Preface

Use the template *preface.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your preface in the Springer layout.

A preface is a book's preliminary statement, usually written by the *author or editor* of a work, which states its origin, scope, purpose, plan, and intended audience, and which sometimes includes afterthoughts and acknowledgments of assistance.

When written by a person other than the author, it is called a foreword. The preface or foreword is distinct from the introduction, which deals with the subject of the work.

Customarily *acknowledgments* are included as last part of the preface.

Place(s), *Firstname Surname*
month year *Firstname Surname*

Contents

Contents iii

Notation ix

1 Introduction 1
- 1.1 Types of machine learning 1
- 1.2 Supervised learning 1
  - 1.2.1 Classification 1
  - 1.2.2 Regression 2
- 1.3 Unsupervised learning 2
  - 1.3.1 Discovering clusters 2
  - 1.3.2 Discovering latent factors 2
  - 1.3.3 Discovering graph structure . . . 2
  - 1.3.4 Matrix completion 2
- 1.4 Three elements of a machine learning model 2
  - 1.4.1 Representation 2
  - 1.4.2 Evaluation 3
  - 1.4.3 Optimization 4
- 1.5 Some basic concepts 4
  - 1.5.1 Parametric vs non-parametric models 4
  - 1.5.2 A simple non-parametric classifier: K-nearest neighbours 4
  - 1.5.3 Overfitting 5
  - 1.5.4 Cross validation 5
  - 1.5.5 Model selection 5

2 Probability and Statistics 7
- 2.1 Frequentists vs. Bayesians 7
- 2.2 Basic concepts of probability theory 7
  - 2.2.1 Discrete random variables 7
  - 2.2.2 Fundamental rules 7
  - 2.2.3 Mutivariate random variables . . 8
  - 2.2.4 Bayes rule 9
  - 2.2.5 Independence and conditional independence 10
  - 2.2.6 Quantiles 10
  - 2.2.7 Mean and variance 11
- 2.3 Some common discrete distributions . . . 11
  - 2.3.1 The Bernoulli and binomial distributions 12
  - 2.3.2 The multinoulli and multinomial distributions 12
  - 2.3.3 The Poisson distribution 12
  - 2.3.4 The empirical distribution 13
- 2.4 Some common continuous distributions . 13
  - 2.4.1 Gaussian (normal) distribution . 13
  - 2.4.2 Student's t-distribution 14
  - 2.4.3 The Laplace distribution 14
  - 2.4.4 The gamma distribution 15
  - 2.4.5 The beta distribution 16
  - 2.4.6 Pareto distribution 18
- 2.5 Joint probability distributions 19
  - 2.5.1 Covariance and correlation 19
  - 2.5.2 Multivariate Gaussian distribution 20
  - 2.5.3 Multivariate Student's t-distribution 20
  - 2.5.4 Dirichlet distribution 22
- 2.6 Transformations of random variables . . . 22
  - 2.6.1 Linear transformations 22
  - 2.6.2 General transformations 24
  - 2.6.3 Central limit theorem 25
- 2.7 Monte Carlo approximation 25
- 2.8 Information theory 26
  - 2.8.1 Entropy 26
  - 2.8.2 KL divergence 26
  - 2.8.3 Mutual information 27

3 Latent linear models 29
- 3.1 Factor analysis 29
  - 3.1.1 FA is a low rank parameterization of an MVN . . 29
  - 3.1.2 Inference of the latent factors . . 30
  - 3.1.3 Unidentifiability 30
  - 3.1.4 Mixtures of factor analysers . . . 31
  - 3.1.5 EM for factor analysis models . 31
  - 3.1.6 Fitting FA models with missing data 32
- 3.2 Principal components analysis (PCA) . . 32
  - 3.2.1 Classical PCA 32
  - 3.2.2 Singular value decomposition (SVD) 34
  - 3.2.3 Probabilistic PCA 35
  - 3.2.4 EM algorithm for PCA 36
- 3.3 Choosing the number of latent dimensions 37
  - 3.3.1 Model selection for FA/PPCA . 37
  - 3.3.2 Model selection for PCA 37
- 3.4 PCA for categorical data 37
- 3.5 PCA for paired and multi-view data 38
  - 3.5.1 Supervised PCA (latent factor regression) 38
  - 3.5.2 Discriminative supervised PCA 38

3.5.3 Canonical correlation analysis . 38
3.6 Independent Component Analysis (ICA) 38
3.6.1 Maximum likelihood estimation 38
3.6.2 The FastICA algorithm 39
3.6.3 Using EM 39
3.6.4 Other estimation principles * . . 39

4 Matrix Decomposition 41
4.1 LU Decompistion 41
4.2 QR Decomposition 41
4.3 Eigen value Decomposition 41
4.4 Singular Value Decomposition 41
4.4.1 definition 41
4.4.2 proof 41

A Optimization methods 43
A.1 Convexity 43
A.2 Gradient descent 43
A.2.1 Stochastic gradient descent . . . 43
A.2.2 Batch gradient descent 43
A.2.3 Line search 43
A.2.4 Momentum term 44
A.3 Lagrange duality 44
A.3.1 Primal form 44
A.3.2 Dual form 44
A.4 Newton's method 44
A.5 Quasi-Newton method 45
A.5.1 DFP 45
A.5.2 BFGS 45
A.5.3 Broyden 46

Glossary 47

List of Contributors

Firstname Surname
ABC Institute, 123 Prime Street, Daisy Town, NA 01234, USA, e-mail: smith@smith.edu

Firstname Surname
XYZ Institute, Technical University, Albert-Schweitzer-Str. 34, 1000 Berlin, Germany, e-mail: meier@tu.edu

Acronyms

Use the template *acronym.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your list(s) of abbreviations or symbols in the Springer layout.

Lists of abbreviations, symbols and the like are easily formatted with the help of the Springer-enhanced `description` environment.

ABC	Spelled-out abbreviation and definition
BABI	Spelled-out abbreviation and definition
CABR	Spelled-out abbreviation and definition

Notation

Introduction

It is very difficult to come up with a single, consistent notation to cover the wide variety of data, models and algorithms that we discuss. Furthermore, conventions differ between machine learning and statistics, and between different books and papers. Nevertheless, we have tried to be as consistent as possible. Below we summarize most of the notation used in this book, although individual sections may introduce new notation. Note also that the same symbol may have different meanings depending on the context, although we try to avoid this where possible.

General math notation

Symbol	Meaning
$\lfloor x \rfloor$	Floor of x, i.e., round down to nearest integer
$\lceil x \rceil$	Ceiling of x, i.e., round down to nearest integer
$\boldsymbol{x} \otimes \boldsymbol{y}$	Convolution of $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{x} \odot \boldsymbol{y}$	Hadamard (elementwise) product of $\boldsymbol{x}$ and $\boldsymbol{y}$
$a \wedge b$	logical AND
$a \vee b$	logical OR
$\neg a$	logical NOT
$\mathbb{I}(x)$	Indicator function, $\mathbb{I}(x) = 1$ if x is true, else $\mathbb{I}(x) = 0$
∞	Infinity
$\rightarrow$	Tends towards, e.g., $n \rightarrow \infty$
$\propto$	Proportional to, so $y = ax$ can be written as $y \propto x$
$\lvert x \rvert$	Absolute value
$\lvert \mathcal{S} \rvert$	Size (cardinality) of a set
$n!$	Factorial function
∇	Vector of first derivatives
∇^2	Hessian matrix of second derivatives
$\triangleq$	Defined as
$O(\cdot)$	Big-O: roughly means order of magnitude
$\mathbb{R}$	The real numbers
$1:n$	Range (Matlab convention): $1:n = 1, 2, ..., n$
$\approx$	Approximately equal to
$\arg\max_x f(x)$	Argmax: the value x that maximizes f
$B(a,b)$	Beta function, $B(a,b) = \dfrac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$
$B(\boldsymbol{\alpha})$	Multivariate beta function, $\dfrac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$
$\binom{n}{k}$	n choose k , equal to $n!/(k!(nk)!)$
$\delta(x)$	Dirac delta function,$\delta(x) = \infty$ if $x = 0$, else $\delta(x) = 0$
$\exp(x)$	Exponential function e^x
$\Gamma(x)$	Gamma function, $\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} \mathrm{d}u$
$\Psi(x)$	Digamma function,$Psi(x) = \dfrac{d}{dx} \log \Gamma(x)$

$\mathcal{X}$	A set from which values are drawn (e.g.,$\mathcal{X} = \mathbb{R}^D$)

Linear algebra notation

We use boldface lower-case to denote vectors, such as $\boldsymbol{x}$, and boldface upper-case to denote matrices, such as $\boldsymbol{X}$. We denote entries in a matrix by non-bold upper case letters, such as X_{ij}.

Vectors are assumed to be column vectors, unless noted otherwise. We use $(x_1, \cdots, x_D)$ to denote a column vector created by stacking D scalars. If we write $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$, where the left hand side is a matrix, we mean to stack the $\boldsymbol{x}_i$ along the columns, creating a matrix.

Symbol	Meaning
$\boldsymbol{X} \succ 0$	$\boldsymbol{X}$ is a positive definite matrix
$tr(\boldsymbol{X})$	Trace of a matrix
$det(\boldsymbol{X})$	Determinant of matrix $\boldsymbol{X}$
$\|\boldsymbol{X}\|$	Determinant of matrix $\boldsymbol{X}$
$\boldsymbol{X}^{-1}$	Inverse of a matrix
$\boldsymbol{X}^{\dagger}$	Pseudo-inverse of a matrix
$\boldsymbol{X}^T$	Transpose of a matrix
$\boldsymbol{x}^T$	Transpose of a vector
$\text{diag}(x)$	Diagonal matrix made from vector $\boldsymbol{x}$
$\text{diag}(X)$	Diagonal vector extracted from matrix $\boldsymbol{X}$
$\boldsymbol{I}$ or $\boldsymbol{I}_d$	Identity matrix of size $d \times d$ (ones on diagonal, zeros of)
$\mathbf{1}$ or $\mathbf{1}_d$	Vector of ones (of length d)
$\mathbf{0}$ or $\mathbf{0}_d$	Vector of zeros (of length d)
$\|\|\boldsymbol{x}\|\| = \|\|\boldsymbol{x}\|\|_2$	Euclidean or ℓ_2 norm $\sqrt{\sum_{j=1}^{d} x_j^2}$
$\|\|\boldsymbol{x}\|\|_1$	ℓ_1 norm $\sum_{j=1}^{d} \|x_j\|$
$\boldsymbol{X}_{:,j}$	j'th column of matrix
$\boldsymbol{X}_{i,:}$	transpose of i'th row of matrix (a column vector)
$\boldsymbol{X}_{i,j}$	Element (i, j) of matrix $\boldsymbol{X}$
$\boldsymbol{x} \otimes \boldsymbol{y}$	Tensor product of $\boldsymbol{x}$ and $\boldsymbol{y}$

Probability notation

We denote random and fixed scalars by lower case, random and fixed vectors by bold lower case, and random and fixed matrices by bold upper case. Occasionally we use non-bold upper case to denote scalar random variables. Also, we use $p()$ for both discrete and continuous random variables

Symbol	Meaning
X, Y	Random variable
$P()$	Probability of a random event
$F()$	Cumulative distribution function(CDF), also called distribution function
$p(x)$	Probability mass function(PMF)
$f(x)$	probability density function(PDF)
$F(x,y)$	Joint CDF
$p(x,y)$	Joint PMF
$f(x,y)$	Joint PDF

$p(X\|Y)$	Conditional PMF, also called conditional probability
$f_{X\|Y}(x\|y)$	Conditional PDF
$X \perp Y$	X is independent of Y
$X \not\perp Y$	X is not independent of Y
$X \perp Y\|Z$	X is conditionally independent of Y given Z
$X \not\perp Y\|Z$	X is not conditionally independent of Y given Z
$X \sim p$	X is distributed according to distribution p
$\boldsymbol{\alpha}$	Parameters of a Beta or Dirichlet distribution
$\text{cov}[X]$	Covariance of X
$\mathbb{E}[X]$	Expected value of X
$\mathbb{E}_q[X]$	Expected value of X wrt distribution q
$\mathbb{H}(X)$ or $\mathbb{H}(p)$	Entropy of distribution $p(X)$
$\mathbb{I}(X;Y)$	Mutual information between X and Y
$\mathbb{KL}(p\|\|q)$	KL divergence from distribution p to q
$\ell(\boldsymbol{\theta})$	Log-likelihood function
$L(\boldsymbol{\theta}, a)$	Loss function for taking action a when true state of nature is $\boldsymbol{\theta}$
λ	Precision (inverse variance) $\lambda = 1/\sigma^2$
$\boldsymbol{\Lambda}$	Precision matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$
$\text{mode}[\boldsymbol{X}]$	Most probable value of $\boldsymbol{X}$
μ	Mean of a scalar distribution
$\boldsymbol{\mu}$	Mean of a multivariate distribution
$\boldsymbol{\Phi}$	cdf of standard normal
ϕ	pdf of standard normal
$\boldsymbol{\pi}$	multinomial parameter vector, Stationary distribution of Markov chain
ρ	Correlation coefficient
$\text{sigm}(x)$	Sigmoid (logistic) function, $\frac{1}{1+e^{-x}}$
σ^2	Variance
$\boldsymbol{\Sigma}$	Covariance matrix
$\text{var}[x]$	Variance of x
ν	Degrees of freedom parameter
Z	Normalization constant of a probability distribution

Machine learning/statistics notation

In general, we use upper case letters to denote constants, such as C, K, M, N, T, etc. We use lower case letters as dummy indexes of the appropriate range, such as $c = 1 : C$ to index classes, $i = 1 : M$ to index data cases, $j = 1 : N$ to index input features, $k = 1 : K$ to index states or clusters, $t = 1 : T$ to index time, etc.

We use x to represent an observed data vector. In a supervised problem, we use y or $\boldsymbol{y}$ to represent the desired output label. We use $\boldsymbol{z}$ to represent a hidden variable. Sometimes we also use q to represent a hidden discrete variable.

Symbol	Meaning
C	Number of classes
D	Dimensionality of data vector (number of features)
N	Number of data cases
N_c	Number of examples of class c,$N_c = \sum_{i=1}^{N} \mathbb{I}(y_i = c)$
R	Number of outputs (response variables)
$\mathcal{D}$	Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \| i = 1 : N\}$
$\mathcal{D}_{test}$	Test data
$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space

K	Number of states or dimensions of a variable (often latent)
$k(x,y)$	Kernel function
$\boldsymbol{K}$	Kernel matrix
$\mathcal{H}$	Hypothesis space
L	Loss function
$J(\boldsymbol{\theta})$	Cost function
$f(\boldsymbol{x})$	Decision function
$P(y\|\boldsymbol{x})$	TODO
λ	Strength of ℓ_2 or ℓ_1 *regularizer*
$\phi(x)$	Basis function expansion of feature vector $\boldsymbol{x}$
$\boldsymbol{\Phi}$	Basis function expansion of design matrix $\boldsymbol{X}$
$q()$	Approximate or proposal distribution
$Q(\boldsymbol{\theta},\boldsymbol{\theta}_{old})$	Auxiliary function in EM
T	Length of a sequence
$T(\mathcal{D})$	Test statistic for data
$\boldsymbol{T}$	Transition matrix of Markov chain
$\boldsymbol{\theta}$	Parameter vector
$\boldsymbol{\theta}^{(s)}$	s'th sample of parameter vector
$\hat{\boldsymbol{\theta}}$	Estimate (usually MLE or MAP) of $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}_{MLE}$	Maximum likelihood estimate of $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}_{MAP}$	MAP estimate of $\boldsymbol{\theta}$
$\bar{\boldsymbol{\theta}}$	Estimate (usually posterior mean) of $\boldsymbol{\theta}$
$\boldsymbol{w}$	Vector of regression weights (called $\boldsymbol{\beta}$ in statistics)
b	intercept (called $\boldsymbol{\varepsilon}$ in statistics)
$\boldsymbol{W}$	Matrix of regression weights
x_{ij}	Component (i.e., feature) j of data case i ,for $i = 1:N, j = 1:D$
$\boldsymbol{x}_i$	Training case, $i = 1:N$
$\boldsymbol{X}$	Design matrix of size $N \times D$
$\bar{\boldsymbol{x}}$	Empirical mean $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$
$\tilde{\boldsymbol{x}}$	Future test case
$\boldsymbol{x}_*$	Feature test case
$\boldsymbol{y}$	Vector of all training labels $\boldsymbol{y} = (y_1,...,y_N)$
z_{ij}	Latent component j for case i

Chapter 1
Introduction

1.1 Types of machine learning

$$\begin{cases} \text{Supervised Learning} \begin{cases} \text{Classification} \\ \text{Regression} \end{cases} \\ \text{Unsupervised Learning} \begin{cases} \text{Discovering clusters} \\ \text{Discovering latent factors} \\ \text{Discovering graph structure} \\ \text{Matrix completion} \end{cases} \\ \text{Reinforcement Learning} \end{cases}$$

In the **predictive** or **supervised learning** approach,the goal is to learn a **mapping** from **inputs x** to **outputs** y,given a labeled set of input-output paris $D = \{(\mathbf{x_i}, y_i)\}_{i=1}N$.Here D is called the **training set**,and N is the number of training examples. In the simplest setting,each training input $\mathbf{x_i}$ is a D-dimensional vector of numbers,representing,say,the height and weight of a person,which are called **features,attributes,or covariates**.

