

IMAGE MODELLING AND ESTIMATION

A STATISTICAL APPROACH

FINN LINDGREN

Third edition, 2006-10-20, Chapter 1–5



LUND UNIVERSITY

Centre for Mathematical Sciences
Mathematical Statistics

Contents

Foreword	5
1 Introduction	7
1.1 Why a statistical approach?	7
2 Statistical modelling and estimation	13
2.1 Statistical modelling	13
2.1.1 Conditional probabilities	13
2.1.2 Hierarchical models	14
2.1.3 Covariance matrices	14
2.1.4 Bayesian methods	15
2.2 Object classification	17
2.2.1 Maximum likelihood classification	18
2.2.2 Maximum A Posteriori classification	20
2.2.3 Classification with loss functions	22
2.2.4 Data reduction	22
Principal component analysis	23
PCA for multi-spectral images	23
2.3 Estimation	24
2.3.1 Training set	26
2.3.2 Covariance estimation	26
2.3.3 The EM-algorithm for mixture distributions	27
Exercises	33
3 Distributions, correlation, and filters	35
Random and fixed image elements	35
3.1 Grey level distributions and histograms	36
3.1.1 Histogram transformations	36
3.2 Covariance and correlation	36
3.2.1 The covariance matrix	36
Covariances and linear operations	36

3.2.2	Model covariance and correlation function	37
3.2.3	Homogeneity and isotropy	38
3.2.4	Data covariance and correlation	39
3.3	Fourier transforms and spectrum	40
3.3.1	Fourier transforms in \mathbb{R} and \mathbb{Z}	41
	Fourier transform of covariance functions; power spectral density	41
3.3.2	Sum of random harmonics	43
3.3.3	Fourier transforms in \mathbb{R}^2 and \mathbb{Z}^2	44
	The Fourier transform of a double indexed sequence	44
	Power spectral densities in \mathbb{R}^2 and \mathbb{Z}^2	44
	Power spectrum for an isotropic field	45
3.3.4	Discrete Fourier transforms of 1D and 2D data	46
	Discrete Fourier transform of a data sequence	46
	The Fourier transform of an observed random sequence	48
	Relation between DFT of a data vector and a spectral density	49
	Discrete Fourier transform of 2D data	51
3.4	Linear filters	51
3.4.1	Random elements in linear filters	52
	Smoothing filter	53
	Sharpening (high boost) filter	53
	Correlation matching	54
	Wiener filter	55
	Exercises	56
4	Random fields with Markov structure	59
4.1	Markov random fields	59
4.1.1	Neighbour structures and the Markov condition	60
	Neighbour structures	60
	The Markov condition for random fields	61
4.1.2	Local Markov transition probabilities	62
4.2	Gaussian Markov random fields	63
4.2.1	Basic properties	64
	A simple example	65
4.2.2	Simulation	65
4.2.3	Block conditioning	66
4.2.4	Soft constraints	67
4.2.5	Intrinsic random fields	68
	Increments	68
4.2.6	Parameter estimation	69
4.3	Gibbs distributions	69

4.3.1	The Gibbs distribution	69
4.3.2	Gibbs distributions and Markov random fields	70
	Cliques	70
	Cliques, potentials, and the Gibbs distribution	71
	Examples of local probabilities	73
4.4	Estimation	74
4.4.1	Iterated Conditional Modes	75
4.4.2	Estimation by simulation	77
4.4.3	Parameter estimation	78
	Partial likelihood estimation	78
	Cross validation	78
	Maximum likelihood estimation	78
	Markov chain Monte Carlo ML-estimation	78
	Exercises	79
5	Markov chain Monte Carlo simulation	83
5.1	Introduction	83
	Why simulate?	83
	How to simulate?	83
5.2	MCMC	84
5.2.1	The Metropolis algorithm	85
5.2.2	The Metropolis-Hastings algorithm	88
5.2.3	Gibbs-sampling	90
	Block-update Gibbs-sampling	91
	Parallel Gibbs-sampling	91
5.2.4	MCMC convergence conditions	92
	Exercises	94
6	Shape analysis	95
6.1	Empirical and Bayesian templates	95
6.1.1	Prior distributions	95
6.1.2	Data likelihood and model estimation	96
6.1.3	Templates and global transformations	96
6.2	Landmark templates	98
6.2.1	Vertex perturbations	98
6.2.2	Edge perturbations	99
6.2.3	Global perturbations and simulation	100
6.3	Free-form templates	100
6.3.1	Curves	100
	Snakes	100
	Splines	104

6.3.2	Surfaces	104
6.4	Shapes from images	104
6.4.1	GMRF-Snakes	104
6.5	Estimation algorithm example	105
6.5.1	The data model	105
6.5.2	Construction of the loss function	105
6.5.3	Loss function derivatives	106
6.5.4	Practical optimisation	108
6.5.5	Uncertainty estimation	109
	Exercises	110
7	Warping	111
7.1	Embedded deformation	111
7.1.1	Warping	111
7.1.2	Morphing	111
	Physical deformation models	112
	Bibliography	113
	Wordlist, notation and formulae	115

Foreword

These fragmentary notes on a statistical approach to Image Modelling and Estimation build on the course in Image analysis given at the Centre for Mathematical Sciences at LTH. They represent an attempt to elucidate the fundamental methods in image analysis by statistical considerations, and to expand the arsenal of thoughts, concepts, and methods with stochastic models for image analysis.

The notes were produced during the very first course in Statistical image analysis, given 2001 at the Centre. This means that our own limited knowledge also limits the contents of the notes – hopefully both will grow during the course work. It also means that many planned sections have not been realized in this version.

The reader is assumed to have some knowledge about basic concepts in Image analysis, as they are presented in Gonzalez and Woods (1992). It is also assumed that the reader is somewhat familiar with stochastic processes, in particular correlation and power spectrum descriptions of stationary processes and of Markov processes. Reminders of the basic definitions are given.

Chapter 3 was written by Georg Lindgren; Anders Malmberg contributed many of the examples and exercises. I am grateful for their help during the preparation of the course. I am also grateful for the comments I received from the students who took the course during the spring semester 2001.

Lund, June 2001

Finn Lindgren

Foreword to the second edition

During the first four years of teaching the course in Statistical Image Analysis, numerous improvements have been made to the lectures and computer exercise, but this book has not been updated to reflect the expanded material and notational changes. This edition is an effort to bring the book up to date. The chapter order has also been changed to get a hopefully more logical reading sequence, and thanks to Johan Lindström, the number of exercises has been increased significantly.

Lund, October 2005

Finn Lindgren

Foreword to the third edition

This edition differs from the previous edition mainly in notation. The use of “ ω ” (“omega”) for “the truth” was too easily confused with weights “ w ” (“double-u”). The truth/data-pairs (ω, \mathbf{x}) have therefore been changed to (\mathbf{x}, \mathbf{y}) in this edition. In the previous edition, the chapter on deformable templates was not brought up-to-date with the lectures. This edition contains an effort to reconcile the old and new material about shape analysis, snakes and warping with the lecture notes, in two new chapters.

Lund, October 2006

Finn Lindgren

Chapter 1

Introduction

1.1 Why a statistical approach?

Statistical methods are coming into frequent use in all kinds of image analysis, for describing both the reality the image is expected to depict, and the disturbances the image has been subject to. Acquisition and transmission of images always introduce errors; the image pixels may be set to the wrong intensity level, or the level is “fuzzy” due to low signal power or motion blur. Using *image enhancement* methods, image errors are removed, in order to give the image a more appealing appearance. In these methods, statistical methods are useful for describing the errors, e.g. by specifying probabilities for large deviations from the true pixel values.

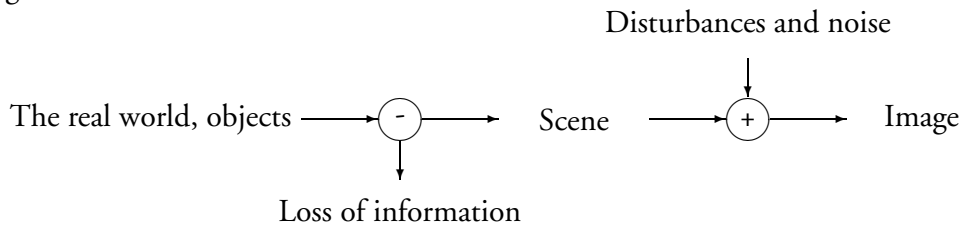
Image restoration methods aim at recreating the original image as faithfully as possible. Here statistical descriptions of the image components again come into play. One method is to use the relations between different image parts. If one part of an image depicts the surface of an ocean, it is more likely that the surrounding areas are water than that they contain land. A distinct local border or edge in an image is likely to continue as an edge in some direction. It may turn or split, with a probability depending on what the image depicts. City blocks often continue in straight lines, or at right angles, while edges in astronomical images of galaxies bend with smooth curves.

Sometimes it is not necessary to make a perfect reconstruction of the image, by instead performing *image interpretation*. In automated document image analysis one can concentrate on identifying symbols such as digits and letters, while an industrial robot can be content with being able to recognise specific objects. When it comes to describing the true contents of an image a statistical framework is also useful. When reading numbers, one can use the fact that the common Arabic digits have specific shapes, and the decoding algorithm does not need to distinguish between letters. Similarly, a text reader for Latin letters does not need to recognise Cyrillic letters.

Generally, an image depicts a *scene*, i.e. the visible part of the real world – the

lions share of reality is hidden behind occluding objects, or lies outside the camera frustum¹. The connection between reality and the scene can also be described using statistical methods.

The basic elements of statistical image analysis can be summarised by the following schematic:



Randomness can affect all links in the schematic. One can, e.g. define a set of possible realities, described by probabilities. In medical cardiographic image analysis this can be used, where the possible variations in the size and shape of human hearts must be considered. In complicated scenes loss of information is common, where interesting details may be hidden behind obstacles and veils. In order to reconstruct the hidden part one must allow different possibilities, and probabilities may come in handy to represent the likelihoods for the different alternatives. Often, the image disturbances are random, e.g. the pixel intensities may exhibit random noise.

As a first example of a random mechanism, look at Figure 1.1. It is a $xx \times yy$ mm piece of paper, where the individual paper fibres obviously have arranged themselves in a random way. It is the randomness in the fibre structure that is of interest here, and not the exact location of an individual fibre. Thus, image analysis here means that we want to extract those properties which are important for the quality and economy of the paper and of the paper making process. Methods for this type of image analysis are correlation Fourier methods, and models for spatial stochastic processes. The paper structure in Figure 1.1 is an example of an *isotropic structure* with the same correlation structure in all directions.

Chapter 2 deals with general statistical methods for modelling and classification of pixels in random images and image elements. The methods are essentially based in likelihood and conditional distributions, *Bayes methods*. Examples of problems are classification of pixel values or image segments according to a multiple characteristic, such as the intensity in several different frequency bands. A Principal Component Analysis (PCA) can then reduce the problem to lower dimension. Another general technique described in this chapter is the EM-algorithm, which can be used for estimation of a mixture of distributions.

Image characterisation by correlation and frequency methods are dealt with in Chapter 3. Here is also the standard Fourier methods for image analysis are re-

¹frustum: a term commonly used in computer graphics, meaning the camera view, specifically the truncated pyramid shape of the view, from Latin, frustum: "piece broken off".

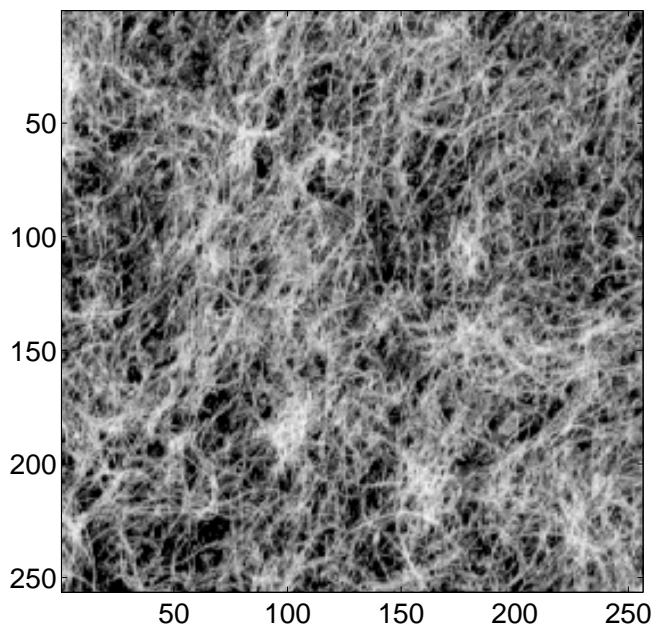


Figure 1.1: A section of a printing paper with fibre structure.

viewed. The standard image filters are also analysed from a statistical viewpoint in that chapter.

Chapter 4 is devoted to image generating stochastic models as a basis for image reconstruction and estimation. Markov random field models are used for images reconstruction. In a Markov random field the distribution of the value at a single point depends on the value at neighbouring points in a very specific way. A neighbour structure is defined, specifying the range of dependence between points. In Figure 1.2 the values in the black points influence the value in the centre.

For two examples of structures generated as Markov random fields, see Figure 1.3

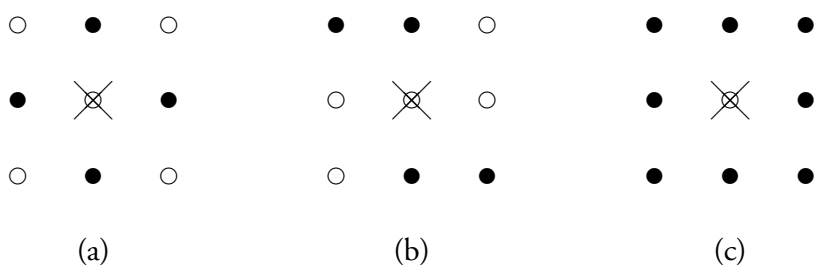


Figure 1.2: Three neighbour structures in a Markov random field. The black points are the neighbours of the centre point.

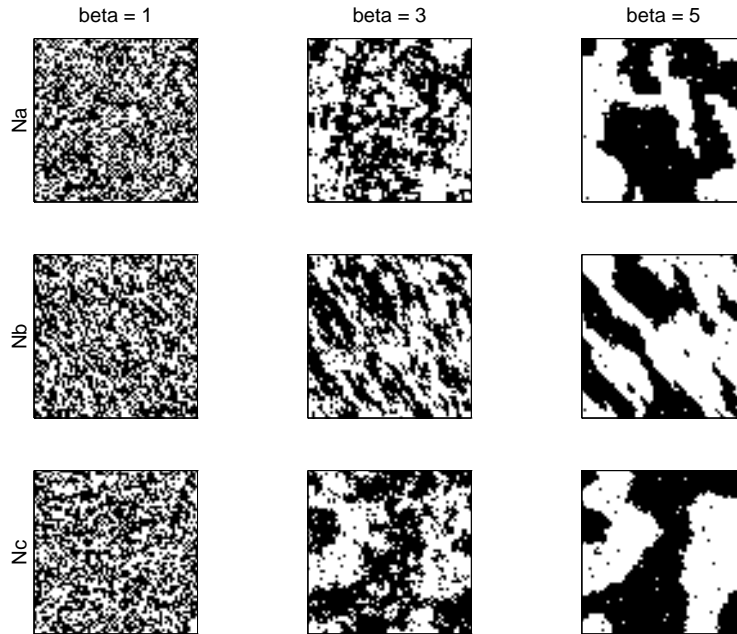


Figure 1.3: Discrete (0/1) Markov random field generated with three different neighbourhood structures and different degrees of dependence. Probability of "black" is given by (1.1).

and Figure 1.4. Figure 1.3 has black and white points only, and the probability of a point being black is equal to

$$\frac{e^{\beta U}}{1 + e^{\beta U}}, \quad (1.1)$$

where U is the number of black neighbours minus the number of white neighbours. The parameter β determines the degree of dependence.

In Figure 1.4, the values have a Gaussian distribution with mean depending on the neighbour points. This is a random field analogue of the autoregressive models used in time series analysis.

Chapter 5 deals with a powerful technique, called *Markov Chain Monte Carlo*, (MCMC) to estimate, or reconstruct, complicated objects from partial observations. The object may be defined as a geometric structure or have any other type of structure. The method includes the use of prior information or assumption in the form of a prior distribution for the possible shapes. Examples of this are reconstruction of three dimensional bodies from two dimensional projections.

In Chapter 6 and 7 we shall deal with statistical shape analysis, with templates and deformations.

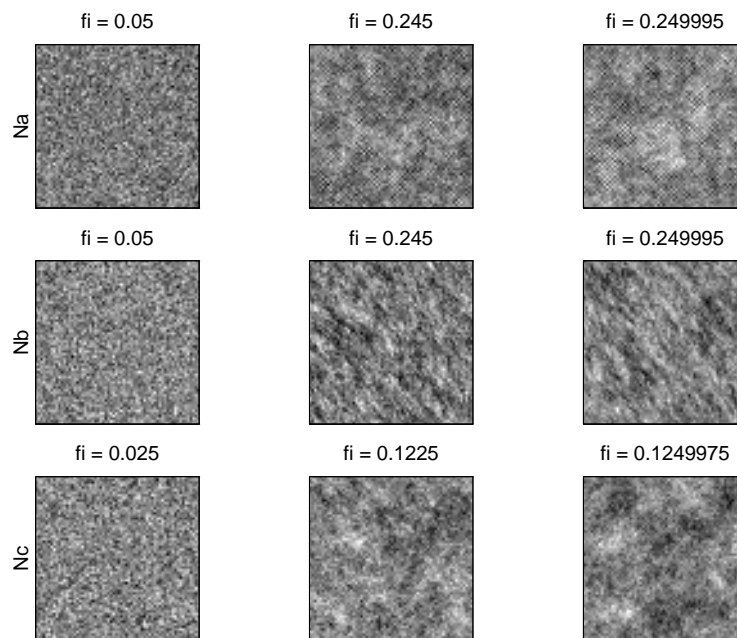


Figure 1.4: Gaussian Markov random fields according to (4.5) with the same neighbourhood structures as in Figure 1.3.

Chapter 2

Statistical modelling and estimation

2.1 Statistical modelling

A common situation is to have a single, joint distribution model for collected data, with unknown parameters, which need to be estimated. At other times, it may be useful to exploit the nature of the process resulting in the data.

When modelling a physical process, it is desirable to use cause and effect relations directly, in order to get a model with components directly connected to real phenomena. Using conditional probabilities, this can often be straightforward in principle. However, the resulting model may be very complex, requiring sophisticated tools to calculate estimates of interesting parameters and properties.

This chapter introduces some modelling and estimation tools, which can be applied to a wide variety of applications.

2.1.1 Conditional probabilities

In statistical image modelling, conditional probabilities and distributions are very useful. They provide a way of designing structured, hierarchical models, which can be difficult to construct as single, multivariate distributions.

Definition 2.1. *The conditional probability of an event A given that event B has occurred, is*

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

□

The most common usage is for probability and density functions,

$$p_{\tilde{\mathbf{y}}|\tilde{\mathbf{x}}=\mathbf{x}}(\mathbf{y}) = \frac{p_{\tilde{\mathbf{y}},\tilde{\mathbf{x}}}(\mathbf{y}, \mathbf{x})}{p_{\tilde{\mathbf{x}}}(\mathbf{x})},$$

where we often use the shorthand notation $p(\mathbf{y}|\mathbf{x}) = p_{\tilde{\mathbf{y}}|\tilde{\mathbf{x}}=\mathbf{x}}(\mathbf{y})$.

2.1.2 Hierarchical models

Using simple conditional distributions, one can build large, complicated hierarchical models.

Example 2.1. Let \tilde{x} be a random quantity, drawn from a population with distribution $N(\mu, \tau^2)$, and let \mathbf{y}_i , $i = 1, \dots, n$ be measurements, with additive noise $N(0, \sigma^2)$, so that \tilde{y}_i given x is $N(x, \sigma^2)$. The joint probability density for $(\tilde{x}, \tilde{y}_1, \dots, \tilde{y}_n)$ is

$$\begin{aligned} p(x, y_1, \dots, y_n) &= p(y_1, \dots, y_n|x)p(x) \\ &= \frac{1}{(2\pi)^{(n+1)/2} \sigma^n \tau} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - x)^2 - \frac{1}{2\tau^2} (x - \mu)^2\right), \end{aligned}$$

whereas the *marginal* distribution for each measurement is $N(\mu, \tau^2 + \sigma^2)$. See Exercise 2.3 for a more thorough investigation of this model. \square

2.1.3 Covariance matrices

Covariances for multidimensional data such as images can be tricky to handle. Computers know little of how to deal with abstract functions, but programs dealing with *matrices* are common. Many operations are therefore simplified if we can use matrix notation.

The elements of the *covariance matrix* Σ for a random vector $\tilde{\mathbf{x}}$ are given by the covariances $\mathbf{C}(\tilde{x}_i, \tilde{x}_j)$, for every pair of components (i, j) . In matrix notation,

$$\Sigma = \mathbf{V}(\tilde{\mathbf{x}}) = \mathbf{C}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) = \mathbf{E}((\tilde{\mathbf{x}} - \mathbf{m})(\tilde{\mathbf{x}} - \mathbf{m})^T).$$

With multidimensional data for each pixel, such as in multi-spectral image acquisition systems, the covariance matrix for each pixel is used to extract discriminating information for each pixel.

One can also construct covariance matrices for entire images, as $\Sigma_{ij} = \mathbf{C}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$, where i and j indicate different (scalar valued) pixels. Covariance matrices for entire images are very large, and can rarely be used directly in numerical computations. For homogeneous covariance functions however, the matrices have a block structure, that can be exploited, often removing the need to store the covariance matrix directly. This is especially important if we want to construct a model for images with vector pixel elements, with dependent pixel values. Chapter 3 deals with the concepts and methods used to describe and analyse image covariances, or *random fields*.

In Chapter 4, random field models with special structure are investigated, with one special case leading to covariance matrices whose *inverses* are sparse matrices, leading to efficient computational methods for estimation and simulation.

In this chapter, all pixels are assumed to be independent, but with covariance matrices describing the relationships between the different colour components of the pixels.

2.1.4 Bayesian methods

In most image analysis applications, the general class of observed images is known. This means that we usually have some idea of what the images may look like, and in particular, what kind of information we should extract from them. Usually we also have some knowledge of the process producing the reality that the images depict. When analysing weather data, we do not need to start from scratch every time; we know how the weather has developed in the past, and may use this information to predict future behaviour.

One way of utilising this *prior information* or *prior belief* is to use *Bayesian methods*¹. The basic principle is to construct a model of several parts:

1. Prior distributions, $p_{\tilde{\mathbf{x}}}(\mathbf{x})$, which indicate how the underlying reality, $\tilde{\mathbf{x}}$, is thought to behave. (Often denoted $\pi(\mathbf{x})$.)
2. Conditional distributions $p(\mathbf{y}|\mathbf{x}) = p_{\tilde{\mathbf{y}}|\tilde{\mathbf{x}}=\mathbf{x}}(\mathbf{y})$, that describe the statistical properties of the measured data \mathbf{y} , given each particular possible reality \mathbf{x} .

A simple consequence of the definition of conditional probabilities, often referred to as *Bayes' rule*, is the relation

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{x})\pi(\mathbf{x})}{p(\mathbf{y})}. \quad (2.1)$$

This formula provides a way to reverse the process of constructing the data model, giving an expression for the distribution of reality given data, the *posterior distribution*.

Note that the *un*-conditional distribution $p(\mathbf{y})$ may be very complicated. In general, we have $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}) \, d\mathbf{x}$ for continuous \mathbf{x} , and $p(\mathbf{y}) = \sum p(\mathbf{y}|\mathbf{x})\pi(\mathbf{x})$ for discrete \mathbf{x} , where the integral and sum are taken over all possible values of \mathbf{x} . Fortunately, in many cases we don't need to explicitly calculate this density.

Example 2.2. (Wet grass) If it has rained today, the grass is wet. If it has not rained today, the grass is dry, unless it rained yesterday, in which case the grass is still wet. The weather is (rather unrealistically) assumed to be independent from day to day.

¹Thomas Bayes, English minister and mathematician, 1702–1761.

We now observe the grass, and learn that it is indeed wet. What is the probability that it has rained today?

This question has no answer, unless we provide some additional information. We therefore access some historical weather data, which tells us that 3 days out of 10 are rainy, on average. This leads us to construct a simple *prior probability* for the event of rain, $\mathbf{P}(\tilde{t} = 1) = 0.3$, where \tilde{t} is a random variable indicating rain today. We let \tilde{w} be the indicator for wet grass, and let \tilde{y} be the indicator for rain yesterday.

First, we calculate the probability of wet grass on any given day:

$$\begin{aligned}\mathbf{P}(\tilde{w} = 1) &= \mathbf{P}(\tilde{w} = 1 | \tilde{t} = 1) \cdot \mathbf{P}(\tilde{t} = 1) \\ &\quad + \mathbf{P}(\tilde{w} = 1 | \tilde{t} = 0, \tilde{y} = 1) \cdot \mathbf{P}(\tilde{t} = 0) \cdot \mathbf{P}(\tilde{y} = 1) \\ &= 1 \cdot 0.3 + 1 \cdot 0.7 \cdot 0.3 \\ &= 0.51.\end{aligned}$$

Now, we are ready to compute the *posterior probability* for rain, given that the grass is wet:

$$\begin{aligned}\mathbf{P}(\tilde{t} = 1 | \tilde{w} = 1) &= \frac{\mathbf{P}(\tilde{t} = 1, \tilde{w} = 1)}{\mathbf{P}(\tilde{w} = 1)} \\ &= \frac{\mathbf{P}(\tilde{w} = 1 | \tilde{t} = 1) \mathbf{P}(\tilde{t} = 1)}{\mathbf{P}(\tilde{w} = 1)} \\ &= \frac{1 \cdot 0.3}{0.51} \approx 0.5882\end{aligned}$$

□

Example 2.3. The most visible difference between Bayesian and “ordinary” statistics is that in Bayesian statistics, almost everything can be, and is, regarded as a random variable. For example, consider the classical problem of estimating μ and σ^2 from a sample x_1, \dots, x_n from a $\mathbf{N}(\mu, \sigma^2)$ -distribution. the joint density is given by

$$p(x_1, \dots, x_n) = \frac{1}{(2\pi)^{n/2} \sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2\right),$$

and the *Maximum likelihood* estimate of μ and σ^2 is obtained by maximising the *likelihood function*, which is simply the density function viewed as a function of the parameters, for fixed sample values. An equivalent Bayesian approach is to assign prior distributions to μ and σ^2 : Let $\tilde{\mu} \in \text{Unif}(-A, A)$ and $\tilde{\sigma}^2 \in \text{Unif}(0, B)$, for some large constants A and B . The prior densities are $\pi(\mu) = \frac{1}{2A} \mathbb{I}\{-A \leq \mu \leq A\}$ and

$\pi(\sigma^2) = \frac{1}{B}\mathbb{I}\{0 \leq \sigma^2 \leq B\}$. The conditional distribution of (μ, σ^2) given x_1, \dots, x_n is

$$p(\mu, \sigma^2 | x_1, \dots, x_n) = \frac{p(x_1, \dots, x_n | \mu, \sigma^2) \pi(\mu) \pi(\sigma^2)}{p(x_1, \dots, x_n)} \\ \propto \frac{\mathbb{I}\{-A \leq \mu \leq A\} \mathbb{I}\{0 \leq \sigma^2 \leq B\}}{2AB(2\pi)^{n/2} \sigma^n} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2\right).$$

The *Maximum posterior estimate* of μ and σ^2 is obtained by maximising the density with respect to μ and σ^2 . If A and B are large, the resulting estimates will be identical to the Maximum likelihood estimates. In practical applications, other prior distributions are used for μ and σ^2 , resulting in different estimates. \square

Example 2.4. (Example 2.1 continued) The conditional distribution of the unknown quantity given the measurements is

$$(x | y_1, \dots, y_n) \in \mathcal{N}\left(\left(\tau^2 \sum_{i=1}^n y_i / n + \mu \sigma^2 / n\right) / (\tau^2 + \sigma^2 / n), \frac{\tau^2 \sigma^2 / n}{\tau^2 + \sigma^2 / n}\right).$$

See Exercise 2.3 for a proof. \square

2.2 Object classification

It is assumed that we have a set of K object categories, labelled $\{1, \dots, K\}$, and that every data point is an observation corresponding to one object. The data points can be each individual pixel, an entire image, or other data. By object we refer to some especially interesting thing or pattern. The purpose of *statistical pattern recognition* is to recognise and determine which of the object types is visible in the data.

