nabla h(\boldsymbol{x}))}{d\boldsymbol{x}} \\ &= g'(h(\boldsymbol{x}))\frac{d\nabla h(\boldsymbol{x})}{d\boldsymbol{x}} + \nabla h(\boldsymbol{x})\frac{d(g'(h(\boldsymbol{x})))}{d\boldsymbol{x}} \\ &= g'(h(\boldsymbol{x}))\mathsf{H} h(\boldsymbol{x}) + \nabla h(\boldsymbol{x})\nabla^T h(\boldsymbol{x}) g''(h(\boldsymbol{x})) \end{aligned}$$

Example 16.5. Let us find the derivatives of $f(\boldsymbol{x}) = \log \sum_{i=1}^m e^{x_i}$. Let $\boldsymbol{z} = (\exp(x_i))_{i=1}^m$ so that $f(\boldsymbol{x}) = \log \mathbf{1}^T \boldsymbol{z}$.

$$\begin{aligned} \nabla f(\boldsymbol{x}) &= \frac{\boldsymbol{z}}{\mathbf{1}^T \boldsymbol{z}}, \\ \mathsf{H} f(\boldsymbol{x}) &= \frac{\operatorname{diag}(\boldsymbol{z})}{\mathbf{1}^T \boldsymbol{z}} - \frac{\boldsymbol{z}\boldsymbol{z}^T}{(\mathbf{1}^T \boldsymbol{z})^2}. \end{aligned}$$

△

Chain rule. Let $\boldsymbol{h} = (h_1, \ldots, h_n) : \mathbb{R}^m \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}$, and $f(\boldsymbol{x}) = g(\boldsymbol{h}(\boldsymbol{x}))$. Then

$$\begin{aligned} \frac{\partial f}{\partial x_i} &= \sum_{j=1}^n \frac{\partial g}{\partial h_j}\frac{\partial h_j}{\partial x_i} = \frac{dg}{d\boldsymbol{h}} \cdot \frac{d\boldsymbol{h}}{dx_i} = \nabla^T g \cdot \frac{d\boldsymbol{h}}{dx_i}, \\ \frac{df}{d\boldsymbol{x}} &= \frac{dg}{d\boldsymbol{h}}\frac{d\boldsymbol{h}}{d\boldsymbol{x}} = \nabla^T g \frac{d\boldsymbol{h}}{d\boldsymbol{x}}, \qquad \nabla_{\boldsymbol{x}} f = \left(\frac{df}{d\boldsymbol{x}}\right)^T = \left(\frac{d\boldsymbol{h}}{d\boldsymbol{x}}\right)^T \nabla g \end{aligned}$$

16.2 Properties of Expectation, Correlation, and Covariance for Vectors

Elementary properties of expectation, correlation, and covariance for vectors follow immediately from similar properties for ordinary scalar random variables. These properties include the following (here A and C are nonrandom matrices and b and d are nonrandom vectors).

1. $E[AX + b] = AE[X] + b$
2. $\text{Cov}(X, Y) = E[X(Y - E[Y])^T] = E[(X - E[X])Y^T] = E[XY^T] - (E[X])(E[Y])^T$
3. $E[(AX)(CY)^T] = AE[XY^T]C^T$
4. $\text{Cov}(AX + b, CY + d) = A\text{Cov}(X, Y)C^T$
5. $\text{Cov}(AX + b) = A\text{Cov}(X)A^T$
6. $\text{Cov}(W + X, Y + Z) = \text{Cov}(W, Y) + \text{Cov}(W, Z) + \text{Cov}(X, Y) + \text{Cov}(X, Z)$.