1.2 Supervised learning

Similarly the form of the output or **response variable** can in principle be anything,but most methods assume that y_i is **categorical or nominal** variable from some finite set,$y_i \in \{1,...,C\}$.When y_i is categorical,the problem is known as **classification or pattern recognition**,and when real-valued,known as **regression**.

1.2.1 Classification

Here the goal is to learn a mapping from inputs x to outputs y,where $y \in \{1,...,C\}$,with C being the number of classes.If $C = 2$,this is called **binary classification**;if $C > 2$,this is called **multiclass classification**.if the class labels are not mutually exclusive,we call it **multi-label classification**,but this is best viewed as predicting multiple related binary class labels(a so-called **multiple output model**). One way to formalize the problem is as **function approximation**:assume $y = f(\mathbf{x})$ for some unknown function f,and the goal of learning is to estimate the function f given a labeled training set,and then to make predictions(estimate) using $\hat{y} = \hat{f}(\mathbf{x})$.Our main goal is to make predictions on novel inputs,meaning ones that we have not seen before(**generalization**.

1.2.1.1 Probabilistic predictions

Given a probabilistic output,we can always compute out "best guess" as to the "true label" using

$$\hat{y} = \hat{f}(\mathbf{x}) = \arg\max_{c=1}^{C} p(y = c|\mathbf{x}, D) \tag{1.1}$$

This corresponds to a **MAP estimate**(MAP stands for**maximum a posteriori**).

1.2.1.2 Applications

1.2.2 Regression

1.3 Unsupervised learning

Descriptive or unsupervised learning approach is sometimes called **knowledge discovery**.We will formalize out task as one of **density estimation**,that is we want to build models of the form $p(\mathbf{x_i}|\theta)$,instead of $p(y_i|\mathbf{x_i},\theta)$.

1.3.1 Discovering clusters

Let $z_i \in \{1,...,K\}$ represent the cluster to which data point i is assigned.(z_i is an exmaple of **hidden or latent** variable).

1.3.2 Discovering latent factors

Although the data may appear high dimensional,there may only be a small number of degrees of variability,corresponding to **latent factors**.The most common approach to dimensionality reduction is called **principal components analysis or PCA**.

1.3.3 Discovering graph structure

1.3.4 Matrix completion

1.3.4.1 Image inpainting

1.3.4.2 Collaborative filtering

1.3.4.3 Market basket analysis

1.4 Three elements of a machine learning model

Model = Representation + Evaluation + Optimization[1]

1.4.1 Representation

In supervised learning, a model must be represented as a conditional probability distribution $P(y|\boldsymbol{x})$(usually we call it classifier) or a decision function $f(x)$. The set of classifiers(or decision functions) is called the hypothesis space of the model. Choosing a representation for a model is tantamount to choosing the hypothesis space that it can possibly learn.

[1] Domingos, P. A few useful things to know about machine learning. Commun. ACM. 55(10):7887 (2012).

1.4.2 Evaluation

In the hypothesis space, an evaluation function (also called objective function or risk function) is needed to distinguish good classifiers(or decision functions) from bad ones.

1.4.2.1 Loss function and risk function

Definition 1.1. In order to measure how well a function fits the training data, a **loss function** $L : Y \times Y \to R \geq 0$ is defined. For training example (x_i, y_i), the loss of predicting the value $\widehat{y}$ is $L(y_i, \widehat{y})$.

The following is some common loss functions:

1. 0-1 loss function
   $L(Y, f(X)) = \mathbb{I}(Y, f(X)) = \begin{cases} 1, & Y = f(X) \\ 0, & Y \neq f(X) \end{cases}$
2. Quadratic(squared) loss function $L(Y, f(X)) = \frac{1}{2}(Y - f(X))^2$
3. Absolute loss function $L(Y, f(X)) = |Y - f(X)|$
4. Exponential loss function $L(Y, f(X) = exp(-\hat{y}_i f(\mathbf{x_i}))$
5. Logarithmic loss function
   $L(Y, P(Y|X)) = -\log P(Y|X)$

Name	Loss	Derivative	f^*	Algorithm
Squared error	$\frac{1}{2}(y_i - f(\mathbf{x_i}))^2$	$y_i - f(\mathbf{x_i})$	$\mathbb{E}[y\|\mathbf{x_i}]$	L2Boosting
Absolute error	$\|y_i - f(\mathbf{x_i})\|$	$sgn(y_i - f(\mathbf{x_i}))$	$median(y\|\mathbf{x_i})$	Gradient boosting
Exponential loss	$exp(-\hat{y}_i f(\mathbf{x_i}))$	$-\hat{y}_i exp(-\hat{y}_i f(\mathbf{x_i}))$	$\frac{1}{2} log \frac{\pi_i}{1-\pi_i}$	AdaBoost
Logloss	$log(1 + e^{-\hat{y}_i f_i})$	$y_i - \pi_i$	$\frac{1}{2} log \frac{\pi_i}{1-\pi_i}$	LogitBoost

Definition 1.2. The risk of function f is defined as the expected loss of f:

$$R_{\exp}(f) = E\left[L(Y, f(X))\right] = \int L(y, f(x)) P(x, y) \mathrm{d}x\mathrm{d}y \tag{1.2}$$

which is also called expected loss or **risk function**.

Definition 1.3. The risk function $R_{\exp}(f)$ can be estimated from the training data as

$$R_{\text{emp}}(f) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) \tag{1.3}$$

which is also called empirical loss or **empirical risk**.

You can define your own loss function, but if you're a novice, you're probably better off using one from the literature. There are conditions that loss functions should meet[2]:

1. They should approximate the actual loss you're trying to minimize. As was said in the other answer, the standard loss functions for classification is zero-one-loss (misclassification rate) and the ones used for training classifiers are approximations of that loss.
2. The loss function should work with your intended optimization algorithm. That's why zero-one-loss is not used directly: it doesn't work with gradient-based optimization methods since it doesn't have a well-defined gradient (or even a subgradient, like the hinge loss for SVMs has).
   The main algorithm that optimizes the zero-one-loss directly is the old perceptron algorithm(chapter §**??**).

[2] http://t.cn/zTrDxLO

1.4.2.2 ERM and SRM

Definition 1.4. ERM(Empirical risk minimization)

$$\min_{f\in\mathcal{F}} R_{\text{emp}}(f) = \min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) \tag{1.4}$$

Definition 1.5. Structural risk

$$R_{\text{smp}}(f) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \tag{1.5}$$

Definition 1.6. SRM(Structural risk minimization)

$$\min_{f\in\mathcal{F}} R_{\text{srm}}(f) = \min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \tag{1.6}$$

1.4.3 Optimization

Finally, we need a **training algorithm**(also called **learning algorithm**) to search among the classifiers in the the hypothesis space for the highest-scoring one. The choice of optimization technique is key to the **efficiency** of the model.

1.5 Some basic concepts

1.5.1 Parametric vs non-parametric models

1.5.2 A simple non-parametric classifier: K-nearest neighbours

1.5.2.1 Representation

$$y = f(\boldsymbol{x}) = \arg\min_{c} \sum_{\boldsymbol{x}_i\in N_k(\boldsymbol{x})} \mathbb{I}(y_i = c) \tag{1.7}$$

where $N_k(\boldsymbol{x})$ is the set of k points that are closest to point $\boldsymbol{x}$.

Usually use **k-d tree** to accelerate the process of finding k nearest points.

1.5.2.2 Evaluation

No training is needed.

1.5.2.3 Optimization

No training is needed.

1.5.3 Overfitting

1.5.4 Cross validation

Definition 1.7. Cross validation, sometimes called *rotation estimation*, is a *model validation* technique for assessing how the results of a statistical analysis will generalize to an independent data set[3].

Common types of cross-validation:

1. K-fold cross-validation. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k 1 subsamples are used as training data.
2. 2-fold cross-validation. Also, called simple cross-validation or holdout method. This is the simplest variation of k-fold cross-validation, k=2.
3. Leave-one-out cross-validation(*LOOCV*). k=M, the number of original samples.

1.5.5 Model selection

When we have a variety of models of different complexity (e.g., linear or logistic regression models with different degree polynomials, or KNN classifiers with different values ofK), how should we pick the right one? A natural approach is to compute the **misclassification rate** on the training set for each method.

[3] http://en.wikipedia.org/wiki/Cross-validation_(statistics)

Chapter 2
Probability and Statistics

2.1 Frequentists vs. Bayesians

There are two different interpretations of probability. One is called the **frequentist** interpretation. In this view, probabilities represent long run frequencies of events. For example, the above statement means that, if we flip the coin many times, we expect it to land heads about half the time.

The other interpretation is called the **Bayesian** interpretation of probability. In this view, probability is used to quantify our **uncertainty** about something; hence it is fundamentally related to information rather than repeated trials (Jaynes 2003). In the Bayesian view, the above statement means we believe the coin is equally likely to land heads or tails on the next toss

One big advantage of the Bayesian interpretation is that it can be used to model our uncertainty about events that do not have long term frequencies. For example, we might want to compute the probability that the polar ice cap will melt by 2020 CE. This event will happen zero or one times, but cannot happen repeatedly. Nevertheless,we thought to be able to quantify our uncertainty about this event. To give another machine learning oriented example, we might have observed a blip on our radar screen, and want to compute the probability distribution over the location of the corresponding target (be it a bird, plane, or missile). In all these cases, the idea of repeated trials does not make sense, but the Bayesian interpretation is valid and indeed quite natural. We shall therefore adopt the Bayesian interpretation in this book. Fortunately, the basic rules of probability theory are the same, no matter which interpretation is adopted.

2.2 Basic concepts of probability theory

2.2.1 Discrete random variables

The expression $p(A)$ denotes the probability that event A is true.We require that $0 \leq p(A) \leq 1$,where 0 means the event definitely will not happen,and $p(A) = 1$ means the event definitely will happen.$p(\hat{A})$ denotes the probability of the event not A;this is defined to be $p(\hat{A}) = 1 - p(A)$.

We denote a random event by defining a **random variable** X. **Descrete random variable**: X ,which can take on any value from a finite or countably infinite set .We denote the probability of the event that $X = x$ by $p(X = x)$,or just $p(x)$ for short.Here $p()$ is called a **probability mass function** or **pmf**.The pmfs are defined one **state space**.$\mathbb{I}$ denotes the binary **indicator funcition**.

Continuous random variable: the value of X is real-valued.

2.2.2 Fundamental rules

In this section,we review the basic rule of probability.

2.2.2.1 Probability of a union of two events

Given two events,A and B,we define the probability of A or B as follows:

$$p(A \cup B) = p(A) + p(B) - p(A \cap B) \tag{2.1}$$
$$= p(A) + p(B) \tag{2.2}$$

if A and B are mutually independent

2.2.2.2 Joint probabilities

We define the probability of the joint event A and B as follows:

$$p(A,B) = p(A \cap B) = p(A|B)p(B) \tag{2.3}$$

This is sometimes called the **product rule**

2.2.2.3 Conditional probability

Define the **conditional probability** of event A,given that event B is true,as follows:

$$p(A|B) = \frac{p(A,B)}{p(B)}, if p(B) > 0 \tag{2.4}$$

2.2.2.4 CDF

$$F(x) \triangleq P(X \leq x) = \begin{cases} \sum_{u \leq x} p(u) & \text{, discrete} \\ \int_{-\infty}^{x} f(u)\mathrm{d}u & \text{, continuous} \end{cases} \tag{2.5}$$

2.2.2.5 PMF and PDF

For descrete random variable, We denote the probability of the event that $X = x$ by $P(X = x)$, or just $p(x)$ for short. Here $p(x)$ is called a **probability mass function** or **PMF**.A probability mass function is a function that gives the probability that a discrete random variable is exactly equal to some value[4]. This satisfies the properties $0 \leq p(x) \leq 1$ and $\sum_{x \in \mathcal{X}} p(x) = 1$.

For continuous variable, in the equation $F(x) = \int_{-\infty}^{x} f(u)\mathrm{d}u$, the function $f(x)$ is called a **probability density function** or **PDF**. A probability density function is a function that describes the relative likelihood for this random variable to take on a given value[5].This satisfies the properties $f(x) \geq 0$ and $\int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1$.

2.2.3 Mutivariate random variables

2.2.3.1 Joint CDF

We denote joint CDF by $F(x,y) \triangleq P(X \leq x \cap Y \leq y) = P(X \leq x, Y \leq y)$.

$$F(x,y) \triangleq P(X \leq x, Y \leq y) = \begin{cases} \sum_{u \leq x, v \leq y} p(u,v) \\ \int_{-\infty}^{x} \int_{-\infty}^{y} f(u,v)\mathrm{d}u\mathrm{d}v \end{cases} \tag{2.6}$$

product rule:

$$p(X,Y) = P(X|Y)P(Y) \tag{2.7}$$

Chain rule:

$$p(X_{1:N}) = p(X_1)p(X_3|X_2,X_1)...p(X_N|X_{1:N-1}) \tag{2.8}$$

[4] http://en.wikipedia.org/wiki/Probability_mass_function

[5] http://en.wikipedia.org/wiki/Probability_density_function

2.2.3.2 Marginal distribution

Marginal CDF:

$$F_X(x) \triangleq F(x,+\infty) = \begin{cases} \sum_{x_i \le x} P(X=x_i) = \sum_{x_i \le x} \sum_{j=1}^{+\infty} P(X=x_i, Y=y_j) \\ \int_{-\infty}^{x} f_X(u)du = \int_{-\infty}^{x} \int_{-\infty}^{+\infty} f(u,v)\mathrm{d}u\mathrm{d}v \end{cases} \tag{2.9}$$

$$F_Y(y) \triangleq F(+\infty,y) = \begin{cases} \sum_{y_j \le y} p(Y=y_j) = \sum_{i=1}^{+\infty} \sum_{y_j \le y} P(X=x_i, Y=y_j) \\ \int_{-\infty}^{y} f_Y(v)dv = \int_{-\infty}^{+\infty} \int_{-\infty}^{y} f(u,v)\mathrm{d}u\mathrm{d}v \end{cases} \tag{2.10}$$

Marginal PMF and PDF:

$$\begin{cases} P(X=x_i) = \sum_{j=1}^{+\infty} P(X=x_i, Y=y_j) & \text{, descrete} \\ f_X(x) = \int_{-\infty}^{+\infty} f(x,y)\mathrm{d}y & \text{, continuous} \end{cases} \tag{2.11}$$

$$\begin{cases} p(Y=y_j) = \sum_{i=1}^{+\infty} P(X=x_i, Y=y_j) & \text{, descrete} \\ f_Y(y) = \int_{-\infty}^{+\infty} f(x,y)\mathrm{d}x & \text{, continuous} \end{cases} \tag{2.12}$$

2.2.3.3 Conditional distribution

Conditional PMF:

$$p(X=x_i|Y=y_j) = \frac{p(X=x_i, Y=y_j)}{p(Y=y_j)} \text{ if } p(Y)>0 \tag{2.13}$$

The pmf $p(X|Y)$ is called **conditional probability**.

Conditional PDF:

$$f_{X|Y}(x|y) = \frac{f(x,y)}{f_Y(y)} \tag{2.14}$$

2.2.4 Bayes rule

$$\begin{aligned} p(Y=y|X=x) &= \frac{p(X=x,Y=y)}{p(X=x)} \\ &= \frac{p(X=x|Y=y)p(Y=y)}{\sum_{y'} p(X=x|Y=y')p(Y=y')} \end{aligned} \tag{2.15}$$

sum rule

$$p(X) = \sum_Y p(X,Y) \tag{2.16}$$

product rule

$$p(X,Y) = p(Y|X)p(X) \tag{2.17}$$

Bayes' theorem

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \tag{2.18}$$

Denominator in Bayes' theorem

$$p(X) = \sum_Y p(X|Y)p(Y) \tag{2.19}$$

probability densities

$$p(x \in (a,b)) = \int_a^b p(x)dx \tag{2.20}$$

The probability density function p(x) must satisfy the two conditions

$$\begin{cases} p(x) \geq 0 \\ \int_{-\infty}^{\infty} p(x)dx = 1 \end{cases} \tag{2.21}$$

Combinations of discrete and continuous variables.

$$p(x) = \int_p (x,y)dy \tag{2.22}$$

$$p(x,y) = p(y|x)p(x) \tag{2.23}$$

Bayesian probabilities So far, we have viewed probabilities in terms of the frequencies of random,repeatable events,which we shall refer to as the classical or frequentist interpretation of probability.Now we turn to the more general Bayesian view,in which probabilities provide a quantification of uncertainty. We can adopt a similar approach when making inferences about quantities such as the parameters **w** in the polynomial curve fitting.We capture our assumptions about **w**,before observing the data,in the form of a prior probability distribution $p(\mathbf{w})$.The effect of the observed data $D = t_1, ..., t_N$ is expressed through the conditional probability $p(D|w)$

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \tag{2.24}$$

The quantity $p(D|w)$ on the right-hand side of Bayes' theorem is evaluated for the observed data set D and can be viewed as a function of the parameter vector **w**,in which case it is called the likelihood function.

$$posteroir \propto likelihood \times prior \tag{2.25}$$

Integrating both side with respect to **w**

$$p(D) = \int p(D|\mathbf{w})p(\mathbf{w})d\mathbf{w} \tag{2.26}$$

A widely used frequentise estimator is maximum likelihood,in which **w** is set to the value that maximizes the likehood function $p(D|w)$.