To aid in the pattern recognition, a vector of observations, \mathbf{x} , is available. It may be the vector of all image pixel values, or the combined intensities of pixels from images acquired using different wavelength filters. It can also be pixels from several images taken at different points in time, or from different viewing angles. Often, we perform an initial *data reduction* step, such as *principal component analysis* (see 2.2.4) and/or feature extraction, so that \mathbf{y} is a vector of much lower dimension than the initial images.

The observation vector is assumed to have been generated by a random mechanism depending on the true pattern. This means that we view the data vector $\mathbf{y} = (y_1, \dots, y_n)^T$ as an observation of a multidimensional stochastic variable $\tilde{\mathbf{y}} = (\tilde{y}_1, \dots, \tilde{y}_n)^T$. (If the observations were perfect replicas of the true pattern we would naturally only need to trivially determine which pattern was observed.) The observations may take *discrete* or *continuous* values.

Example 2.5. A satellite with d image channels takes photographs of the surface of the Earth, using e.g. infrared and visible light filters. The goal is to classify the ground in each pixel. If we analyse each pixel separately, the dimension of each data point is d . The data vector elements are now *themselves* vectors. Depending on the circumstances, we may want write the data as a 3-dimensional matrix, with element x_{ijl} holding image pixel (i, j) from image channel l . \square

Throughout this section, up to Section 2.2.4, we assume that we have determined a suitable set of features that are to be used for classifying independent observations into different categories. We also assume that the distributions of data points corresponding to each model class $k = 1, \dots, K$ are known, including parameters such as expected values and covariances. Section 2.3 is devoted to methods for estimating these distributions.

Remark 2.1: Assuming that the true pixel class x_i and the data for all other pixels are independent is not always realistic. Chapter 4 and onward deal with models and methods where the true underlying structure have dependent random components. \square

In the examples provided, we will make use of the multivariate Normal (or Gaussian) distribution, with probability density

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

where $\boldsymbol{\mu}$ is the expected value, Σ is the covariance matrix, d is the dimension of the data vectors, and $|\Sigma|$ is the determinant of Σ . The covariance matrix must be invertible and positive definite.

2.2.1 Maximum likelihood classification

Viewing the true model classes, $\mathbf{x} = \{x_1, \dots, x_n\}$, as parameters to be estimated, we construct the *likelihood function*

$$L(\mathbf{x}; \mathbf{y}) = p(\mathbf{y}; \mathbf{x}) = \prod_{i=1}^n p(\mathbf{y}_i; x_i),$$

where $p(\mathbf{y}_i; x_i)$ is the probability density for pixel i , given that it belongs to the model determined by x_i .

We now proceed as in ordinary statistical parameter estimation, by computing the *maximum likelihood (ML)* estimate of the parameters $\mathbf{x} = \{x_1, \dots, x_n\}$. Maximising each term of the likelihood function, we see that we should choose the model x_i for each \mathbf{y}_i that gives the largest value of $p(\mathbf{y}_i; x_i)$.

Example 2.6. (Linear discriminant analysis) Assume that we want to distinguish between data from two Normal distributions, with expected values $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and common covariance matrix $\boldsymbol{\Sigma}$. For each data point, \mathbf{y}_i , we choose model k_2 over k_1 when

$$\frac{p(\mathbf{y}_i; x_i = k_2)}{p(\mathbf{y}_i; x_i = k_1)} > 1.$$

Taking the logarithm of the ratio, we obtain

$$\begin{aligned} \log \left(\frac{p(\mathbf{y}_i; x_i = k_2)}{p(\mathbf{y}_i; x_i = k_1)} \right) &= -\frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_2) + \frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_1) \\ &= \dots = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1} \left(\mathbf{y}_i - \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2} \right), \end{aligned}$$

so that the discriminating border is given by a straight line through $(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$, with direction perpendicular to $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$.

Figure 2.1 shows contour curves drawn at half the maximum height of the probability density, and the linear discrimination line, for two Normal densities with $\boldsymbol{\mu}_1 = (7, 7)$, $\boldsymbol{\mu}_2 = (3, 3)$, and

$$\boldsymbol{\Sigma} = \begin{pmatrix} 9 & -1.5 \\ -1.5 & 4 \end{pmatrix}.$$

□

Example 2.7. (Quadratic discriminant analysis) We modify the assumptions in Example 2.6, so that each model has its own covariance matrix, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ respectively. The criterion for choosing model k_2 now becomes

$$\begin{aligned} 0 &< \log \frac{p(\mathbf{y}_i; x_i = k_2)}{p(\mathbf{y}_i; x_i = k_1)} \\ &= \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|} - \frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_2^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_2) + \frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_1^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_1) \\ &= \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|} + \frac{1}{2} \mathbf{y}_i^\top (\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_2^{-1}) \mathbf{y}_i + (\boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}_2^{-1} - \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}_1^{-1}) \mathbf{y}_i \\ &\quad + \frac{1}{2} \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 - \frac{1}{2} \boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2. \end{aligned}$$

This means that the discriminating border is a conic section, see Figure 2.2. □

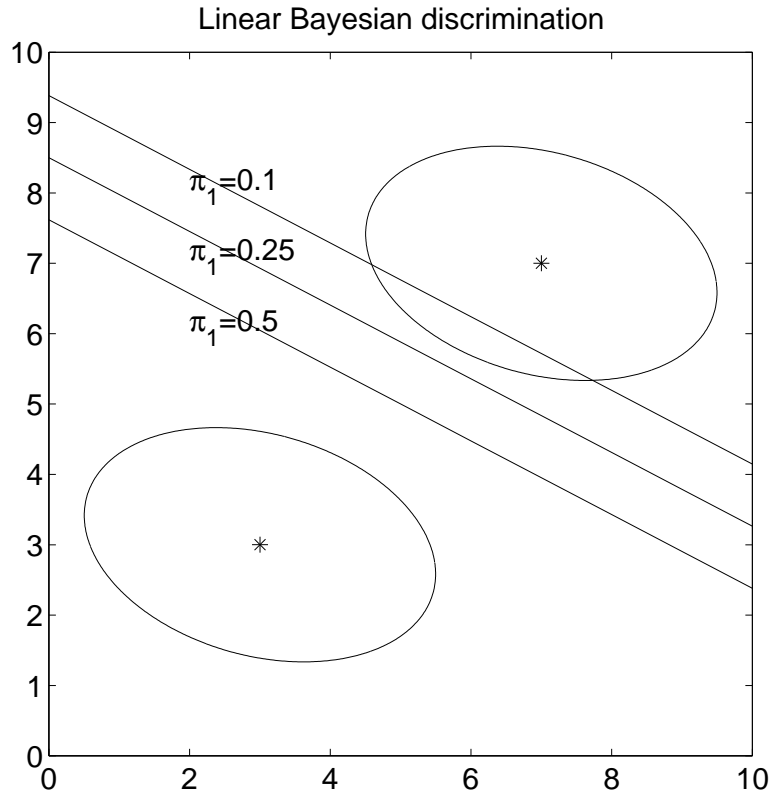


Figure 2.1: Contour curves at half the maximum pdf-height for two Normal densities with common covariance, and the discriminating border, for pure likelihood classification as in Example 2.6, and for Bayesian classification, as in Example 2.8. The pure likelihood classification is equivalent to the Bayesian variant with $\pi_1 = 0.5$.

2.2.2 Maximum A Posteriori classification

Pure likelihood classification is rarely a method to be recommended, due to the fact that no account is taken for how common objects from different classes are. Adopting a Bayesian perspective, however, opens the doors to better methods.

Assume a prior distribution for the relative frequencies of the different models, with probabilities $\pi_k = \mathbf{P}(\tilde{x} = k)$. Assuming that the π_k -probabilities are known (e.g. from a training set, see Section 2.3.1), we obtain the posterior probability function

$$\mathbf{P}(\tilde{\mathbf{x}} = \mathbf{x}|\mathbf{y}) = \prod_{i=1}^n \mathbf{P}(\tilde{x}_i = x_i|\mathbf{y}_i) = \prod_{i=1}^n p(x_i|\mathbf{y}_i) = \prod_{i=1}^n \frac{p(\mathbf{y}_i|x_i)\pi_{x_i}}{p(\mathbf{y}_i)}.$$

The counterpart to ML-estimation is now to simply maximise each factor, by choosing the x_i that maximises $p(\mathbf{y}_i|x_i)\pi_{x_i}$. This technique, *Maximum A Posteriori (MAP)*

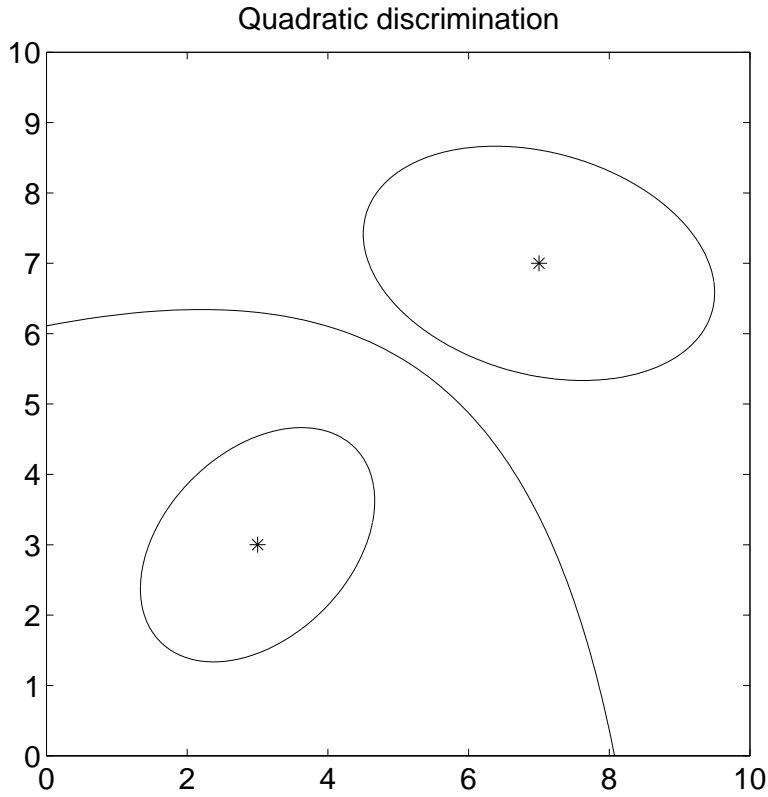


Figure 2.2: Contour curves for two Normal densities with different covariances, and the discriminating border, for pure likelihood classification.

estimation, can be used when the probability of finding the correct answer should be maximised, without considering what happens if an incorrect answer is given. However, the posterior distribution can also be used in other ways, such as in the method described in Section 2.2.3.

Example 2.8. (Linear discriminant analysis, cont.) With the Bayesian approach, the classification criterion in Example 2.6 becomes; Choose model k_2 if

$$\begin{aligned}
 0 &< \log \frac{p(k_2|\mathbf{y}_i)}{p(k_1|\mathbf{y}_i)} \\
 &= (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1} \left(\mathbf{y}_i - \frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2} \right) + \log \frac{\pi_2}{\pi_1} \\
 &= (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1} \left(\mathbf{y}_i - \frac{w_1 \boldsymbol{\mu}_1 + w_2 \boldsymbol{\mu}_2}{w_1 + w_2} \right)
 \end{aligned}$$

where $w_1 = 2 \log(\pi_2/\pi_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ and $w_2 = 2 \log(\pi_1/\pi_2) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$. The discriminating border is still a straight line, but in a

different location; see Figure 2.1 for an example with $\boldsymbol{\mu}_1 = (7, 7)$, $\boldsymbol{\mu}_2 = (3, 3)$, and

$$\boldsymbol{\Sigma} = \begin{pmatrix} 9 & -1.5 \\ -1.5 & 4 \end{pmatrix}.$$

□

2.2.3 Classification with loss functions

Minimum posterior expected loss means that every combination of true (x^0) and estimated (\hat{x}) patterns is associated with a cost, or loss, $C(\hat{x}; x^0)$. The classifications are then performed by minimising $\mathbf{E}(C(\mathbf{x}; \tilde{\mathbf{x}}^0) | \mathbf{y})$ with respect to \mathbf{x} . If the observations are independent we minimise

$$\mathbf{E}(C(x_i; \tilde{x}_i^0) | \mathbf{y}_i) = \sum_{k=1}^K C(x_i; k) \mathbf{P}(\tilde{x}_i^0 = k | \mathbf{y}_i)$$

for every data point i . This technique is suitable when some types of error are more dangerous or costly than other errors.

Example 2.9. (A special loss function) Let

$$C(x; x^0) = \begin{cases} 0, & \text{if } x = x^0, \\ 1, & \text{if } x \neq x^0, \end{cases}$$

so that the loss is 0 for correct classifications, and 1 for errors. Then,

$$\mathbf{E}(C(x; \tilde{x}^0) | \mathbf{y}) = \sum_{k=1}^K C(x; k) \mathbf{P}(\tilde{x}^0 = k | \mathbf{y}) = 1 - \mathbf{P}(\tilde{x}^0 = x | \mathbf{y}).$$

Minimisation of this particular choice of loss function yields the same result as maximisation of the posterior probability function. This means that the MAP estimate can be seen as a special case of the “minimum posterior expected loss” estimate. □

2.2.4 Data reduction

In multivariate (multidimensional) data, the *information* does not always use all possible degrees of freedom. E.g. when measuring the circumference of a rectangle, we don’t need to store the individual side-lengths, only their sum. In higher dimensions, it may be difficult to manually determine a suitable subset, or suitable functions, of the variables to use in further analysis.

The issue is how to perform the *feature selection* required to obtain the vectors with as much information as possible about the objects to be recognised; See also Gonzalez and Woods (1992), Section 9.2, pp. 574–579.

Principal component analysis

Principal component analysis (PCA) is an automatic method for finding optimal linear functions of the data, with the property that the resulting data functionals are *uncorrelated*.

We make use of the information encoded in the covariance matrix Σ for the joint distribution of the data vectors.

Definition 2.2. (Principal component analysis, PCA) A principal component is a linear combination of data,² $\sum_l a_l x_l$, which differs significantly for different patterns or objects.

In order to find the principal components of the observation vector \mathbf{x} , we diagonalise the covariance matrix Σ ,

$$\Sigma = \mathbf{P}\Lambda\mathbf{P}^T,$$

with a diagonal matrix Λ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \lambda_n$, and an orthogonal matrix \mathbf{P} . The rows and columns can be ordered so that λ_1 is the largest eigenvalue. Since covariance matrices are symmetric and positive-semidefinite, such a diagonalisation always exists, with all eigenvalues non-negative.

By the first principal component is meant the scalar $\mathbf{P}_1^T \mathbf{x}$, where \mathbf{P}_1 is the vector in the orthogonal matrix \mathbf{P} corresponding to the largest eigenvalue, λ_1 , of the diagonal matrix Λ . One then continues with the second principal component, $\mathbf{P}_2^T \mathbf{x}$, etc. \square

The variance of $\mathbf{P}_l^T \mathbf{x}$, or, more precisely, of the random variable $\mathbf{P}_l^T \tilde{\mathbf{x}}$ of which $\mathbf{P}_l^T \mathbf{x}$ is an observation, is

$$\mathbf{V}(\mathbf{P}_l^T \tilde{\mathbf{x}}) = \mathbf{P}_l^T (\mathbf{P}\Lambda\mathbf{P}^T) \mathbf{P}_l = \lambda_l,$$

so that $\mathbf{P}_1^T \mathbf{x}$ is the linear combination of x -values displaying the largest variance.

Since \mathbf{P} is an orthogonal matrix, the covariance $\mathbf{C}(\mathbf{P}_j^T \tilde{\mathbf{x}}, \mathbf{P}_l^T \tilde{\mathbf{x}}) = 0$ for all $j \neq l$, i.e. the principal component vectors are uncorrelated.

The principal components corresponding to the largest eigenvalues are used in the further data analysis.

PCA for multi-spectral images

Example 2.10. The Landsat satellite (<http://landsat.gsfc.nasa.gov/>) takes photographs of the surface of the Earth using a number of filters, such as for infrared and visible light. We obtain an image using d such image channels, and the goal is to classify the ground in each pixel. The data are given as a 3-dimensional matrix \mathbf{x} ,

²In this section, \mathbf{x} and \mathbf{y} both denote data.

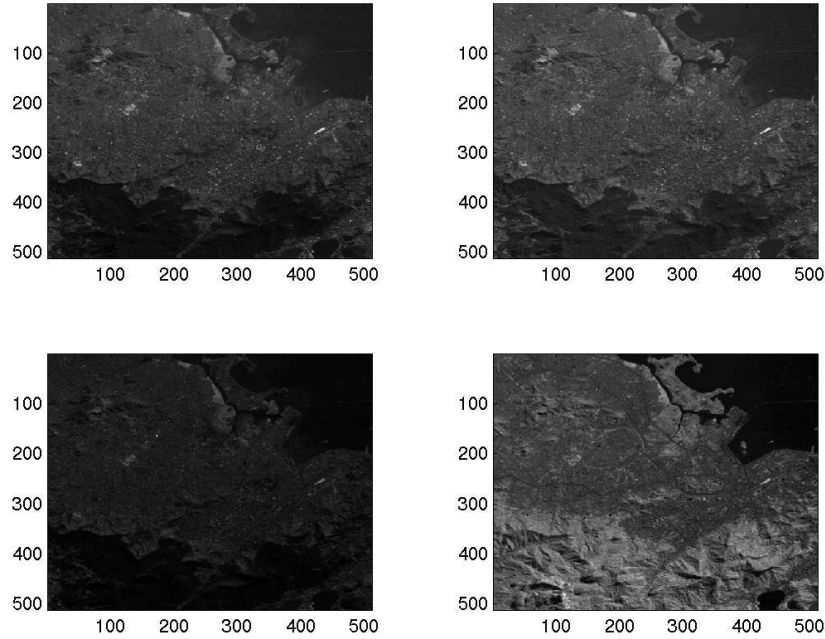


Figure 2.3: Some Landsat images of Rio de Janeiro with three visible (blue, green, and red) and one non-visible (infrared) light channel (out of seven available).

with element x_{ijl} holding image pixel (i, j) from image channel l . We view this as a set of independent d -dimensional vectors \mathbf{x}_{ij} , with covariance matrix Σ .

We compute the principal components for all pixels, obtaining new vectors $\mathbf{y}_{ij} = \mathbf{P}^T \mathbf{x}_{ij}$, with $y_{ijl} = \mathbf{P}_l^T \mathbf{x}_{ij}$.

□

2.3 Estimation

Most data models depend on some parameters. Sometimes they are assumed to be known from previous experience, but often they need to be estimated from the data itself.

If the functions are to be estimated, a *training set* of images is needed, where the object types are known. The prior distributions indicate how often the various object types occur in a certain population (set) of images. These probabilities can be estimated by counting the proportion of images containing each object type, and use this as an estimate of $\mathbf{P}(\tilde{x} = k)$. An alternative method is to assume that all objects are equally likely, i.e. that $\mathbf{P}(\tilde{x} = k) = 1/K$ for all k .

When using prior probabilities, caution is needed. It is often difficult to determine which population a certain image belongs to, and the proportions in the

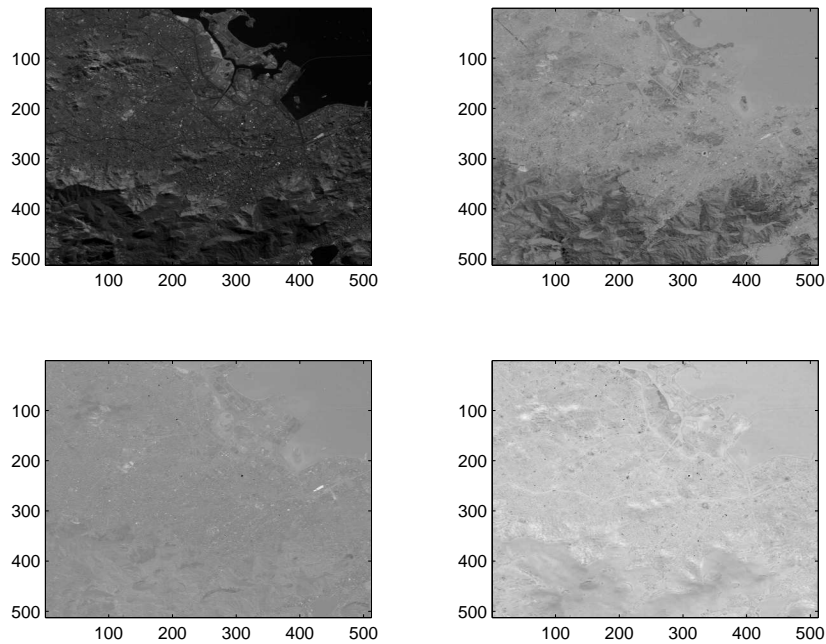


Figure 2.4: Rio de Janeiro as portrayed by the first four principal components based on images from seven images in different wavelength bands.

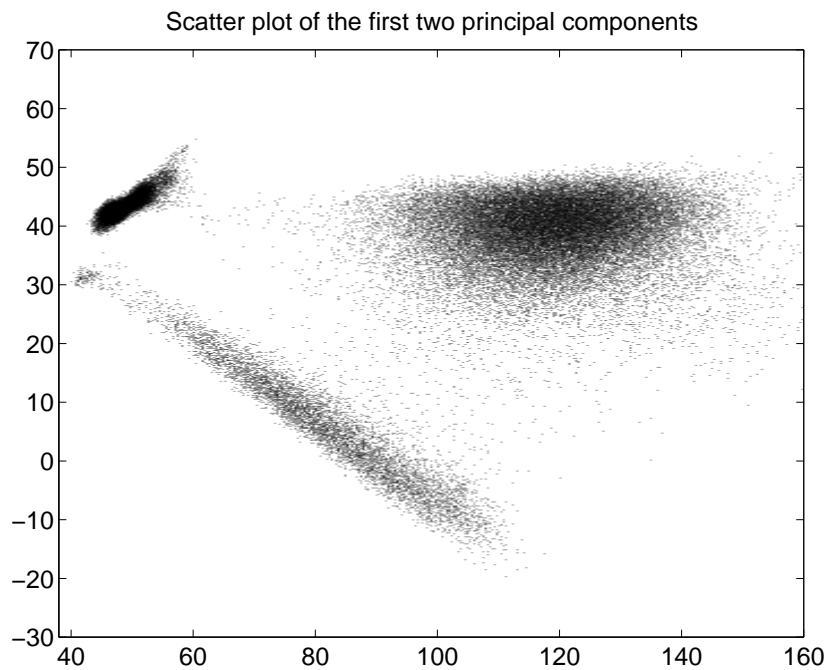


Figure 2.5: Scatter plot of the first two principal components of Rio de Janeiro, with three almost distinct sub-distributions.

training set may be different from the true proportions. Furthermore, it is desirable to be able to classify the image without regard for a particular population. It may be important to be able to discover very rare objects, or even new objects, that are not present in the training set.

The prior probabilities are equal to the expected proportions of different object types in the images. The likelihood-functions tells which data vectors the different objects can generate, and with how large probabilities, e.g. how smooth the image of a snow field can be, or the expected intensity variation in images showing a forest with a particular kinds of trees.

The prior probability functions and the likelihood-functions both often use parameters, θ , which need to be estimated, i.e. $\mathbf{P}(\tilde{x} = k) = \mathbf{P}(\tilde{x} = k|\theta)$ and $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x}, \theta)$.

2.3.1 Training set

The classical approach to determining appropriate prior distributions is to obtain a *training set* of data. This typically mean using more expensive methods for data acquisition, that give information on the true object classification, linked to the ordinary data. An example might be to embark on an expedition to remote locations and chart vegetation, to provide correct classifications to images obtained by a satellite. The information obtained in this manner is then used for estimation of object proportions, $\mathbf{P}(\tilde{x} = k)$, and of measurement distributions, $p(\mathbf{y}|x)$.

2.3.2 Covariance estimation

Given a set of observations, $(\mathbf{y}_1, \dots, \mathbf{y}_n)$, of d -dimensional vectors, how may we estimate the covariance matrix, Σ , of the corresponding random variable, $\tilde{\mathbf{y}}$?

With $\boldsymbol{\mu} = \mathbf{E}(\tilde{\mathbf{y}})$, we have $\Sigma = \mathbf{E}((\tilde{\mathbf{y}} - \boldsymbol{\mu})(\tilde{\mathbf{y}} - \boldsymbol{\mu})^\top)$. Introducing $\bar{\mathbf{y}} = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i$ and $\mathbf{y}_i^* = \mathbf{y}_i - \bar{\mathbf{y}}$, each product $\mathbf{y}_i^* \mathbf{y}_i^{*\top}$ is a (biased) estimate of Σ , leading to a final (biased) estimate $\frac{1}{n} \sum_{i=1}^n \mathbf{y}_i^* \mathbf{y}_i^{*\top}$. For the Normal distribution, this estimate is the same as the maximum likelihood estimate.

Example 2.11. Given a Landsat image with d channels, we want to estimate the expected value and the covariance matrix in the d -dimensional distribution of $\tilde{\mathbf{y}}_{ij}$. With an image size of $m \times n$, we obtain

$$\hat{\boldsymbol{\mu}} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \mathbf{y}_{ij}, \quad \text{and} \quad \hat{\Sigma} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (\mathbf{y}_{ij} - \hat{\boldsymbol{\mu}})(\mathbf{y}_{ij} - \hat{\boldsymbol{\mu}})^\top.$$

□

2.3.3 The EM-algorithm for mixture distributions

In Section 2.2, we assumed that all distribution parameters were known. We will now relax this requirement, to allow more realistic modelling.

We assume that the distribution *type* for each model class is known, but that the distributions parameters, $\Theta = \{\Theta_1, \dots, \Theta_K\}$, are unknown. The relative frequencies π are also unknown. Thus, we need to estimate $\Psi = \{\pi, \Theta\}$, using data $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$.

Here, notation indicates that each model, $k = 1, \dots, K$, has its own unique parameter set Θ_k . However, in general this is not absolutely necessary, and there may be common elements in different Θ_k . A natural example is when different models have the same variance or covariance structure.

When each data point is sampled from a randomly selected model class, the resulting density is a mixture,

$$p(\mathbf{y}|\Psi) = \prod_{i=1}^n \sum_{k=1}^K p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k) \pi_k,$$

which can be very difficult to handle analytically. However, if the true class x_i for each pixel was known, the likelihood would have the much nicer form

$$p(\mathbf{y}, \mathbf{x}|\Psi) = \prod_{i=1}^n p(\mathbf{y}_i | x_i, \Theta_{x_i}) \pi_{x_i}.$$

The *EM-algorithm* is a general method for calculating maximum likelihood estimators by augmenting (extending) the observed data with unknown quantities, which can often yield more easily handled likelihood expressions.

Definition 2.3. (The EM-algorithm) *Augment the data set \mathbf{y} with the random variables for unknown quantities, $\tilde{\mathbf{x}}$. The joint, or complete, variable $(\mathbf{y}, \tilde{\mathbf{x}})$ is an extended version of \mathbf{y} . Let $L(\Psi | \mathbf{y}, \mathbf{x}) = p(\mathbf{y}, \mathbf{x} | \Psi)$ be the joint likelihood for (\mathbf{y}, \mathbf{x}) , Given an initial parameter estimate $\Psi^{(0)}$, iterate the following steps.*

1. *E-step: Evaluate $Q(\Psi, \Psi^{(t)}) = \mathbf{E}(\log p(\mathbf{y}, \tilde{\mathbf{x}} | \Psi) | \mathbf{y}, \Psi^{(t)})$, i.e. with the expectation taken over the conditional (or posterior) distribution for \mathbf{x} given the observed data \mathbf{y} and the old parameter estimate $\Psi^{(t)}$.*
2. *M-step: Find the $\Psi = \Psi^{(t+1)}$ which maximises $Q(\Psi, \Psi^{(t)})$.*

□

It can be shown that the original likelihood $L(\Psi | \mathbf{y})$ is non-decreasing in each step.

In our mixture model, the variable $(\mathbf{y}, \tilde{\mathbf{x}})$ is a version of \mathbf{y} complete with class for each \mathbf{y}_i . The log-likelihood is given by

$$\log p(\mathbf{y}, \mathbf{x} | \Psi) = \sum_{i=1}^n (\log p(\mathbf{y}_i | x_i, \Theta_{x_i}) + \log(\pi_{x_i})),$$

and the posterior distribution for $\mathbf{x} | \mathbf{y}$, $\Psi^{(t)}$ is determined by the posterior probabilities

$$\begin{aligned} p_{i,k}^{(t)} &= \mathbf{P}(\tilde{x}_i = k | \mathbf{y}, \Psi^{(t)}) \\ &= \frac{p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k^{(t)}) \pi_k^{(t)}}{p(\mathbf{y}_i | \Psi^{(t)})} \\ &= \frac{p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k^{(t)}) \pi_k^{(t)}}{\sum_{j=1}^K p(\mathbf{y}_i | \tilde{x}_i = j, \Theta_j^{(t)}) \pi_j^{(t)}}. \end{aligned}$$

We can now calculate Q , and obtain

$$\begin{aligned} Q(\Psi, \Psi^{(t)}) &= \sum_{i=1}^n \sum_{k=1}^K (\log p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k) + \log(\pi_k)) p_{i,k}^{(t)} \quad (2.2) \\ &= \sum_{k=1}^K \sum_{i=1}^n p_{i,k}^{(t)} \log p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k) + \sum_{k=1}^K \log(\pi_k) \sum_{i=1}^n p_{i,k}^{(t)} \end{aligned}$$

Maximising $Q(\Psi, \Psi^{(t)})$ with respect to π , under the conditions $\sum_{k=1}^K \pi_k = 1$ and $\pi_k \geq 0$, we find (see Exercise 2.7) new probabilities

$$\pi_k^{(t+1)} = \frac{1}{n} \sum_{i=1}^n p_{i,k}^{(t)},$$

for $k = 1, \dots, K$. For notational convenience, we introduce the *apparent sample size* for each model, denoted $n_k^{(t+1)} = n \pi_k^{(t+1)}$.

To find the new estimates of the model parameters one should maximise $Q(\Psi, \Psi^{(t)})$ as a function of the Θ_k . If we assume separate parameters for each model this is done by maximising the sum for each k separately, i.e. the new distribution parameters $\Theta^{(t+1)}$ are found by maximising

$$\sum_{i=1}^n p_{i,k}^{(t)} \log p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k)$$

with respect to Θ_k , for $k = 1, \dots, K$.

Example 2.12. We have a set of one-dimensional data, $\mathbf{y} = \{y_1, \dots, y_n\}$, which is to be classified into K different categories, each corresponding to a Gaussian distribution with unknown mean and variance. Let μ_k and σ_k^2 be the mean and variance for category k . Recall that

$$p(y_i | \tilde{x}_i = k, \Theta_k^{(t)}) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(y_i - \mu_k)^2}{2\sigma_k^2}\right).$$

Maximising

$$\begin{aligned} & \sum_{i=1}^n p_{i,k}^{(t)} \log p(y_i | \tilde{x}_i = k, \Theta_k) \\ &= \sum_{i=1}^n p_{i,k}^{(t)} \left(-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma_k^2 - \frac{(y_i - \mu_k)^2}{2\sigma_k^2} \right) \end{aligned}$$

with respect to μ_k and σ_k^2 yields new estimates

$$\mu_k^{(t+1)} = \frac{1}{n_k^{(t+1)}} \sum_{i=1}^n p_{i,k}^{(t)} y_i$$

and

$$(\sigma_k^{(t+1)})^2 = \frac{1}{n_k^{(t+1)}} \sum_{i=1}^n p_{i,k}^{(t)} (y_i - \mu_k^{(t+1)})^2.$$

□

Example 2.13. We have a set of d -dimensional data, $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$, which is to be classified into K different categories, each corresponding to a Gaussian distribution with unknown mean and variance. Let $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ be the mean and covariance matrix for category k . Recall that

$$p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k)\right).$$

Maximising

$$\begin{aligned} & \sum_{i=1}^n p_{i,k}^{(t)} \log p(\mathbf{y}_i | \tilde{x}_i = k, \Theta_k) \\ &= \sum_{i=1}^n p_{i,k}^{(t)} \left(-\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k) \right) \end{aligned}$$

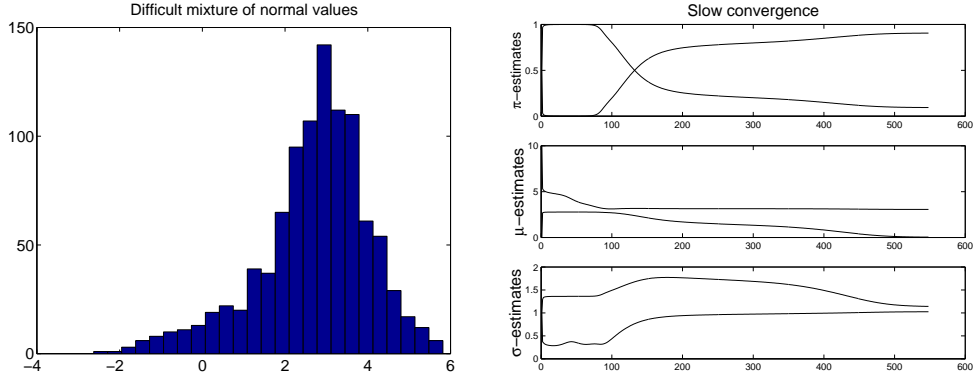


Figure 2.6: Slow but safe convergence of parameter estimates in a difficult normal mixture with $n = 1000$ observations. True parameters: $\boldsymbol{\pi} = (0.1, 0.9)$, $\boldsymbol{\mu} = (0, 3)$, $\boldsymbol{\sigma} = (1, 1)$. Starting values $\boldsymbol{\pi} = (0.5, 0.5)$, $\boldsymbol{\mu} = (0, 10)$, $\boldsymbol{\sigma} = (1, 1)$. The algorithm stayed at a stable value with less than 0.001 variation after 548 steps.

with respect to $\boldsymbol{\mu}_k$ yields a new estimates

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{1}{n_k^{(t+1)}} \sum_{i=1}^n p_{i,k}^{(t)} \mathbf{y}_i.$$

Since the expression to be maximised contains the determinant and the inverse of the covariance matrix, it is difficult to construct a covariance estimate by differentiating with respect to the elements of $\boldsymbol{\Sigma}$. However, it can be shown that

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{1}{n_k^{(t+1)}} \sum_{i=1}^n p_{i,k}^{(t)} (\mathbf{y}_i - \boldsymbol{\mu}_k^{(t+1)})(\mathbf{y}_i - \boldsymbol{\mu}_k^{(t+1)})^\top,$$

which is a natural generalisation of the one-dimensional variance estimate, is in fact the true maximiser. \square

Example 2.14. As an example of a mixture where two models have a common parameter, we change Example 2.12 and assume that the variances are the same in the two models, $\sigma_1^2 = \sigma_2^2 = \sigma^2$. In that example they were actually equal, but we did not use that in the estimation algorithm.

The expression to be maximised in the M-step is now

$$\sum_{k=1}^2 \sum_{i=1}^n p_{i,k}^{(t)} \left(-\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma^2 - \frac{(y_i - \mu_k)^2}{2\sigma^2} \right),$$

which gives the iteration

$$\begin{aligned} (\sigma^{(t+1)})^2 &= \frac{1}{n_1^{(t+1)} + n_2^{(t+1)}} \sum_{k=1}^2 \sum_{i=1}^n p_{i,k}^{(t)} (y_i - \mu_k^{(t+1)})^2 \\ &= \frac{n_1^{(t+1)} (\sigma_1^{(t+1)})^2 + n_2^{(t+1)} (\sigma_2^{(t+1)})^2}{n_1^{(t+1)} + n_2^{(t+1)}}, \end{aligned}$$

in complete analogy with pooled variance estimates in two sample statistics. \square

Example 2.15. In the Rio de Janeiro example, the first two principal components seemed to fall in three almost distinct classes. By assuming three Normal distributions as an appropriate model, estimating the parameters by the EM-algorithm, we can assign each pixel to one of the three distributions by the (Bayesian) quadratic discriminant technique. (Obviously, Normal distributions are not perfect in this case, but they work reasonably well for classification purposes.) Figure 2.7 is the same as Figure 2.5 but now we have drawn the discrimination curves between the three regions. These curves are pieces of conical sections.

Finally, Figure 2.8 shows the map of Rio de Janeiro with each pixel classified as belonging to one of the three classes – these can be interpreted as water (notice the little lake), urban area, and forest. \square

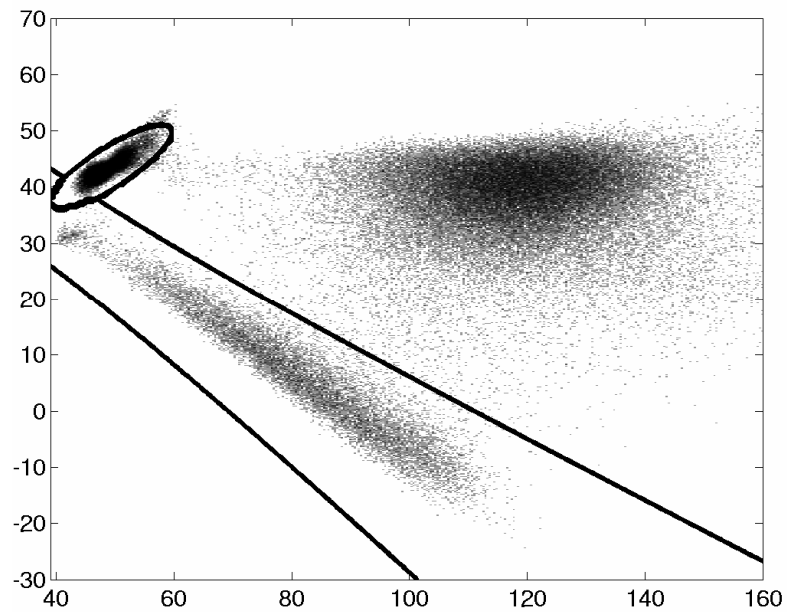


Figure 2.7: Bayesian quadratic discrimination by means of three Normal distributions fitted to the first two principal components of Rio de Janeiro.

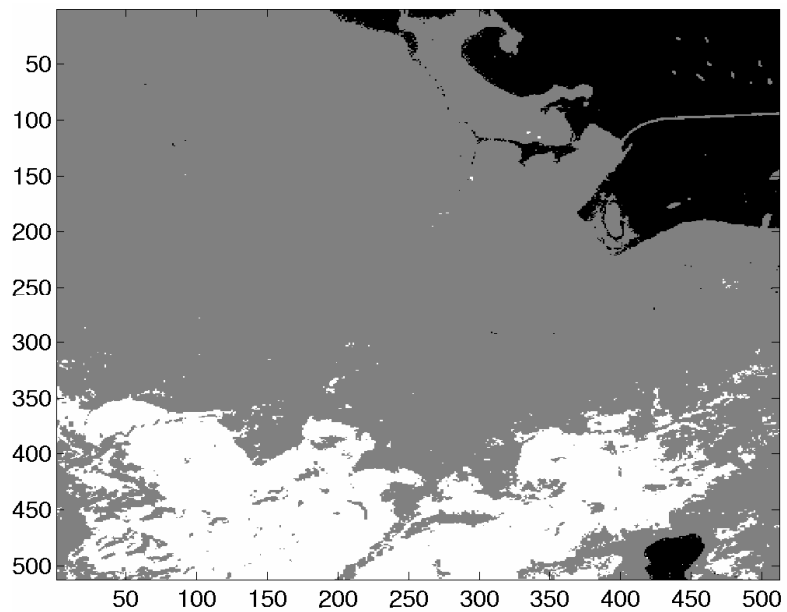


Figure 2.8: Rio de Janeiro map with pixels classified into three classes; water (black), urban area (grey), and forest (white).

Exercises

Exercise 2.1: Given n independent observations $x_1, \dots, x_n \in \mathbb{R}$ from a $N(\mu, \sigma^2)$ distribution, find the ML-estimates of μ and σ^2 .

Exercise 2.2: Assume that $x_i | \lambda$, $i = 1 \dots n$, are independent and exponentially distributed with expectation $1/\lambda$, i.e. $p(x_i | \lambda) = \lambda e^{-x_i \lambda}$.

- Find the ML-estimator of λ .
- Now introduce a $\Gamma(a, b)$ prior on λ , with density $\pi(\lambda) = \lambda^{a-1} e^{-\lambda/b} / (\Gamma(a) b^a)$. Find the posterior distribution of $\lambda | x_1, \dots, x_n$ (Hint: the posterior is also a Gamma distribution).

Exercise 2.3: Let $x_i | m$, $i = 1 \dots n$, be independent $N(m, \sigma^2)$, with a $N(\mu, \tau^2)$ prior on m and assume that σ^2, μ, τ^2 are known constants.

- If \bar{x} is the mean of x_1, \dots, x_N find the density of $\bar{x} | m$, i.e. $p(\bar{x} | m)$.
(Note that \bar{x} is a *sufficient statistic* for m , implying that all information needed to estimate m is contained in \bar{x})
- Find the joint density for \bar{x} and m , i.e. $p(\bar{x}, m)$.
- Motivate that $p(m | \bar{x}) \propto p(\bar{x}, m)$ and motivate that the normalising constant does not depend on m .
- Use the result in (c) to find the MAP estimate of m .
- Use the result in (c) to show that the posterior density $p(m | \bar{x})$ is the density of

$$N\left(\bar{x} \frac{\tau^2}{\tau^2 + \sigma^2/n} + \mu \frac{\sigma^2/n}{\tau^2 + \sigma^2/n}, \frac{\tau^2 \sigma^2/n}{\tau^2 + \sigma^2/n}\right).$$

- What does it imply for the prior and for the posterior distribution $m | \bar{x}$, that $\tau^2 \rightarrow \infty$, or that $\tau^2 \rightarrow 0$? What happens if the measurement variance is increased, $\sigma^2 \rightarrow \infty$?

Exercise 2.4: The lifetime for a certain type of light bulb is assumed to follow an exponential distribution with expectation a . To estimate a , n independent light bulbs are observed and their lifetimes x_i are recorded. However, the experiment was run for a pre-determined duration of h hours, and at the end of the experiment m light bulbs are still working. This gives $n - m$ observations of lifetimes, x_i , and m unknown lifetimes y_i that are only known to be larger than h .

One way of estimating a is to use an ML-estimator.

- a) If \tilde{y} follows an exponential distribution with $\mathbf{E}(\tilde{y}) = a$, find $\mathbf{P}(\tilde{y} > h)$.
- b) Use the result in (a) to get the entire likelihood for a , i.e. state

$$L(a \mid x_1, \dots, x_{n-m}, m, h).$$

- c) Use the result in (b) to find the ML-estimate of a .

Exercise 2.5: Another way of solving the previous problem is to derive an EM-algorithm.

- a) If \tilde{y} follows an exponential distribution with $\mathbf{E}(\tilde{y}) = a$ calculate the conditional expectation $\mathbf{E}(\tilde{y} \mid \tilde{y} > h)$.
- b) State the log-likelihood function assuming that all x_i and y_i are known, i.e. state $l(a \mid \dots) = \log L(a \mid m, x_1, \dots, x_{n-m}, y_1, \dots, y_m)$.
- c) Use the results in (a) and (b) to calculate

$$Q(a, a_t) = \mathbf{E} (l(a \mid \dots) \mid a_t, m, h, x_1, \dots, x_{n-m}),$$

which is the E-step in an EM algorithm.

- d) Find the $a = a_{t+1}$ that maximises Q . This is the M-step of the EM-algorithm.
- e) Solve the recursion equation for a found in (d) to find the ML-estimate of a and compare it to the result obtained in the previous exercise.

Exercise 2.6: Show that the Equation (2.2) given for Q in the EM-algorithm for mixtures in Section 2.3.3 is correct.

Exercise 2.7: Show that the expression given for $\pi_m^{(t+1)}$ in the EM-algorithm in Section 2.3.3 is correct.

Chapter 3

Distributions, correlation, and filters

Many of the standard methods used in image analysis are based on statistical ideas such as distribution entropy or information, or on simple measures of similarity and connectedness, such as correlation. The simplest of these use only the single pixel values and their distributions, without regards to neighbouring values. More common is that one uses the information from neighbour pixels to improve on the pixel values. In this chapter we shall investigate some of these methods - which we might call *classical* in contrast to the more fancier methods which make up the rest of this book.

The methods we shall deal with here are typically image manipulating methods, which either enhance the visual impression already present in the image, or reduce the effect of disturbance or noise in the image. The techniques will be correlations and Fourier transform methods for random and fixed image elements. In particular we shall study the effect of randomness in common image filters such as edge detection filters, smoothing filters, and matching filters.

Random and fixed image elements

Following the paradigm in this book, we shall make it clear from the beginning, when we work with observed data and when we work with the stochastic model, which we assume to have generated the data by some random mechanism. The stochastic model can contain deterministic elements, which remain the same regardless of disturbances and distortions – we call these *fixed image elements*. Other elements are the result of a random process, for example creating blurring noise or translation motion noise – we call these *random image elements*. A fixed image element can become a random element if its location becomes random, for example by an unstable camera.

Example 3.1. In a text image the letter A in black lies on top of a smooth white background disturbed by random grey level noise. The *model* is then $\tilde{x}(\mathbf{u}) = a(\mathbf{u}) + \tilde{n}(\mathbf{u})$, where $a(\mathbf{u}) = \text{"black"}$ for $\mathbf{u} \in i_A$, and $a(\mathbf{u}) = \text{"white"}$ for $\mathbf{u} \notin i_A$, and $\tilde{n}(\mathbf{u}) \in$

$GLD(\theta)$ for $\mathbf{u} \in \mathcal{I}$. For the data in an observed image, we use $x(\mathbf{u}) = a(\mathbf{u}) + n(\mathbf{u})$, where the $n(\mathbf{u})$ are observations of the random variables $\tilde{n}(\mathbf{u})$. \square

3.1 Grey level distributions and histograms

3.1.1 Histogram transformations

3.2 Covariance and correlation

3.2.1 The covariance matrix

The *covariance* between two random variables \tilde{x} and \tilde{y} is defined as

$$\mathbf{C}(\tilde{x}, \tilde{y}) = \mathbf{E}((\tilde{x} - m_{\tilde{x}})(\tilde{y} - m_{\tilde{y}})),$$

where $m_{\tilde{x}} = \mathbf{E}(\tilde{x})$, $m_{\tilde{y}} = \mathbf{E}(\tilde{y})$ are the expected values. With variances $\sigma_{\tilde{x}}^2 = \mathbf{V}(\tilde{x})$, $\sigma_{\tilde{y}}^2 = \mathbf{V}(\tilde{y})$, the *correlation coefficient* is

$$\rho_{\tilde{x}, \tilde{y}} = \frac{\mathbf{C}(\tilde{x}, \tilde{y})}{\sigma_{\tilde{x}} \sigma_{\tilde{y}}}.$$

Note that ρ is a normalised dimensionless quantity with $-1 \leq \rho \leq 1$.

If $\tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_p)^\top$ is a (column) vector of random variables with pairwise covariances $\mathbf{C}(\tilde{x}_j, \tilde{x}_k) = \Sigma_{jk}$, the matrix $\Sigma = (\Sigma_{jk})$ is called the *covariance matrix*. In matrix notation,

$$\Sigma = \mathbf{V}(\tilde{\mathbf{x}}) = \mathbf{C}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) = \mathbf{E}((\tilde{\mathbf{x}} - \mathbf{m})(\tilde{\mathbf{x}} - \mathbf{m})^\top).$$

The covariance matrix is a non-negative definite, symmetric matrix.

Covariances and linear operations

That a covariance matrix is a non-negative definite follows simply from the observation that the covariance operator is *bi-linear*: for any two random vectors $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$, and compatible real matrices \mathbf{A} and \mathbf{B} , one has

$$\mathbf{C}(\mathbf{A}\tilde{\mathbf{x}}, \mathbf{B}\tilde{\mathbf{y}}) = \mathbf{A}\mathbf{C}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})\mathbf{B}^\top. \quad (3.1)$$

Taking $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}$ and letting \mathbf{a} be a real vector, we have

$$\mathbf{V}(\mathbf{a}^\top \tilde{\mathbf{x}}) = \mathbf{C}(\mathbf{a}^\top \tilde{\mathbf{x}}, \mathbf{a}^\top \tilde{\mathbf{x}}) = \mathbf{a}^\top \mathbf{C}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \mathbf{a} = \mathbf{a}^\top \Sigma \mathbf{a} \geq 0,$$

since the left hand side is always non-negative. We also see that the covariance matrix is semi-definite if there is a linear combination of variables which has variance 0, i.e. is constant.

Without matrix notation, the covariance formula between two linear combinations reads,

$$\mathbf{C}\left(\sum_j a_j \tilde{x}_j, \sum_k b_k \tilde{y}_k\right) = \sum_j \sum_k a_j b_k \mathbf{C}(\tilde{x}_j, \tilde{y}_k). \quad (3.2)$$

Using the covariance function definition in Section 3.2.2, we can construct the elements of the covariance matrix of an entire image with scalar pixel values as $\Sigma_{jk} = \mathbf{r}(\tilde{\mathbf{x}}(\mathbf{u}_j), \tilde{\mathbf{x}}(\mathbf{u}_k))$. Covariance matrices for entire images are very large, and can rarely be used directly in numerical computations. For homogeneous covariance functions however, the matrices have a block structure, that can be exploited, often removing the need to store the covariance matrix directly. This is especially important if we want to construct a model for images with vector pixel elements, with dependent pixel values.

3.2.2 Model covariance and correlation function

In signal analysis the correlation between signal elements is described by the *covariance function*, or by its normalised version, the *correlation function*. Both are defined for random image elements where the pixel values are considered as the result of a random procedure. They describe the *expected average behaviour* of the signal or image element.

The terms *covariance* or *correlation function* are also used to describe properties of a recorded signal, or of an observed image element. They are dependent on data and will be different for different images or signals even when these are generated by the same mechanism. We shall be careful in distinguishing these two meanings of the word correlation.

Definition 3.1. Let $\tilde{x}(\mathbf{u}), \mathbf{u} \in \mathcal{A}$ be a random image element with expectation $E(\tilde{x}(\mathbf{u})) = m_{\tilde{x}}(\mathbf{u})$ and variance $V(\tilde{x}(\mathbf{u})) = E((\tilde{x}(\mathbf{u}) - m_{\tilde{x}}(\mathbf{u}))^2) = \sigma_{\tilde{x}}^2(\mathbf{u})$. The statistical covariance between image elements $\tilde{x}(\mathbf{u})$ and $\tilde{x}(\mathbf{v})$ is

$$r_{\tilde{x}}(\mathbf{u}, \mathbf{v}) = C(\tilde{x}(\mathbf{u}), \tilde{x}(\mathbf{v})) = E((\tilde{x}(\mathbf{u}) - m_{\tilde{x}}(\mathbf{u}))(\tilde{x}(\mathbf{v}) - m_{\tilde{x}}(\mathbf{v}))).$$

The correlation between $\tilde{x}(\mathbf{u})$ and $\tilde{x}(\mathbf{v})$ is

$$\rho_{\tilde{x}}(\mathbf{u}, \mathbf{v}) = \frac{r_{\tilde{x}}(\mathbf{u}, \mathbf{v})}{\sigma_{\tilde{x}}(\mathbf{u})\sigma_{\tilde{x}}(\mathbf{v})} = \frac{r_{\tilde{x}}(\mathbf{u}, \mathbf{v})}{\sqrt{r_{\tilde{x}}(\mathbf{u}, \mathbf{u})r_{\tilde{x}}(\mathbf{v}, \mathbf{v})}}$$

The functions $r_{\tilde{x}}(\mathbf{u}, \mathbf{v})$ and $\rho_{\tilde{x}}(\mathbf{u}, \mathbf{v})$ are called the (model) covariance function, mcf, and correlation function, respectively. The index \tilde{x} in $m_{\tilde{x}}, \sigma_{\tilde{x}}, r_{\tilde{x}}, \rho_{\tilde{x}}$ is omitted when no mistake can occur. \square

The expectation, variance etc., are here to be understood as expectation over repeated images. The model covariance function is a characteristic of the random mechanism that generates the images and not of the images themselves. If several images are produced by the same procedure, they still have the same covariance function, since that is a property of the image generation.

3.2.3 Homogeneity and isotropy

The structure of a random image segment can be more or less *homogeneous*, which means that its statistical properties remain the same after a translation of the segment. A typical example of a homogeneous image segment is a section of a forest or of the sea surface as imaged by a satellite. Another example is a uniform image segment disturbed by pixel-wise added noise.

A homogeneous image element may exhibit different structure in different directions. A more strict structural property is *isotropy*, which means that the statistical properties of the image element remain the same also after rotation.

Definition 3.2. A random image element $\tilde{x}(\mathbf{u})$, $\mathbf{u} \in \mathcal{A}$ is called *homogeneous* if the distribution of $(\tilde{x}(\mathbf{u}_1), \dots, \tilde{x}(\mathbf{u}_p))$ is the same as that of $(\tilde{x}(\mathbf{u}_1 + \boldsymbol{\tau}), \dots, \tilde{x}(\mathbf{u}_p + \boldsymbol{\tau}))$ for all $(\mathbf{u}_1, \dots, \mathbf{u}_p)$ and $\boldsymbol{\tau}$ such that $\mathbf{u}_j, \mathbf{u}_j + \boldsymbol{\tau} \in \mathcal{A}$. It is called *isotropic* if it is homogeneous and the distributions remain unchanged after an arbitrary rotation of the plane; in particular the distribution of $\tilde{x}(\mathbf{u})$ is the same as that of $\tilde{x}(\mathbf{B}\mathbf{u} + \boldsymbol{\tau})$ for all rotation matrices \mathbf{B} .

The random element $\tilde{x}(\mathbf{u})$, $\mathbf{u} \in \mathcal{A}$ is called *weakly homogeneous* if the covariance between $\tilde{x}(\mathbf{u})$ and $\tilde{x}(\mathbf{v})$ is the same as that between $\tilde{x}(\mathbf{u} + \boldsymbol{\tau})$ and $\tilde{x}(\mathbf{v} + \boldsymbol{\tau})$ for all $\mathbf{u}, \mathbf{v}, \boldsymbol{\tau}$. Then the covariance function is a function only of the difference $\mathbf{v} - \mathbf{u}$,

$$r_{\tilde{x}}(\mathbf{v} - \mathbf{u}) = \mathbf{C}(\tilde{x}(\mathbf{u}), \tilde{x}(\mathbf{v})).$$

The normalised covariance function is called the (model) correlation function,

$$\rho_{\tilde{x}}(\mathbf{v} - \mathbf{u}) = \rho(\tilde{x}(\mathbf{u}), \tilde{x}(\mathbf{v})) = \frac{r_{\tilde{x}}(\mathbf{v} - \mathbf{u})}{r_{\tilde{x}}(\mathbf{0})}.$$

□

We use the same notation $r_{\tilde{x}}$ to denote both the function of two arguments and that of one argument. No confusion should occur. Note that the covariance function is always symmetric and has its maximum at the origin,

$$r_{\tilde{x}}(\mathbf{u}) = r_{\tilde{x}}(-\mathbf{u}), \quad (3.3)$$

$$|r_{\tilde{x}}(\mathbf{u})| \leq r_{\tilde{x}}(\mathbf{0}). \quad (3.4)$$

For an isotropic random element, the covariance function is rotationally symmetric,

$$r_{\tilde{x}}(\mathbf{u}) = r_{\tilde{x}}(\mathbf{B}\mathbf{u}) = \tilde{r}_{\tilde{x}}(\|\mathbf{u}\|),$$

for all rotation matrices \mathbf{B} . This has implications for the possible shapes of isotropic covariance functions. We shall prove in Section 3.3 that for any isotropic $r_{\tilde{x}}(\mathbf{u})$ there is a bounded non-decreasing function $G(\lambda)$, defined for $\lambda \geq 0$, such that

$$r_{\tilde{x}}(\mathbf{u}) = \int_0^\infty J_0(\lambda\|\mathbf{u}\|) dG(\lambda), \quad (3.5)$$

where $J_0(x)$ is a first order Bessel function,

$$J_0(x) = \sum_{k=0}^{\infty} (-1)^k \frac{(x/2)^{2k}}{k!^2}.$$

There are no other restrictions on $G(\lambda)$, and all functions of the form (3.5) can be the covariance function for an isotropic field. Explicit and simple isotropic covariance functions are

$$\begin{aligned} r(\mathbf{u}) &= \sigma^2 e^{-a^2\|\mathbf{u}\|^2}, \\ r(\mathbf{u}) &= \sigma^2 (1 + \|\mathbf{u}\|^2/b^2)^{-\beta}. \end{aligned} \quad (3.6)$$

3.2.4 Data covariance and correlation

The model correlation is a characteristic of a data generation mechanism. One can also define a covariance function and a correlation function in an image or other recorded signal. Suppose we have a data set in the form of an observed image element $x(\mathbf{u})$, $\mathbf{u} \in \mathcal{A}$, where \mathcal{A} is a uniform gitter. To find the data covariance between values $x(\mathbf{t})$, $\mathbf{t} \in \mathcal{A}$ and $x(\mathbf{t} + \mathbf{u})$, $\mathbf{t} + \mathbf{u} \in \mathcal{A}$, in a data set one identifies all pairs of points $(\mathbf{t}, \mathbf{t} + \mathbf{u})$ with vector difference \mathbf{u} , and then take the average:

$$r_x(\mathbf{u}) = \frac{1}{n} \sum_{\mathbf{t} \in \mathcal{N}(\mathbf{u})} (x(\mathbf{t}) - m)(x(\mathbf{t} + \mathbf{u}) - m), \quad (3.7)$$

where the sum is taken over all points \mathbf{t} such that both \mathbf{t} and $\mathbf{t} + \mathbf{u}$ belong to \mathcal{A} .