2.2.5 Independence and conditional independence

We say X and Y are unconditionally independent or marginally independent, denoted $X \perp Y$, if we can represent the joint as the product of the two marginals, i.e.,

$$X \perp Y = P(X,Y) = P(X)P(Y) \tag{2.27}$$

We say X and Y are conditionally independent(CI) given Z if the conditional joint can be written as a product of conditional marginals:

$$X \perp Y|Z = P(X,Y|Z) = P(X|Z)P(Y|Z) \tag{2.28}$$

2.2.6 Quantiles

Since the cdf F is a monotonically increasing function, it has an inverse; let us denote this by F^{-1}. If F is the cdf of X , then $F^{-1}(\alpha)$ is the value of x_α such that $P(X \leq x_\alpha) = \alpha$; this is called the α quantile of F. The value $F^{-1}(0.5)$ is the **median** of the distribution, with half of the probability mass on the left, and half on the right. The values $F^{-1}(0.25)$

and $F^1(0.75)$are the lower and upper **quartiles**.

2.2.7 Mean and variance

The most familiar property of a distribution is its **mean**,or **expected value**, denoted by μ. For discrete rvs, it is defined as $\mathbb{E}[X] \triangleq \sum_{x \in \mathcal{X}} xp(x)$, and for continuous rvs, it is defined as $\mathbb{E}[X] \triangleq \int_{\mathcal{X}} xp(x)\mathrm{d}x$. If this integral is not finite, the mean is not defined (we will see some examples of this later).

The **variance** is a measure of the spread of a distribution, denoted by σ^2. This is defined as follows:

$$\begin{aligned} var[X] &= \mathbb{E}[(X-\mu)^2] \qquad (2.29)\\ &= \int (x-\mu)^2 p(x)\mathrm{d}x \\ &= \int x^2 p(x)\mathrm{d}x + \mu^2 \int p(x)\mathrm{d}x - 2\mu \int xp(x)\mathrm{d}x \\ &= \mathbb{E}[X^2] - \mu^2 \qquad (2.30)\end{aligned}$$

from which we derive the useful result

$$\mathbb{E}[X^2] = \sigma^2 + \mu^2 \qquad (2.31)$$

The **standard deviation** is defined as

$$std[X] \triangleq \sqrt{var[X]} \qquad (2.32)$$

This is useful since it has the same units as X itself.

Expectations and covariances The average value of some function $f(x)$ under a probability distribution p(x) is called the expectation of f(x) and will be denoted by

$$\mathbb{E}[f] = \sum_x p(x)f(x)\mathbb{E}[f] = \int p(x)f(x)dx \qquad (2.33)$$

approximation

$$\mathbb{E}[f] \simeq \frac{1}{N}\sum_{n=1}^{N} f(x_n) \qquad (2.34)$$

conditional expectation with respect to a conditional distribution

$$\mathbb{E}_x[f|y] = \sum_x p(x|y)f(x) \qquad (2.35)$$

variance of f(x) is defined by

$$var[f] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2 var[f] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2 \qquad (2.36)$$

convariance

$$cov[x,y] = \mathbb{E}_{x,y}[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}] \qquad (2.37)$$

In the case of two vectors of random variables **x** and **y**

$$cov[\mathbf{x},\mathbf{y}] = \mathbb{E}_{x,y}[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T]\}]cov[\mathbf{x},\mathbf{y}] = \mathbb{E}_{x,y}[\mathbf{x}\mathbf{y}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^T] \qquad (2.38)$$

2.3 Some common discrete distributions

In this section, we review some commonly used parametric distributions defined on discrete state spaces, both finite and countably infinite.

2.3.1 The Bernoulli and binomial distributions

Definition 2.1. Now suppose we toss a coin only once. Let $X \in \{0,1\}$ be a binary random variable, with probability of success or heads of θ. We say that X has a **Bernoulli distribution**. This is written as $X \sim \text{Ber}(\theta)$, where the pmf is defined as

$$\text{Ber}(x|\theta) \triangleq \theta^{\mathbb{I}(x=1)}(1-\theta)^{\mathbb{I}(x=0)} \tag{2.39}$$

Definition 2.2. Suppose we toss a coin n times. Let $X \in \{0,1,\cdots,n\}$ be the number of heads. If the probability of heads is θ, then we say X has a **binomial distribution**, written as $X \sim \text{Bin}(n,\theta)$. The pmf is given by

$$\text{Bin}(k|n,\theta) \triangleq \binom{n}{k}\theta^k(1-\theta)^{n-k} \tag{2.40}$$

2.3.2 The multinoulli and multinomial distributions

Definition 2.3. The Bernoulli distribution can be used to model the outcome of one coin tosses. To model the outcome of tossing a K-sided dice, let $\boldsymbol{x} = (\mathbb{I}(x=1),\cdots,\mathbb{I}(x=K)) \in \{0,1\}^K$ be a random vector(this is called **dummy encoding** or **one-hot encoding**), then we say X has a **multinoulli distribution**(or **categorical distribution**), written as $X \sim \text{Cat}(\theta)$. The pmf is given by:

$$p(\boldsymbol{x}) \triangleq \prod_{k=1}^{K}\theta_k^{\mathbb{I}(x_k=1)} \tag{2.41}$$

Definition 2.4. Suppose we toss a K-sided dice n times. Let $\boldsymbol{x} = (x_1,x_2,\cdots,x_K) \in \{0,1,\cdots,n\}^K$ be a random vector, where x_j is the number of times side j of the dice occurs, then we say X has a **multinomial distribution**, written as $X \sim \text{Mu}(n,\boldsymbol{\theta})$. The pmf is given by

$$p(\boldsymbol{x}) \triangleq \binom{n}{x_1\cdots x_k}\prod_{k=1}^{K}\theta_k^{x_k} \tag{2.42}$$

where $\binom{n}{x_1\cdots x_k} \triangleq \dfrac{n!}{x_1!x_2!\cdots x_K!}$

Bernoulli distribution is just a special case of a Binomial distribution with $n=1$, and so is multinoulli distribution as to multinomial distribution. See Table 2.1 for a summary.

Table 2.1: Summary of the multinomial and related distributions.

Name	K	n	X
Bernoulli	1	1	$x \in \{0,1\}$
Binomial	1	-	$\boldsymbol{x} \in \{0,1,\cdots,n\}$
Multinoulli	-	1	$\boldsymbol{x} \in \{0,1\}^K, \sum_{k=1}^K x_k = 1$
Multinomial	-	-	$\boldsymbol{x} \in \{0,1,\cdots,n\}^K, \sum_{k=1}^K x_k = n$

2.3.3 The Poisson distribution

Definition 2.5. We say that $X \in \{0,1,2,\cdots\}$ has a **Poisson distribution** with parameter $\lambda > 0$, written as $X \sim \text{Poi}(\lambda)$, if its pmf is

$$p(x|\lambda) = e^{-\lambda}\frac{\lambda^x}{x!} \tag{2.43}$$

The first term is just the normalization constant, required to ensure the distribution sums to 1.

The Poisson distribution is often used as a model for counts of rare events like radioactive decay and traffic accidents.

Table 2.2: Summary of Bernoulli, binomial multinoulli and multinomial distributions.

Name	Written as	X	$p(x)$(or $p(\boldsymbol{x})$)	$\mathbb{E}[X]$	var$[X]$
Bernoulli	$X \sim \text{Ber}(\theta)$	$x \in \{0,1\}$	$\theta^{\mathbb{I}(x=1)}(1-\theta)^{\mathbb{I}(x=0)}$	θ	$\theta(1-\theta)$
Binomial	$X \sim \text{Bin}(n,\theta)$	$x \in \{0,1,\cdots,n\}$	$\binom{n}{k}\theta^k(1-\theta)^{n-k}$	$n\theta$	$n\theta(1-\theta)$
Multinoulli	$X \sim \text{Cat}(\boldsymbol{\theta})$	$\boldsymbol{x} \in \{0,1\}^K, \sum_{k=1}^K x_k = 1$	$\prod_{k=1}^K \theta_j^{\mathbb{I}(x_j=1)}$	-	-
Multinomial	$X \sim \text{Mu}(n,\boldsymbol{\theta})$	$\boldsymbol{x} \in \{0,1,\cdots,n\}^K, \sum_{k=1}^K x_k = n$	$\binom{n}{x_1 \cdots x_k}\prod_{k=1}^K \theta_j^{x_j}$	-	-
Poisson	$X \sim \text{Poi}(\lambda)$	$x \in \{0,1,2,\cdots\}$	$e^{-\lambda}\frac{\lambda^x}{x!}$	λ	λ

2.3.4 The empirical distribution

The **empirical distribution function**[6], or **empirical cdf**, is the cumulative distribution function associated with the empirical measure of the sample. Let $\mathcal{D} = \{x_1, x_2, \cdots, x_N\}$ be a sample set, it is defined as

$$F_n(x) \triangleq \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(x_i \leq x) \tag{2.44}$$

2.4 Some common continuous distributions

In this section we present some commonly used univariate (one-dimensional) continuous probability distributions.

2.4.1 Gaussian (normal) distribution

Table 2.3: Summary of Gaussian distribution.

Written as	$f(x)$	$\mathbb{E}[X]$	mode	var$[X]$
$X \sim \mathcal{N}(\mu,\sigma^2)$	$\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$	μ	μ	σ^2

If $X \sim N(0,1)$,we say X follows a **standard normal** distribution.

The Gaussian distribution is the most widely used distribution in statistics. There are several reasons for this.

1. First, it has two parameters which are easy to interpret, and which capture some of the most basic properties of a distribution, namely its mean and variance.

[6] http://en.wikipedia.org/wiki/Empirical_distribution_function

2. Second,the central limit theorem (Section TODO) tells us that sums of independent random variables have an approximately Gaussian distribution, making it a good choice for modeling residual errors or noise.
3. Third, the Gaussian distribution makes the least number of assumptions (has maximum entropy), subject to the constraint of having a specified mean and variance, as we show in Section TODO; this makes it a good default choice in many cases.
4. Finally, it has a simple mathematical form, which results in easy to implement, but often highly effective, methods, as we will see.

See (Jaynes 2003, ch 7) for a more extensive discussion of why Gaussians are so widely used.

2.4.2 Student's t-distribution

Table 2.4: Summary of Student's t-distribution.

Written as	$f(x)$	$\mathbb{E}[X]$	mode	var$[X]$
$X \sim \mathcal{T}(\mu, \sigma^2, \nu)$	$\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}\left[1+\frac{1}{\nu}\left(\frac{x-\mu}{\nu}\right)^2\right]$	μ	μ	$\frac{\nu\sigma^2}{\nu-2}$

where $\Gamma(x)$ is the gamma function:

$$\Gamma(x) \triangleq \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t \tag{2.45}$$

μ is the mean, $\sigma^2 > 0$ is the scale parameter, and $\nu > 0$ is called the **degrees of freedom**. See Figure 2.1 for some plots.

The variance is only defined if $\nu > 2$. The mean is only defined if $\nu > 1$.

As an illustration of the robustness of the Student distribution, consider Figure 2.2. We see that the Gaussian is affected a lot, whereas the Student distribution hardly changes. This is because the Student has heavier tails, at least for small ν(see Figure 2.1).

If $\nu = 1$, this distribution is known as the **Cauchy** or **Lorentz** distribution. This is notable for having such heavy tails that the integral that defines the mean does not converge.

To ensure finite variance, we require $\nu > 2$. It is common to use $\nu = 4$, which gives good performance in a range of problems (Lange et al. 1989). For $\nu \gg 5$, the Student distribution rapidly approaches a Gaussian distribution and loses its robustness properties.

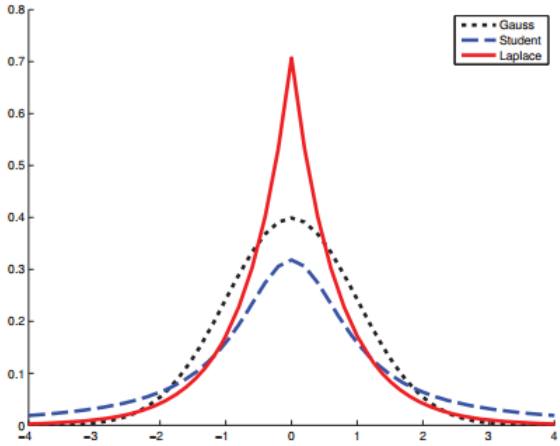
2.4.3 The Laplace distribution

Table 2.5: Summary of Laplace distribution.

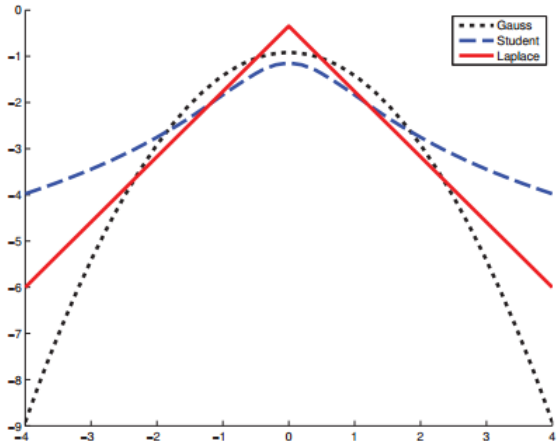
Written as	$f(x)$	$\mathbb{E}[X]$	mode	var$[X]$
$X \sim \text{Lap}(\mu, b)$	$\frac{1}{2b}\exp\left(-\frac{\lvert x-\mu\rvert}{b}\right)$	μ	μ	$2b^2$

Here μ is a location parameter and $b > 0$ is a scale parameter. See Figure 2.1 for a plot.

Its robustness to outliers is illustrated in Figure 2.2. It also put mores probability density at 0 than the Gaussian. This property is a useful way to encourage sparsity in a model, as we will see in Section TODO.



(a)



(b)

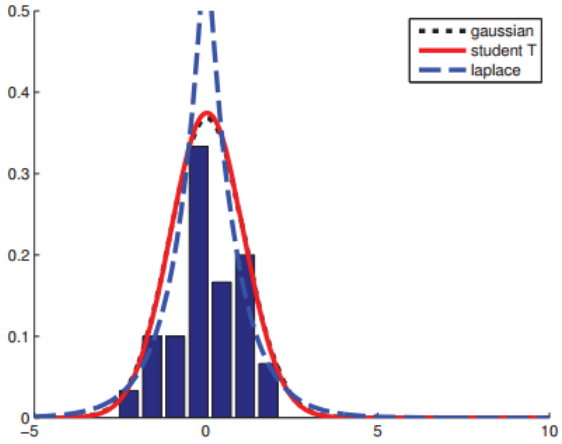
Fig. 2.1: (a) The pdfs for a $\mathcal{N}(0,1)$, $\mathcal{T}(0,1,1)$ and $Lap(0,1/\sqrt{2})$. The mean is 0 and the variance is 1 for both the Gaussian and Laplace. The mean and variance of the Student is undefined when $\nu = 1$.(b) Log of these pdfs. Note that the Student distribution is not log-concave for any parameter value, unlike the Laplace distribution, which is always log-concave (and log-convex...) Nevertheless, both are unimodal.

2.4.4 The gamma distribution

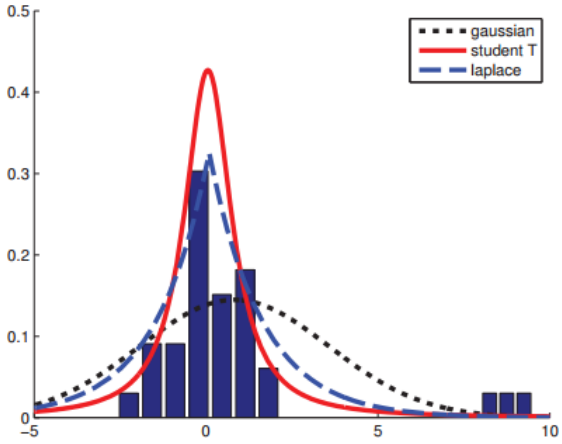
Table 2.6: Summary of gamma distribution

Written as	X	$f(x)$	$\mathbb{E}[X]$	mode	var$[X]$
$X \sim \text{Ga}(a,b)$	$x \in \mathbb{R}^+$	$\frac{b^a}{\Gamma(a)} x^{a-1} e^{-xb}$	$\frac{a}{b}$	$\frac{a-1}{b}$	$\frac{a}{b^2}$

Here $a > 0$ is called the shape parameter and $b > 0$ is called the rate parameter. See Figure 2.3 for some plots.



(a)



(b)

Fig. 2.2: Illustration of the effect of outliers on fitting Gaussian, Student and Laplace distributions. (a) No outliers (the Gaussian and Student curves are on top of each other). (b) With outliers. We see that the Gaussian is more affected by outliers than the Student and Laplace distributions.