The denominator n in (3.7) should be chosen in such a way that for increasing regions \mathcal{A} , the ratio $|\mathcal{N}(\mathbf{u})|/n \rightarrow 1$ where $|\mathcal{N}(\mathbf{u})|$ is the number of terms in the sum. Then

$$\mathbf{E}(r_x(\mathbf{u})) \rightarrow r_{\tilde{x}}(\mathbf{u}). \quad (3.8)$$

Model	Data
Covariance function, mcf: $r_{\tilde{x}}(\mathbf{u}) = \mathbf{C}(\tilde{x}(\mathbf{t}), \tilde{x}(\mathbf{t} + \mathbf{u}))$	Data covariance function, dcf: $r_x(\mathbf{u}) = \frac{1}{n_{\mathbf{u}}} \sum_{\mathbf{t} \in \mathcal{N}(\mathbf{u})} (x(\mathbf{t}) - m)(x(\mathbf{t} + \mathbf{u}) - m)$
Correlation function: $\rho_{\tilde{x}}(\mathbf{u}) = \rho(\tilde{x}(\mathbf{t}), \tilde{x}(\mathbf{t} + \mathbf{u})) = r_{\tilde{x}}(\mathbf{u})/r_{\tilde{x}}(\mathbf{0})$	Data correlation function: $\rho_x(\mathbf{u}) = r_x(\mathbf{u})/r_x(\mathbf{0})$

Table 3.1: Summary of model and data correlation measures.

The normalised covariance function is called the (data) correlation function,

$$\rho_x(\mathbf{u}) = \frac{r_x(\mathbf{u})}{r_x(\mathbf{0})},$$

where $s_x^2 = r_x(\mathbf{0})$ is the sample variance.

Remark 3.1: The terms covariance and correlation are used in this work in their common statistical sense, i.e. a correlation is a covariance normalised by standard deviations. A covariance between two random quantities U and V has the same dimension as the product UV , while the correlation is dimensionless.

In signal processing and image analysis literature the term correlation is mostly used for the function

$$\tilde{r}_x(\mathbf{u}) = \frac{1}{n} \sum_{\mathbf{t} \in \mathcal{N}(\mathbf{u})} x(\mathbf{t})x(\mathbf{t} + \mathbf{u}).$$

The most notable difference from our convention is that the average m is not subtracted in the cross product. A uniform additive increase in signal level will therefore increase the values of $\tilde{r}_x(\mathbf{u})$ without changing the real correlation structure of the signal. Also a multiplicative change of scaling units will change $\tilde{r}_x(\mathbf{u})$, without really affecting the correlation structure. \square

Figure 3.1 shows the data covariance function for the paper structure in Figure 1.1 estimated by (3.7). Because the large number of pixels we only used a segment of the figure of size 128×128 pixels. As seen in the figure, the covariance function extends only a few pixels out from the centre. The almost exponential decrease with increasing distance is typical for many images with a random structure. The correlations can not represent the typical fibre structure with random fibre orientation. In Section 3.3 we shall see examples with more structure, where the covariance function also is more informative.

3.3 Fourier transforms and spectrum

The Fourier transform is one of the most useful of all image analysis tools. It can be defined for data vectors and image pixel values, as well as for the functions, which

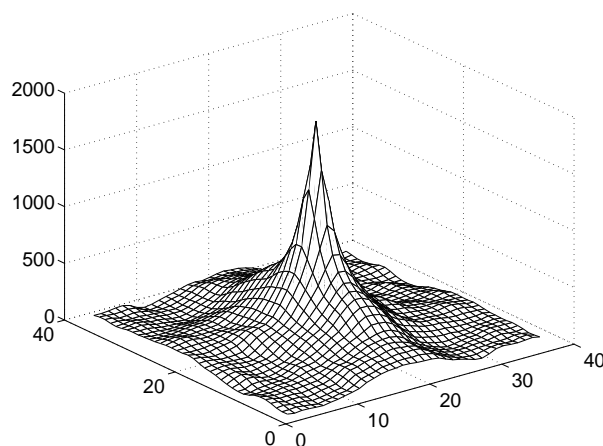


Figure 3.1: Data covariance function of the paper structure in Figure 1.1.

describe the data generating mechanism.

3.3.1 Fourier transforms in \mathbb{R} and \mathbb{Z}

Let $g(u)$ be a continuous function of a real variable u . The *Fourier transform* of g is a function of a real variable, denoted by $\mathcal{F}g$, and defined by,

$$(\mathcal{F}g)(f) = G(f) = \int_{-\infty}^{\infty} g(u) e^{-i2\pi fu} du, \quad (3.9)$$

where $i = \sqrt{-1}$. The Fourier transform exists if, for example, $g(u)$ is absolutely integrable, $\int |g(u)| du < \infty$. If $\int |G(f)| df < \infty$, then the *Fourier inversion formula* holds, and $g(u)$ is given by

$$g(u) = (\mathcal{F}^{-1}G)(u) = \int_{-\infty}^{\infty} G(f) e^{i2\pi fu} df. \quad (3.10)$$

Equations (3.9) and (3.10) make $g(u)$ and $G(f)$ a *Fourier transform pair*.

Fourier transform of covariance functions; power spectral density

In signal and image generation the Fourier transform of the covariance function is particularly important, since it carries information about the distribution of signal energy over frequencies. Let $r_{\tilde{x}}(u) = \mathbf{C}(\tilde{x}(t), \tilde{x}(t + u))$ be the covariance function of a stationary random process of a one-dimensional parameter t . The following famous theorem characterises the possible covariance functions as those which are the (inverse) Fourier transforms of a symmetric integrable non-negative functions.

Theorem 3.1. (a) If $r_{\tilde{x}}(u)$ is a continuous covariance function of a stationary stochastic process $\tilde{x}(t)$, $t \in \mathbb{R}$, then there exists a symmetric integrable non-negative function $R_{\tilde{x}}(f)$, called the power spectral density, such that

$$r_{\tilde{x}}(u) = \int_{f=-\infty}^{\infty} e^{i2\pi fu} R_{\tilde{x}}(f) df, \quad (3.11)$$

$$R_{\tilde{x}}(f) = \int_{u=-\infty}^{\infty} e^{-i2\pi fu} r_{\tilde{x}}(u) du, \quad (3.12)$$

where (3.12) holds if $\int |r_{\tilde{x}}(u)| du < \infty$.

(b) If $r_{\tilde{x}}(u)$ is a covariance function of a stationary stochastic sequence $\tilde{x}(t)$, $t \in \mathbb{Z}$, then there exists a symmetric integrable non-negative function $R_{\tilde{x}}(f)$, defined for $-1/2 < f \leq 1/2$, called the power spectral density, such that

$$r_{\tilde{x}}(u) = \int_{f=-1/2}^{1/2} e^{i2\pi fu} R_{\tilde{x}}(f) df, \quad (3.13)$$

$$R_{\tilde{x}}(f) = \sum_{u=-\infty}^{\infty} e^{-i2\pi fu} r_{\tilde{x}}(u), \quad (3.14)$$

where (3.14) holds if $\sum |r_{\tilde{x}}(u)| < \infty$.

Since the covariance function has the dimension (unit of $\tilde{x}(u)$)², and f has the dimension (unit of u)⁻¹, the power spectral density has the dimension (unit of $\tilde{x}(u)$)²(unit of u). For example, if u is time in seconds and $\tilde{x}(u)$ is in meter then $R_{\tilde{x}}(f)$ has the unit $m^2(\text{Hz})^{-1}$. The variable f is usually referred to as *frequency*, if u is time, or as *wave number*, if u is length.

The power spectral density expresses the variance $\mathbf{V}(\tilde{x}(\mathbf{u}))$ as an integral over all frequencies. Taking $\mathbf{u} = \mathbf{0}$ we get,

$$\mathbf{V}(\tilde{x}(\mathbf{t})) = \mathbf{E}((\tilde{x}(\mathbf{t}) - m_{\tilde{x}})^2) = r_{\tilde{x}}(\mathbf{0}) = \int_{-\infty}^{\infty} R_{\tilde{x}}(f) df. \quad (3.15)$$

As we shall see in the section on linear filters, Section 3.4, the interpretation of the power spectral density as a distribution of variation over frequencies, has more to it than appears from (3.15).

Example 3.2. (a) The Gaussian shaped covariance function $r(u) = \sigma^2 e^{-au^2}$ has the Fourier transform

$$G(f) = \sigma^2 \sqrt{\pi/a} e^{-\frac{(2\pi f)^2}{4a}}.$$

(b) The exponential covariance function $r(u) = \sigma^2 e^{-\alpha|u|}$ has the Fourier transform

$$G(f) = \frac{2\alpha}{\alpha^2 + (2\pi f)^2}.$$

□

Remark 3.2: The condition that $g(u)$ and $G(f)$ be absolutely integrable is not a necessary condition for them to be a Fourier transform pair. An example is the function $g(u) = \frac{\sin(2\pi u)}{2\pi u}$ which is the Fourier transform of the box function

$$G(f) = \begin{cases} 1/2 & \text{for } |f| < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.16)$$

Then $g(u)$ is given by (3.10) for all u , while 3.9 has to be interpreted as the Cauchy principal value

$$G(f) = \lim_{T \rightarrow \infty} \int_{u=-T}^T g(u) e^{-i2\pi f u} du,$$

except at $f = \pm 1$. □

3.3.2 Sum of random harmonics

This section treats one particularly useful building block in random image elements, namely the sum of harmonics with random phase and amplitude. A sum of cosine functions with different frequencies can become very complex even with a small number of terms. To make it a genuine random sequence one can let amplitudes and phases be random.¹ Thus, take fixed positive frequencies, f_1, \dots, f_n , and define

$$\tilde{x}(t) = \sum_{j=1}^n A_j \cos(2\pi f_j t + \varphi_j), \quad (3.17)$$

where we let the A_j 's be random amplitudes and the φ_j 's random phases, all A_j and φ_j being independent of each other. The phases φ_j determine the location of the cosine functions relative the origin. In order that the process $X(t)$ be statistically stationary – i.e. the time origin should play no particular role – the phases need to be uniformly distributed between 0 and 2π .

It is then easily seen that $\tilde{x}(t)$ has expectation 0 and that its covariance function is

$$r_{\tilde{x}}(u) = \sum_{j=1}^n E(A_j^2/2) \cos 2\pi f_j u.$$

The power spectral density is a sum of Dirac delta functions $\delta_{\pm f_j}(f)$:

$$R_{\tilde{x}}(f) = \sum_{j=1}^n E(A_j^2/2) \left\{ \frac{1}{2} \delta_{f_j}(f) + \frac{1}{2} \delta_{-f_j}(f) \right\}. \quad (3.18)$$

¹Even random frequencies would be possible, but then the process would not be as easily described mathematically.

To check this, just note that $\int_{f=-\infty}^{\infty} \delta_{\pm f_j}(f) e^{i2\pi f u} df = e^{i2\pi \pm f_j u}$.

The form of the power spectral density hints at another way of representing $\tilde{x}(t)$, namely as, for example,

$$\tilde{x}(t) = \sum_{k=1}^n \left\{ \frac{A_j}{2} \cos(2\pi f_k t + \varphi_k) + \frac{A_j}{2} \cos(2\pi(-f_k)t - \varphi_k) \right\}. \quad (3.19)$$

Here we encounter negative frequencies and phases; this is of importance when we generalise the model to a moving image.

3.3.3 Fourier transforms in \mathbb{R}^2 and \mathbb{Z}^2

The Fourier transform can be defined for continuous functions $g(\mathbf{u})$ on \mathbb{R}^2 as

$$(\mathcal{F}g)(\mathbf{f}) = G(\mathbf{f}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(\mathbf{u}) e^{-i2\pi(f_1 u_1 + f_2 u_2)} d\mathbf{u}. \quad (3.20)$$

The Fourier transform exists if $g(\mathbf{u})$ is absolutely integrable, $\int |g(\mathbf{u})| d\mathbf{u} < \infty$. If $\int |G(\mathbf{f})| d\mathbf{f} < \infty$, then the inversion formula holds, and

$$g(\mathbf{u}) = (\mathcal{F}^{-1}G)(\mathbf{u}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(\mathbf{f}) e^{i2\pi(f_1 u_1 + f_2 u_2)} d\mathbf{f}. \quad (3.21)$$

The Fourier transform of a double indexed sequence

The Fourier transform of a double indexed sequence $g(\mathbf{u})$, $\mathbf{u} \in \mathbb{Z}$ can also be defined:

$$(\mathcal{F}g)(\mathbf{f}) = G(\mathbf{f}) = \sum_{u_1=-\infty}^{\infty} \sum_{u_2=-\infty}^{\infty} g(\mathbf{u}) e^{-i2\pi(f_1 u_1 + f_2 u_2)}, \quad (3.22)$$

$$g(\mathbf{u}) = (\mathcal{F}^{-1}G)(\mathbf{u}) = \int_{f_1=-1/2}^{1/2} \int_{f_2=-1/2}^{1/2} G(\mathbf{f}) e^{i2\pi(f_1 u_1 + f_2 u_2)} d\mathbf{f}. \quad (3.23)$$

Power spectral densities in \mathbb{R}^2 and \mathbb{Z}^2

If $r_{\tilde{x}}(\mathbf{u})$ is the covariance of a homogeneous random field over \mathbb{R}^2 or \mathbb{Z}^2 , the power spectral density $R_{\tilde{x}}(\mathbf{f})$ is defined to form a Fourier transform pair with $r_{\tilde{x}}(\mathbf{u})$:

$$r_{\tilde{x}}(\mathbf{u}) = \begin{cases} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{\tilde{x}}(\mathbf{f}) e^{i2\pi(f_1 u_1 + f_2 u_2)} d\mathbf{f}, & \mathbf{u} \in \mathbb{R}^2, \\ \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} R_{\tilde{x}}(\mathbf{f}) e^{i2\pi(f_1 u_1 + f_2 u_2)} d\mathbf{f}, & \mathbf{u} \in \mathbb{Z}^2, \end{cases} \quad (3.24)$$

$$R_{\tilde{x}}(\mathbf{f}) = \begin{cases} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} r_{\tilde{x}}(\mathbf{u}) e^{-i2\pi(f_1 u_1 + f_2 u_2)} d\mathbf{u}, & \mathbf{f} \in \mathbb{R}^2, \\ \sum_{u_1=-\infty}^{\infty} \sum_{u_2=-\infty}^{\infty} r_{\tilde{x}}(\mathbf{u}) e^{-i2\pi(f_1 u_1 + f_2 u_2)}, & \mathbf{f} \in [-1/2, 1/2]^2. \end{cases} \quad (3.25)$$

The two-dimensional power spectrum $R_{\bar{x}}(\mathbf{f})$ is useful when it comes to description of a homogeneous random surface or a homogeneous random image element, like the structure of an imaged region. The spectral density in two dimensions can be interpreted in analogy with the one-dimensional case as a distribution function, which describes how a surface is built from random harmonics with different frequencies. Each individual component extends with plane wave front in its specified direction, independently of the other components. The argument $\mathbf{f} = (f_1, f_2)$ gives the wave numbers in Cartesian co-ordinates. The power spectral density if other given in polar co-ordinates, $R_{\bar{x}}^p(r, \theta) = r F_{\bar{x}}(r \cos \theta, r \sin \theta)$.

Power spectrum for an isotropic field

In an isotropic field the random structure remains the same after a rotation. This imposes a specific structure to the covariance function, which has to be rotationally symmetric, $r_{\bar{x}}(\mathbf{u}) = r_{\bar{x}}(\mathbf{B}\mathbf{u})$, for all rotation matrices \mathbf{B} . We can now prove the statement in Section 3.2.3 that for any isotropic $r(\mathbf{u})$ there is a bounded non-decreasing function $G(\lambda)$, defined for $\lambda \geq 0$, such that

$$r(\mathbf{u}) = \int_0^\infty J_0(\lambda \|\mathbf{u}\|) dG(\lambda),$$

where $J_0(x)$ is a first order Bessel function,

$$J_0(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \cos(x \sin \theta) d\theta = \sum_{k=0}^{\infty} (-1)^k \frac{(x/2)^{2k}}{k!^2}.$$

If $r(\mathbf{u})$ is rotationally symmetric, then the power spectral density is also rotationally symmetric,

$$\begin{aligned} F(\mathbf{B}\mathbf{f}) &= \int r(\mathbf{u}) \exp(-i2\pi(\mathbf{B}\mathbf{f})^\top \mathbf{u}) d\mathbf{u} \\ &= \int r(\mathbf{B}^{-1}\mathbf{u}) \exp(-i2\pi\mathbf{f}\mathbf{u}) d\mathbf{u} = F(\mathbf{f}) = \tilde{F}(\|\mathbf{f}\|), \end{aligned}$$

for some non-negative function \tilde{F} . With a change to polar co-ordinates, $f_1 =$

$\frac{\lambda}{2\pi} \cos \theta, f_2 = \frac{\lambda}{2\pi} \sin \theta$, we obtain

$$\begin{aligned}
r((u_1, u_2)) &= r((0, \|\mathbf{u}\|)) \\
&= \int \int e^{i2\pi(0+f_2\|\mathbf{u}\|)} F((f_1, f_2)) \, d\mathbf{f} \\
&= \int_{\lambda=0}^{\infty} \int_{\theta=-\pi}^{\pi} e^{i\lambda\|\mathbf{u}\| \sin \theta} \frac{\lambda}{(2\pi)^2} \tilde{F}\left(\frac{\lambda}{2\pi}\right) \, d\theta \, d\lambda \\
&= \int_0^{\infty} \frac{\lambda}{(2\pi)^2} \tilde{F}\left(\frac{\lambda}{2\pi}\right) \left\{ \int_{-\pi}^{\pi} \cos(\lambda\|\mathbf{u}\| \sin \theta) \, d\theta \right\} \, d\lambda \\
&= \int_0^{\infty} \frac{\lambda}{2\pi} \tilde{F}\left(\frac{\lambda}{2\pi}\right) J_0(\lambda\|\mathbf{u}\|) \, d\lambda \\
&= \int_0^{\infty} J_0(\lambda\|\mathbf{u}\|) \, dG(\lambda),
\end{aligned}$$

for $G(\lambda) = \int_0^{\lambda} \frac{x}{2\pi} \tilde{F}\left(\frac{x}{2\pi}\right) \, dx$.

3.3.4 Discrete Fourier transforms of 1D and 2D data

In the previous section we dealt with Fourier transforms of covariance functions of homogeneous, 2D functions, or of stationary processes or sequences. We shall now examine Fourier transforms of data, in particular of data, which are observations of a homogeneous process in the plane, or of a stationary sequence. We shall also be concerned with the difference between the Fourier pair $r_{\tilde{x}}(\mathbf{u}) \leftrightarrow R_{\tilde{x}}(\mathbf{f})$, which is a property of the data generating mechanism, and the relation between data and its Fourier transform.

Discrete Fourier transform of a data sequence

Consider a data sequence, $x_k = x(u_k), k = 0, 1, \dots, N-1$, which may, but need not, represent sampled values of a continuous function $x(u)$. The sampling points u_k are assumed to be equidistant, with $u_k - u_{k-1} = \Delta u$.

We first define the Fourier transform of the sequence x_k , without considering the sampling rate, i.e. $\Delta u = 0$. The *discrete Fourier transform* is defined as the sequence²

$$F_j = \frac{1}{N} \sum_{k=0}^{N-1} x_k e^{-i2\pi jk/N}, \quad j = 0, 1, \dots, N-1. \quad (3.26)$$

²The factor $1/N$ in the Fourier transform is somewhat arbitrary; some literature places it in the forward transform, and some in the inverse transform. Matlab uses the latter convention.

The corresponding *inverse* relation recovers x_k from F_j ,

$$x_k = \sum_{j=0}^{N-1} F_j e^{i2\pi jk/N}, \quad k = 0, 1, \dots, N-1. \quad (3.27)$$

Example 3.3. Take $f_n = n/N$, $n = 0, 1, \dots, N-1$, and consider a data sequence defined for $k = 0, 1, \dots, N-1$,

$$\begin{aligned} x_k &= \sum_{n=0}^{N-1} A_n e^{i(2\pi f_n k + \varphi_n)} \\ &= \sum_{n=0}^{N-1} A_n \cos(2\pi f_n k + \varphi_n) + i \sum_{n=0}^{N-1} A_n \sin(2\pi f_n k + \varphi_n), \end{aligned}$$

for $A_n \geq 0$, $0 \leq \varphi_n \leq 2\pi$. Then

$$\begin{aligned} F_j &= \frac{1}{N} \sum_{k=0}^{N-1} e^{-i2\pi jk/N} \sum_{n=0}^{N-1} A_n e^{i(2\pi nk/N + \varphi_n)} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} A_n e^{i\varphi_n} \sum_{k=0}^{N-1} e^{i2\pi k(n-j)/N} \\ &= A_j e^{i\varphi_j} = A_j (\cos \varphi_j + i \sin \varphi_j). \end{aligned}$$

Thus, the DFT generates the amplitude and phase of the complex functions that make up the data sequence.

For a real sequence

$$x_k = \sum_{n=0}^{N-1} A_n \cos(2\pi f_n k + \varphi_n), \quad k = 0, 1, \dots, N-1, \quad (3.28)$$

one has

$$F_j = \begin{cases} \frac{A_j e^{i\varphi_j} + A_{N-j} e^{-i\varphi_{N-j}}}{2}, & j = 1, 1, \dots, N-1, \\ A_0 \cos \varphi_0, & j = 0. \end{cases} \quad (3.29)$$

Similarly, for

$$x_k = \sum_{n=0}^{N-1} A_n \sin(2\pi f_n k + \varphi_n), \quad k = 0, 1, \dots, N-1, \quad (3.30)$$

one has

$$F_j = \begin{cases} \frac{A_j e^{i\varphi_j} - A_{N-j} e^{-i\varphi_{N-j}}}{2i}, & j = 1, \dots, N-1. \\ A_0 \sin \varphi_0, & j = 0. \end{cases} \quad (3.31)$$

As usual, the x_k sequence can be recovered from the F_j -sequence. To recover also the A_k and φ_k in the cosine sequence (3.23) from a given complex F_j -sequence obviously requires some extra conditions. One natural restriction is to require

$$A_j = A_{N-j} \quad \text{and} \quad \varphi_j = -\varphi_{N-j}. \quad (3.32)$$

Then, $F_j = A_j e^{i\varphi_j}$, would uniquely define A_j and φ_j in 3.23.

For the sinus sum 3.30, take $A_j = -A_{N-j}$ and $\varphi_j = -\varphi_{N-j}$. \square

In (3.26) we introduced the sequence F_j . We now define the function $z_N(f)$, for $f = j/N, j = 0, 1, \dots, N-1$, as $z_N(j/N) = NF_j$. Obviously, $z_N(f)$ can be extended to all f by $z_N(f) = \sum_{k=0}^{N-1} x_k e^{-i2\pi fk}$. One reason for the special notation F_j is that there is a one-to-one relationship between the data series and its discrete Fourier transform – the one can be constructed from the other without loss of information. A reason to be interested in the *whole* function $z_N(f)$ will be apparent when we consider a sequence $x_k, k = 0, 1, \dots$, which is a realization of a stationary random sequence, $\tilde{x}_k, k = 0, 1, \dots$; see Theorem 3.2.

The Fourier transform of an observed random sequence

The Fourier transform F_j of a data series is in general complex, $F_j = |F_j| e^{i \arg F_j}$. Here, $|F_j|$ is of special interest when data are observations of a stationary random sequence.

Theorem 3.2. (a) *If the data series $x_k, k = 0, 1, \dots, N-1$, represents observations of a stationary stochastic sequence $\tilde{x}_k, k = 0, 1, \dots$, then, for $0 < f < 1/2$, as $N \rightarrow \infty$,*

$$\mathbf{E}(N^{-1}|z_N(f)|^2) \rightarrow R_{\tilde{x}}(f), \quad N \rightarrow \infty. \quad (3.33)$$

Thus, $N^{-1}|z_N(f)|^2$ is, on the average an unbiased estimator of the power spectral density $R_{\tilde{x}}(f)$.

(b) *(If the sequence \tilde{x}_k is Gaussian ???) then, as $N \rightarrow \infty$, $2|z_N(f)|^2/R_{\tilde{x}}(f)$ has asymptotically a χ^2 -distribution with $df = 2$ degrees of freedom:*

$$P(2N^{-1}|z_N(f)|^2/R_{\tilde{x}}(f) > u) = e^{-u/2}, \quad (3.34)$$

$$P(\log N^{-1}|z_N(f)|^2 > \log R_{\tilde{x}}(f) - \log 2 + \log u) = e^{-u/2}, \quad (3.35)$$

$$P(\log N^{-1}|z_N(f)|^2 < \log R_{\tilde{x}}(f) - \log 2 + \log u) = 1 - e^{-u/2}. \quad (3.36)$$

Hence, $N^{-1}|z_N(f)|^2$ is not a consistent estimator³ of $R_{\tilde{x}}(f)$.

We now turn to the case when the sequence x_k has been sampled at a sampling rate of $1/\Delta u$, i.e. $x_k = x(k\Delta u)$, $k = 0, 1, \dots, N-1$, for some (continuous) function $x(u)$, $0 \leq u \leq (N-1)d$. Then the basic formula (3.26) remains the same, but the $z_N(f)$ -function is defined for $0 \leq f \leq N\Delta f$, where

$$\Delta f = \frac{1}{N\Delta u}, \quad (3.37)$$

and $NF_j = z_N(j\Delta f) = z_N(\frac{j}{N} \cdot \frac{1}{\Delta u})$.

Relation between DFT of a data vector and a spectral density

We have encountered two basically different Fourier transforms, the power spectral density function $R_{\tilde{x}}(f)$ as Fourier transform of a covariance function $r_{\tilde{x}}(u)$, and the discrete Fourier transform (DFT) $z_N(f_j) = F_j$ of a data sequence. As seen in Theorem 3.2, if data are generated from a stationary random sequence, it is the squared absolute value of the DFT (multiplied by n^{-1}) that converges *on average* towards $R_{\tilde{x}}(f_j)$. To understand what is actually going on, let us consider the process from Section 3.3.2,

$$\tilde{x}(t) = \sum_{j=1}^n A_j \cos(2\pi f_j t + \varphi_j).$$

where, f_1, \dots, f_n are fixed frequencies, the A_j 's independent random amplitudes and the φ_j 's independent random phases, uniformly distributed between 0 and 2π . By 3.18, the power spectral density of this process is

$$R_{\tilde{x}}(f) = \sum_{j=1}^n E(A_j^2/2) \left\{ \frac{1}{2} \delta_{f_j}(f) + \frac{1}{2} \delta_{-f_j}(f) \right\}.$$

We assume $N \geq 2n$ and that the frequencies are exact multiples of $1/N$ of the form $f_j = n_j/N$, $0 \leq n_j \leq N/2$. We can then always extend the series to be of the form

$$\tilde{x}(t) = \sum_{j=0}^{\lfloor N/2 \rfloor} A_j \cos(2\pi f_j t + \varphi_j),$$

for *all* $f_j = j/N$, with $0 \leq j \leq N/2$, (possibly by filling in with some $A_j = 0$).

³A consistent estimator θ_N^* of an unknown quantity θ has the property that $\mathbf{P}(|\theta_N^* - \theta| > \varepsilon) \rightarrow 0$ as $N \rightarrow \infty$.

Now consider $\mathbf{x} = (x_0, x_2, \dots, x_{N-1})^\top$ as a vector of observations of

$$\tilde{\mathbf{x}} = (\tilde{x}(0), \tilde{x}(1), \dots, \tilde{x}(N-1))^\top,$$

i.e. $x_k = \sum_{j=0}^{\lfloor N/2 \rfloor} A_j \cos(2\pi f_j k + \varphi_j)$, for $k = 0, 1, \dots, N-1$. Then, by Example 3.3, the Fourier transform coefficients are

$$\begin{aligned} F_j &= \frac{A_j e^{i\varphi_j} + A_{N-j} e^{-i\varphi_{N-j}}}{2} \\ &= \begin{cases} A_0 \cos \varphi_0, & j = 0, \\ \frac{A_j}{2} e^{i\varphi_j}, & j = 1, \dots, \lfloor N/2 \rfloor, \\ \frac{A_{N-j}}{2} e^{-i\varphi_{N-j}}, & j = \lfloor N/2 \rfloor + 1, \dots, N-1. \end{cases} \end{aligned} \quad (3.38)$$

Now $|F_j|^2 = |F_{N-j}|^2 = A_j^2/4$ for $j = 1, \dots, \lfloor N/2 \rfloor$. Thus, the DFT of the observations recovers the amplitude (and the phases) of the separate terms in the cosine sum, and $|F_j|^2 + |F_{N-j}|^2 = A_j^2/2$. On the other hand, the power spectral density is by 3.18 proportional to $E(A_j^2/2)$, i.e. to the *expectation* of $A_j^2/2$.

Also note the relation between

$$\begin{aligned} |F_0|^2 + \sum_{j=1}^{\lfloor N/2 \rfloor} (|F_j|^2 + |F_{N-j}|^2) &= A_0^2 \cos(\varphi_0)^2 + \sum_{j=1}^{\lfloor N/2 \rfloor} A_j^2/2 \\ \int_{-1/2}^{1/2} R_{\tilde{\mathbf{x}}}(f) df &= E(A_0^2 \cos(\varphi_0)^2) + \sum_{j=1}^{\lfloor N/2 \rfloor} E(A_j^2/2) = \sum_{j=0}^{\lfloor N/2 \rfloor} E(A_j^2/2). \end{aligned}$$

Remark 3.3: The relation between the Fourier transform pair *model covariance function* \leftrightarrow *power spectral density* in the model and the Fourier transform pair *data covariance function* \leftrightarrow *data power spectrum* can be expressed in a rather compact mathematical form. Let \circ denote the *correlation operator* between two sequences, i.e. if $\mathbf{x} = (x_0, \dots, x_{n-1})$, $\mathbf{y} = (y_0, \dots, y_{n-1})$. Then we define a new sequence $\mathbf{z} = \mathbf{x} \circ \mathbf{y} = (z_{-n+1}, \dots, z_{n-1})$ and its Fourier transform by

$$z_k = \sum_j x_j y_{j+k}, \quad \mathcal{F}\mathbf{z} = \mathcal{F}\mathbf{x} \cdot \overline{\mathcal{F}\mathbf{y}}.$$

Another formulation is that \mathbf{z} is the convolution of \mathbf{x} with the *reversed* \mathbf{y} -sequence. The data covariance function and its Fourier transform are then

$$r_{\mathbf{x}} = n^{-1} \mathbf{x} \circ \mathbf{x}, \quad (3.39)$$

$$\mathcal{F}r_{\mathbf{x}} = n^{-1} \mathcal{F}\mathbf{x} \cdot \overline{\mathcal{F}\mathbf{x}} = n^{-1} |\mathcal{F}\mathbf{x}|^2. \quad (3.40)$$

Asymptotically, by (3.8) and (3.33), as $n \rightarrow \infty$, $\mathbf{E}(r_{\mathbf{x}}) \rightarrow r_{\tilde{\mathbf{x}}}$ and $\mathbf{E}(\mathcal{F}r_{\mathbf{x}}) \rightarrow R_{\tilde{\mathbf{x}}} = \mathcal{F}r_{\tilde{\mathbf{x}}}$. \square

Remark 3.4: The reader may have been worried about the ambiguity in the choice of A_j and φ_j . The Fourier coefficients and also the x_k -sequence would have been exactly the same if we had changed some of the A_j 's and φ_j 's, as long as the right hand side of (3.29) was kept the same. One possibility to make the formulation definite is to take the choice of (3.32), i.e. $A_j = A_{N-j}$, $\varphi_j = -\varphi_{N-j}$. Another choice is to assume $A_j = 0$ for $j \geq N/2$, as we did in the calculations leading to (3.38). The ambiguity is of no importance as long as we deal with only one sequence, as we will do when dealing with a single image. However, when we deal with a sequence of images, where each cosine component in (3.28) will have both a time and a space parameter, it will be important to be able to separate the different members of (3.29). We shall return to this in a later chapter. \square

Discrete Fourier transform of 2D data

For a two-dimensional data set, $x_{jk} = x(\mathbf{u}_{jk})$, $\mathbf{u}_{jk} = (u_{jk}^{(1)}, u_{jk}^{(2)})$, where \mathbf{u}_{jk} , $j = 0, \dots, M-1$, $k = 0, \dots, N-1$ form an equidistant grid in the plane, the discrete Fourier transform and inverse transform are defined as

$$F_{jk} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{mn} e^{-i2\pi(jm+kn)/(MN)}, \quad (3.41)$$

$$x_{mn} = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} F_{jk} e^{i2\pi(jm+kn)/(MN)}. \quad (3.42)$$

3.4 Linear filters

Linear and non-linear filters are used for image improvement, to reduce disturbances, for image enhancements, to emphasise certain characteristics such as edges or other simple image elements, and for detection, to identify predefined objects in a disturbed image. Linear filters are simply analysed by statistical means, in particular when it comes to their correlation of spectral properties.

A *mask* is a centred linear filter acting on an image $\mathbf{x}(\mathbf{u})$. It is defined as a convolution between the image and an *impulse response* function h_{jk} , $\mathbf{y} = \mathbf{h} * \mathbf{x}$,

$$y_{jk} = \sum_{m=-M}^M \sum_{n=-N}^N h_{j-m, k-n} x_{m,n}. \quad (3.43)$$

Here M, N may be infinite, but are in image analysis usually finite. The constants $h_{j,k}$ are also called the filter *weights*.

A linear filter can also be defined in the frequency domain, by the *transfer* or *frequency response function*, $H = \mathcal{F}h$,

$$H(j, k) = \sum_m \sum_n h_{m,n} e^{-i2\pi(jm+kn)}. \quad (3.44)$$

Examples of simple image filters are

a point spread function, which is the mathematical formulation of the optical phenomenon of spreading the light aiming at a specific point in the image. The point spread function is often assumed to have a Gaussian shape,

$$h_{jk} = \frac{1}{2\pi\sigma^2} \exp(-(j^2 + k^2)/2\sigma^2),$$

for some spreading parameter σ .

a smoothing filter, which is used to soften contours. Besides the Gaussian filter, a *moving average* can be used. If the weights are all equal the moving average is a proper average, otherwise it is an improper average. An examples of an improper moving average is an average over an approximate circle.

a high boost filter, which are filters that sharpens contours and other high frequency objects.

a matching filter, which is designed to detect specified objects in a signal or image. It can be optimised to best performance in the presence of uncorrelated or correlated noise.

a frequency matched filter, which is a frequency based filter, designed to separate noise from a random image element by using the different frequency characteristics.

Non-linear filters are also used, the *median filter* being the most well know. There the moving average is replaced by a moving median, which can be an effective mean to remove isolated disturbed pixel values.

3.4.1 Random elements in linear filters

All linear filters act on deterministic image elements, as well as on random elements and on random noise. The effect on a random element or noise can be described by its averaging effect and by its effect on the correlation structure.

By elementary properties of expectation, $\mathbf{E}(\mathbf{h} * \tilde{\mathbf{x}}) = \mathbf{h} * \mathbf{E}(\tilde{\mathbf{x}})$, while the covariances in a filtered random structure follow from (3.1). The covariance matrix for the different pixels in an image is very large with $(MN)^2$ elements if the image has $M \times N$

pixels. Since we shall need merely the covariance function of the filtered image, i.e. the covariances between filtered values at two separate pixels, we can use (3.1) on the vectorised image.

For large filters, the best way to get the covariance function after a linear filter is via the power spectral density $R_{\tilde{y}}(\mathbf{f})$, since

$$R_{\tilde{y}}(\mathbf{f}) = |H(\mathbf{f})|^2 R_{\tilde{x}}(\mathbf{f}),$$

$$r_{\tilde{y}} = \mathcal{F}^{-1} (|H|^2 \mathcal{F} r_{\tilde{x}}).$$

Smoothing filter

For a small smoothing moving average filter with coefficients

$$\mathbf{h}_S = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

direct calculation of covariance is possible. It acts as a low pass filter decreasing the amplitude of high frequency oscillations. In practical applications, the more isotropic filter defined by

$$\mathbf{h}_S = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

are usually preferred.

Sharpening (high boost) filter

A derivation filter is an example of a high boost filter, which enhances variation in the image. Several types are used, to get derivative in different directions, often together with some smoothing. Examples from the literature are

$$\mathbf{h}_A = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad \mathbf{h}_B = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix},$$

which approximate the derivatives $\partial x/\partial u_2$ and $\partial x/\partial u_1$ in vertical and horizontal directions, including some smoothing in the perpendicular direction.

The filters

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

approximate the derivatives along the diagonals.

Derivative filters are often used to detect borders between separate image elements. As an example consider a light left half plane with mean intensity 0 and a darker right half plane with mean intensity 1, and on top of that random disturbances which form a homogeneous random (Gaussian) field with mean 0 and covariance function $r_{\tilde{x}}(\mathbf{u})$. Thus $\mathbf{E}(\tilde{x}_{j,k}) = 0, 1$ depending on whether $j \leq -1$ or $j \geq 0$. Then, after a filter of type \mathbf{h}_B , the output $y_{j,k}$ is Gaussian with mean and variance (dropping the subscript on $r_{\tilde{x}}$),

$$\mathbf{E}(y_{j,k}) = \begin{cases} 0, & j \leq -2, \\ -4, & j = -1, \\ -4, & j = 0, \\ 0, & j \geq 1, \end{cases}$$

$$\mathbf{V}(y_{j,k}) = 12r(0, 0) + 2r(2, 0) + 8r(1, 0) \\ - 4r(1, -2) - 2r(2, -2) - 4r(1, 2) - 2r(2, 2).$$

Another popular filter has

$$\mathbf{h}_C = \begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 4 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array},$$

used to approximate the Laplacian, $\nabla^2 x = \partial^2 x / \partial u_1^2 + \partial^2 x / \partial u_2^2$. The variance after this filter is

$$\mathbf{V}(y_{jk}) = 20r(0, 0) + 4r(1, 1) + 4r(1, -1) \\ + 2r(2, 0) + 2r(0, 2) - 8r(1, 0) - 8r(0, 1).$$

We shall return to the detection capabilities in the chapter on random fields and their extremes.

Correlation matching

Another type of filter are the correlation matching filters which are used to detect finite objects of specified form in a blurred image. If the object is defined as $A(\mathbf{u}) = a_{\mathbf{u}}$ for $\mathbf{u} \in \mathbf{u}_0 + \mathbf{i}_A$ and 0 otherwise, then the matching filter has

$$\mathbf{h}_A(j, k) = a_{-j, -k}.$$

If the image only consists of the object plus noise $N(\mathbf{u})$, then the output of the filter is

$$y_{j,k} = \sum_{m,n} h_A(j-m, k-m)N(m, n) + \sum_{\mathbf{u}} a_{\mathbf{u}}^2,$$

when $(j, k) = \mathbf{u}_0$,

$$y_{j,k} = \sum_{m,n} h_A(j-m, k-n)N(m, n),$$

when the mask does not intersect the object.

If the noise has mean zero and zero correlation (we have white noise) then this filter maximises the signal to noise ratio $\mathbf{E}(\tilde{y}_{\mathbf{u}_0})^2 / \mathbf{V}(\tilde{y}_{\mathbf{u}_0})$.

Wiener filter

The Wiener filter is by its design a statistically based filter which aims at removing random elements from a non-random information by a frequency dependent filter. Usually it is formulated in terms of an interesting *random* image element \tilde{s} with known power spectral density $R_{\tilde{s}}$, which is first distorted by a known point spread filter h and then disturbed by an un-interesting random noise \tilde{n} with power spectral density $R_{\tilde{n}}$. The Wiener filter acts on

$$\tilde{x} = h * \tilde{s} + \tilde{n},$$

which has power spectral density

$$R_{\tilde{x}} = |H|^2 R_{\tilde{s}} + R_{\tilde{n}}.$$

and it is defined as the linear filter that minimises the mean square error in the reconstruction of \tilde{s} from \tilde{x} , $\mathbf{E}((\tilde{s} - \hat{\tilde{s}})^2)$. It has the transfer function

$$\frac{1}{H} \frac{|H|^2}{|H|^2 + R_{\tilde{n}}/R_{\tilde{s}}}.$$

Exercises

Exercise 3.1: Variance calculations.

- Suppose that we have two random variables \tilde{x} and \tilde{y} with variances $\mathbf{V}(\tilde{x}) = 9$ and $\mathbf{V}(\tilde{y}) = 25$ and covariance $\mathbf{C}(\tilde{x}, \tilde{y}) = -17$. Calculate $\mathbf{V}(\tilde{x} + \tilde{y})$.
- The random vector $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2)^T$ has a two dimensional normal distribution with $\mathbf{V}(\tilde{x}_1) = \mathbf{V}(\tilde{x}_2) = \sigma^2$ and $\mathbf{C}(\tilde{x}_1, \tilde{x}_2) = r$. Show that $\tilde{x}_1 + \tilde{x}_2$ and $\tilde{x}_1 - \tilde{x}_2$ are uncorrelated. Are they independent?
- Let $(\tilde{x}, \tilde{y})^T \in \mathcal{N}\left(\begin{pmatrix} m_{\tilde{x}} \\ m_{\tilde{y}} \end{pmatrix}, \begin{pmatrix} \sigma^2 & r \\ r & \sigma^2 \end{pmatrix}\right)$. Derive the following expression:

$$\mathbf{C}(a_1\tilde{x} + a_2\tilde{y}, b_1\tilde{x} + b_2\tilde{y}) = \mathbf{a}^T \begin{pmatrix} \sigma^2 & r \\ r & \sigma^2 \end{pmatrix} \mathbf{b},$$

where $\mathbf{a}^T = (a_1, a_2)$, $\mathbf{b}^T = (b_1, b_2)$. Solve the previous exercise using this relation.

Exercise 3.2: Consider an image built up by a rectangular gitter with pixels $(i, j) = \mathbf{u} \in \mathcal{A} = \{(1, 1), (1, 2), \dots, (16, 16)\}$. Each pixel shows a noisy version of the true scene pixel, in other words $\tilde{x}(\mathbf{u}) = s(\mathbf{u}) + \tilde{n}(\mathbf{u})$, $\mathbf{u} \in \mathcal{A}$ where \tilde{n} has covariance function:

$$C(\tilde{n}(\mathbf{u}), \tilde{n}(\mathbf{v})) = \rho^{|\mathbf{u}-\mathbf{v}|} \cdot \sigma^2,$$

where $|\mathbf{u} - \mathbf{v}|$ is the normal Euclidean distance⁴.

- What can you say about this covariance function?

A simple way to estimate the value of $s(\mathbf{u})$, $\mathbf{u} \in \mathcal{A}$ (\mathbf{u} assumed not to be on the boundary of \mathcal{A}) is to put

$$s^*(\mathbf{u}) = s^*(i, j) = \frac{1}{5} (x(i-1, j) + x(i+1, j) + x(i, j) + x(i, j-1) + x(i, j+1))$$

- Calculate the variance of $s^*(\mathbf{u})$ if $\sigma = 1$ and $\rho = 0.5$.

Exercise 3.3: Let each pixel in an image be an observation $x(i, j) = s(i, j) + n(i, j)$, $(i, j) = \mathbf{u} \in \mathcal{A} = \{(1, 1), (1, 2), \dots, (n, n)\}$ of a stochastic system $\tilde{x}(i, j) = \tilde{s}(i, j) + \tilde{n}(i, j)$. We can estimate the value of s at an interior point (i, j) by centering a mask:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

⁴The Euclidean distance between \mathbf{u} and \mathbf{v} is $\sqrt{(u_i - v_i)^2 + (u_j - v_j)^2}$.

at (i, j) and then calculate

$$\begin{aligned} 16s^*(i, j) &= x(i-1, j-1) + 2x(i-1, j) + x(i-1, j+1) \\ &\quad + 2x(i, j-1) + 4x(i, j) + 2x(i, j+1) \\ &\quad + x(i+1, j-1) + 2x(i+1, j) + x(i+1, j+1). \end{aligned}$$

Assume that the covariance function of \tilde{s} is $r_{\tilde{s}}(\mathbf{u}, \mathbf{v}) = \sigma^2 e^{-|\mathbf{v}-\mathbf{u}|^2} = r_{\tilde{s}}(\mathbf{v}-\mathbf{u}) = r_{\tilde{s}}(\boldsymbol{\tau})$ since we chosen an isotropic covariance function. The noise is assumed to have covariance function $r_{\tilde{n}} = \delta_{\mathbf{0}}$. What is the variance of our estimate s^* ? What is the variance of the error $s^* - s$?

Exercise 3.4: A Wiener filter in 1D. Assume that we observe $\tilde{x}_k = \tilde{s}_k + \tilde{n}_k$, $k = 0, \pm 1, \dots$ where \tilde{s}_k is an unknown signal with covariance function

$$r_{\tilde{s}}(k) = \begin{cases} 4, & k = 0, \\ -1, & k = \pm 1, \\ 0, & \text{otherwise,} \end{cases}$$

and the covariance function of the independent noise \tilde{n}_k is

$$r_{\tilde{n}}(k) = \begin{cases} 4, & k = 0, \\ 1, & k = \pm 1, \\ 0, & \text{otherwise.} \end{cases}$$

One way of estimating \tilde{s}_k is to filter the signal through a Wiener filter $H(f)$ which will maximize the signal to noise ratio $\text{SNR} = \frac{\mathbb{E}(\tilde{s}_k)^2}{\mathbb{E}(\tilde{s}_{u_k} + \tilde{n}_{u_k} - \tilde{s}_k)^2}$ where \tilde{s}_{u_k} and \tilde{n}_{u_k} are the filtered signal and filtered noise.

a) Write down the Wiener filter $H(f)$ in this setup.

Taking the inverse Fourier transform of $H(f)$ allows us to write the output $\tilde{y}_k = \sum_{j=-\infty}^{\infty} h_j \cdot (\tilde{s}_{k-j} + \tilde{n}_{k-j})$.

b) Calculate the impulse response h_k .

Chapter 4

Random fields with Markov structure

4.1 Markov random fields

In the previous chapter, we did not use the obvious fact that neighbouring pixels in an image may be highly dependent of each other. This dependence can be described either, as in Chapter 3, by means of a correlation function or a power spectral density, or as we shall describe here, by means of a Markov model. The correlation/spectrum technique is useful to describe seemingly random variations within an otherwise rather homogeneous image element. The influence in a correlation model extends, at least in principle, over the whole image element. The Markov technique is more local and is suited to describe connectiveness and abrupt changes between image elements.

In the Markov approach the relation between different image elements is described by a *Markov random field*. One advantage with this approach is that it is relatively easy to improve a distorted or incompletely observed image by means of a simulation technique, which uses this local dependence.

The distribution of the whole field, including the dependence between neighbouring pixels, is called *the global model* in contrast to *the local model*, which describes the dependence between pixels. Every global model defines a local model, but not every local model is compatible with any global model. This means that one can easily define *impossible* dependence structures, which contain inner conflicts.¹ One way to avoid this is to specify the model as a *Gibbs distribution*, where the local models are consistent, but where the global probability function may not be integrable. In image analysis it is not uncommon that one uses such impossible global models!

We start with a simple model for local dependence in images. As usual we denote

¹This corresponds to the fact that not every matrix can be a covariance matrix, but it has to be non-negative semidefinite. For example, it is impossible that $V(\tilde{x}) = V(\tilde{y}) = V(\tilde{z}) = 1$, $C(\tilde{x}, \tilde{y}) = 0.9$, $C(\tilde{z}, \tilde{y}) = 0.9$, $C(\tilde{x}, \tilde{z}) = -0.9$.

the random pixel values by $\tilde{x}(\mathbf{u})$, where $\mathbf{u} = (u_1, u_2) \in \mathbb{Z}^2$.

4.1.1 Neighbour structures and the Markov condition

A Markov random field is often a useful model for a sampled image with local dependence. To define a Markov random field we need a *neighbour structure* \mathcal{N} , which defines the range of *immediate influence* from one pixel to other pixels. (Note that we talk about the immediate influence, in order to emphasise that the actual influence is not restricted to the neighbour points. There is usually an indirect influence far beyond the immediate neighbours. The precise definition of a Markov field makes this clear.)

Neighbour structures

A neighbour structure is a symmetric relation which defines which pixels are neighbours of a certain pixel \mathbf{u} ,

$$\mathcal{N}(\mathbf{u}) = \{\mathbf{v}; \text{ such that } \mathbf{v} \text{ is a neighbour of } \mathbf{u}\}$$

$$\mathbf{v} \in \mathcal{N}(\mathbf{u}) \leftrightarrow \mathbf{u} \in \mathcal{N}(\mathbf{v}).$$

A *homogeneous* neighbour structure can be defined by a matrix, also denoted by \mathcal{N} , with a centre element, where a value 1 signifies the elements which are neighbours to the centre pixel.

Non-homogeneous neighbour structures define general, non-directed graphs, with edges indicating neighbour relationships. There are numerous ways to represent and handle such models.

Example 4.1. Figure 4.1 shows three different possible neighbour structures.

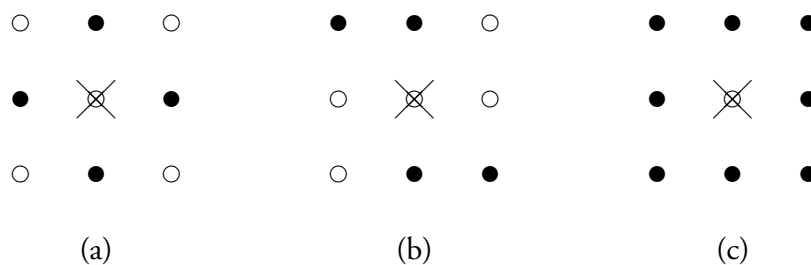


Figure 4.1: Three neighbour definitions. The black points are neighbours to the middle point.

The corresponding matrices are

$$\mathcal{N}_a = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \mathcal{N}_b = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad \mathcal{N}_c = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

□

The Markov condition for random fields

The Markov condition for a Markov random field is stated in terms of the conditional distribution of a pixel value, given the value at *all other points* and how that conditional distribution relates to the conditional distribution given the values at *the neighbour points*. In a Markov random field these conditional distributions are the same.

$$\mathbf{P}(\tilde{x}(\mathbf{u}) = x \mid \tilde{x}(\mathbf{v}), \mathbf{v} \neq \mathbf{u}) = \mathbf{P}(\tilde{x}(\mathbf{u}) = x \mid \tilde{x}(\mathbf{v}), \mathbf{v} \in \mathcal{N}(\mathbf{u})), \quad (4.1)$$

i.e. the conditional distribution of $\tilde{x}(\mathbf{u})$ depends on the rest of the fields only through its neighbours.

Another formulation of the Markov condition is that two pixel values $\tilde{x}(\mathbf{u})$ and $\tilde{x}(\mathbf{v})$ are conditionally independent, given the values at all neighbour points to \mathbf{u} (and also given the values at the neighbour points to \mathbf{v}).

Remark 4.1: The condition (4.1) is a natural generalisation of the condition for discrete time Markov chains. This can be seen as follows. Let $\{\tilde{x}_n\}_{n=0}^{\infty} = \{\tilde{x}(t); t \in \mathbb{Z}\}$ be a Markov chain, and let the immediate time points be the neighbours. Then,

$$\mathbf{P}(\tilde{x}(u) = x \mid \tilde{x}(v), v \neq u) \quad (4.2)$$

$$\begin{aligned} &= \mathbf{P}(\tilde{x}(u) = x \mid \tilde{x}(s) = x_s, s < u, \tilde{x}(t) = x_t, t > u) \\ &= \frac{\mathbf{P}(\tilde{x}(s) = x_s, 0 \leq s < u, \tilde{x}(u) = x, \tilde{x}(t) = x_t, t > u)}{\mathbf{P}(\tilde{x}(s) = x_s, 0 \leq s < u, \tilde{x}(t) = x_t, t > u)}. \end{aligned} \quad (4.3)$$

Here, the nominator is a product of conditional probabilities for events concerning $\{\tilde{x}(s), 0 \leq s < u - 1\}$, $\tilde{x}(u - 1)$, $\tilde{x}(u)$, $\tilde{x}(u + 1)$ $\{\tilde{x}(t), t > u + 1\}$. The Markov property implies that the conditions can be reduced to just the latest one. The denominator is a similar product, but, without $\tilde{x}(u)$. Cancelling common factors, we get that (4.3) is equal to

$$\begin{aligned} &\frac{\mathbf{P}(\tilde{x}(u) = x \mid \tilde{x}(u - 1) = x_{u-1}) \mathbf{P}(\tilde{x}(u + 1) = x_{u+1} \mid \tilde{x}(u) = x)}{\mathbf{P}(\tilde{x}(u + 1) = x_{u+1} \mid \tilde{x}(u - 1) = x_{u-1})} \\ &= \frac{\mathbf{P}(\tilde{x}(u - 1) = x_{u-1}, \tilde{x}(u) = x, \tilde{x}(u + 1) = x_{u+1})}{\mathbf{P}(\tilde{x}(u - 1) = x_{u-1}, \tilde{x}(u + 1) = x_{u+1})} \\ &= \mathbf{P}(\tilde{x}(u) = x \mid \tilde{x}(u - 1) = x_{u-1}, \tilde{x}(u + 1) = x_{u+1}). \end{aligned}$$

This shows that the conditional probability (4.2) only depends on the closest neighbours. The Markov condition (4.1) is therefore the natural two-dimensional generalisation of the one-dimensional Markov condition. \square

4.1.2 Local Markov transition probabilities

In a Markov random field the neighbour influence on a pixel is expressed by local transition probabilities, similar to the transition probabilities in a one-dimensional Markov chain. These local probabilities define the conditional distribution of a pixel value, given the values at all its neighbours, and they are consequences of the global distribution that describes the whole field. In Section 4.3 we shall formulate a general technique to define the global Markov field distributions, and then what conditions the local probabilities need to satisfy. Here we shall only give two examples, one for binary fields with possible values 0 and 1, and one continuous model which can generate a field with Gaussian pixel values.

Example 4.2. Suppose an image can hold only two values, black ($x = 0$) and white ($x = 1$), and assume that the probability of a pixel being black depends on the number of black neighbours. A constant β determines the strength of the dependence.

For the particular neighbour structure defined, write

$$\begin{aligned} W(\mathbf{u}) &= \text{the proportion of black neighbours to } \mathbf{u} \\ &\quad - \text{the proportion of white neighbours to } \mathbf{u}. \end{aligned}$$

In this model

$$P(\text{pixel } \mathbf{u} \text{ is black} \mid \tilde{x}(\mathbf{v}), \mathbf{v} \text{ and } \mathbf{u} \text{ are neighbours}) = \frac{e^{\beta W(\mathbf{u})}}{1 + e^{\beta W(\mathbf{u})}}. \quad (4.4)$$

Here, the larger we take β , the stronger is the dependence. When $\beta = 0$ there is no dependence at all, and the image is a clutter of independent black and white pixels.

Figure 4.2 shows some simulated black and white fields where the local dependence is defined by (4.4). The neighbour structures are those in Example 4.1. The neighbour structure is clearly visible in the point pattern, in particularly in case (b), with its dependence in SE-NW direction. The degree of homogeneity is strongly dependent on the β -value. \square

Example 4.3. In a grey-scale image the dependence between pixels can be described by a Gaussian distribution. The following model is often used for images disturbed by random Gaussian noise. Then the value at a pixel has a Gaussian distribution with an expectation equal to a linear combination of the neighbour values. Let the

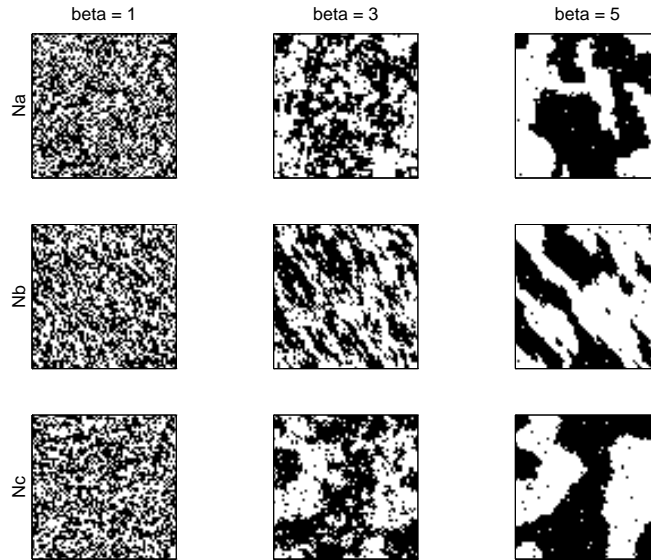


Figure 4.2: Discrete Markov fields with neighbour structures Example 4.1 and different degree of dependence. Probability of black is given by (4.4).

neighbour structure be defined by a neighbour matrix \mathcal{N} as in Example 4.1, such that $N_{\mathbf{u},\mathbf{v}} = 1$ if \mathbf{u} and \mathbf{v} are neighbours and with $N_{\mathbf{u},\mathbf{u}} = 0$, and let

$$C_{\mathbf{u},\mathbf{v}} = \varphi N_{\mathbf{u},\mathbf{v}},$$

for some constant φ which determines the strength of the dependence. In this model

$$\left(\tilde{x}(\mathbf{u}) \mid \tilde{x}(\mathbf{v}) = x(\mathbf{v}), \mathbf{u} \text{ and } \mathbf{v} \text{ are neighbours} \right) \in \mathcal{N} \left(\sum_{\mathbf{v}} C_{\mathbf{u},\mathbf{v}} x(\mathbf{v}), \sigma_p^2 \right), \quad (4.5)$$

i.e. the value at \mathbf{u} has a Gaussian distribution with mean $\sum_{\mathbf{v}} C_{\mathbf{u},\mathbf{v}} x(\mathbf{v})$ and a certain standard deviation σ_p , independent of location. The parameter φ has to be chosen with some care – for a too large value the field becomes unstable and there will be no global distribution (4.5) that makes the field homogeneous. A necessary condition for stability is that $1/\varphi$ has to be larger than the number of neighbour points \mathcal{N} .