2.4.5 The beta distribution

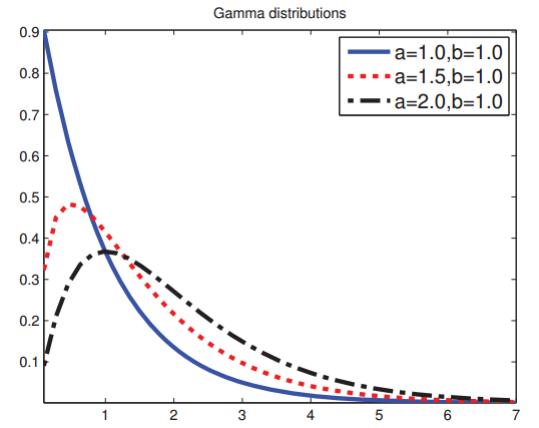
Table 2.7: Summary of Beta distribution

Name	Written as	X	$f(x)$	$\mathbb{E}[X]$	mode	var$[X]$
Beta distribution	$X \sim \text{Beta}(a,b)$	$x \in [0,1]$	$\frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}$	$\frac{a}{a+b}$	$\frac{a-1}{a+b-2}$	$\frac{ab}{(a+b)^2(a+b+1)}$

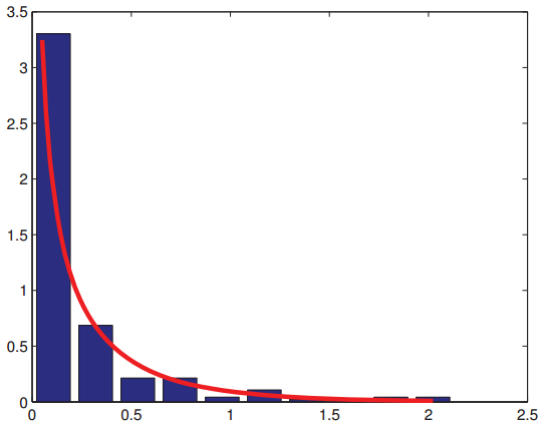
Here $B(a,b)$is the beta function,

$$B(a,b) \triangleq \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \tag{2.46}$$

See Figure 2.4 for plots of some beta distributions. We require $a,b > 0$ to ensure the distribution is integrable (i.e., to ensure $B(a,b)$ exists). If $a = b = 1$, we get the uniform distirbution. If a and b are both less than 1, we get a bimodal distribution with spikes at 0 and 1; if a and b are both greater than 1, the distribution is unimodal.



(a)



(b)

Fig. 2.3: Some Ga$(a,b=1)$ distributions. If $a \leq 1$, the mode is at 0, otherwise it is > 0.As we increase the rate b, we reduce the horizontal scale, thus squeezing everything leftwards and upwards. (b) An empirical pdf of some rainfall data, with a fitted Gamma distribution superimposed.

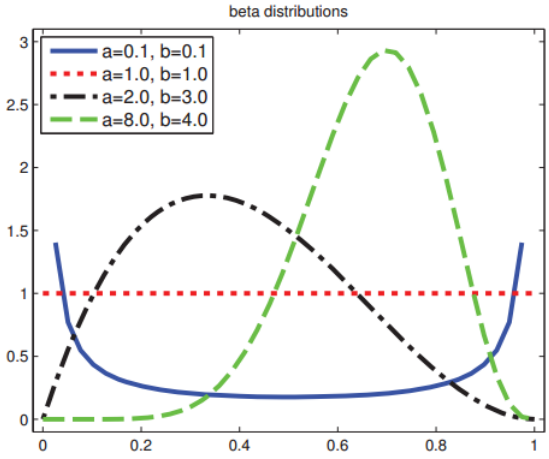


Fig. 2.4: Some beta distributions.

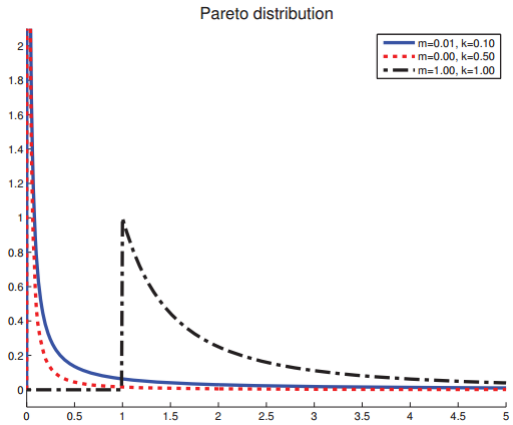
2.4.6 Pareto distribution

Table 2.8: Summary of Pareto distribution

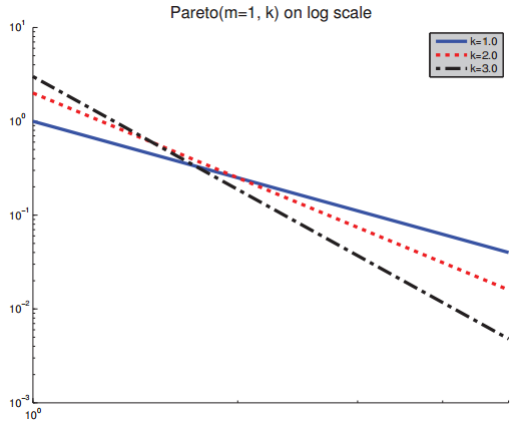
Name	Written as	X	$f(x)$	$\mathbb{E}[X]$	mode	var$[X]$
Pareto distribution	$X \sim \text{Pareto}(k,m)$	$x \geq m$	$km^k x^{-(k+1)} \mathbb{I}(x \geq m)$	$\frac{km}{k-1}$ if $k>1$	m	$\frac{m^2 k}{(k-1)^2(k-2)}$ if $k>2$

The **Pareto distribution** is used to model the distribution of quantities that exhibit **long tails**, also called **heavy tails**.

As $k \to \infty$, the distribution approaches $\delta(x-m)$. See Figure 2.5(a) for some plots. If we plot the distribution on a log-log scale, it forms a straight line, of the form $\log p(x) = a \log x + c$ for some constants a and c. See Figure 2.5(b) for an illustration (this is known as a **power law**).



(a)



(b)

Fig. 2.5: (a) The Pareto distribution Pareto$(x|m,k)$ for $m=1$. (b) The pdf on a log-log scale.

2.5 Joint probability distributions

Given a **multivariate random variable** or **random vector** [7] $X \in \mathbb{R}^D$, the **joint probability distribution**[8] is a probability distribution that gives the probability that each of $X_1, X_2, \cdots, X_D$ falls in any particular range or discrete set of values specified for that variable. In the case of only two random variables, this is called a **bivariate distribution**, but the concept generalizes to any number of random variables, giving a **multivariate distribution**.

The joint probability distribution can be expressed either in terms of a **joint cumulative distribution function** or in terms of a **joint probability density function** (in the case of continuous variables) or **joint probability mass function** (in the case of discrete variables).

2.5.1 Covariance and correlation

Definition 2.6. The **covariance** between two rvs X and Y measures the degree to which X and Y are (linearly) related. Covariance is defined as

$$\begin{aligned} \text{cov}[X,Y] &\triangleq \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right] \\ &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \end{aligned} \tag{2.47}$$

Definition 2.7. If X is a D-dimensional random vector, its **covariance matrix** is defined to be the following symmetric, positive definite matrix:

$$\text{cov}[X] \triangleq \mathbb{E}\left[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T\right] \tag{2.48}$$

$$= \begin{pmatrix} \text{var}[X_1] & \text{Cov}[X_1, X_2] & \cdots & \text{Cov}[X_1, X_D] \\ \text{Cov}[X_2, X_1] & \text{var}[X_2] & \cdots & \text{Cov}[X_2, X_D] \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}[X_D, X_1] & \text{Cov}[X_D, X_2] & \cdots & \text{var}[X_D] \end{pmatrix} \tag{2.49}$$

Definition 2.8. The (Pearson) **correlation coefficient** between X and Y is defined as

$$\text{corr}[X,Y] \triangleq \frac{\text{Cov}[X,Y]}{\sqrt{\text{var}[X], \text{var}[Y]}} \tag{2.50}$$

A **correlation matrix** has the form

$$\mathbf{R} \triangleq \begin{pmatrix} \text{corr}[X_1, X_1] & \text{corr}[X_1, X_2] & \cdots & \text{corr}[X_1, X_D] \\ \text{corr}[X_2, X_1] & \text{corr}[X_2, X_2] & \cdots & \text{corr}[X_2, X_D] \\ \vdots & \vdots & \ddots & \vdots \\ \text{corr}[X_D, X_1] & \text{corr}[X_D, X_2] & \cdots & \text{corr}[X_D, X_D] \end{pmatrix} \tag{2.51}$$

The correlation coefficient can viewed as a degree of linearity between X and Y, see Figure 2.6.

Uncorrelated does not imply independent. For example, let $X \sim U(-1,1)$ and $Y = X^2$. Clearly Y is dependent on X(in fact, Y is uniquely determined by X), yet one can show that $\text{corr}[X,Y] = 0$. Some striking examples of this fact are shown in Figure 2.6. This shows several data sets where there is clear dependence between X and Y, and yet the correlation coefficient is 0. A more general measure of dependence between random variables is mutual information, see Section TODO.

[7] http://en.wikipedia.org/wiki/Multivariate_random_variable

[8] http://en.wikipedia.org/wiki/Joint_probability_distribution

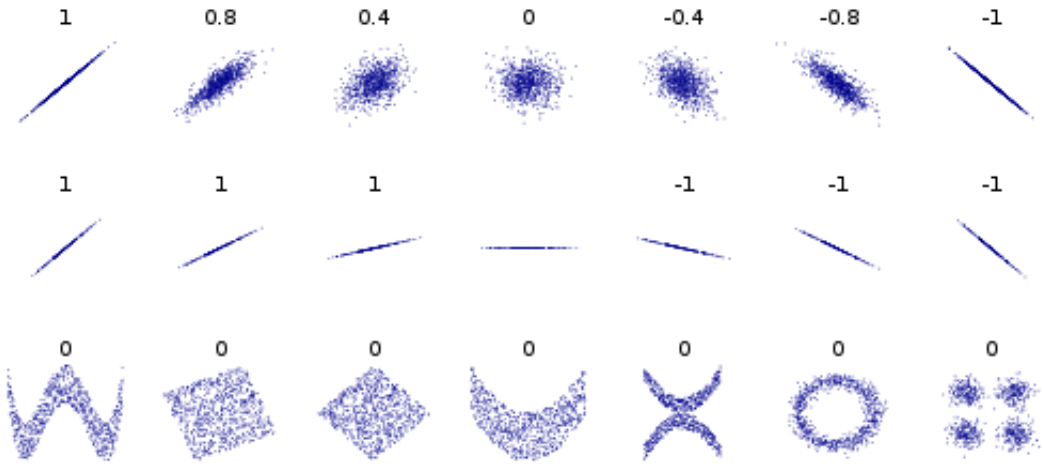


Fig. 2.6: Several sets of (x,y) points, with the Pearson correlation coefficient of x and y for each set. Note that the correlation reflects the noisiness and direction of a linear relationship (top row), but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom). N.B.: the figure in the center has a slope of 0 but in that case the correlation coefficient is undefined because the variance of Y is zero.Source:http://en.wikipedia.org/wiki/Correlation

2.5.2 Multivariate Gaussian distribution

The **multivariate Gaussian** or **multivariate normal**(MVN) is the most widely used joint probability density function for continuous variables. We discuss MVNs in detail in Chapter 4; here we just give some definitions and plots.

The pdf of the MVN in D dimensions is defined by the following:

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\Sigma) \triangleq \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{2.52}$$

where $\boldsymbol{\mu} = \mathbb{E}[X] \in \mathbb{R}^D$ is the mean vector, and $\Sigma = \mathrm{Cov}[X]$ is the $D \times D$ covariance matrix. The normalization constant $(2\pi)^{D/2}|\Sigma|^{1/2}$ just ensures that the pdf integrates to 1.

Figure 2.7 plots some MVN densities in 2d for three different kinds of covariance matrices. A full covariance matrix has A $D(D+1)/2$ parameters (we divide by 2 since Σ is symmetric). A diagonal covariance matrix has D parameters, and has 0s in the off-diagonal terms. A spherical or isotropic covariance,$\Sigma = \sigma^2\boldsymbol{I}_D$, has one free parameter.

2.5.3 Multivariate Student's t-distribution

A more robust alternative to the MVN is the multivariate Student's t-distribution, whose pdf is given by

$$\begin{aligned}
&\mathcal{T}(x|\boldsymbol{\mu},\Sigma,\nu) \\
&\triangleq \frac{\Gamma(\frac{\nu+D}{2})}{\Gamma(\frac{\nu}{2})}\frac{|\Sigma|^{-\frac{1}{2}}}{(\nu\pi)^{\frac{D}{2}}}\left[1+\frac{1}{\nu}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{-\frac{\nu+D}{2}} &(2.53)\\
&= \frac{\Gamma(\frac{\nu+D}{2})}{\Gamma(\frac{\nu}{2})}\frac{|\Sigma|^{-\frac{1}{2}}}{(\nu\pi)^{\frac{D}{2}}}\left[1+(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]^{-\frac{\nu+D}{2}} &(2.54)
\end{aligned}$$

where Σ is called the scale matrix (since it is not exactly the covariance matrix) and $\boldsymbol{V} = \nu\Sigma$. This has fatter tails than a Gaussian. The smaller ν is, the fatter the tails. As $\nu \to \infty$, the distribution tends towards a Gaussian. The distribution

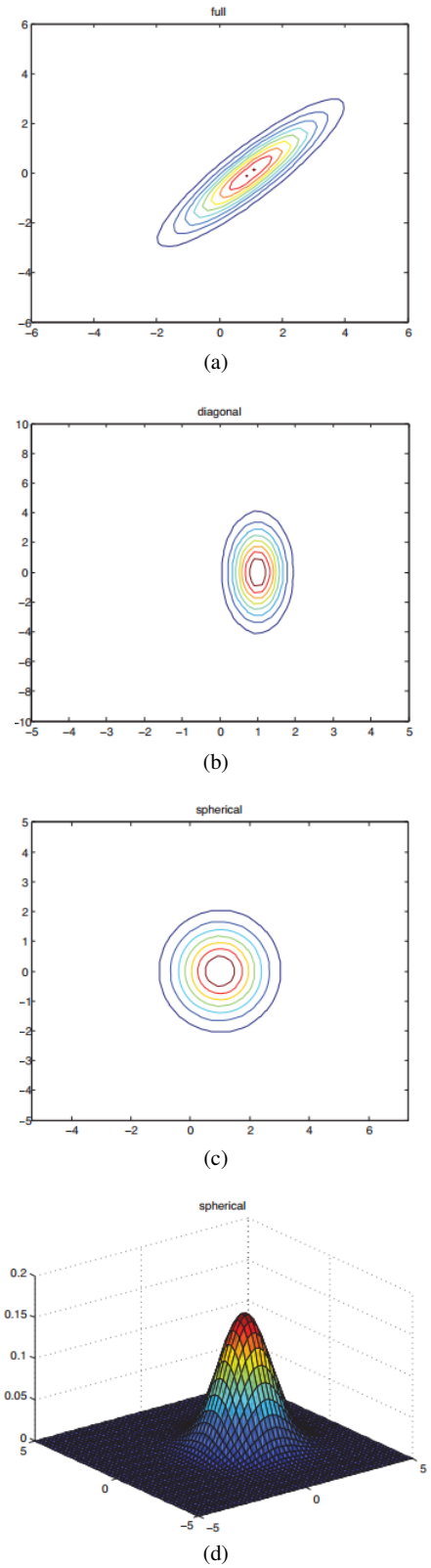


Fig. 2.7: We show the level sets for 2d Gaussians. (a) A full covariance matrix has elliptical contours.(b) A diagonal covariance matrix is an axis aligned ellipse. (c) A spherical covariance matrix has a circular shape. (d) Surface plot for the spherical Gaussian in (c).

has the following properties

$$\text{mean} = \boldsymbol{\mu} \text{ , mode} = \boldsymbol{\mu} \text{ , Cov} = \frac{\nu}{\nu - 2}\Sigma \tag{2.55}$$

2.5.4 Dirichlet distribution

A multivariate generalization of the beta distribution is the **Dirichlet distribution**, which has support over the probability simplex, defined by

$$S_K = \left\{ \boldsymbol{x} : 0 \leq x_k \leq 1, \sum_{k=1}^{K} x_k = 1 \right\} \tag{2.56}$$

The pdf is defined as follows:

$$\text{Dir}(\boldsymbol{x}|\boldsymbol{\alpha}) \triangleq \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} x_k^{\alpha_k - 1} \mathbb{I}(\boldsymbol{x} \in S_K) \tag{2.57}$$

where $B(\alpha_1, \alpha_2, \cdots, \alpha_K)$ is the natural generalization of the beta function to K variables:

$$B(\alpha) \triangleq \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\alpha_0)} \text{ where } \alpha_0 \triangleq \sum_{k=1}^{K} \alpha_k \tag{2.58}$$

Figure 2.8 shows some plots of the Dirichlet when $K = 3$, and Figure 2.9 for some sampled probability vectors. We see that α_0 controls the strength of the distribution (how peaked it is), and thekcontrol where the peak occurs. For example, $\text{Dir}(1,1,1)$ is a uniform distribution, $\text{Dir}(2,2,2)$ is a broad distribution centered at $(1/3, 1/3, 1/3)$, and $\text{Dir}(20,20,20)$ is a narrow distribution centered at $(1/3, 1/3, 1/3)$.If $\alpha_k < 1$ for all k, we get spikes at the corner of the simplex.