This model is called a *CAR-model*, (Conditional Auto Regression). Figure 4.3 shows simulated Markov fields with Gaussian grey-scale values. The neighbour structure is as in Example 4.1 and the diagonal structure is obvious in model (b). \square

4.2 Gaussian Markov random fields

In Section 4.3, the general theory of Markov random fields will be investigated, but first we will discuss the special case of Gaussian Markov random fields (GMRF:s).

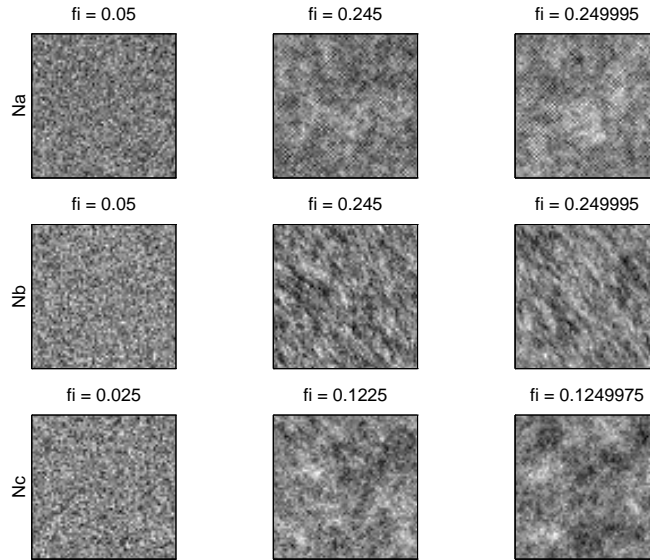


Figure 4.3: Gaussian Markov fields according to (4.5) with neighbour structures as in Example 4.1; the dependence is defined by the value of φ .

This is an important subclass, with applications ranging from spatial statistics to stochastic shape modelling and estimation. For a thorough approach to Gaussian Markov random fields, the reader is referred to Rue and Held (2005), the book that inspired much of this material. An efficient computer implementation of the computational methods resulting from using the sparse matrices used to define Gaussian Markov random field models is available in the free software library GMRFLib (<http://www.math.ntnu.no/~hrue/GMRFLib/>).

Throughout the section, the column stacked version of an image \mathbf{x} will be denoted \mathbf{X} , and the number of nodes in \mathbf{X} is denoted N . For an $m \times n$ image, $N = mn$.

4.2.1 Basic properties

Definition 4.1. A Gaussian Markov random field (GMRF) $\tilde{\mathbf{X}}$ is a random field with probability density

$$p(\mathbf{X}) = \frac{|\mathbf{Q}|^{1/2}}{(2\pi)^{N/2}} \exp\left(-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^T \mathbf{Q}(\mathbf{X} - \boldsymbol{\mu})\right),$$

for some column vector $\boldsymbol{\mu}$ and a sparse, symmetric, and positive (semi-)definite matrix \mathbf{Q} . In other words, $\tilde{\mathbf{X}} \in N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$. The sparsity structure of the precision matrix \mathbf{Q} determines the neighbourhood system for the Markov random field. \square

The full conditional density of X_i given all other X_j is given by

$$p(X_i|X_j, j \neq i) = \frac{Q_{ii}^{1/2}}{\sqrt{2\pi}} \exp \left(-\frac{Q_{ii}}{2} \left(X_i - \mu_i + \sum_{j \neq i} \frac{Q_{ij}}{Q_{ii}} (X_j - \mu_j) \right)^2 \right),$$

so that

$$(X_i|X_j, j \neq i) \in \mathcal{N} \left(\mu_i - \sum_{j \neq i} \frac{Q_{ij}}{Q_{ii}} (X_j - \mu_j), Q_{ii}^{-1} \right).$$

The conditional distribution depends only on X_j for which Q_{ij} is non-zero. This means that the Markov property $p(X_i|X_j, j \neq i) = p(X_i|X_j, j \in \mathcal{N}_i)$ is fulfilled, for the neighbour structure defined by $\mathcal{N}_i = \{j; Q_{ij} \neq 0, j \neq i\}$.

For an image of size $n \times n$, the precision matrix is of size $n^2 \times n^2$. However, for a neighbour system with at most k neighbours per pixel, \mathbf{Q} has $\mathcal{O}(kn^2)$ elements, making storage feasible even for large n . Unfortunately, the inverse of a sparse matrix is in general a dense matrix, making it unfeasible to compute and store the entire covariance matrix, $\mathbf{\Sigma} = \mathbf{Q}^{-1}$. This makes it necessary to exploit the sparsity structure of \mathbf{Q} as much as possible, even in situations where $\mathbf{\Sigma}$ would normally have been used.

A simple example

A simple GMRF structure is defined by $\boldsymbol{\mu} = \mathbf{0}$ and precision parameters

$$\mathbf{q} = \begin{bmatrix} & -1 & & \\ -1 & 4 + \alpha & -1 & \\ & -1 & & \end{bmatrix},$$

that are interpreted as follows. The precision matrix elements $Q_{i,j}$ are -1 for the four nearest neighbours j of each pixel i , and $Q_{i,i} = 4 + \alpha$, so that the conditional expectation of X_i given its neighbours is $\frac{1}{4+\alpha} \sum_{j \in \mathcal{N}_i} X_j$, and the conditional variance is $\frac{1}{4+\alpha}$.

4.2.2 Simulation

The common method for generating samples from a Gaussian distribution is to compute the Cholesky factorisation of the covariance matrix, $\mathbf{\Sigma}$, and multiply with a vector of independent samples from $\mathcal{N}(0, 1)$. Here, we do not have access to $\mathbf{\Sigma}$ directly, but we can apply a similar principle to the precision matrix \mathbf{Q} instead.

The Cholesky factor \mathbf{R} of \mathbf{Q} is an upper right triangular matrix with non-negative diagonal elements, fulfilling

$$\mathbf{Q} = \mathbf{R}^T \mathbf{R}.$$

A unique matrix with these properties exists for every positive (semi-)definite matrix \mathbf{Q} . In addition, if the order of the components in \mathbf{X} is adapted to the neighbourhood system, \mathbf{R} will also be a sparse matrix. Now, let $\tilde{\mathbf{E}} \in \mathcal{N}(\mathbf{0}, \mathbf{I}_N)$ be independent Gaussian noise with unit variance, and compute

$$\mathbf{X} = \boldsymbol{\mu} + \mathbf{R}^{-1}\mathbf{E}$$

for a sample \mathbf{E} from $\tilde{\mathbf{E}}$. The matrix inverse itself needs not be computed, since the triangular structure allows the expression to be computed using simple back-substitution. The expectation of $\tilde{\mathbf{X}}$ is

$$\mathbf{E}(\boldsymbol{\mu} + \mathbf{R}^{-1}\tilde{\mathbf{E}}) = \boldsymbol{\mu} + \mathbf{R}^{-1}\mathbf{E}(\tilde{\mathbf{E}}) = \boldsymbol{\mu},$$

and the covariance is

$$\begin{aligned} \mathbf{C}(\tilde{\mathbf{X}} - \boldsymbol{\mu}, \tilde{\mathbf{X}} - \boldsymbol{\mu}) &= \mathbf{E}((\tilde{\mathbf{X}} - \boldsymbol{\mu})(\tilde{\mathbf{X}} - \boldsymbol{\mu})^\top) \\ &= \mathbf{E}(\mathbf{R}^{-1}\tilde{\mathbf{E}}\tilde{\mathbf{E}}^\top\mathbf{R}^{-\top}) = \mathbf{R}^{-1}\mathbf{E}(\tilde{\mathbf{E}}\tilde{\mathbf{E}}^\top)\mathbf{R}^{-\top} \\ &= \mathbf{R}^{-1}\mathbf{R}^{-\top} = (\mathbf{R}^\top\mathbf{R})^{-1} = \mathbf{Q}^{-1}, \end{aligned}$$

which means that $\tilde{\mathbf{X}} \in \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}^{-1})$, as intended. Even if several samples are needed, the \mathbf{R} matrix need only be computed once, so the additional cost of making more than one sample is small.

4.2.3 Block conditioning

As seen above, the conditional distribution of a single pixel given all others is simple to express in terms of the GMRF parameter matrices. In addition, the expressions for the conditional distribution of several pixels is almost as simple.

Partition the field into two sub-groups, \mathbf{X}_1 and \mathbf{X}_2 , and partition the expectation and precision matrix accordingly:

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} \in \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}^{-1} \right).$$

The conditional distribution of $\mathbf{X}_1|\mathbf{X}_2$ is then given by

$$(\mathbf{X}_1|\mathbf{X}_2) \in \mathcal{N}(\boldsymbol{\mu}_1 - \mathbf{Q}_{11}^{-1}\mathbf{Q}_{12}(\mathbf{X}_2 - \boldsymbol{\mu}_2), \mathbf{Q}_{11}^{-1}).$$

Note in particular that the conditional precision is simply the sub-block of \mathbf{Q} corresponding to \mathbf{X}_1 , while the corresponding expression in terms of sub-blocks of the covariance matrix is much more complicated. As before, the matrix inverse \mathbf{Q}_{11}^{-1} need not be computed, since back-substitution with the Cholesky factorisation can be used instead, by noting that with $\mathbf{Q}_{11} = \mathbf{R}_{11}^\top\mathbf{R}_{11}$, the inverse is given by $\mathbf{Q}_{11}^{-1} = \mathbf{R}_{11}^{-1}(\mathbf{R}_{11}^\top)^{-1}$.

4.2.4 Soft constraints

We now assume that only a small number of linear combinations of $\tilde{\mathbf{X}}_i$ can be observed, denoted $\tilde{\mathbf{Y}} = \mathbf{A}\tilde{\mathbf{X}} + \tilde{\mathbf{E}}$, where $\tilde{\mathbf{E}} \in \mathcal{N}(\mathbf{0}, \Sigma_E)$ is measurement noise and \mathbf{A} is a $K \times N$ weight matrix. If the field is measured at locations i_1, \dots, i_K , the matrix \mathbf{A} will be zero except for $A_{i,i_k} = 1$.

With $\tilde{\mathbf{X}} \in \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}^{-1})$, the conditional distribution of $\tilde{\mathbf{X}}|\mathbf{Y}$ is also Gaussian, with conditional precision matrix

$$\mathbf{Q}_{\tilde{\mathbf{X}}|\mathbf{Y}} = \mathbf{Q} + \mathbf{A}^\top \Sigma_E^{-1} \mathbf{A},$$

and conditional expectation

$$\boldsymbol{\mu}_{\tilde{\mathbf{X}}|\mathbf{Y}} = \mathbf{E}(\tilde{\mathbf{X}}|\mathbf{Y}) = \mathbf{Q}_{\tilde{\mathbf{X}}|\mathbf{Y}}^{-1}(\mathbf{Q}\boldsymbol{\mu} + \mathbf{A}^\top \Sigma_E^{-1} \mathbf{Y}).$$

If the conditional precision matrix is sparse, the expressions above can be used both to compute the conditional expectation and to simulate samples from the conditional distribution, via the sparse Cholesky factorisation of $\mathbf{Q}_{\tilde{\mathbf{X}}|\mathbf{Y}}$. However, if $\mathbf{Q}_{\tilde{\mathbf{X}}|\mathbf{Y}}$ is dense, the formulae above have prohibitive memory and computational requirements. Fortunately, if $K \ll N$, it is still feasible to compute the conditional expectation, as well as to simulate samples from the conditional distribution. Using the matrix inversion lemma, it can be shown that

$$\boldsymbol{\mu}_{\tilde{\mathbf{X}}|\mathbf{Y}} = \boldsymbol{\mu} - \mathbf{Q}^{-1} \mathbf{A}^\top (\mathbf{A} \mathbf{Q}^{-1} \mathbf{A}^\top + \Sigma_E)^{-1} (\mathbf{A}\boldsymbol{\mu} - \mathbf{Y}),$$

which only requires computation of a sparse Cholesky factor and the inverse of a $K \times K$ -matrix:

1. Compute \mathbf{R} such that $\mathbf{Q} = \mathbf{R}^\top \mathbf{R}$.
2. Calculate $\mathbf{B} = \mathbf{Q}^{-1} \mathbf{A}^\top = \mathbf{R}^{-1} (\mathbf{R}^\top)^{-1} \mathbf{A}^\top$ through back-substitution.
3. Compute \mathbf{R}_2 such that $\mathbf{A}\mathbf{B} + \Sigma_E = \mathbf{R}_2^\top \mathbf{R}_2$.
4. Compute $\mathbf{C} = \mathbf{B}(\mathbf{A}\mathbf{B} + \Sigma_E)^{-1} = \mathbf{B}\mathbf{R}_2^{-1} (\mathbf{R}_2^\top)^{-1}$ through back-substitution.
5. Compute $\hat{\tilde{\mathbf{X}}} = \mathbf{E}(\tilde{\mathbf{X}}|\mathbf{Y}) = \boldsymbol{\mu} - \mathbf{C}(\mathbf{A}\boldsymbol{\mu} - \mathbf{Y})$.

Note that the size of \mathbf{B} and \mathbf{C} is $N \times K$, and that the size of $\mathbf{A}\mathbf{B} + \Sigma_E$ is $K \times K$, so that none of the stored dense matrices are of size $N \times N$.

By replacing \mathbf{Y} with a sample from $\mathcal{N}(\mathbf{Y}, \Sigma_E)$, and $\boldsymbol{\mu}$ with a sample from $\mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}^{-1})$ in the above algorithm, a sample from the conditional distribution $\tilde{\mathbf{X}}|\mathbf{Y}$ is generated. This affects only step 5 in the algorithm, making it almost as easy to generate conditional samples as unconditional samples, by storing and reusing \mathbf{R} and \mathbf{C} .

The above procedures make it possible to deal with the conditional distribution even for cases where the conditional precision matrix is dense. For example, if the sum of the field is observed, with unit noise variance, $\mathbf{Q}_{\tilde{\mathbf{X}}|\mathbf{Y}} = \mathbf{Q} + \mathbf{1}\mathbf{1}^\top$, a completely dense matrix, but we only have $K = 1$ in the above algorithm.

4.2.5 Intrinsic random fields

Let \mathbf{Q} be a positive semi-definite (sparse) matrix with rank $N - r$. An *Intrinsic Gaussian markov random field* (IGMRF) is a field \mathbf{X} with “density”

$$p(\mathbf{X}) = \frac{|\mathbf{Q}|_*^{1/2}}{(2\pi)^{(n-r)/2}} \exp\left(-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^\top \mathbf{Q}(\mathbf{X} - \boldsymbol{\mu})\right),$$

where $|\mathbf{Q}|_*$ denotes the *generalised determinant* of \mathbf{Q} , defined as the product of all non-zero eigenvalues of \mathbf{Q} . The “density” function is not a true density, since it is invariant to adding multiples of the eigenvectors corresponding to the zero eigenvalues of \mathbf{Q} to the field \mathbf{X} , and therefore cannot be normalised to integrate to 1. The *rank deficiency* r is the *order* of the IGMRF.

The generalised determinant behaves similarly to the ordinary determinant in most cases, but care needs to be taken if the rank of \mathbf{Q} changes. A simple, but important, computational rule is that $|\lambda\mathbf{Q}|_* = \lambda^{N-r}|\mathbf{Q}|_*$ for all $\lambda \neq 0$.

An IGMRF does not have an expectation (since the density cannot be integrated), so the interpretation of $\boldsymbol{\mu}$ and \mathbf{Q} is not quite the standard one. Here, we will only use intrinsic fields with $\boldsymbol{\mu} = \mathbf{0}$.

Increments

A useful way to construct an IGMRF is to use *increments*. This typically means that discrete versions of linear differential operators are constructed for each pixel, and combined to form a precision matrix with the desired invariances. The main models are the regular 1:st and 2:nd order IGMRF models, defined by the increments

$$\begin{bmatrix} -1 & 1 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

respectively. The 1:st order increments approximate the first order derivatives of the field, and the second order increment approximates the Laplacian of the field (the sum of the second derivatives). In the 2:nd order case, special increments are needed at the field boundary:

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$$

for the horizontal and vertical edges, respectively, and for the corners.

The increment coefficients are gathered in a matrix \mathbf{W} , where each row holds the weights for a single increment, and the IGMRF precision is then defined as $\mathbf{Q} = \mathbf{W}^T \mathbf{W}$. It is straightforward to check that the field using only the first order increments is invariant to addition of a constant to the entire field. Similarly, the second order field is invariant to addition of a plane of arbitrary slope.

Intrinsic GMRF:s are useful in environmental statistics, where slowly varying features can be dealt with as invariant effects in the model.

4.2.6 Parameter estimation

The general problem of estimating both $\boldsymbol{\mu}$ and the elements of \mathbf{Q} is difficult, even if the neighbourhood structure is assumed known. Recent approaches include finding parameter values that make the covariances of the GMRF approximate the covariances generated by some parametric covariance function.

Assume that $\boldsymbol{\mu} = \mathbf{0}$, and that the precision matrix is known up to an unknown scaling factor, χ . The likelihood function is given by

$$\begin{aligned} L(\chi; \mathbf{X}) &\propto p(\mathbf{X}|\chi) \\ &= \frac{|\chi \mathbf{Q}_*|^{1/2}}{(2\pi)^{(n-r)/2}} \exp\left(-\frac{1}{2} \mathbf{X}^T (\chi \mathbf{Q}) \mathbf{X}\right) \\ &= \frac{\chi^{(N-r)/2} |\mathbf{Q}_*|^{1/2}}{(2\pi)^{(n-r)/2}} \exp\left(-\frac{\chi}{2} \mathbf{X}^T \mathbf{Q} \mathbf{X}\right). \end{aligned}$$

The ML-estimate of χ can be found by taking the logarithm and differentiating with respect to χ . Note in particular that the (generalized) determinant of \mathbf{Q} does not need to be calculated.

4.3 Gibbs distributions

4.3.1 The Gibbs distribution

The Markov random field models (4.4) and (4.5) are but two examples of a general model in spatial statistics and statistical physics; from its originator, J. Willard Gibbs, it is called a *Gibbs distribution*.² A Gibbs distribution describes the global distribution of the whole field.

A Gibbs distribution always generates a Markov random field. The converse is also true – every Markov random field can be expressed by means of a Gibbs distribution. The neighbour structures and local transition distributions we encountered in

²J. Willard Gibbs, American physicist, one of the great names in mathematical physics. He derived many of the laws of thermodynamics and their implications for chemical reactions.

Section 4.1 imply specific Gibbs distributions for the global field distribution. Other types of Gibbs distributions can be constructed modifying the dependence by varying the weights given to the different neighbour points, i.e. $N_{\mathbf{u},\mathbf{v}}$ need not be either 0 or 1.

A general Gibbs distribution is a probability distribution for a whole random field or an even more general structure. We denote the possible states (outcome) of the field by the letter \mathbf{x} ; this is usually a vector of large dimension, $\mathbf{x} = \{x_{\mathbf{u}}\}$. In image language, $x_{\mathbf{u}}$ specifies the value at pixel \mathbf{u} .

In a Gibbs distribution the probability for an outcome \mathbf{x} is defined by an expression

$$\mathbf{P}(\tilde{\mathbf{x}} = \mathbf{x}) = \frac{e^{-W(\mathbf{x})/kT}}{Z}, \quad (4.6)$$

where $Z = \sum_{\mathbf{x}} \exp(-W(\mathbf{x})/kT)$ is a normalising constant. The function $W(\mathbf{x})$ is in statistical physics called the *energy* of the state \mathbf{x} . The parameter T is called the *temperature*, and it acts to change the relative likelihood of different states. If $T \rightarrow \infty$ all states will become equally likely. The constant k is introduced in statistical physics to give the other quantities physical meaning. In probability theory, both k and T can be incorporated in the function $W(\mathbf{x})$.

Of course, every probability distribution with a finite number of possible outcomes can be written in the form (4.6). To be called a Gibbs distribution the energy $W(\mathbf{x})$ has to be defined in a special way, very closely related to the Markov random field definition.

4.3.2 Gibbs distributions and Markov random fields

One can define the local structure of a Markov random field $\tilde{\mathbf{x}}(\mathbf{u})$, $\mathbf{u} \in \mathbb{Z}^2$ via a more restricted definition of a Gibbs distribution, as follows.

Cliques

Besides the neighbouring structure, the concept of *cliques* is central for the construction of dependence between pixels. Each neighbour structure will have its own well defined set of cliques.

By one clique is meant a subset \mathcal{C} of points in \mathbb{Z}^2 such that if \mathbf{u} and \mathbf{v} both belong to \mathcal{C} then they must be neighbours. Single points and the empty set are also called cliques. For a specified neighbouring structure, the cliques will all have very typical forms, determined by the neighbour structure. The set of all cliques will be used in the definition of the Gibbs distribution.

Example 4.4. All the possible clique shapes for the neighbour structures in Example 4.1 are shown in Figure 4.4. The cliques are all combinations of points located in any of the patterns shown. \square

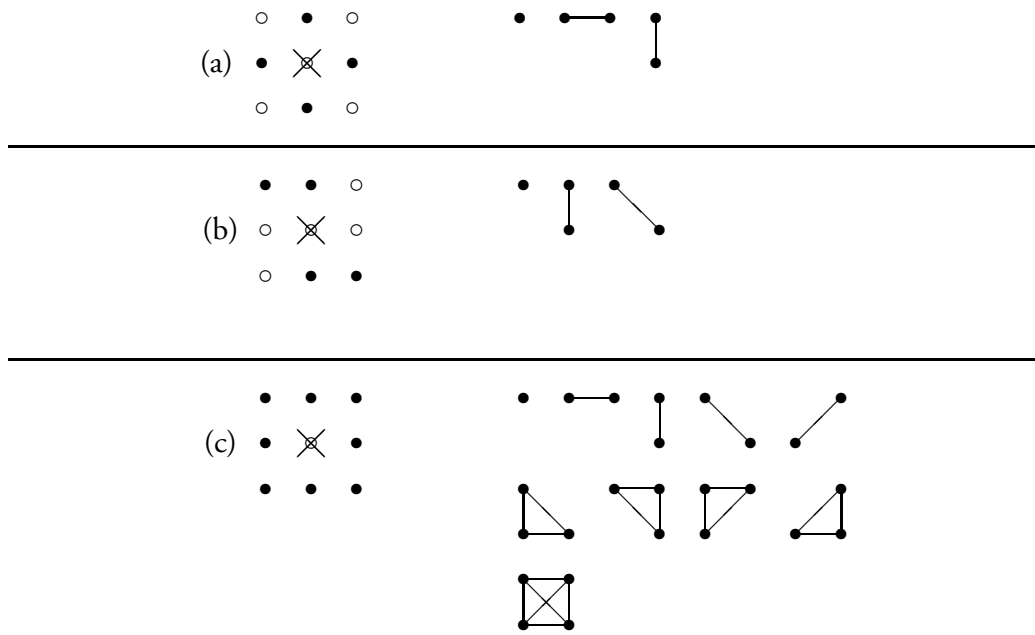


Figure 4.4: The cliques (right) for three neighbour definitions (left).

Cliques, potentials, and the Gibbs distribution

Each clique \mathcal{C} in a point net can be assigned a *potential*, a terms also borrowed from statistical physics. The potential for a clique \mathcal{C} in a field \mathbf{x} is denoted by $V_{\mathcal{C}}(\mathbf{x})$, and the total energy of the field is then the negative sum of all potentials over all cliques,

$$W(\mathbf{x}) = - \sum_{\mathcal{C}} V_{\mathcal{C}}(\mathbf{x}).$$

Finally, a Gibbs distribution with a finite number of possible states \mathbf{x} on a finite subset of \mathbb{Z}^2 has the probability function

$$p(\mathbf{x}) = \frac{e^{-W(\mathbf{x})}}{Z} \tag{4.7}$$

where $Z = \sum_{\omega} e^{-W(\omega)}$. The state space for each pixel, Λ , need in general not be either finite or countable, but the theory is simpler for finite state spaces.

A Gibbs distribution defined by the potentials of the clique generated by a neighbourhood structure is a Markov random field, with local dependence as in the following theorem.

Theorem 4.1. *Let $\{\tilde{\mathbf{x}}_{\mathbf{u}}, \mathbf{u} \in \mathbf{T}\}$ be a random field defined on a finite subset $\mathbf{T} \subset \mathbb{Z}^2$. If $\tilde{\mathbf{x}}_{\mathbf{u}}$ has a Gibbs distribution with probability function*

$$p(\mathbf{x}) = \frac{\exp(-W(\mathbf{x}))}{\sum_{\mathbf{x}'} \exp(-W(\mathbf{x}'))}, \tag{4.8}$$

then

$$\mathbf{P}(\tilde{\mathbf{x}}_{\mathbf{u}} = \mathbf{x}_{\mathbf{u}} \mid \tilde{\mathbf{x}}_{\mathbf{v}}, \mathbf{v} \neq \mathbf{u}) = \frac{\exp \left\{ \sum_{\mathcal{C} \ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}) \right\}}{\sum_{\mathbf{x}'} \exp \left\{ \sum_{\mathcal{C} \ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}') \right\}} \quad (4.9)$$

$$= \mathbf{P}(\tilde{\mathbf{x}}_{\mathbf{u}} = \mathbf{x}_{\mathbf{u}} \mid \tilde{\mathbf{x}}_{\mathbf{v}}, \mathbf{v} \text{ and } \mathbf{u} \text{ are neighbours}). \quad (4.10)$$

The sum $\sum_{\mathcal{C} \ni \mathbf{u}} V_{\mathcal{C}}$ is over all cliques \mathcal{C} that contain \mathbf{u} . Hence $\tilde{\mathbf{x}}_{\mathbf{u}}$ is a Markov random field.

Proof: The conditional probability on the left hand side of (4.9) is equal to the ratio

$$p(\mathbf{x}) / \sum_{\mathbf{x}'_{\mathbf{u}} \in \Lambda} p(\mathbf{x}'),$$

where the sum in the denominator is taken over all fields \mathbf{x}' which are equal to \mathbf{x} except possibly for the value at \mathbf{u} i.e. $\mathbf{x}'_{\mathbf{v}} = \mathbf{x}_{\mathbf{v}}$ for all $\mathbf{v} \neq \mathbf{u}$. With the special form of the Gibbs distribution in (4.8), we obtain

$$\begin{aligned} \frac{p(\mathbf{x})}{\sum_{\mathbf{x}'_{\mathbf{u}} \in \Lambda} p(\mathbf{x}')} &= \frac{\exp \left(\sum_{\mathcal{C} \ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}) + \sum_{\mathcal{C} \not\ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}) \right)}{\sum_{\mathbf{x}'_{\mathbf{u}} \in \Lambda} \exp \left(\sum_{\mathcal{C} \ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}') + \sum_{\mathcal{C} \not\ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}') \right)} \\ &= \frac{\exp \left(\sum_{\mathcal{C} \ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}) \right)}{\sum_{\mathbf{x}'_{\mathbf{u}} \in \Lambda} \exp \left(\sum_{\mathcal{C} \ni \mathbf{u}} V_{\mathcal{C}}(\mathbf{x}') \right)}, \end{aligned}$$

since \mathbf{x} and \mathbf{x}' are identical, except for a possible difference at \mathbf{u} , and therefore $V_{\mathcal{C}}(\mathbf{x}) = V_{\mathcal{C}}(\mathbf{x}')$ if the clique \mathcal{C} does not contain \mathbf{u} . The final expression is equal to the right hand side of (4.9), and since the union of the cliques containing \mathbf{u} only contain neighbours of \mathbf{u} , and \mathbf{u} itself, we have proved (4.10). \square

Every Gibbs distribution of the form (4.6) defines a Markov random field, as we have just shown. The converse is also true, but that is harder to prove. The reader is referred to Kinderman and Snell (1980), which contains further examples on the relation between Gibbs distributions and Markov random fields. Our presentation in the theorem follows in principle that of Kinderman and Snell (1980).