For future reference, the distribution has these properties

$$\mathbb{E}(x_k) = \frac{\alpha_k}{\alpha_0}, \text{mode}[x_k] = \frac{\alpha_k - 1}{\alpha_0 - K}, \text{var}[x_k] = \frac{\alpha_k(\alpha_0 - \alpha_k)}{\alpha_0^2(\alpha_0 + 1)} \tag{2.59}$$

2.6 Transformations of random variables

If $\boldsymbol{x} \sim P()$ is some random variable, and $\boldsymbol{y} = f(\boldsymbol{x})$, what is the distribution of Y? This is the question we address in this section.

2.6.1 Linear transformations

Suppose $g()$ is a linear function:

$$g(\boldsymbol{x}) = A\boldsymbol{x} + b \tag{2.60}$$

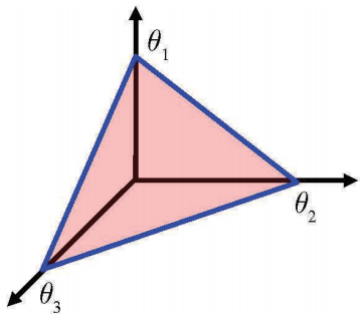
First, for the mean, we have

$$\mathbb{E}[\boldsymbol{y}] = \mathbb{E}[A\boldsymbol{x} + b] = A\mathbb{E}[\boldsymbol{x}] + b \tag{2.61}$$

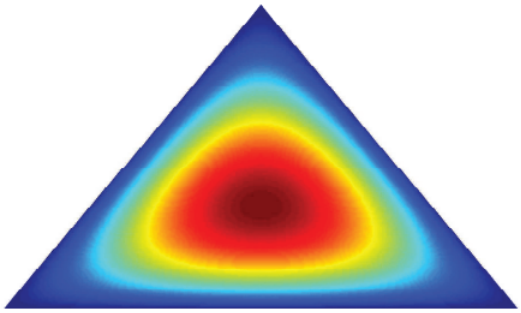
this is called the **linearity of expectation**.

For the covariance, we have

$$\text{Cov}[\boldsymbol{y}] = \text{Cov}[A\boldsymbol{x} + b] = A\Sigma A^T \tag{2.62}$$

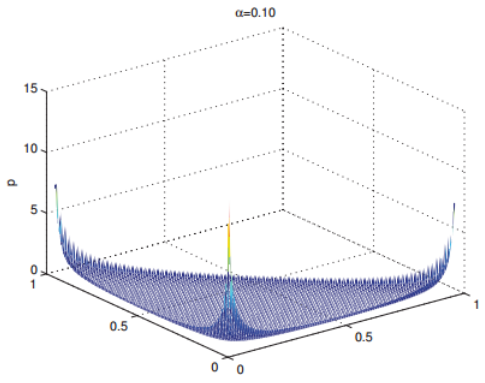


(a)



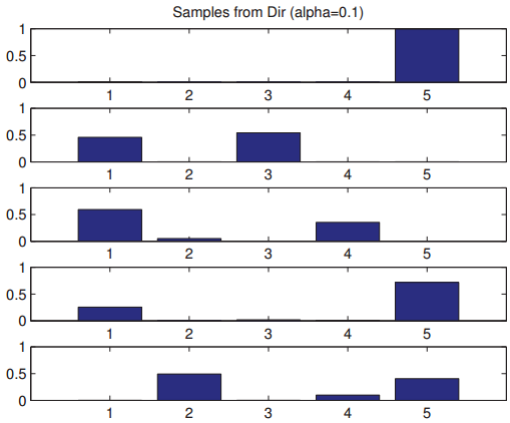
(b)

(c)

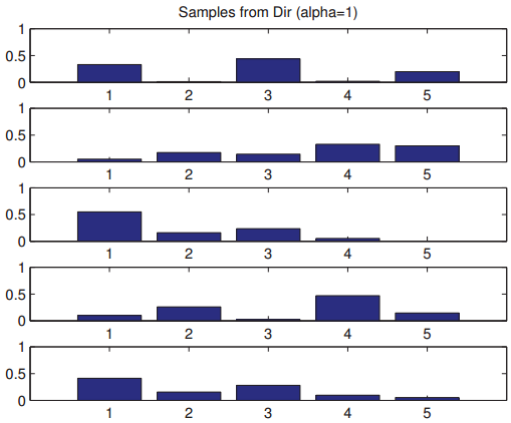


(d)

Fig. 2.8: (a) The Dirichlet distribution when $K = 3$ defines a distribution over the simplex, which can be represented by the triangular surface. Points on this surface satisfy $0 \leq \theta_k \leq 1$ and $\sum_{k=1}^{K} \theta_k = 1$. (b) Plot of the Dirichlet density when $\boldsymbol{\alpha} = (2,2,2)$. (c) $\boldsymbol{\alpha} = (20,2,2)$.



(a) $\boldsymbol{\alpha} = (0.1, \cdots, 0.1)$. This results in very sparse distributions, with many 0s.



(b) $\boldsymbol{\alpha} = (1, \cdots, 1)$. This results in more uniform (and dense) distributions.

Fig. 2.9: Samples from a 5-dimensional symmetric Dirichlet distribution for different parameter values.

2.6.2 General transformations

If X is a discrete rv, we can derive the pmf for y by simply summing up the probability mass for all the xs such that $f(x) = y$:

$$p_Y(y) = \sum_{x:g(x)=y} p_X(x) \tag{2.63}$$

If X is continuous, we cannot use Equation 2.63 since $p_X(x)$ is a density, not a pmf, and we cannot sum up densities. Instead, we work with cdfs, and write

$$F_Y(y) = P(Y \leq y) = P(g(X) \leq y) = \int_{g(X) \leq y} f_X(x)\mathrm{d}x \tag{2.64}$$

We can derive the pdf of Y by differentiating the cdf:

$$f_Y(y) = f_X(x)|\frac{dx}{dy}| \tag{2.65}$$

This is called **change of variables** formula. We leave the proof of this as an exercise.

For example, suppose $X \sim U(1,1)$, and $Y = X^2$. Then $p_Y(y) = \frac{1}{2} y^{-\frac{1}{2}}$.

2.6.2.1 Multivariate change of variables *

Let f be a function $f : \mathbb{R}^n \to \mathbb{R}^n$, and let $\boldsymbol{y} = f(\boldsymbol{x})$. Then its Jacobian matrix $\boldsymbol{J}$ is given by

$$\boldsymbol{J}_{\boldsymbol{x} \to \boldsymbol{y}} \triangleq \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \triangleq \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{pmatrix} \tag{2.66}$$

$|\det(\boldsymbol{J})|$ measures how much a unit cube changes in volume when we apply f.

If f is an invertible mapping, we can define the pdf of the transformed variables using the Jacobian of the inverse mapping $\boldsymbol{y} \to \boldsymbol{x}$:

$$p_y(\boldsymbol{y}) = p_x(\boldsymbol{x}) |\det(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}})| = p_x(\boldsymbol{x}) |\det(\boldsymbol{J}_{\boldsymbol{y} \to \boldsymbol{x}})| \tag{2.67}$$

2.6.3 Central limit theorem

Given N random variables $X_1, X_2, \cdots, X_N$, each variable is **independent and identically distributed**[9](**iid** for short), and each has the same mean μ and variance σ^2, then

$$\frac{\sum_{i=1}^{n} X_i - N\mu}{\sqrt{N}\sigma} \sim \mathcal{N}(0,1) \tag{2.68}$$

this can also be written as

$$\frac{\bar{X} - \mu}{\sigma / \sqrt{N}} \sim \mathcal{N}(0,1) \quad \text{, where } \bar{X} \triangleq \frac{1}{N} \sum_{i=1}^{n} X_i \tag{2.69}$$

2.7 Monte Carlo approximation

In general, computing the distribution of a function of an rv using the change of variables formula can be difficult. One simple but powerful alternative is as follows. First we generate S samples from the distribution, call them $x_1, \cdots, x_S$. (There are many ways to generate such samples; one popular method, for high dimensional distributions, is called Markov chain Monte Carlo or MCMC; this will be explained in Chapter TODO.) Given the samples, we can approximate the distribution of $f(X)$ by using the empirical distribution of $\{f(x_s)\}_{s=1}^{S}$. This is called a **Monte Carlo approximation**[10], named after a city in Europe known for its plush gambling casinos.

We can use Monte Carlo to approximate the expected value of any function of a random variable. We simply draw samples, and then compute the arithmetic mean of the function applied to the samples. This can be written as follows:

$$\mathbb{E}[g(X)] = \int g(x) p(x) \mathrm{d}x \approx \frac{1}{S} \sum_{s=1}^{S} f(x_s) \tag{2.70}$$

where $x_s \sim p(X)$.

[9] http://en.wikipedia.org/wiki/Independent_identically_distributed

[10] http://en.wikipedia.org/wiki/Monte_Carlo_method

This is called **Monte Carlo integration**[11], and has the advantage over numerical integration (which is based on evaluating the function at a fixed grid of points) that the function is only evaluated in places where there is non-negligible probability.

2.8 Information theory

2.8.1 Entropy

The entropy of a random variable X with distribution p, denoted by $\mathbb{H}(X)$ or sometimes $\mathbb{H}(p)$, is a measure of its uncertainty. In particular, for a discrete variable with K states, it is defined by

$$\mathbb{H}(X) \triangleq -\sum_{k=1}^{K} p(X=k)\log_2 p(X=k) \tag{2.71}$$

Usually we use log base 2, in which case the units are called **bits**(short for binary digits). If we use log base e , the units are called **nats**.

The discrete distribution with maximum entropy is the uniform distribution (see Section XXX for a proof). Hence for a K-ary random variable, the entropy is maximized if $p(x=k) = 1/K$; in this case, $\mathbb{H}(X) = \log_2 K$.

Conversely, the distribution with minimum entropy (which is zero) is any **delta-function** that puts all its mass on one state. Such a distribution has no uncertainty.

2.8.2 KL divergence

One way to measure the dissimilarity of two probability distributions, p and q , is known as the **Kullback-Leibler divergence**(**KL divergence**)or **relative entropy**. This is defined as follows:

$$\mathbb{KL}(P||Q) \triangleq \sum_x p(x)\log_2 \frac{p(x)}{q(x)} \tag{2.72}$$

where the sum gets replaced by an integral for pdfs[12]. The KL divergence is only defined if P and Q both sum to 1 and if $q(x) = 0$ implies $p(x) = 0$ for all x(absolute continuity). If the quantity $0\ln 0$ appears in the formula, it is interpreted as zero because $\lim_{x\to 0} x\ln x$. We can rewrite this as

$$\begin{aligned}\mathbb{KL}(p||q) &\triangleq \sum_x p(x)\log_2 p(x) - \sum_{k=1}^{K} p(x)\log_2 q(x) \\ &= \mathbb{H}(p) - \mathbb{H}(p,q)\end{aligned} \tag{2.73}$$

where $\mathbb{H}(p,q)$ is called the **cross entropy**,

$$\mathbb{H}(p,q) = \sum_x p(x)\log_2 q(x) \tag{2.74}$$

One can show (Cover and Thomas 2006) that the cross entropy is the average number of bits needed to encode data coming from a source with distribution p when we use model q to define our codebook. Hence the regular entropy $\mathbb{H}(p) = \mathbb{H}(p,p)$, defined in section §2.8.1,is the expected number of bits if we use the true model, so the KL divergence is the diference between these. In other words, the KL divergence is the average number of *extra* bits needed to encode

[11] http://en.wikipedia.org/wiki/Monte_Carlo_integration

[12] The KL divergence is not a distance, since it is asymmetric. One symmetric version of the KL divergence is the **Jensen-Shannon divergence**, defined as $JS(p_1,p_2) = 0.5\mathbb{KL}(p_1||q) + 0.5\mathbb{KL}(p_2||q)$,where $q = 0.5p_1 + 0.5p_2$

the data, due to the fact that we used distribution q to encode the data instead of the true distribution p.

The extra number of bits interpretation should make it clear that $\mathbb{KL}(p||q) \geq 0$, and that the KL is only equal to zero if $q = p$. We now give a proof of this important result.

Theorem 2.1. ***(Information inequality)*** $\mathbb{KL}(p||q) \geq 0$ *with equality iff* $p = q$.

One important consequence of this result is that *the discrete distribution with the maximum entropy is the uniform distribution*.

2.8.3 Mutual information

Definition 2.9. Mutual information or **MI**, is defined as follows:

$$\begin{aligned} \mathbb{I}(X;Y) &\triangleq \mathbb{KL}(P(X,Y)||P(X)P(X)) \\ &= \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \end{aligned} \tag{2.75}$$

We have $\mathbb{I}(X;Y) \geq 0$ with equality if $P(X,Y) = P(X)P(Y)$. That is, the MI is zero if the variables are independent.

To gain insight into the meaning of MI, it helps to re-express it in terms of joint and conditional entropies. One can show that the above expression is equivalent to the following:

$$\mathbb{I}(X;Y) = \mathbb{H}(X) - \mathbb{H}(X|Y) \tag{2.76}$$
$$= \mathbb{H}(Y) - \mathbb{H}(Y|X) \tag{2.77}$$
$$= \mathbb{H}(X) + \mathbb{H}(Y) - \mathbb{H}(X,Y) \tag{2.78}$$
$$= \mathbb{H}(X,Y) - \mathbb{H}(X|Y) - \mathbb{H}(Y|X) \tag{2.79}$$

where $\mathbb{H}(X)$ and $\mathbb{H}(Y)$ are the **marginal entropies**, $\mathbb{H}(X|Y)$ and $\mathbb{H}(Y|X)$ are the **conditional entropies**, and $\mathbb{H}(X,Y)$ is the **joint entropy** of X and Y, see Fig. 2.10[13].

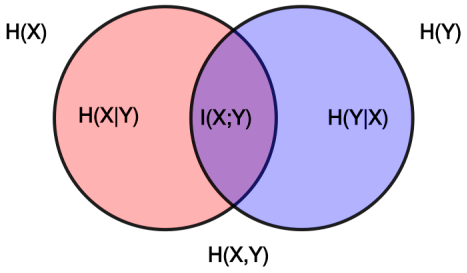


Fig. 2.10: Individual $\mathbb{H}(X), \mathbb{H}(Y)$, joint $\mathbb{H}(X,Y)$, and conditional entropies for a pair of correlated subsystems X, Y with mutual information $\mathbb{I}(X;Y)$.

Intuitively, we can interpret the MI between X and Y as the reduction in uncertainty about X after observing Y, or, by symmetry, the reduction in uncertainty about Y after observing X.

A quantity which is closely related to MI is the **pointwise mutual information** or **PMI**. For two events (not random variables) x and y, this is defined as

[13] http://en.wikipedia.org/wiki/Mutual_information

$$PMI(x,y) \triangleq \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)} \tag{2.80}$$

This measures the discrepancy between these events occuring together compared to what would be expected by chance. Clearly the MI of X and Y is just the expected value of the PMI. Interestingly, we can rewrite the PMI as follows:

$$PMI(x,y) = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)} \tag{2.81}$$

This is the amount we learn from updating the prior $p(x)$ into the posterior $p(x|y)$, or equivalently, updating the prior $p(y)$ into the posterior $p(y|x)$.

Chapter 3
Latent linear models

3.1 Factor analysis

One problem with mixture models is that they only use a single latent variable to generate the observations. In particular, each observation can only come from one of K prototypes. One can think of a mixture model as using K hidden binary variables, representing a one-hot encoding of the cluster identity. But because these variables are mutually exclusive, the model is still limited in its representational power.

An alternative is to use a vector of real-valued latent variables,$\boldsymbol{z}_i \in \mathbb{R}^L$. The simplest prior to use is a Gaussian (we will consider other choices later):

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{3.1}$$

If the observations are also continuous, so $\boldsymbol{x}_i \in \mathbb{R}^D$, we may use a Gaussian for the likelihood. Just as in linear regression, we will assume the mean is a linear function of the (hidden) inputs, thus yielding

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}) \tag{3.2}$$

where $\boldsymbol{W}$ is a $D \times L$ matrix, known as the **factor loading matrix**, and $\boldsymbol{\Psi}$ is a $D \times D$ covariance matrix. We take $\boldsymbol{\Psi}$ to be diagonal, since the whole point of the model is to force $\boldsymbol{z}_i$ to explain the correlation, rather than baking it in to the observations covariance. This overall model is called **factor analysis** or **FA**. The special case in which $\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}$ is called **probabilistic principal components analysis** or **PPCA**. The reason for this name will become apparent later.

3.1.1 FA is a low rank parameterization of an MVN

FA can be thought of as a way of specifying a joint density model on $\boldsymbol{x}$ using a small number of parameters. To see this, note that from Equation **??**, the induced marginal distribution $p(\boldsymbol{x}_i|\boldsymbol{\theta})$ is a Gaussian:

$$\begin{aligned} p(\boldsymbol{x}_i|\boldsymbol{\theta}) &= \int \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\mathrm{d}\boldsymbol{z}_i \\ &= \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}) \end{aligned} \tag{3.3}$$

From this, we see that we can set $\boldsymbol{\mu}_0 = 0$ without loss of generality, since we can always absorb $\boldsymbol{W}\boldsymbol{\mu}_0$ into $\boldsymbol{\mu}$. Similarly, we can set $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$ without loss of generality, because we can always emulate a correlated prior by using defining a new weight matrix, $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{\Sigma}_0^{-\frac{1}{2}}$. So we can rewrite Equation 3.6 and 3.2 as:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I}) \tag{3.4}$$

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}) \tag{3.5}$$

We thus see that FA approximates the covariance matrix of the visible vector using a low-rank decomposition:

$$\boldsymbol{C} \triangleq \mathrm{cov}[\boldsymbol{x}] = \boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{\Psi} \tag{3.6}$$

This only uses $O(LD)$ parameters, which allows a flexible compromise between a full covariance Gaussian, with $O(D^2)$ parameters, and a diagonal covariance, with $O(D)$ parameters. Note that if we did not restrict $\boldsymbol{\Psi}$ to be diagonal, we could trivially set $\boldsymbol{\Psi}$ to a full covariance matrix; then we could set $\boldsymbol{W} = 0$, in which case the latent factors would not be required.

3.1.2 Inference of the latent factors

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i,\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \tag{3.7}$$

$$\boldsymbol{\Sigma}_i \triangleq (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W})^{-1} \tag{3.8}$$

$$= (\boldsymbol{I} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W})^{-1} \tag{3.9}$$

$$\boldsymbol{\mu}_i \triangleq \boldsymbol{\Sigma}_i[\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0] \tag{3.10}$$

$$= \boldsymbol{\Sigma}_i\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) \tag{3.11}$$

Note that in the FA model, $\boldsymbol{\Sigma}_i$ is actually independent of i, so we can denote it by $\boldsymbol{\Sigma}$. Computing this matrix takes $O(L^3 + L^2D)$ time, and computing each $\boldsymbol{\mu}_i = \mathbb{E}[\boldsymbol{z}_i|\boldsymbol{x}_i,\boldsymbol{\theta}]$ takes $O(L^2 + LD)$ time. The $\boldsymbol{\mu}_i$ are sometimes called the **latent scores**, or **latent factors**.

3.1.3 Unidentifiability

Just like with mixture models, FA is also unidentifiable. To see this, suppose $\boldsymbol{R}$ is an arbitrary orthogonal rotation matrix, satisfying $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$. Let us define $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{R}$, then the likelihood function of this modified matrix is the same as for the unmodified matrix, since $\boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{W}^T + \boldsymbol{\Psi} = \boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{\Psi}$. Geometrically, multiplying $\boldsymbol{W}$ by an orthogonal matrix is like rotating $\boldsymbol{z}$ before generating $\boldsymbol{x}$.

To ensure a unique solution, we need to remove $L(L-1)/2$ degrees of freedom, since that is the number of orthonormal matrices of size $L \times L$.[14] In total, the FA model has $D + LD - L(L-1)/2$ free parameters (excluding the mean), where the first term arises from $\boldsymbol{\Psi}$. Obviously we require this to be less than or equal to $D(D+1)/2$, which is the number of parameters in an unconstrained (but symmetric) covariance matrix. This gives us an upper bound on L, as follows:

$$L_{\max} = \lfloor D + 0.5(1 - \sqrt{1+8D}) \rfloor \tag{3.12}$$

For example, $D = 6$ implies $L \leq 3$. But we usually never choose this upper bound, since it would result in overfitting (see discussion in Section 3.3 on how to choose L).

Unfortunately, even if we set $L < L_{\max}$, we still cannot uniquely identify the parameters, since the rotational ambiguity still exists. Non-identifiability does not affect the predictive performance of the model. However, it does affect the loading matrix, and hence the interpretation of the latent factors. Since factor analysis is often used to uncover structure in the data, this problem needs to be addressed. Here are some commonly used solutions:

- **Forcing $\boldsymbol{W}$ to be orthonormal** Perhaps the cleanest solution to the identifiability problem is to force $\boldsymbol{W}$ to be orthonormal, and to order the columns by decreasing variance of the corresponding latent factors. This is the approach adopted by PCA, which we will discuss in Section 3.2. The result is not necessarily more interpretable, but at least it is unique.
- **Forcing $\boldsymbol{W}$ to be lower triangular** One way to achieve identifiability, which is popular in the Bayesian community (e.g., (Lopes and West 2004)), is to ensure that the first visible feature is only generated by the first latent factor, the second visible feature is only generated by the first two latent factors, and so on. For example, if $L = 3$ and $D = 4$, the correspond factor loading matrix is given by

$$\boldsymbol{W} = \begin{pmatrix} w_{11} & 0 & 0 \\ w_{21} & w_{22} & 0 \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{32} & w_{43} \end{pmatrix}$$

  We also require that $w_{jj} > 0$ for $j = 1 : L$. The total number of parameters in this constrained matrix is $D + DL - L(L-1)/2$, which is equal to the number of uniquely identifiable parameters. The disadvantage of this method is

[14] To see this, note that there are $L-1$ free parameters in $\boldsymbol{R}$ in the first column (since the column vector must be normalized to unit length), there are $L-2$ free parameters in the second column (which must be orthogonal to the first), and so on.

that the first L visible variables, known as the **founder variables**, affect the interpretation of the latent factors, and so must be chosen carefully.

- **Sparsity promoting priors on the weights** Instead of pre-specifying which entries in $\boldsymbol{W}$ are zero, we can encourage the entries to be zero, using ℓ_1 regularization (Zou et al. 2006), ARD (Bishop 1999; Archambeau and Bach 2008), or spike-and-slab priors (Rattray et al. 2009). This is called sparse factor analysis. This does not necessarily ensure a unique MAP estimate, but it does encourage interpretable solutions. See Section 13.8 TODO.
- **Choosing an informative rotation matrix** There are a variety of heuristic methods that try to find rotation matrices $\boldsymbol{R}$ which can be used to modify $\boldsymbol{W}$(and hence the latent factors) so as to try to increase the interpretability, typically by encouraging them to be (approximately) sparse. One popular method is known as **varimax**(Kaiser 1958).
- **Use of non-Gaussian priors for the latent factors** In Section 3.6, we will dicuss how replacing $p(\boldsymbol{z}_i)$ with a non-Gaussian distribution can enable us to sometimes uniquely identify $\boldsymbol{W}$ as well as the latent factors. This technique is known as ICA.

3.1.4 Mixtures of factor analysers

The FA model assumes that the data lives on a low dimensional linear manifold. In reality, most data is better modeled by some form of low dimensional *curved* manifold. We can approximate a curved manifold by a piecewise linear manifold. This suggests the following model: let the k'th linear subspace of dimensionality L_k be represented by $\boldsymbol{W}_k$, for $k = 1 : K$. Suppose we have a latent indicator $qi \in \{1, \cdots, K\}$ specifying which subspace we should use to generate the data. We then sample $\boldsymbol{z}_i$ from a Gaussian prior and pass it through the $\boldsymbol{W}_k$ matrix (where $k = q_i$), and add noise. More precisely, the model is as follows:

$$p(q_i|\boldsymbol{\theta}) = \text{Cat}(q_i\boldsymbol{\pi}) \tag{3.13}$$

$$p(\boldsymbol{z}_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I}) \tag{3.14}$$

$$p(\boldsymbol{x}_i|q_i = k, \boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}_k, \boldsymbol{\Psi}) \tag{3.15}$$

This is called a **mixture of factor analysers**(MFA) (Hinton et al. 1997).

Another way to think about this model is as a low-rank version of a mixture of Gaussians. In particular, this model needs $O(KLD)$ parameters instead of the $O(KD^2)$ parameters needed for a mixture of full covariance Gaussians. This can reduce overfitting. In fact, MFA is a good generic density model for high-dimensional real-valued data.

3.1.5 EM for factor analysis models

Below we state the results without proof. The derivation can be found in (Ghahramani and Hinton 1996a). To obtain the results for a single factor analyser, just set $r_{ic} = 1$ and $c = 1$ in the equations below. In Section 3.2.4 we will see a further simplification of these equations that arises when fitting a PPCA model, where the results will turn out to have a particularly simple and elegant interpretation.

In the E-step, we compute the posterior responsibility of cluster k for data point i using

$$r_{ik} \triangleq p(q_i = k|\boldsymbol{x}_i, \boldsymbol{\theta}) \propto \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{W}_k\boldsymbol{W}_k^T\boldsymbol{\Psi}_k) \tag{3.16}$$

The conditional posterior for $\boldsymbol{z}_i$ is given by

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i, q_i = k, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \tag{3.17}$$

$$\boldsymbol{\Sigma}_{ik} \triangleq (\boldsymbol{I} + \boldsymbol{W}_k^T\boldsymbol{\Psi}_k^{-1}\boldsymbol{W})_k^{-1} \tag{3.18}$$

$$\boldsymbol{\mu}_{ik} \triangleq \boldsymbol{\Sigma}_{ik}\boldsymbol{W}_k^T\boldsymbol{\Psi}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) \tag{3.19}$$

In the M step, it is easiest to estimate $\boldsymbol{\mu}_k$ and $\boldsymbol{W}_k$ at the same time, by defining $\tilde{\boldsymbol{W}}_k = (\boldsymbol{W}_k, \boldsymbol{\mu}_k)$, $\tilde{\boldsymbol{z}} = (\boldsymbol{z}, 1)$, also, define

$$\tilde{\boldsymbol{W}}_k = (\boldsymbol{W}_k, \boldsymbol{\mu}_k) \tag{3.20}$$

$$\tilde{\boldsymbol{z}} = (\boldsymbol{z}, 1) \tag{3.21}$$

$$\boldsymbol{b}_{ik} \triangleq \mathbb{E}[\tilde{\boldsymbol{z}}|\boldsymbol{x}_i, q_i = k] = \mathbb{E}[(\boldsymbol{\mu}_{ik}; 1)] \tag{3.22}$$

$$\boldsymbol{C}_{ik} \triangleq \mathbb{E}[\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}^T|\boldsymbol{x}_i, q_i = k] \tag{3.23}$$

$$= \begin{pmatrix} \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T|\boldsymbol{x}_i, q_i = k] & \mathbb{E}[\boldsymbol{z}|\boldsymbol{x}_i, q_i = k] \\ \mathbb{E}[\boldsymbol{z}|\boldsymbol{x}_i, q_i = k]^T & 1 \end{pmatrix} \tag{3.24}$$

Then the M step is as follows:

$$\hat{\boldsymbol{\pi}}_k = \frac{1}{N}\sum_{i=1}^{N} r_{ik} \tag{3.25}$$

$$\hat{\tilde{\boldsymbol{W}}}_k = \left(\sum_{i=1}^{N} r_{ik}\boldsymbol{x}_i\boldsymbol{b}_{ik}^T\right)\left(\sum_{i=1}^{N} r_{ik}\boldsymbol{x}_i\boldsymbol{C}_{ik}^T\right)^{-1} \tag{3.26}$$

$$\hat{\boldsymbol{\Psi}} = \frac{1}{N}\text{diag}\left[\sum_{i=1}^{N} r_{ik}(\boldsymbol{x}_i - \hat{\tilde{\boldsymbol{W}}}_{ik}\boldsymbol{b}_{ik})\boldsymbol{x}_i^T\right] \tag{3.27}$$

Note that these updates are for vanilla EM. A much faster version of this algorithm, based on ECM, is described in (Zhao and Yu 2008).

3.1.6 Fitting FA models with missing data

In many applications, such as collaborative filtering, we have missing data. One virtue of the EM approach to fitting an FA/PPCA model is that it is easy to extend to this case. However, overfitting can be a problem if there is a lot of missing data. Consequently it is important to perform MAP estimation or to use Bayesian inference. See e.g., (Ilin and Raiko 2010) for details.

3.2 Principal components analysis (PCA)

Consider the FA model where we constrain $\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}$, and $\boldsymbol{W}$ to be orthonormal. It can be shown (Tipping and Bishop 1999) that, as $\sigma^2 \to 0$, this model reduces to classical (nonprobabilistic) **principal components analysis**(PCA), also known as the Karhunen Loeve transform. The version where $\sigma^2 > 0$ is known as **probabilistic PCA**(**PPCA**) (Tipping and Bishop 1999), orsensible PCA(Roweis 1997).

3.2.1 Classical PCA

3.2.1.1 Statement of the theorem

The synthesis viewof classical PCA is summarized in the forllowing theorem.

Theorem 3.1. *Suppose we want to find an orthogonal set of L linear basis vectors $\boldsymbol{w}_j \in \mathbb{R}^D$, and the corresponding scores $\boldsymbol{z}_i \in \mathbb{R}^L$, such that we minimize the average* ***reconstruction error***

$$J(\boldsymbol{W}, \boldsymbol{Z}) = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 \tag{3.28}$$

where $\hat{\boldsymbol{x}}_i = \boldsymbol{W}\boldsymbol{z}_i$, subject to the constraint that $\boldsymbol{W}$ is orthonormal. Equivalently, we can write this objective as follows

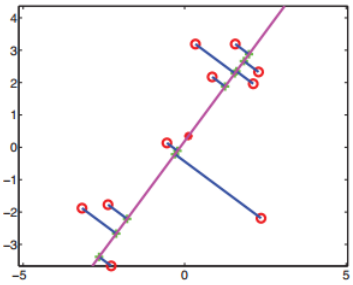
$$J(\boldsymbol{W},\boldsymbol{Z}) = \frac{1}{N}\|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{Z}^T\|^2 \tag{3.29}$$

where $\boldsymbol{Z}$ is an $N \times L$ matrix with the $\boldsymbol{z}_i$ in its rows, and $\|A\|_F$ is the ***Frobenius norm*** *of matrix $\boldsymbol{A}$, defined by*

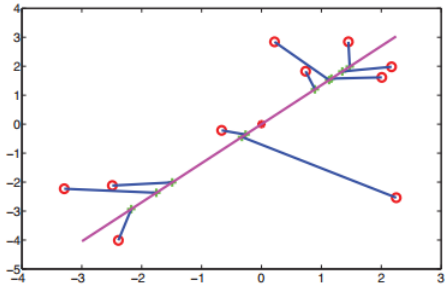
$$\|A\|_F \triangleq \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} a_{ij}^2} = \sqrt{\text{tr}(\boldsymbol{A}^T\boldsymbol{A})} \tag{3.30}$$

The optimal solution is obtained by setting $\hat{\boldsymbol{W}} = \boldsymbol{V}_L$, where $\boldsymbol{V}_L$ contains the L eigenvectors with largest eigenvalues of the empirical covariance matrix, $\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T$. (We assume the $\boldsymbol{x}_i$ have zero mean, for notational simplicity.) Furthermore, the optimal low-dimensional encoding of the data is given by $\hat{\boldsymbol{z}}_i = \boldsymbol{W}^T\boldsymbol{x}_i$, which is an orthogonal projection of the data onto the column space spanned by the eigenvectors.

An example of this is shown in Figure 3.1(a) for $D = 2$ and $L = 1$. The diagonal line is the vector $\boldsymbol{w}_1$; this is called the first principal component or principal direction. The data points $\boldsymbol{x}_i \in \mathbb{R}^2$ are orthogonally projected onto this line to get $\boldsymbol{z}_i \in \mathbb{R}$. This is the best 1-dimensional approximation to the data. (We will discuss Figure 3.1(b) later.)



(a)



(b)

Fig. 3.1: An illustration of PCA and PPCA where $D = 2$ and $L = 1$. Circles are the original data points, crosses are the reconstructions. The red star is the data mean. (a) PCA. The points are orthogonally projected onto the line. (b) PPCA. The projection is no longer orthogonal: the reconstructions are shrunk towards the data mean (red star).

The principal directions are the ones along which the data shows maximal variance. This means that PCA can be misled by directions in which the variance is high merely because of the measurement scale. It is therefore standard practice to standardize the data first, or equivalently, to work with correlation matrices instead of covariance matrices.

3.2.1.2 Proof *

See Section 12.2.2 of MLAPP.

3.2.2 Singular value decomposition (SVD)

We have defined the solution to PCA in terms of eigenvectors of the covariance matrix. However, there is another way to obtain the solution, based on the **singular value decomposition**, or **SVD**. This basically generalizes the notion of eigenvectors from square matrices to any kind of matrix.

Theorem 3.2. *(SVD). Any matrix can be decomposed as follows*

$$\underbrace{\boldsymbol{X}}_{N\times D} = \underbrace{\boldsymbol{U}}_{N\times N}\underbrace{\boldsymbol{\Sigma}}_{N\times D}\underbrace{\boldsymbol{V}^T}_{D\times D} \tag{3.31}$$

where $\boldsymbol{U}$ is an $N\times N$ matrix whose columns are orthornormal(so $\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}$), $\boldsymbol{V}$ is $D\times D$ matrix whose rows and columns are orthonormal (so $\boldsymbol{V}^T\boldsymbol{V}=\boldsymbol{V}\boldsymbol{V}^T=\boldsymbol{I}_D$), and $\boldsymbol{\Sigma}$ is a $N\times D$ matrix containing the $r=\min(N,D)$ singular values $\sigma_i\geq 0$ on the main diagonal, with 0s filling the rest of the matrix.

This shows how to decompose the matrix X into the product of three matrices: $\boldsymbol{V}$ describes an orthonormal basis in the domain, and $\boldsymbol{U}$ describes an orthonormal basis in the co-domain, and $\boldsymbol{\Sigma}$ describes how much the vectors in $\boldsymbol{V}$ are stretched to give the vectors in $\boldsymbol{U}$.