Example 4.5. We assume some neighbour structure, and use cliques of order ≤ 2 . We let the pixels take values on \mathbb{R} , and let $V_{\mathbf{u}}(x_{\mathbf{u}}) = ax_{\mathbf{u}}^2$, and $V_{\mathbf{u},\mathbf{v}}(x_{\mathbf{u}}, x_{\mathbf{v}}) = bx_{\mathbf{u}}x_{\mathbf{v}}$, for some constants a and b . The probability density for \mathbf{x} is

$$\begin{aligned} p(\mathbf{x}) &\propto \exp \left(\sum_{\mathbf{v}} V_{\mathbf{v}}(x_{\mathbf{v}}) + \frac{1}{2} \sum_{\mathbf{v}} \sum_{\mathbf{w} \in \mathcal{N}(\mathbf{v})} V_{\mathbf{v},\mathbf{w}}(x_{\mathbf{v}}, x_{\mathbf{w}}) \right) \\ &= \exp \left(a \sum_{\mathbf{v}} x_{\mathbf{v}}^2 + \frac{b}{2} \sum_{\mathbf{v}} \sum_{\mathbf{w} \in \mathcal{N}(\mathbf{v})} x_{\mathbf{v}}x_{\mathbf{w}} \right), \end{aligned}$$

where the factor $\frac{1}{2}$ is due to the fact that the second sum includes every two-pixel clique twice. We compute the local, conditional density:

$$p(x_{\mathbf{u}} | \mathbf{x}_{(\mathbf{u})}) \propto \exp \left(ax_{\mathbf{u}}^2 + \frac{b}{2} x_{\mathbf{u}} \sum_{\mathbf{w} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{w}} \right)$$

We simplify the expression for a Gaussian density with mean μ and variance σ^2 , and obtain

$$\begin{aligned} p(y) &\propto \exp \left(-\frac{1}{2\sigma^2} (y - \mu)^2 \right) \\ &\propto \exp \left(-\frac{1}{2\sigma^2} y^2 + \frac{\mu}{\sigma^2} y \right), \end{aligned}$$

so that the Gibbs distribution for the Markov field results in Gaussian conditional distributions, with $a = -1/(2\sigma^2)$ and $b = \mu/\sigma^2$ (i.e. $\mu = -2ab$ and $\sigma^2 = -2a$). \square

Examples of local probabilities

Example 4.6. In the example with a binary image, the conditional probability (4.4) that a pixel \mathbf{u} is black, can be written as

$$\begin{aligned} &\mathbf{P}(\tilde{x}_{\mathbf{u}} = 0 \mid \tilde{x}_{\mathbf{v}}, \mathbf{v} \text{ and } \mathbf{u} \text{ are neighbours}) \\ &= \frac{\exp(\beta \#\{\text{black neighbours}\})}{\exp(\beta \#\{\text{black neighbours}\}) + \exp(\beta \#\{\text{white neighbours}\})}, \end{aligned}$$

which can be obtained from a Gibbs distribution where the potential for a pair of neighbouring points is β if they are of the same colour, and 0 otherwise,

$$V_{\mathbf{u},\mathbf{v}}(x_{\mathbf{u}}, x_{\mathbf{v}}) = \begin{cases} \beta & \text{if } x_{\mathbf{v}} = x_{\mathbf{u}}, \\ 0 & \text{if } x_{\mathbf{v}} \neq x_{\mathbf{u}}. \end{cases}$$

This model is symmetric and will produce, on the average, equally many black as white points. Cliques which are singletons, have the same potential regardless of their colour.

A more general model, with different probabilities for black and white, is obtained by letting

$$V_{\mathbf{u}}(x_{\mathbf{u}}) = \begin{cases} \alpha_1 & \text{if } x_{\mathbf{u}} = 1, \\ \alpha_0 & \text{if } x_{\mathbf{u}} = 0. \end{cases}$$

Then

$$\begin{aligned} \mathbf{P}(\tilde{x}_{\mathbf{u}} = 0 \mid \tilde{x}_{\mathbf{v}}, \mathbf{v} \text{ and } \mathbf{u} \text{ are neighbours}) \\ = \frac{\exp(\alpha_0 + \beta \#\{\text{black neighbours}\})}{\exp(\alpha_0 + \beta \#\{\text{black neighbours}\}) + \exp(\alpha_1 + \beta \#\{\text{white neighbours}\})} \end{aligned}$$

is the local distribution for the Gibbs distribution defined through $V_{\mathbf{u}}$ and $V_{\mathbf{u},\mathbf{v}}$. \square

Example 4.7. For images with continuous pixel values, a CAR (Conditional Auto Regression) Gaussian distribution may be a useful model. The conditional expectation is assumed to be a linear combination of neighbouring pixel values,

$$\mathbf{E}(\tilde{x}_{\mathbf{u}} \mid \tilde{x}_{\mathbf{v}}, \mathbf{v} \neq \mathbf{u}) = \mu_{\mathbf{u}} + \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} c_{\mathbf{v},\mathbf{u}}(\tilde{x}_{\mathbf{v}} - \mu_{\mathbf{v}}),$$

and the conditional variance is assumed to be constant,

$$\mathbf{V}(\tilde{x}_{\mathbf{u}} \mid \tilde{x}_{\mathbf{v}}, \mathbf{v} \neq \mathbf{u}) = \sigma^2.$$

In the CAR model the local probability density is

$$\begin{aligned} p(x_{\mathbf{u}} \mid \tilde{x}_{\mathbf{v}} = x_{\mathbf{v}}, \mathbf{v} \neq \mathbf{u}) \\ = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} \left[x_{\mathbf{u}} - \left(\mu_{\mathbf{u}} + \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} c_{\mathbf{v},\mathbf{u}}(x_{\mathbf{v}} - \mu_{\mathbf{v}})\right)\right]^2\right). \end{aligned}$$

\square

Example 4.8. Other useful models are the SAR (Simultaneous Auto Regression) or the Geman-McClure prior distribution. A Geman-McClure prior density

$$p(x_{\mathbf{u}} \mid x_{\mathbf{v}}, \mathbf{v} \neq \mathbf{u})$$

is proportional to

$$\exp\left(\sum_{\mathbf{v}} \frac{\beta_{\mathbf{v},\mathbf{u}}}{1 + \left(\frac{x_{\mathbf{u}} - x_{\mathbf{v}}}{\delta}\right)^2}\right)$$

where δ and $\beta_{\mathbf{v},\mathbf{u}}$ are parameters. \square

4.4 Estimation

For convenience, we introduce the notation $\mathbf{x}_{(\mathbf{u})} = \{x_{\mathbf{v}}, \mathbf{v} \in \mathcal{N}(\mathbf{u})\}$ for all field nodes, or pixels, except for \mathbf{u} .

4.4.1 Iterated Conditional Modes

We assume that the true reality, $\tilde{\mathbf{x}}$, behaves as a random Markov field, with local distribution defined by $p(\tilde{\mathbf{x}}_{\mathbf{u}} = \mathbf{x}_{\mathbf{u}} | \tilde{\mathbf{x}}_{(\mathbf{u})} = \mathbf{x}_{(\mathbf{u})})$. We obtain a measurement \mathbf{y} of \mathbf{x} , where each pixel is disturbed by some noise mechanism, resulting in some distribution $p(\tilde{y}_{\mathbf{u}} = y_{\mathbf{u}} | \tilde{\mathbf{x}} = \mathbf{x}) = p(\tilde{y}_{\mathbf{u}} = y_{\mathbf{u}} | \tilde{\mathbf{x}}_{\mathbf{u}} = \mathbf{x}_{\mathbf{u}})$.

Following the pattern of previous estimation methods, we attempt to construct a MAP-estimator of \mathbf{x} given \mathbf{y} . We obtain the posterior density

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

If $\tilde{\mathbf{x}}$ is modelled via a Gibbs distribution, the problem is how to maximise this over the set of all possible outcomes of \mathbf{x} . We solve this problem by an iterative procedure. Instead of the total posterior densities, we use the local, conditional densities

$$p(\mathbf{x}_{\mathbf{u}} | y_{\mathbf{u}}, \mathbf{x}_{(\mathbf{u})}) = \frac{p(y_{\mathbf{u}} | \mathbf{x}_{\mathbf{u}})p(\mathbf{x}_{\mathbf{u}} | \mathbf{x}_{(\mathbf{u})})}{p(y_{\mathbf{u}} | \mathbf{x}_{(\mathbf{u})})}.$$

We make a starting guess for \mathbf{x} (usually, we can take $\mathbf{x} = \mathbf{y}$) and iterate through all pixels \mathbf{u} , replacing $\mathbf{x}_{\mathbf{u}}$ with the value that maximises the conditional posterior density. We repeat this procedure until no change needs to be made for any pixel. (Note that we may not reach the *global* maximum of the posterior density.)

This simple method is called Iterated Conditional Modes (ICM), referring to the maximisation steps as locating the *mode* of the conditional distributions.

Example 4.9. Let $\mathbf{x} \in \{0, 1\}^n$ be a random field with Markov structure (a) from Figure 4.1, with local conditional distribution

$$p(\tilde{x}_{\mathbf{u}} = 0 | \tilde{\mathbf{x}}_{(\mathbf{u})} = \mathbf{x}_{(\mathbf{u})}) = \frac{\exp(\sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \beta \mathbb{I}(x_{\mathbf{v}} = 0))}{\exp(\sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \beta \mathbb{I}(x_{\mathbf{v}} = 0)) + \exp(\sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \beta \mathbb{I}(x_{\mathbf{v}} = 1))}.$$

When this field is measured, giving a measured image \mathbf{y} , each pixel is measured incorrectly with probability p_e , so that

$$y_{\mathbf{u}} = \begin{cases} x_{\mathbf{u}} & \text{with probability } 1 - p_e, \\ 1 - x_{\mathbf{u}} & \text{with probability } p_e. \end{cases}$$

We want to reconstruct the original field, \mathbf{x} . Writing down the entire density for $\mathbf{x} | \mathbf{y}$ is unworkable, but if we instead take smaller steps, things become manageable; we compute the conditional density for a single pixel $x_{\mathbf{u}}$ given not only the measured

pixels, but also given the surrounding pixels (which are unknown!):

$$\begin{aligned}
p(\tilde{x}_{\mathbf{u}} = y_{\mathbf{u}} | \tilde{y}_{\mathbf{u}} = y_{\mathbf{u}}, \mathbf{x}(\mathbf{u})) &= \frac{p(\tilde{y}_{\mathbf{u}} = y_{\mathbf{u}} | \tilde{x}_{\mathbf{u}} = y_{\mathbf{u}}, \mathbf{x}(\mathbf{u})) p(\tilde{x}_{\mathbf{u}} = y_{\mathbf{u}} | \mathbf{x}(\mathbf{u}))}{p(\tilde{y}_{\mathbf{u}} = y_{\mathbf{u}} | \mathbf{x}(\mathbf{u}))} = \dots \\
&= \frac{(1 - p_e) \exp(\sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \beta \mathbb{I}(x_{\mathbf{v}} = y_{\mathbf{u}}))}{(1 - p_e) \exp(\sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \beta \mathbb{I}(x_{\mathbf{v}} = y_{\mathbf{u}})) + p_e \exp(\sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \beta \mathbb{I}(x_{\mathbf{v}} = 1 - y_{\mathbf{u}}))} \\
&\propto (1 - p_e) \exp\left(\sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \beta \mathbb{I}(x_{\mathbf{v}} = y_{\mathbf{u}})\right).
\end{aligned}$$

Assuming that β and p_e are known, we can now estimate the values of \mathbf{x} with the ICM algorithm. \square

Example 4.10. Now, we let \mathbf{x} be a Markov field with Gaussian conditional distributions.

$$\begin{aligned}
p(x_{\mathbf{u}} | \mathbf{x}(\mathbf{u})) &\propto \exp\left(-\frac{1}{2\sigma^2} \left(x_{\mathbf{u}} - \varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)^2\right) \\
&= \exp\left(-\frac{1}{2\sigma^2} \left(x_{\mathbf{u}}^2 + \left(\varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)^2 - 2x_{\mathbf{u}}\varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)\right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2} \left(x_{\mathbf{u}}^2 - 2x_{\mathbf{u}}\varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)\right)
\end{aligned}$$

Now we introduce $y_{\mathbf{u}} = x_{\mathbf{u}} + e_{\mathbf{u}}$, where $e_{\mathbf{u}}$ are independent, $N(0, \sigma_e^2)$, so that

$$p(y_{\mathbf{u}} | x_{\mathbf{u}}) \propto \exp\left(-\frac{1}{2\sigma_e^2} (y_{\mathbf{u}}^2 + x_{\mathbf{u}}^2 - 2y_{\mathbf{u}}x_{\mathbf{u}})\right)$$

The posterior density for $x_{\mathbf{u}}$ given everything else is

$$\begin{aligned}
p(x_{\mathbf{u}} | y_{\mathbf{u}}, \mathbf{x}_{(\mathbf{u})}) &\propto p(y_{\mathbf{u}} | x_{\mathbf{u}}) p(x_{\mathbf{u}} | \mathbf{x}_{(\mathbf{u})}) \\
&\propto \exp\left(-\frac{1}{2\sigma_e^2} (y_{\mathbf{u}}^2 + x_{\mathbf{u}}^2 - 2y_{\mathbf{u}}x_{\mathbf{u}})\right) \\
&\quad \cdot \exp\left(-\frac{1}{2\sigma^2} \left(x_{\mathbf{u}}^2 - 2x_{\mathbf{u}}\varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)\right) \\
&= \exp\left(-\frac{1}{2} \left(\frac{1}{\sigma_e^2} + \frac{1}{\sigma^2}\right) x_{\mathbf{u}}^2 + x_{\mathbf{u}} \left(\frac{1}{\sigma_e^2} y_{\mathbf{u}} + \frac{1}{\sigma^2} \varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)\right) \\
&= \exp\left(-\frac{1}{2\gamma^2} \left(x_{\mathbf{u}}^2 - 2x_{\mathbf{u}}\gamma^2 \left(\frac{1}{\sigma_e^2} y_{\mathbf{u}} + \frac{1}{\sigma^2} \varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)\right)\right) \\
&\propto \exp\left(-\frac{1}{2\gamma^2} \left[x_{\mathbf{u}} - \gamma^2 \left(\frac{1}{\sigma_e^2} y_{\mathbf{u}} + \frac{1}{\sigma^2} \varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right)\right]^2\right),
\end{aligned}$$

where $\gamma^2 = 1/(\frac{1}{\sigma_e^2} + \frac{1}{\sigma^2})$. We recognise this as the density function for a Gaussian distribution, i.e.

$$x_{\mathbf{u}} | y_{\mathbf{u}}, \mathbf{x}_{(\mathbf{u})} \in \mathcal{N}\left(\gamma^2 \left(\frac{1}{\sigma_e^2} y_{\mathbf{u}} + \frac{1}{\sigma^2} \varphi \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} x_{\mathbf{v}}\right), \gamma^2\right).$$

We can now easily perform the ICM algorithm to obtain a denoised estimate of \mathbf{x} , from the measurement \mathbf{y} . (However, note that for Gaussian Markov random fields, direct calculation is often possible; See Section 4.2.4 for a generalisation of this example.) \square

4.4.2 Estimation by simulation

What would happen if we in the ICM algorithm replaced the pixels with a sample from the conditional posterior distribution, instead of the mode? After a number of iterations over all the pixels, we would obtain a sample from (approximately) the global posterior distribution! Using this method, called *Gibbs sampling*, we can answer questions related to the certainty of our reconstruction, that was not available with the single estimate obtain by the ICM algorithm.

Chapter 5 is devoted to methods for generating samples from complicated distributions, such as Markov fields.

4.4.3 Parameter estimation

In the previous sections, we assumed that the parameters of the Markov model, as well as the noise distribution parameters, were known. This section will deal with methods for estimating these parameters.

(To appear in a future edition.)

Partial likelihood estimation

Cross validation

Maximum likelihood estimation

Markov chain Monte Carlo ML-estimation

Exercises

Exercise 4.1: A Markov process as a Markov random field. Let us assume we observe a first order Markov process in N steps: $\tilde{\mathbf{x}} = \{\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_N\}$. The process takes values in the state space³ $\Lambda = \{-1, 1\}$, and we assume that

$$p_{ij} = \mathbf{P}(\tilde{x}_{n+1} = j | \tilde{x}_n = i) > 0$$

for all $i, j \in \Lambda$ and that $\mathbf{P}(\tilde{x}_0 = i) = p_0(i) > 0$ for all $i \in \Lambda$.

- Calculate the local characteristics $\mathbf{P}(\tilde{x}_n = j | \tilde{x}_m = x_m, x_m \in \Lambda, m \neq n)$.
- If we would like to define the Markov process as a MRF, what would the neighbourhood system $\mathcal{N} = \{\mathcal{N}_0, \mathcal{N}_1, \dots, \mathcal{N}_N\}$ be?

Exercise 4.2: A Markov process and the Gibbs distribution. Assume we observe our Markov chain in exercise 4.1 in stationarity and that we have the following transition matrix:

$$P = \frac{1}{e^\beta + e^{-\beta}} \begin{pmatrix} e^\beta & e^{-\beta} \\ e^{-\beta} & e^\beta \end{pmatrix}$$

- Calculate the probability of observing the sequence $\mathbf{x} = (x_1, x_2, \dots, x_6) = (-1, 1, 1, -1, 1, 1)$.
- Motivate that $\mathbf{P}(\tilde{\mathbf{x}} = \mathbf{x})$ is a Gibbs distribution.

Exercise 4.3: Gibbs' distributions, which can be viewed as a normalised *energy* of the different states in $\mathbf{x} \in \Omega$ is defined with respect to a neighbourhood system \mathcal{N} and a potential V_C . Assume we have the same Markov process as in exercise 4.1 and that we have a Gibbs distribution

$$p(\mathbf{x}) = \frac{e^{-U(\mathbf{x})}}{\sum_{\mathbf{x}} e^{-U(\mathbf{x})}}$$

where $U(\mathbf{x}) = -\sum_C V_C(\mathbf{x})$. Use the neighbourhood system you found and the potential $V_C(\mathbf{x}) = \beta x_i x_{i+1}$, $0 \leq i \leq N-1$, to show that the Gibbs distribution will give us a Markov random field.

Exercise 4.4: Let $\tilde{\mathbf{x}}$ be random vector with multivariate Gaussian density

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^3 |\Sigma|}} \exp \left(-\frac{(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}{2} \right),$$

³Generally for a Markov process; if each \tilde{x}_i can take values in Λ_i then $\mathbf{x} = \{x_1, \dots, x_N\} \in \Omega = \prod_{i=1}^N \Lambda_i$, where Ω is the sample space.

where $\mathbf{x} = (x_1 \ x_2 \ x_3)^\top$, $\boldsymbol{\mu} = (0 \ 0 \ 0)^\top$ and

$$\boldsymbol{\Sigma} = \frac{\sigma^2}{1-a^2} \begin{pmatrix} 1 & a & a^2 \\ a & 1 & a \\ a^2 & a & 1 \end{pmatrix},$$

where we also require that $|a| < 1$, in order for $\boldsymbol{\Sigma}$ to exist. After calculations the density can be rewritten as

$$p(\mathbf{x}) = \frac{\sqrt{1-a^2}}{\sqrt{(2\pi\sigma^2)^3}} \exp\left(-\frac{x_1^2 + x_2^2(1+a^2) + x_3^2 - 2(ax_1x_2 + ax_2x_3)}{2\sigma^2}\right).$$

Find the conditional density $p(x_3 \mid x_1, x_2)$.

Exercise 4.5: Conditional density in a bivariate Gaussian distribution. A common problem for Gaussian Markov random fields is to find the expected value and variance at a point given measurements at neighbouring locations. The simplest formulation of the problem is when only two points are considered. Therefore let \tilde{x}, \tilde{y} be random variables with a joint bivariate Gaussian distribution

$$p(x, y) = \frac{1}{\sqrt{(2\pi)^2 |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2} (x \ y) \boldsymbol{\Sigma}^{-1} \begin{pmatrix} x \\ y \end{pmatrix}\right),$$

where

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix}$$

- Determine the (unconditional) expectations $\mathbf{E}(\tilde{x})$ and $\mathbf{E}(\tilde{y})$.
- What are the interpretations of the parameters σ_x^2 , σ_y^2 , and ρ ?
- Find the conditional density $p(x|y)$.
- Find the conditional expectation $\mathbf{E}(\tilde{x}|\tilde{y} = y)$ and variance $\mathbf{V}(\tilde{x}|\tilde{y} = y)$.

Exercise 4.6: A simple example of a process with Markov structure is an AR(1)-process. In an AR(1)-process the value at any given time point, x_t , is equal to a constant, a , times the previous value x_{t-1} plus some noise $\tilde{\varepsilon}_t$, i.e.

$$\tilde{x}_t = a\tilde{x}_{t-1} + \tilde{\varepsilon}_t.$$

If the noise is Gaussian with $\mathbf{E}(\tilde{\varepsilon}_t) = 0$, $\mathbf{V}(\tilde{\varepsilon}_t) = \sigma^2$ and independent for different time points, then the conditional distributions, $p(x_t|x_{t-1})$, will be Gaussian with $\tilde{x}_t|\tilde{x}_{t-1} = x_{t-1} \in \mathcal{N}(ax_{t-1}, \sigma^2)$. To complete the model an initial distribution, $p(x_0)$, for \tilde{x}_0 is also needed. Let $p(x_0)$ be the stationary distribution of an AR(1)-process, i.e. $\tilde{x}_0 \in \mathcal{N}(0, \sigma^2/(1-a^2))$.

Now assume that we have observed x_0, x_1, x_2, x_4 and x_5 and want to reconstruct the missing value x_3 , that is we want to find $\mathbf{E}(\tilde{x}_3|x_0, x_1, x_2, x_4, x_5)$.

- State the conditional density $p(x_3|x_0, x_1, x_2, x_4, x_5)$ using only $p(x_0), p(x_t|x_{t-1}), t = 1, \dots, 5$ and (one) integral over x_3 .
- Use the result in a) to show that $p(x_3|x_0, x_1, x_2, x_4, x_5) = p(x_3|x_2, x_4)$.
- Calculate the conditional density $p(x_3|x_2, x_4)$.
- Find the conditional expectation, $\mathbf{E}(\tilde{x}_3|x_2, x_4)$, and the conditional variance, $\mathbf{V}(\tilde{x}_3|x_2, x_4)$.
- Motivate that $\mathbf{E}(\tilde{x}_3|x_2, x_4) = \mathbf{E}(\tilde{x}_3|x_0, x_1, x_2, x_4, x_5)$.

Exercise 4.7: Conditional probabilities Let \tilde{x}, \tilde{y} and \tilde{z} be three discrete random variables.

- Use Bayes' formula to show that

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

- Use the result in a) to show that $p(y|z) = \sum_x p(y|x, z) p(x|z)$.
- Combine the results in a) and b) to show that

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

Exercise 4.8: Markovian reconstruction Given an image \mathbf{y} , the task is to reconstruct the underlying field \mathbf{x} , when $\tilde{\mathbf{x}}$ is assumed to be a Markov field.

Assuming that $\tilde{x}_{\mathbf{u}}$ takes a value in $\{1, 2\}$ and that the Markov structure for $\tilde{\mathbf{x}}$ is given by

$$\mathbf{P}(\tilde{x}_{\mathbf{u}} = k|x_{(\mathbf{u})}) = \frac{\exp(\beta \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \mathbb{I}(x_{\mathbf{v}} = k))}{\sum_{j=1}^2 \exp(\beta \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} \mathbb{I}(x_{\mathbf{v}} = j))}, \quad \text{for } k=1,2,$$

where \mathcal{N} is some (arbitrary) neighbourhood structure. Conditioned on \mathbf{x} the pixels are assumed to be independent and Gaussian with $(\tilde{y}_{\mathbf{u}}|\tilde{\mathbf{x}} = \mathbf{x}) = (\tilde{y}_{\mathbf{u}}|\tilde{x}_{\mathbf{u}} = k) \in \mathbf{N}(\mu_k, \sigma_k^2)$. That is, the conditional distribution of $\tilde{y}_{\mathbf{u}}|\mathbf{x}$ depends only on the Markov field at \mathbf{u} , i.e. only on $x_{\mathbf{u}}$.

- Motivate that $p(y_{\mathbf{u}}|\tilde{x}_{\mathbf{u}} = k, x_{(\mathbf{u})}) = p(y_{\mathbf{u}}|\tilde{x}_{\mathbf{u}} = k)$.

- b) Motivate that the conditional posterior probabilities for $x_{\mathbf{u}}$ can be written as,

$$\mathbf{P}(\tilde{x}_{\mathbf{u}} = k | x_{\mathbf{u}}, x_{(\mathbf{u})}) = \frac{p(y_{\mathbf{u}} | \tilde{x}_{\mathbf{u}} = k) \cdot \mathbf{P}(\tilde{x}_{\mathbf{u}} = k | x_{(\mathbf{u})})}{Z_{\tilde{w}_{\mathbf{u}}}},$$

where $Z_{\tilde{w}_{\mathbf{u}}}$ is a normalising constant that does not depend on k .

- c) Use the result in b) and the densities for $p(y_{\mathbf{u}} | \tilde{x}_{\mathbf{u}} = k)$ and $\mathbf{P}(\tilde{x}_{\mathbf{u}} = k | x_{(\mathbf{u})})$ given above to find an expression for $\mathbf{P}(\tilde{x}_{\mathbf{u}} = k | y_{\mathbf{u}}, x_{(\mathbf{u})})$, where the (unknown) normalising constant $Z_{\tilde{w}_{\mathbf{u}}}$ contains everything that does not depend on k .
- d) Use the sum to one property of probabilities to find $Z_{\tilde{w}_{\mathbf{u}}}$.
- e) Generalise the result in c) and d) to let $\tilde{x}_{\mathbf{u}}$ take a value in $\{1, \dots, K\}$.
- f) Generalise the result in e) allowing each pixel to be multivariate Gaussian with $(\tilde{\mathbf{y}}_{\mathbf{u}} | \tilde{x}_{\mathbf{u}} = k) \in \mathbf{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Note that this basically is classification where the underlying class belongings follow a Markov field, and the distribution of the pixels given the class belongings is assumed to be known.

Chapter 5

Markov chain Monte Carlo simulation

5.1 Introduction

Why simulate?

Simulation of images from prior distributions is an important tool for investigating the appropriateness of the distributions used to model image elements. Complicated elements in disturbed images can also be estimated efficiently by simulation. A simulated sample from a posterior distribution can in fact give more information than a pure maximisation estimate, since it provides an idea about the precision of the estimates. However, distributions for entire images are often complicated, and it may be difficult to find direct methods for generating samples.

Example 5.1. (Binary image) As an example, take a small binary image of 8×8 pixels. Then there are 2^{64} different pixel configurations possible. In order to directly generate random samples of such an image, the probability for each of these 2^{64} images needs to be generated and stored in advance. This is very time and memory consuming, even for such a tiny image. Therefore, special simulation techniques need to be employed. \square

How to simulate?

In Chapter 4 we constructed a useful class of distributions for images, namely the Gibbs distributions. A characteristic feature of a Gibbs distribution is that it is relatively easy to *compare* the probability or density function (likelihood) $p(\omega)$ for two different outcomes. This fact can be used to simulate an expanding series of outcomes as follows:

- At each step in the series with present outcome \mathbf{x} , use some random mechanism to suggest a new outcome \mathbf{x}' . Compare the suggestion with the present outcome and accept the new suggestion as the next outcome, with a probability that depends on the likelihoods of the outcomes.

One useful class of methods of this type is *Markov chain Monte Carlo* (MCMC) simulation. The basis of these methods is to construct a (usually) time-reversible Markov chain, and simulate transitions between different states (images). The transition probabilities, $q(\mathbf{x}, \mathbf{x}')$, are chosen so that the stationary distribution, $\pi(\mathbf{x})$, is equal to the desired distribution, i.e. $\pi(\mathbf{x}) = p(\mathbf{x})$. Under mild conditions on $q(\mathbf{x}, \mathbf{x}')$, the distribution of the simulated images will converge to $\pi(\mathbf{x})$. To be efficient, the transition structure in the Markov chain has to be such that comparison of the likelihoods of the old and new states can be made without difficulty.