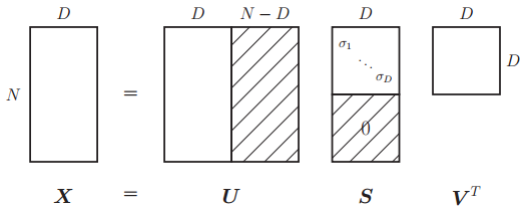
Since there are at most D singular values (assuming $N>D$), the last ND columns of $\boldsymbol{U}$ are irrelevant, since they will be multiplied by 0. The **economy sized SVD**, or **thin SVD**, avoids computing these unnecessary elements. Let us denote this decomposition by $\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^T$ .If $N>D$, we have

$$\underbrace{\boldsymbol{X}}_{N\times D} = \underbrace{\hat{\boldsymbol{U}}}_{N\times D}\underbrace{\hat{\boldsymbol{\Sigma}}}_{D\times D}\underbrace{\hat{\boldsymbol{V}}^T}_{D\times D} \tag{3.32}$$

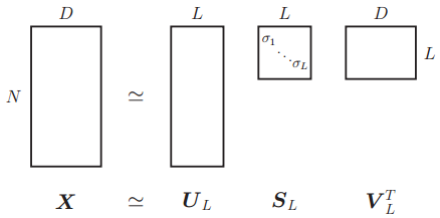
as in Figure 3.2(a). If $N<D$, we have

$$\underbrace{\boldsymbol{X}}_{N\times D} = \underbrace{\hat{\boldsymbol{U}}}_{N\times N}\underbrace{\hat{\boldsymbol{\Sigma}}}_{N\times N}\underbrace{\hat{\boldsymbol{V}}^T}_{N\times D} \tag{3.33}$$

Computing the economy-sized SVD takes $O(ND\min(N,D))$ time (Golub and van Loan 1996, p254).



(a)



(b)

Fig. 3.2: (a) SVD decomposition of non-square matrices $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The shaded parts of $\boldsymbol{\Sigma}$, and all the off-diagonal terms, are zero. The shaded entries in $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$ are not computed in the economy-sized version, since they are not needed. (b) Truncated SVD approximation of rank L.

The connection between eigenvectors and singular vectors is the following:

$$\boldsymbol{U} = \text{evec}(\boldsymbol{X}\boldsymbol{X}^T) \tag{3.34}$$

$$\boldsymbol{V} = \text{evec}(\boldsymbol{X}^T\boldsymbol{X}) \tag{3.35}$$

$$\boldsymbol{\Sigma}^2 = \text{eval}(\boldsymbol{X}\boldsymbol{X}^T) = \text{eval}(\boldsymbol{X}^T\boldsymbol{X}) \tag{3.36}$$

For the proof please read Section 12.2.3 of MLAPP.

Since the eigenvectors are unaffected by linear scaling of a matrix, we see that the right singular vectors of $\boldsymbol{X}$ are equal to the eigenvectors of the empirical covariance $\hat{\boldsymbol{\Sigma}}$. Furthermore, the eigenvalues of $\hat{\boldsymbol{\Sigma}}$ are a scaled version of the squared singular values.

However, the connection between PCA and SVD goes deeper. From Equation 4.1, we can represent a rank r matrix as follows:

$$\boldsymbol{X} = \sigma_1 \begin{pmatrix} | \\ \boldsymbol{u}_1 \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_1 & - \end{pmatrix} + \cdots + \sigma_r \begin{pmatrix} | \\ \boldsymbol{u}_r \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_r^T & - \end{pmatrix}$$

If the singular values die off quickly, we can produce a rank L approximation to the matrix as follows:

$$\begin{aligned} \boldsymbol{X} &\approx \sigma_1 \begin{pmatrix} | \\ \boldsymbol{u}_1 \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_1 & - \end{pmatrix} + \cdots + \sigma_r \begin{pmatrix} | \\ \boldsymbol{u}_L \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_L^T & - \end{pmatrix} \\ &= \boldsymbol{U}_{:,1:L}\boldsymbol{\Sigma}_{1:L,1:L}\boldsymbol{V}_{:,1:L}^T \end{aligned} \tag{3.37}$$

This is called a **truncated SVD** (see Figure 3.2(b)).

One can show that the error in this approximation is given by

$$\|\boldsymbol{X} - \boldsymbol{X}_L\|_F \approx \sigma_L \tag{3.38}$$

Furthermore, one can show that the SVD offers the best rank L approximation to a matrix (best in the sense of minimizing the above Frobenius norm).

Let us connect this back to PCA. Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be a truncated SVD of $\boldsymbol{X}$. We know that $\hat{\boldsymbol{W}} = \boldsymbol{V}$, and that $\hat{\boldsymbol{Z}} = \boldsymbol{X}\hat{\boldsymbol{W}}$, so

$$\hat{\boldsymbol{Z}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma} \tag{3.39}$$

Furthermore, the optimal reconstruction is given by $\hat{\boldsymbol{X}} = \boldsymbol{Z}\hat{\boldsymbol{W}}$,so we find

$$\hat{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{3.40}$$

This is precisely the same as a truncated SVD approximation! This is another illustration of the fact that PCA is the best low rank approximation to the data.

3.2.3 Probabilistic PCA

Theorem 3.3. *((**Tipping and Bishop 1999**)). Consider a factor analysis model in which $\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}$ and $\boldsymbol{W}$ is orthogonal. The observed data log likelihood is given by*

$$\begin{aligned} \log p(\boldsymbol{X}|\boldsymbol{W}, \sigma^2\boldsymbol{I}) &= -\frac{N}{2}\ln|\boldsymbol{C}| - \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{x}_i^T\boldsymbol{C}^{-1}\boldsymbol{x}_i \\ &= -\frac{N}{2}\ln|\boldsymbol{C}| + \text{tr}(\boldsymbol{C}^{-1}\boldsymbol{\Sigma}) \end{aligned} \tag{3.41}$$

where $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$ and $\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$. (We are assuming centred data, for notational simplicity.) The maxima of the log-likelihood are given by

$$\hat{\boldsymbol{W}} = \boldsymbol{V}(\boldsymbol{\Lambda} - \sigma^2\boldsymbol{I})^{\frac{1}{2}}\boldsymbol{R} \tag{3.42}$$

where $\boldsymbol{R}$ is an arbitrary $L \times L$ orthogonal matrix, $\boldsymbol{V}$ is the $D \times L$ matrix whose columns are the first L eigenvectors of $\boldsymbol{\Sigma}$, and $\boldsymbol{\Lambda}$ is the corresponding diagonal matrix of eigenvalues. Without loss of generality, we can set $\boldsymbol{R} = \boldsymbol{I}$. Furthermore, the MLE of the noise variance is given by

$$\hat{\sigma}^2 = \frac{1}{D-L} \sum_{j=L+1}^{D} \lambda_j \tag{3.43}$$

which is the average variance associated with the discarded dimensions.

Thus, as $\sigma^2 \to 0$, we have $\hat{\boldsymbol{W}} \to \boldsymbol{V}$, as in classical PCA. What about $\hat{\boldsymbol{Z}}$? It is easy to see that the posterior over the latent factors is given by

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}) = \mathcal{N}(\boldsymbol{z}_i|\hat{\boldsymbol{F}}^{-1}\hat{\boldsymbol{W}}^T\boldsymbol{x}_i, \sigma^2\hat{\boldsymbol{F}}^{-1}) \tag{3.44}$$

$$\hat{\boldsymbol{F}} \triangleq \hat{\boldsymbol{W}}^T\hat{\boldsymbol{W}} + \sigma^2\boldsymbol{I} \tag{3.45}$$

(Do not confuse $\boldsymbol{F} = \boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I}$ with $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$.) Hence, as $\sigma^2 \to 0$, we find $\hat{\boldsymbol{W}} \to \boldsymbol{V}$, $\hat{\boldsymbol{F}} \to \boldsymbol{I}$ and $\boldsymbol{z}_i \to \boldsymbol{V}^T\boldsymbol{x}_i$. Thus the posterior mean is obtained by an orthogonal projection of the data onto the column space of $\boldsymbol{V}$, as in classical PCA.

Note, however, that if $\sigma^2 \to 0$, the posterior mean is not an orthogonal projection, since it is shrunk somewhat towards the prior mean, as illustrated in Figure 3.1(b). This sounds like an undesirable property, but it means that the reconstructions will be closer to the overall data mean, $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{x}}$.

3.2.4 EM algorithm for PCA

Although the usual way to fit a PCA model uses eigenvector methods, or the SVD, we can also use EM, which will turn out to have some advantages that we discuss below. EM for PCA relies on the probabilistic formulation of PCA. However the algorithm continues to work in the zero noise limit, $\sigma^2 = 0$, as shown by (Roweis 1997).

Let $\tilde{\boldsymbol{Z}}$ be a $L \times N$ matrix storing the posterior means (low-dimensional representations) along its columns. Similarly, let $\tilde{\boldsymbol{X}} = \boldsymbol{X}^T$ store the original data along its columns. From Equation 3.44, when $\sigma^2 = 0$, we have

$$\tilde{\boldsymbol{Z}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\tilde{\boldsymbol{X}} \tag{3.46}$$

This constitutes the E step. Notice that this is just an orthogonal projection of the data.

From Equation 3.26, the M step is given by

$$\hat{\boldsymbol{W}} = \left(\sum_{i=1}^{N} \boldsymbol{x}_i\mathbb{E}[\boldsymbol{z}_i]^T\right)\left(\sum_{i=1}^{N} \mathbb{E}[\boldsymbol{z}_i]\mathbb{E}[\boldsymbol{z}_i]^T\right)^{-1} \tag{3.47}$$

where we exploited the fact that $\boldsymbol{\Sigma} = \text{cov}[\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}] = \boldsymbol{0}$ when $\sigma^2 = 0$.

(Tipping and Bishop 1999) showed that the only stable fixed point of the EM algorithm is the globally optimal solution. That is, the EM algorithm converges to a solution where $\boldsymbol{W}$ spans the same linear subspace as that defined by the first L eigenvectors. However, if we want $\boldsymbol{W}$ to be orthogonal, and to contain the eigenvectors in descending order of eigenvalue, we have to orthogonalize the resulting matrix (which can be done quite cheaply). Alternatively, we can modify EM to give the principal basis directly (Ahn and Oh 2003).

This algorithm has a simple physical analogy in the case $D = 2$ and $L = 1$(Roweis 1997). Consider some points in $\mathbb{R}^2$ attached by springs to a rigid rod, whose orientation is defined by a vector $\boldsymbol{w}$. Let $\boldsymbol{z}_i$ be the location where the i'th spring attaches to the rod. See Figure 12.11 of MLAPP for an illustration.

Apart from this pleasing intuitive interpretation, EM for PCA has the following advantages over eigenvector methods:

- EM can be faster. In particular, assuming $N, D \gg L$, the dominant cost of EM is the projection operation in the E step, so the overall time is $O(TLND)$, where T is the number of iterations. This is much faster than the

$O(\min(ND^2, DN^2))$ time required by straightforward eigenvector methods, although more sophisticated eigenvector methods, such as the Lanczos algorithm, have running times comparable to EM.

- EM can be implemented in an online fashion, i.e., we can update our estimate of $\boldsymbol{W}$ as the data streams in.
- EM can handle missing data in a simple way (see Section 3.1.6).
- EM can be extended to handle mixtures of PPCA/ FA models.
- EM can be modified to variational EM or to variational Bayes EM to fit more complex models.

3.3 Choosing the number of latent dimensions

In Section **??**, we discussed how to choose the number of components K in a mixture model. In this section, we discuss how to choose the number of latent dimensions L in a FA/PCA model.

3.3.1 Model selection for FA/PPCA

TODO

3.3.2 Model selection for PCA

TODO

3.4 PCA for categorical data

In this section, we consider extending the factor analysis model to the case where the observed data is categorical rather than real-valued. That is, the data has the form $y_{ij} \in \{1, ..., C\}$, where $j = 1 : R$ is the number of observed response variables. We assume each y_{ij} is generated from a latent variable $\boldsymbol{z}_i \in \mathbb{R}^L$, with a Gaussian prior, which is passed through the softmax function as follows:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i | \boldsymbol{0}, \boldsymbol{I}) \tag{3.48}$$

$$p(\boldsymbol{y}_i | \boldsymbol{z}_i, \boldsymbol{\theta}) = \prod_{j=1}^{R} \text{Cat}(y_{ir} | \mathcal{S}(\boldsymbol{W}_r^T \boldsymbol{z}_i + \boldsymbol{w}_{0r})) \tag{3.49}$$

where $\boldsymbol{W}_r \in \mathbb{R}^L$ is the factor loading matrix for response j, and $\boldsymbol{W}_{0r} \in \mathbb{R}^M$ is the offset term for response r, and $\boldsymbol{\theta} = (\boldsymbol{W}_r, \boldsymbol{W}_{0r})_{r=1}^R$. (We need an explicit offset term, since clamping one element of $\boldsymbol{z}_i$ to 1 can cause problems when computing the posterior covariance.) As in factor analysis, we have defined the prior mean to be $\boldsymbol{\mu}_0 = \boldsymbol{0}$ and the prior covariance $\boldsymbol{V}_0 = \boldsymbol{I}$, since we can capture non-zero mean by changing $\boldsymbol{w}_{0j}$ and non-identity covariance by changing $\boldsymbol{W}_r$. We will call this categorical PCA. See Chapter 27 TODO for a discussion of related models.

In (Khan et al. 2010), we show that this model outperforms finite mixture models on the task of imputing missing entries in design matrices consisting of real and categorical data. This is useful for analysing social science survey data, which often has missing data and variables of mixed type.

3.5 PCA for paired and multi-view data

3.5.1 Supervised PCA (latent factor regression)

3.5.2 Discriminative supervised PCA

3.5.3 Canonical correlation analysis

3.6 Independent Component Analysis (ICA)

Let $\boldsymbol{x}_t \in \mathbb{R}^D$ be the observed signal at the sensors at time t, and $\boldsymbol{z}_t \in \mathbb{R}^L$ be the vector of source signals. We assume that

$$\boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{z}_t + \boldsymbol{\epsilon}_t \tag{3.50}$$

where $\boldsymbol{W}$ is an $D \times L$ matrix, and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$. In this section, we treat each time point as an independent observation, i.e., we do not model temporal correlation (so we could replace the t index with i, but we stick with t to be consistent with much of the ICA literature). The goal is to infer the source signals, $p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$. In this context, $\boldsymbol{W}$ is called the **mixing matrix**. If $L = D$ (number of sources = number of sensors), it will be a square matrix. Often we will assume the noise level, $|\boldsymbol{\Psi}|$, is zero, for simplicity.

So far, the model is identical to factor analysis. However, we will use a different prior for $p(\boldsymbol{z}_t)$. In PCA, we assume each source is independent, and has a Gaussian distribution. We will now relax this Gaussian assumption and let the source distributions be any *non-Gaussian* distribution

$$p(\boldsymbol{z}_t) = \prod_{j=1}^{L} p_j(z_{tj}) \tag{3.51}$$

Without loss of generality, we can constrain the variance of the source distributions to be $\mathbf{1}$, because any other variance can be modelled by scaling the rows of $\boldsymbol{W}$ appropriately. The resulting model is known as **independent component analysis** or **ICA**.

The reason the Gaussian distribution is disallowed as a source prior in ICA is that it does not permit unique recovery of the sources. This is because the PCA likelihood is invariant to any orthogonal transformation of the sources $\boldsymbol{z}_t$ and mixing matrix $\boldsymbol{W}$. PCA can recover the best linear subspace in which the signals lie, but cannot uniquely recover the signals themselves.

ICA requires that $\boldsymbol{W}$ is square and hence invertible. In the non-square case (e.g., where we have more sources than sensors), we cannot uniquely recover the true signal, but we can compute the posterior $p(\boldsymbol{z}_t|\boldsymbol{x}_t, \hat{\boldsymbol{W}})$, which represents our beliefs about the source. In both cases, we need to estimate Was well as the source distributions p_j. We discuss how to do this below.

3.6.1 Maximum likelihood estimation

In this section, we discuss ways to estimate square mixing matrices $\boldsymbol{W}$ for the noise-free ICA model. As usual, we will assume that the observations have been centered; hence we can also assume $\boldsymbol{z}$ is zero-mean. In addition, we assume the observations have been whitened, which can be done with PCA.

If the data is centered and whitened, we have $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = \boldsymbol{I}$. But in the noise free case, we also have

$$\text{cov}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = \boldsymbol{W}\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T]\boldsymbol{W}^T \tag{3.52}$$

Hence we see that $\boldsymbol{W}$ must be orthogonal. This reduces the number of parameters we have to estimate from D^2 to $D(D-1)/2$. It will also simplify the math and the algorithms.

Let $\boldsymbol{V} = \boldsymbol{W}^{-1}$; these are often called the recognition weights, as opposed to $\boldsymbol{W}$, which are the generative weights.

Since $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z}$, we have, from Equation 2.67,

$$\begin{aligned} p_x(\boldsymbol{W}\boldsymbol{z}_t) &= p_z(\boldsymbol{z}_t)|\det(\boldsymbol{W}^{-1})| \\ &= p_z(\boldsymbol{V}\boldsymbol{x}_t)|\det(\boldsymbol{V})| \end{aligned} \tag{3.53}$$

Hence we can write the log-likelihood, assuming T iid samples, as follows:

$$\frac{1}{T}\log p(\mathcal{D}|\boldsymbol{V}) = \log|\det(\boldsymbol{V})| + \frac{1}{T}\sum_{j=1}^{L}\sum_{t=1}^{T}\log p_j(\boldsymbol{v}_j^T\boldsymbol{x}_t)$$

where $\boldsymbol{v}_j$ is the j'th row of $\boldsymbol{V}$. Since we are constraining $\boldsymbol{V}$ to be orthogonal, the first term is a constant, so we can drop it. We can also replace the average over the data with an expectation operator to get the following objective

$$\text{NLL}(\boldsymbol{V}) = \sum_{j=1}^{L}\mathbb{E}[G_j(\boldsymbol{z}_j)] \tag{3.54}$$

where $\boldsymbol{z}_j = \boldsymbol{v}_j^T\boldsymbol{x}$ and $G_j(\boldsymbol{z}) \triangleq -\log p_j(\boldsymbol{z})$. We want to minimize this subject to the constraint that the rows of $\boldsymbol{V}$ are orthogonal. We also want them to be unit norm, since this ensures that the variance of the factors is unity (since, with whitened data, $\mathbb{E}[\boldsymbol{v}_j^T\boldsymbol{x}] = \|\boldsymbol{v}_j\|^2$, which is necessary to fix the scale of the weights. In otherwords, $\boldsymbol{V}$ should be an orthonormal matrix.

It is straightforward to derive a gradient descent algorithm to fit this model; however, it is rather slow. One can also derive a faster algorithm that follows the natural gradient; see e.g., (MacKay 2003, ch 34) for details. A popular alternative is to use an approximate Newton method, which we discuss in Section 3.6.2. Another approach is to use EM, which we discuss in Section 3.6.3.

3.6.2 The FastICA algorithm

3.6.3 Using EM

3.6.4 Other estimation principles *

Chapter 4
Matrix Decomposition

4.1 LU Decompistion

4.2 QR Decomposition

4.3 Eigen value Decomposition

Definition 4.1. An **eigenvector** of an $n \times n$ matrix A is a nonzero vector $\boldsymbol{x}$ such that $A\boldsymbol{x} = \lambda\boldsymbol{x}$ for some scalar λ.A scalar λ is called **eigenvalue** of A if there is a nontrivial solution $\boldsymbol{x}$ of $A\boldsymbol{x} = \lambda\boldsymbol{x}$;such an $\boldsymbol{x}$ is called an eigenvector corresponding to λ[15].

4.4 Singular Value Decomposition

4.4.1 definition

Definition 4.2. Any matrix can be decomposed as follows

$$\underbrace{\boldsymbol{X}}_{N\times D} = \underbrace{\boldsymbol{U}}_{N\times N}\underbrace{\boldsymbol{\Sigma}}_{N\times D}\underbrace{\boldsymbol{V}^T}_{D\times D} \tag{4.1}$$

where $\boldsymbol{U}$ is an $N \times N$ matrix whose columns are orthornormal(so $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$), $\boldsymbol{V}$ is $D \times D$ matrix whose rows and columns are orthonormal (so $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_D$), and $\boldsymbol{\Sigma}$ is a $N \times D$ matrix containing the $r = \min(N, D)$ singular values $\sigma_i \geq 0$ on the main diagonal, with 0s filling the rest of the matrix.

4.4.2 proof

Let A be an $m \times n$ matrix.Then A^TA is symmetric and can be orthogonally diagonalized with eigenvectors.The **singular values** of A are the square root of the eigenvalues of A^TA,denoted by $\sigma_1, \sigma_2, ..., \sigma_n$.That is $\sigma_i = \sqrt{\lambda_i}$ for $1 \leq i \leq n$. The eigenvalues are usually arranged so that

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0 \tag{4.2}$$

Theorem 4.1. *Suppose $\boldsymbol{v}_1, \boldsymbol{v_2}..., \boldsymbol{v_n}$ is an orthogonal basis of $\mathbb{R}^n$ consisting of eigenvector of A^TA,arranged so that the corresponding eigenvalues of A^TA satisfy $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ and suppose A has r nonzero singular values.Then $A\boldsymbol{v_1}, ..., A\boldsymbol{v_r}$ is an orthogonal basis for ColA,and rankA = r.*

Proof. Because $\mathbf{v_i}$ and $\boldsymbol{v_j}$ are orthogonal for $i \neq j$,

$$(A\boldsymbol{v_i})^T(A\boldsymbol{v_j}) = \boldsymbol{v_i}^TA^TA\boldsymbol{v_j} = \boldsymbol{v_i}^T(\lambda_j\boldsymbol{v_j}) = 0 \tag{4.3}$$

We therefore have

$$A\boldsymbol{v_i} = \sigma_i\boldsymbol{u_i} \tag{4.4}$$

For a general vector $\boldsymbol{x}$,since eigenvectors are orthogonal unit vectors,we have

[15] An eigenvalue may be zero

$$\boldsymbol{x} = (\boldsymbol{v_1} \cdot \boldsymbol{x})\boldsymbol{v_1} + (\boldsymbol{v_2} \cdot \boldsymbol{x})\boldsymbol{v_2} + ... + (\boldsymbol{v_n} \cdot \boldsymbol{x})\boldsymbol{v_n} \tag{4.5}$$

This means that

$$M\boldsymbol{x} = (\boldsymbol{v_1} \cdot \boldsymbol{x})M\boldsymbol{v_1} + (\boldsymbol{v_2} \cdot \boldsymbol{x})M\boldsymbol{v_2} + ... + (\boldsymbol{v_n} \cdot \boldsymbol{x})M\boldsymbol{v_n} \tag{4.6}$$

$$M\boldsymbol{x} = (\boldsymbol{v_1} \cdot \boldsymbol{x})\sigma_1\boldsymbol{u_1} + (\boldsymbol{v_2} \cdot \boldsymbol{x})\sigma_2\boldsymbol{u_2} + ... + (\boldsymbol{v_n} \cdot \boldsymbol{x})\sigma_n\boldsymbol{u_n} \tag{4.7}$$

Remember that dot product can be computed using the vector transpose

$$\boldsymbol{v} \cdot \boldsymbol{u} = \boldsymbol{v^T u} \tag{4.8}$$

which leads to

$$M\boldsymbol{x} = \boldsymbol{u_1}\sigma_1\boldsymbol{v_1^T x} + \boldsymbol{u_2}\sigma_2\boldsymbol{v_2^T x} + ... + \boldsymbol{u_n}\sigma_n\boldsymbol{v_n^T x} \tag{4.9}$$

$$M = \boldsymbol{u_1}\sigma_1\boldsymbol{v_1^T} + \boldsymbol{u_2}\sigma_2\boldsymbol{v_2^T} + ... + \boldsymbol{u_n}\sigma_n\boldsymbol{v_n^T} \tag{4.10}$$

And this is usually expressed by writing

$$M = U\Sigma V^T \tag{4.11}$$

As for $\boldsymbol{u_i}$,we have

$$\begin{cases} A^TA\boldsymbol{v_i} & = \lambda\boldsymbol{v_i} \\ A\boldsymbol{v_i} & = \sigma_i\boldsymbol{u_i} \end{cases} \Rightarrow A^T\sigma_i\boldsymbol{u_i} = \lambda\boldsymbol{v_i} \tag{4.12}$$

$$\Rightarrow \sigma_i A^T\boldsymbol{u_i} = \lambda\boldsymbol{v_i} \tag{4.13}$$

$$\Rightarrow AA^T\boldsymbol{u_i} = \lambda_i A\boldsymbol{v_i} \tag{4.14}$$

$$= \lambda_i\sigma_i\boldsymbol{u_i} \tag{4.15}$$

$$\tag{4.16}$$

So we can see that $\boldsymbol{u_i}$ is the eigenvector of symmetric matrix AA^T,and $\boldsymbol{v_i}$ is the eigenvector of symmetric matrix A^TA.In summary,$\boldsymbol{u_i}$ and $\boldsymbol{v_i}$ are the **left-eigenvector** and **right-eigenvectors** of matrix A.

Appendix A
Optimization methods

A.1 Convexity

Definition A.1. (**Convex set**) We say aset $\mathcal{S}$ is convex if for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}$, we have

$$\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \in \mathcal{S}, \forall \lambda \in [0,1] \tag{A.1}$$

Definition A.2. (**Convex function**) A function $f(\boldsymbol{x})$ is called convex if its **epigraph**(the set of points above the function) defines a convex set. Equivalently, a function $f(\boldsymbol{x})$ is called convex if it is defined on a convex set and if, for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}$, and any $\lambda \in [0,1]$, we have

$$f(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2) \leq \lambda f(\boldsymbol{x}_1) + (1-\lambda) f(\boldsymbol{x}_2) \tag{A.2}$$

Definition A.3. A function $f(\boldsymbol{x})$ is said to be **strictly convex** if the inequality is strict

$$f(\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2) < \lambda f(\boldsymbol{x}_1) + (1-\lambda) f(\boldsymbol{x}_2) \tag{A.3}$$

Definition A.4. A function $f(\boldsymbol{x})$ is said to be (strictly) **concave** if $-f(\boldsymbol{x})$ is (strictly) convex.

Theorem A.1. *If $f(x)$ is twice differentiable on $[a,b]$ and $f''(x) \geq 0$ on $[a,b]$ then $f(x)$ is convex on $[a,b]$.*

Proposition A.1. $\log(x)$ *is strictly convex on* $(0,\infty)$.

Intuitively, a (strictly) convex function has a bowl shape, and hence has a unique global minimum x^* corresponding to the bottom of the bowl. Hence its second derivative must be positive everywhere, $\frac{\mathrm{d}^2}{\mathrm{d}x^2} f(x) > 0$. A twice-continuously differentiable, multivariate function f is convex iff its Hessian is positive definite for all $\boldsymbol{x}$. In the machine learning context, the function f often corresponds to the NLL.

Models where the NLL is convex are desirable, since this means we can always find the globally optimal MLE. We will see many examples of this later in the book. However, many models of interest will not have concave likelihoods. In such cases, we will discuss ways to derive locally optimal parameter estimates.

A.2 Gradient descent

A.2.1 Stochastic gradient descent

A.2.2 Batch gradient descent

A.2.3 Line search

The **line search**[1] approach first finds a descent direction along which the objective function f will be reduced and then computes a step size that determines how far $\boldsymbol{x}$ should move along that direction. The descent direction can be computed by various methods, such as gradient descent(Section A.2), Newton's method(Section A.4) and Quasi-Newton method(Section A.5). The step size can be determined either exactly or inexactly.

[1] http://en.wikipedia.org/wiki/Line_search

input : Training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) | i = 1 : N\}$
output: A linear model: $y_i = \boldsymbol{\theta}^T \boldsymbol{x}$
$\boldsymbol{w} \leftarrow 0$; $b \leftarrow 0$; $k \leftarrow 0$;
while *no mistakes made within the for loop* **do**
  for $i \leftarrow 1$ **to** N **do**
    if $y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \leq 0$ **then**
      $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta y_i \boldsymbol{x}_i$;
      $b \leftarrow b + \eta y_i$;
      $k \leftarrow k + 1$;
    end
  end
end

Algorithm 1: Stochastic gradient descent

A.2.4 Momentum term

A.3 Lagrange duality

A.3.1 Primal form

Consider the following, which we'll call the **primal** optimization problem:

$$xyz \tag{A.4}$$

A.3.2 Dual form

A.4 Newton's method

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}_k) + \boldsymbol{g}_k^T (\boldsymbol{x} - \boldsymbol{x}_k) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_k)^T \boldsymbol{H}_k (\boldsymbol{x} - \boldsymbol{x}_k)$$

$$\text{where } \boldsymbol{g}_k \triangleq \boldsymbol{g}(\boldsymbol{x}_k) = f'(\boldsymbol{x}_k), \boldsymbol{H}_k \triangleq \boldsymbol{H}(\boldsymbol{x}_k),$$

$$\boldsymbol{H}(\boldsymbol{x}) \triangleq \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{D \times D} \quad \text{(Hessian matrix)}$$

$$f'(\boldsymbol{x}) = \boldsymbol{g}_k + \boldsymbol{H}_k (\boldsymbol{x} - \boldsymbol{x}_k) = 0 \Rightarrow \tag{A.5}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{H}_k^{-1} \boldsymbol{g}_k \tag{A.6}$$

Initialize $\boldsymbol{x}_0$
while *(!convergency)* **do**
  Evaluate $\boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k)$
  Evaluate $\boldsymbol{H}_k = \nabla^2 f(\boldsymbol{x}_k)$
  $\boldsymbol{d}_k = -\boldsymbol{H}_k^{-1} \boldsymbol{g}_k$
  Use line search to find step size η_k along $\boldsymbol{d}_k$
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k \boldsymbol{d}_k$
end

Algorithm 2: Newtons method for minimizing a strictly convex function

A.5 Quasi-Newton method

From Equation A.5 we can infer out the **quasi-Newton condition** as follows:

$$\begin{aligned} f'(\boldsymbol{x}) - \boldsymbol{g}_k &= \boldsymbol{H}_k(\boldsymbol{x} - \boldsymbol{x}_k) \\ \boldsymbol{g}_{k-1} - \boldsymbol{g}_k &= \boldsymbol{H}_k(\boldsymbol{x}_{k-1} - \boldsymbol{x}_k) \Rightarrow \\ \boldsymbol{g}_k - \boldsymbol{g}_{k-1} &= \boldsymbol{H}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \\ \boldsymbol{g}_{k+1} - \boldsymbol{g}_k &= \boldsymbol{H}_{k+1}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \quad \text{(quasi-Newton condition)} \end{aligned} \tag{A.7}$$

The idea is to replace $\boldsymbol{H}_k^{-1}$ with a approximation $\boldsymbol{B}_k$, which satisfies the following properties:

1. $\boldsymbol{B}_k$ must be symmetric
2. $\boldsymbol{B}_k$ must satisfies the quasi-Newton condition, i.e., $\boldsymbol{g}_{k+1} - \boldsymbol{g}_k = \boldsymbol{B}_{k+1}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)$.
   Let $\boldsymbol{y}_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$, $\boldsymbol{\delta}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$, then
   $$\boldsymbol{B}_{k+1}\boldsymbol{y}_k = \boldsymbol{\delta}_k \tag{A.8}$$
3. Subject to the above, $\boldsymbol{B}_k$ should be as close as possible to $\boldsymbol{B}_{k-1}$.

Note that we did not require that $\boldsymbol{B}_k$ be positive definite. That is because we can show that it must be positive definite if $\boldsymbol{B}_{k-1}$ is. Therefore, as long as the initial Hessian approximation $\boldsymbol{B}_0$ is positive definite, all $\boldsymbol{B}_k$ are, by induction.

A.5.1 DFP

Updating rule:

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \boldsymbol{P}_k + \boldsymbol{Q}_k \tag{A.9}$$

From Equation A.8 we can get

$$\boldsymbol{B}_{k+1}\boldsymbol{y}_k = \boldsymbol{B}_k\boldsymbol{y}_k + \boldsymbol{P}_k\boldsymbol{y}_k + \boldsymbol{Q}_k\boldsymbol{y}_k = \boldsymbol{\delta}_k$$

To make the equation above establish, just let

$$\begin{aligned} \boldsymbol{P}_k\boldsymbol{y}_k &= \boldsymbol{\delta}_k \\ \boldsymbol{Q}_k\boldsymbol{y}_k &= -\boldsymbol{B}_k\boldsymbol{y}_k \end{aligned}$$

In DFP algorithm, $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$ are

$$\boldsymbol{P}_k = \frac{\boldsymbol{\delta}_k\boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T\boldsymbol{y}_k} \tag{A.10}$$

$$\boldsymbol{Q}_k = -\frac{\boldsymbol{B}_k\boldsymbol{y}_k\boldsymbol{y}_k^T\boldsymbol{B}_k}{\boldsymbol{y}_k^T\boldsymbol{B}_k\boldsymbol{y}_k} \tag{A.11}$$

A.5.2 BFGS

Use $\boldsymbol{B}_k$ as a approximation to $\boldsymbol{H}_k$, then the quasi-Newton condition becomes

$$\boldsymbol{B}_{k+1}\boldsymbol{\delta}_k = \boldsymbol{y}_k \tag{A.12}$$

The updating rule is similar to DFP, but $\boldsymbol{P}_k$ and $\boldsymbol{Q}_k$ are different. Let

$$\boldsymbol{P}_k \boldsymbol{\delta}_k = \boldsymbol{y}_k$$
$$\boldsymbol{Q}_k \boldsymbol{\delta}_k = -\boldsymbol{B}_k \boldsymbol{\delta}_k$$

Then

$$\boldsymbol{P}_k = \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{\delta}_k} \tag{A.13}$$

$$\boldsymbol{Q}_k = -\frac{\boldsymbol{B}_k \boldsymbol{\delta}_k \boldsymbol{\delta}_k^T \boldsymbol{B}_k}{\boldsymbol{\delta}_k^T \boldsymbol{B}_k \boldsymbol{\delta}_k} \tag{A.14}$$

A.5.3 Broyden

Broyden's algorithm is a linear combination of DFP and BFGS.

Glossary

Use the template *glossary.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your glossary in the Springer layout.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.