5.2 MCMC

How can one generate a random image such as in Figure 4.2 or 4.3? One has to choose an image from a large number of possible images. A binary image with 64×64 pixels may take $2^{64 \times 64} \approx 10^{1233}$ different appearances, each with its own probability. How can one draw an image with the correct probability, and is it really necessary to compute all these probabilities?

Markov chain Monte Carlo (MCMC) methods are useful for simulating samples from complex distributions. Markov field models¹, and Gibbs distributions in general, have properties that can be exploited by the MCMC simulation.

Assume that we want to simulate a random sample from a complicated distribution, e.g. a binary image. We then need not calculate the entire distribution, but only the *relative* probability, $p(\mathbf{x}')/p(\mathbf{x})$, for two different outcomes, which usually only differ in a few components. For Gibbs distributions this is often straightforward.

We now assume that we have a distribution defined by $\pi(\mathbf{x})$ and want to simulate samples from this distribution. In contrast to the usual situation, where a Markov chain is given, and the stationary distribution unknown, we now have to construct a Markov chain with the given distribution as stationary distribution.

Definition 5.1. *By a MCMC-simulation from a distribution π is meant a simulation of a Markov chain with stationary distribution π .* \square

The main issue in MCMC-simulation is how to construct an appropriate Markov chain, and there are several related general methods, most of which can be seen as

¹Note the distinction between the Markov *field* (which is what we want to simulate) and the Markov *chain* that will be used as a tool in the MCMC-simulation.

special cases of the Metropolis-Hastings algorithm, see Hastings (1970) and Section 5.2.2.

At each point in the simulation, we have a configuration (image) \mathbf{x}^t , which is replaced with a new configuration, \mathbf{x}^{t+1} , with probability or density $m(\mathbf{x}^t, \mathbf{x}^{t+1})$, defined by the *transition kernel*, \mathbf{M} .² Algorithms of Metropolis-Hastings type construct this final transition kernel with the aid of a *proposal transition kernel*, \mathbf{Q} . First, a new configuration \mathbf{x}' is generated according to the proposal kernel $q(\mathbf{x}^t, \mathbf{x}')$. Then, this change is either accepted or rejected, with a certain probability, defined so that the final transition kernel \mathbf{M} has certain properties that guarantees that samples from the correct distribution will be generated.

We are now ready to present the following algorithm, which was proposed as early as 1953 by Metropolis et al. for calculating state equations in theoretical physics; see Metropolis et al. (1953).

5.2.1 The Metropolis algorithm

Let $\pi(\mathbf{x})$ be the desired distribution, where we index the set Ω of all possible configurations of \mathbf{x} by i, j , etc. The Metropolis algorithm for simulation from the distribution π is then given by the following.

- (i) Choose a symmetric transition kernel \mathbf{Q} between elements in \mathbf{x} , i.e. a kernel with $q_{ij} = q_{ji}$. It should be chosen so that all elements are easily calculable. Also choose an initial configuration \mathbf{x}^0 .
- (ii) For $t = 1, 2, \dots$ generate \mathbf{x}^t as follows:

If $\mathbf{x}^{t-1} = i$, draw a new state j with probabilities given by q_{ij} . Let

$$\mathbf{x}^t = \begin{cases} j & \text{with probability } \min(1, \pi_j/\pi_i), \\ i & \text{with probability } 1 - \min(1, \pi_j/\pi_i), \end{cases}$$

i.e. a switch to state j is made if $\pi_j \geq \pi_i$, or with probability π_j/π_i if $\pi_j < \pi_i$.

The final transition kernel \mathbf{M} is given by the probabilities

$$\begin{aligned} m_{ij} &= \begin{cases} q_{ii} + \sum_{k \neq i} q_{ik}(1 - \min(1, \pi_k/\pi_i)) & \text{if } i = j, \\ q_{ij} \min(1, \pi_j/\pi_i) & \text{if } i \neq j, \end{cases} \\ &= q_{ij} \min(1, \pi_j/\pi_i) + \mathbb{I}(i = j) \left(1 - \sum_{k \in \Omega} q_{ik} \min(1, \pi_k/\pi_i) \right) \end{aligned} \quad (5.1)$$

²In accordance with common practise in the MCMC-world we use the term transition kernel to cover both discrete and continuous models. For a model with discrete (finite or infinite) state space the transition kernel is the same as the transition *matrix* in a Markov chain.

for discrete distributions, and the densities

$$\begin{aligned} m_{ij} &= q_{ij} \min(1, \pi_j / \pi_i) + \delta_i(j) \int_{\Omega} q_{ik} (1 - \min(1, \pi_k / \pi_i)) dk \\ &= q_{ij} \min(1, \pi_j / \pi_i) + \delta_i(j) \left(1 - \int_{\Omega} q_{ik} \min(1, \pi_k / \pi_i) dk \right) \end{aligned}$$

for continuous distributions.

We now show that π really is a stationary distribution of the constructed Markov chain. The following definition will be of use.

Definition 5.2. Let $\{\tilde{\mathbf{x}}^t\}$ be a Markov chain with transition kernel \mathbf{M} , and let π be the distribution of $\tilde{\mathbf{x}}^t$. If the condition

$$\pi_i m_{ij} = \pi_j m_{ji} \tag{5.2}$$

holds for all i and j , the chain is said to be time-reversible. \square

A simple consequence of Eq. 5.2 is that

$$\sum_i \pi_i m_{ij} = \sum_i \pi_j m_{ji} = \pi_j, \quad \text{for all } j, \tag{5.3}$$

so that π is a stationary distribution. With π as a row vector of probabilities for a discrete variable, Equation 5.3 can be written as $\pi \mathbf{M} = \pi$, which may be familiar from courses on Markov chains.

Theorem 5.1. The chain $\{\tilde{\mathbf{x}}^t\}_{t=0}^{\infty}$ defined by the Metropolis algorithm, started in the distribution π , is time-reversible, and hence has π as stationary distribution.

Proof: For transitions from a state i back to i , condition 5.2 is trivially true. For two different states i and j , we first calculate $\pi_i m_{ij}$. The transition probability from i to j is $m_{ij} = q_{ij} \min(1, \pi_j / \pi_i)$, so $\pi_i m_{ij} = q_{ij} \min(\pi_i, \pi_j)$. Similarly, the right-hand side of 5.2 is $\pi_j m_{ji} = q_{ji} \min(\pi_j, \pi_i)$. Since \mathbf{Q} was required to be symmetric, the condition holds, so the chain is time-reversible. \square

We have shown that the Metropolis algorithm generates a stationary Markov chain, if started in the distribution, π , and thus we would have an algorithm for generating multiple, dependent samples from π , if we could generate *one* sample. An important result that makes the algorithm useful in practice is that with only mild conditions on \mathbf{Q} , the distribution will *converge* to π , for all starting configurations.

If the transition kernel \mathbf{M} defines an *irreducible* and *recurrent* Markov chain, then the distribution will converge to π , for all starting configurations. This means that

\mathbf{Q} must be chosen so that there is a positive probability or density for moving from between two arbitrary states, i.e. for each pair of states \mathbf{x} and \mathbf{x}' , there must exist a finite-length sequence $(\mathbf{x}^0, \dots, \mathbf{x}^n)$, with $\mathbf{x}^0 = \mathbf{x}$ and $\mathbf{x}^n = \mathbf{x}'$, such that $m(\mathbf{x}^{t-1}, \mathbf{x}^t) > 0$ for all $t = 1, \dots, n$. More strict mathematical conditions are given in Section 5.2.4.

One might wonder why it is useful to simulate samples from the distribution π , when we already assume that it is known. One answer is that the algorithms described in this chapter need only know the relative probabilities π_j/π_i , so that we can use distributions known only up to a common normalising constant. Gibbs distributions are prime examples of this: For a Gibbs distribution, the probability for an outcome \mathbf{x} is proportional to $e^{-W(\mathbf{x})}$, and therefore $\pi_j/\pi_i = \exp\{W(i) - W(j)\}$.

Example 5.2. (Poisson simulation) As a first example of a symmetric updating kernel we shall show how to simulate samples from a Poisson distribution with mean μ , $\pi_i = e^{-\mu} \mu^i / i!$. This is a very simple example compared to the image simulations which are our real topic, since the possible values of x are just the non-negative integers \mathbb{Z} , and hence the distribution π is a one-dimensional distribution on \mathbb{Z} . To define the Markov transition kernel we therefore only have to define how one shall move around among the possible states on the non-negative integers.

We use the simplest possibility, the symmetric random walk, and allow transitions to negative x -values, even if these are not possible in the Poisson distribution – this will be taken care of by the acceptance mechanism. Thus, take

$$q_{i,j} = 1/2, \quad \text{for } j = i \pm 1, \\ = 0, \quad \text{otherwise.}$$

Since

$$\pi_{i+1}/\pi_i = \mu/(i+1), \\ \pi_{i-1}/\pi_i = i/\mu,$$

for $i = 0, 1, \dots$, we have the following rule.

- If $x^{t-1} = i$, the possible candidates are $j = i \pm 1$, with equal probabilities.
- If the Markov update suggests $i + 1$, accept it with probability $\min(1, \frac{\mu}{i+1})$. This means that we always accept a suggestion to move upwards if $i + 1 \leq \mu$, but that a move upwards is accepted only with probability $\mu/(i+1)$ if $i+1 > \mu$.
- Similarly, if the Markov update suggests a move downwards to $i - 1$, this is always accepted if $i \geq \mu$, but accepted with probability i/μ if $i < \mu$. Obviously, if $i = 0$, a move to the left is never accepted.

Here it is easy to formulate the simulation rule explicitly and find the final transition kernel according to (5.1). Note, however, that one idea with the MCMC routine is that one need not have such an explicit formula. One has

$$\mathbf{x}^t = \begin{cases} i + 1 & \text{with probability } \min(1, \mu/(i + 1))/2, \\ i & \text{with probability } 1 - (\min(1, \mu/(i + 1)) + \min(1, i/\mu))/2, \\ i - 1 & \text{with probability } \min(1, i/\mu)/2. \end{cases}$$

For future use, we give two of the final transition probabilities:

$$\begin{aligned} m_{01} &= \min(1, \mu)/2, \\ m_{10} &= \min(1, 1/\mu)/2. \end{aligned}$$

□

Example 5.3. (Bivariate normal simulation) A continuous bivariate distribution with density

$$p(\mathbf{x}) = \frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(\left(\frac{x_1}{\sigma_1}\right)^2 - \frac{2\rho x_1 x_2}{\sigma_1 \sigma_2} + \left(\frac{x_2}{\sigma_2}\right)^2\right)\right)$$

can be simulated by the Metropolis algorithm by a symmetric kernel, e.g. if $\mathbf{x}^{t-1} = (x_1, x_2)$, then a candidate is $(x_1, x_2) + (\xi_1, \xi_2)$, where ξ_1 and ξ_2 are independent standard normal variables. The suggestion is accepted depending on the ratio between the old and the new likelihood. □

5.2.2 The Metropolis-Hastings algorithm

In practice, the requirement of a symmetric update proposal kernel \mathbf{Q} in the Metropolis algorithm may be difficult, or even undesirable, to fulfil. The Metropolis-Hastings algorithm (mentioned in Hastings (1970)) is a generalisation which permits arbitrary \mathbf{Q} (although subject to the same time-reversibility and irreducibility conditions).

- (i) Choose a transition kernel \mathbf{Q} , and a starting configuration \mathbf{x}^0 .
- (ii) For $t = 1, 2, \dots$ generate \mathbf{x}^t as follows:

If $\mathbf{x}^{t-1} = i$, draw a new state j with probabilities given by q_{ij} . Let

$$\mathbf{x}^t = \begin{cases} j & \text{with probability } \alpha_{ij}, \\ i & \text{with probability } 1 - \alpha_{ij}, \end{cases}$$

where

$$\alpha_{ij} = \min \left(1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \right),$$

i.e. a switch to state j is made with probability α_{ij} .

The final transition kernel \mathbf{M} is now given by the probabilities

$$m_{ij} = q_{ij} \alpha_{ij} + \mathbb{I}(i = j) \left(1 - \sum_{k \in \Omega} q_{ik} \alpha_{ik} \right)$$

for discrete distributions, and the densities

$$m_{ij} = q_{ij} \alpha_{ij} + \delta_i(j) \left(1 - \int_{\Omega} q_{ik} \alpha_{ik} dk \right)$$

for continuous distributions.

Theorem 5.2. *The chain $\{\tilde{\mathbf{x}}^t\}_{t=0}^{\infty}$ defined by the Metropolis-Hastings algorithm, started in the distribution π , is time-reversible.*

Proof: We follow the same line of arguments as for the Metropolis algorithm. For transitions from a state i back to the same state, condition 5.2 is trivially true. For two different states i and j , we calculate $\pi_i m_{ij}$. The transition probability from i to j is $m_{ij} = q_{ij} \alpha_{ij}$, so $\pi_i m_{ij} = \min(\pi_i q_{ij}, \pi_j q_{ji})$. Similarly, the right-hand side of 5.2 is $\pi_j m_{ji} = \min(\pi_j q_{ji}, \pi_i q_{ij})$, so condition 5.2 holds. \square

Example 5.4. We make a super-simple modification to the Metropolis simulation of the Poisson distribution in Example 5.2 by preventing the Markov update to suggest a negative value. Thus, we take

$$\begin{aligned} q_{i,j} &= 1/2, & \text{for } j = i \pm 1, i = 1, 2, \dots, \\ &= 1, & \text{for } j = 1, i = 0, \\ &= 0, & \text{otherwise.} \end{aligned}$$

This leads to modification of the final transition probabilities between states 0 and 1, compared to those in Example 5.2:

$$\begin{aligned} m_{01} &= \min(1, \mu/2), \\ m_{10} &= \min(1, 2/\mu)/2. \end{aligned}$$

□

Example 5.5. (Poisson simulation with large μ) For a Poisson distribution with large μ -value, and hence large standard deviation, $\mathbf{D} = \sqrt{\mu}$, the random walk simulation with step size ± 1 is very inefficient. The small step size makes successive observations very dependent, in fact they are almost equal, relative to the large variability in the distribution.

To remedy this in the Metropolis-Hastings algorithm one can for example let a positive (negative) step have a geometric (truncated geometric) distribution with parameter p , i.e. $\mathbf{P}(\tilde{\xi} = x) = p(1-p)^{x-1}$, $x = 1, 2, \dots$. Here p should be chosen of the same order as $1/\sqrt{\mu}$ in order for the simulation to have a chance to visit the possible states in a reasonable time. □

5.2.3 Gibbs-sampling

There are many special ways of constructing the proposal kernel \mathbf{Q} in the Metropolis-Hastings algorithm. A simple method called *Gibbs-sampling* is to first choose a random component, k , and then propose an update according to the conditional distribution, so that the transition probabilities become $q(\mathbf{x}, \mathbf{x}') = \pi(x'_k | \mathbf{x}_{(k)})$, where \mathbf{x} and \mathbf{x}' differ only at component k . The acceptance probability is

$$\begin{aligned} \alpha(\mathbf{x}, \mathbf{x}') &= \min \left(1, \frac{\pi(\mathbf{x}')q(\mathbf{x}', \mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x}, \mathbf{x}')} \right) \\ &= \min \left(1, \frac{\pi(x'_k | \mathbf{x}'_{(k)})\pi(\mathbf{x}'_{-k})q(\mathbf{x}', \mathbf{x})}{\pi(x_k | \mathbf{x}_{(k)})\pi(\mathbf{x}_{-k})q(\mathbf{x}, \mathbf{x}')} \right) \\ &= \min \left(1, \frac{\pi(x'_k | \mathbf{x}_{(k)})\pi(\mathbf{x}_{-k})\pi(x_k | \mathbf{x}_{(k)})}{\pi(x_k | \mathbf{x}_{(k)})\pi(\mathbf{x}_{-k})\pi(x'_k | \mathbf{x}_{(k)})} \right) \\ &= 1, \end{aligned}$$

so that all proposed updates are accepted. Thus, each iteration updates one random component of \mathbf{x} .

An alternative is to obtain a (random) permutation k_1, \dots, k_n of the components, and update them in the order $k_1, \dots, k_n, k_n, \dots, k_1$. In this way, the components are updated in regular intervals, but time-reversibility is retained.

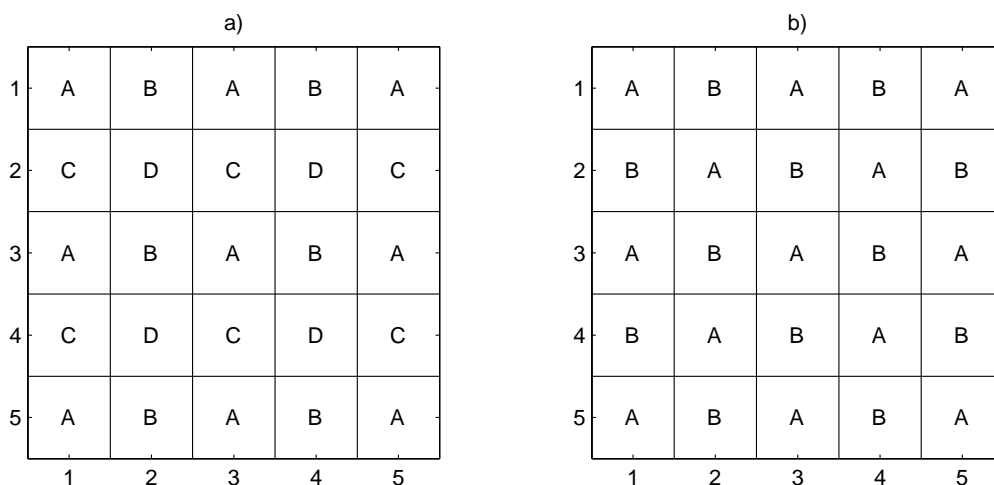


Figure 5.1: a) The conditional independent pixel groups for a general 8-neighbour structure. b) Using the 4-neighbour structure it is possible to find only two conditional independent groups.

Block-update Gibbs-sampling

Pure Gibbs-sampling is often very slow. If the conditional distribution of a group of components can be computed, and simulated from, we can update entire blocks at once. If the conditional distribution is simple enough, this may lead to faster convergence to the desired distribution.

Parallel Gibbs-sampling

Another way to speed up the simulation of variables with Markov structure is to exploit the neighbour structure to enable updates of several components independently of each other. We divide the components into N_A sets $\mathcal{A}_1, \dots, \mathcal{A}_A$, such that, if $i, j \in \mathcal{A}_a$,

$$\pi(x_i, x_j | x_k, k \notin \mathcal{A}_a) = \pi(x_i | x_k, k \notin \mathcal{A}_a) \pi(x_j | x_k, k \notin \mathcal{A}_a)$$

for all $a = 1, \dots, A$, i.e. the components in \mathcal{A}_a are independent given all components not in \mathcal{A}_a . This means that the intersection between \mathcal{A}_a and the set of neighbours $\mathcal{N}(\mathcal{A}_a) = \{k; k \in \mathcal{N}(i), i \in \mathcal{A}_a\}$ is empty. We now go through the sets $\mathcal{A}_1, \dots, \mathcal{A}_A$ (in random or forward/backward order), for each set \mathcal{A}_a updating all components at once.

Example 5.6. (Parallel sampling for images) In the case of a 8-neighbour structure we divide the pixels into 4 sets $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and \mathcal{D} , see Figure 5.1 (a). Choosing two

pixels from group \mathcal{B} their simultaneous conditional density is equal to the product of their densities:

$$\begin{aligned} & \pi(\mathbf{x}_{(2,3)}, \mathbf{x}_{(4,3)} | \mathbf{x}_{(i,j)}, (i,j) \notin \mathcal{B}) \\ &= \pi(\mathbf{x}_{(2,3)} | \mathbf{x}_{(4,3)}, \mathbf{x}_{(i,j)}, (i,j) \notin \mathcal{B}) \pi(\mathbf{x}_{(4,3)} | \mathbf{x}_{(i,j)}, (i,j) \notin \mathcal{B}) \\ &= \pi(\mathbf{x}_{(2,3)} | \mathbf{x}_{(i,j)}, (i,j) \notin \mathcal{B}) \pi(\mathbf{x}_{(4,3)} | \mathbf{x}_{(i,j)}, (i,j) \notin \mathcal{B}). \end{aligned} \quad (5.4)$$

This implies that they are independent and we can update them at the same time. When we use the 4-neighbour structure it is possible to use only two pixel groups when updating the picture, see Figure 5.1 (b). \square

5.2.4 MCMC convergence conditions

In this section we give a brief overview of the mathematical concepts used for defining necessary conditions for convergence of MCMC algorithms. See Gilks et al. (1996) for a more thorough discussion on these topics.

Let $\tau(\mathbf{x}, \mathcal{W})$ be the transition time from a state \mathbf{x} to a state in the set \mathcal{W} .

Definition 5.3. A Markov chain is φ -irreducible for a probability distribution φ on Ω if $\varphi(\mathcal{W}) > 0$ for a set $\mathcal{W} \subset \Omega$ implies that

$$\mathbf{P}(\tau(\omega, \mathcal{W}) < \infty) > 0$$

for all $\mathbf{x} \in \Omega$. A chain is irreducible if it is φ -irreducible for some distribution φ . If a chain is φ -irreducible, then φ is called an irreducibility distribution for the chain. \square

For each irreducible chain, there is at least one *maximal* irreducibility distribution, ψ , such that all other irreducibility distributions are absolute continuous with respect to ψ .

Definition 5.4. An irreducible Markov chain with maximal irreducibility ψ is (Harris) recurrent if for any set $\mathcal{W} \subset \Omega$ with $\psi(\mathcal{W}) > 0$ the condition

$$\mathbf{P}(\mathbf{x}^t \in \mathcal{W} \text{ for infinitely many } t | \mathbf{x}^0 = \mathbf{x}) = 1 \text{ for all } \mathbf{x} \in \Omega \quad (5.5)$$

is satisfied.³ \square

Remark 5.1: In the usual definition of recurrence, there may be a null set for which the probability in (5.5) is less than one. This makes it easier to prove recurrence, but the stronger Harris recurrence is simpler to use, in its shorter formulation. \square

³An event occurring infinitely many times is in probability theory usually referred to as occurring *infinitely often*.

Theorem 5.3. *Suppose that the Markov chain $\{\mathbf{x}^t\}$ is irreducible and Harris recurrent, and has π as a stationary distribution. Then the chain is π -irreducible, π is a maximal irreducibility distribution, and π is the unique stationary distribution of the chain.*

Theorem 5.4. *Let $\{\mathbf{x}^t\}$ be an irreducible, Harris recurrent Markov chain with transition kernel \mathbf{M} and stationary distribution π . Define the average transition kernel of order T by*

$$\bar{\mathbf{M}}^T(\mathbf{x}, \mathcal{W}) = \frac{1}{T+1} \sum_{t=0}^T \mathbf{M}^t(\mathbf{x}, \mathcal{W})$$

for all $x \in \Omega$ and $\mathcal{W} \subset \Omega$. Then

$$\|\bar{\mathbf{M}}^T(\mathbf{x}, \cdot) - \pi(\cdot)\| \rightarrow 0$$

for all \mathbf{x} .

Exercises

Bibliography

- Dryden, I. L. and Mardia, K. V. (1998). *Statistical Shape Analysis*. Wiley.
- Farin, G. (1996). *Curves and Surfaces for Computer Aided Geometric Design: a practical guide*. Academic Press, fourth edition.
- Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall.
- Glasbye, C. A. and Horgan, G. W. (1995). *Image Analysis for the Biological Sciences*. Wiley.
- Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing*. Addison-Wesley.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.
- Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, 1:321–331.
- Kinderman, R. and Snell, J. L. (1980). *Markov Random Fields and their Applications*, volume 1 of *Contemporary mathematics*. American Mathematical Society, Providence, Rhode Island.
- Lindgren, F. (1997). Flame reconstruction by Markov chain Monte Carlo simulation. Master's thesis, Department of Mathematical Statistics, Lund University, Sweden.
- Lindgren, F. (2001). A stochastic surface modelling system. In *Proc. Scandinavian Conference on Image Analysis*, Bergen, Norway. SCIA.
- Lindgren, F., Johansson, B., and Holst, J. (1997). Flame reconstruction in spark ignition engines. In *SAE Fall Fuels and Lubricants Meeting and Exposition*. SAE. SAE paper 972825.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–92.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. Number 104 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC Press.

Wordlist, notation and formulae

English	svenska	Notation
scalar value	skalär	x, y
vector or matrix	vektor eller matris	\mathbf{x}, \mathbf{Y}
random variable	stokastisk variabel	\tilde{x}, \tilde{y}
probability	sannolikhet	$\mathbf{P}(\tilde{x} = x)$
probability function	sannolikhetsfunktion	$p_{\tilde{x}}(x)$
density function	täthetsfunktion	$p_{\tilde{x}}(x)$
expectation	väntevärde	$\mu_{\tilde{x}} = \mathbf{E}(\tilde{x})$
mean value	medelvärde	\bar{x}
variance	varians	$\sigma_{\tilde{x}}^2 = \mathbf{V}(\tilde{x})$
covariance	kovarians	$\mathbf{C}(\tilde{x}_1, \tilde{x}_2)$
covariance matrix	kovariansmatris	Σ
precision matrix	precisionsmatris	$\mathbf{Q} = \Sigma^{-1}$
standard deviation	standardavvikelse	
has a Gaussian/Normal distribution	är Normalfördelad	$\tilde{x} \in \mathbf{N}(\mu, \sigma^2)$
conditional probability	betingad sannolikhet	
the prior distribution (usually)	a priori-fördelningen ¹	
the posterior distribution (usually)	a posteriori-fördelningen ²	
Bayes' fomula	Bayes formel	
Principal Component Analysis	principalkomponentanalys	
training set	träningssmängd	
discriminant analysis	diskriminantanalys	
random field	stokastiskt fält	
Markov random field	Markovfält	
Neighbourhood structure	grannstruktur	
the Markov condition	Markovvillkoret	
clique	“klick”	\mathcal{C}
Gibbs distribution	Gibbsfördelning	
Markov chain Monte Carlo simulation	Markovkedje-Monte-Carlo-simulering	
transition kernel	övergångskärna	
deformable templates	deformerbara mallar	
column/row stacking	kolumn/radstapling	

¹ ²

¹From *latin*, “a priori”, meaning “beforehand” / “på förhand”.

²From *latin*, “a posteriori”, meaning “after the fact” / “i efterhand”.

Expectation $E(\tilde{\mathbf{x}}) = \int p_{\tilde{\mathbf{x}}}(x) dx$

Covariance for row vectors $C(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) = E((\tilde{\mathbf{x}}_1 - \mu_{\tilde{\mathbf{x}}_1})^T(\tilde{\mathbf{x}}_2 - \mu_{\tilde{\mathbf{x}}_2}))$

Covariance for column vectors $C(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2) = E((\tilde{\mathbf{x}}_1 - \mu_{\tilde{\mathbf{x}}_1})(\tilde{\mathbf{x}}_2 - \mu_{\tilde{\mathbf{x}}_2})^T)$

Probability function Convenience notation: $p(x) = P(\tilde{x} = x)$

Density function Convenience notation: $p(x) = p_{\tilde{x}}(x)$

Conditional probability

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

Conditional density

$$p(x_1|x_2) = p_{x_1|\tilde{x}_2=x_2}(x_1) = \frac{p(x_1, x_2)}{p(x_2)}$$

Bayes' formula

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x)p(x) dy} \propto p(y|x)p(x)$$

Prior density $\pi(x) = p_{\tilde{x}}(x)$

Neighbours $\mathcal{N}(\mathbf{u}) = \mathcal{N}_{\mathbf{u}}$

The Markov condition $p(x_{\mathbf{u}}|x_{\mathbf{v}}, \mathbf{v} \neq \mathbf{u}) = p(x_{\mathbf{u}}|x_{\mathbf{v}}, \mathbf{v} \in \mathcal{N}_{\mathbf{u}})$

Gaussian MRF Precision matrix $\mathbf{Q} = \Sigma^{-1} = \mathbf{S}^{-1}(\mathbf{I} - \mathbf{C})\mathbf{S}^{-1}$, $\mathbf{S} = \text{diag}(\tau_{\mathbf{u}})$,
 $\tau_{\mathbf{u}}^2 = 1/Q_{\mathbf{u},\mathbf{u}}$, $C_{\mathbf{u},\mathbf{v}} = -Q_{\mathbf{u},\mathbf{v}}/Q_{\mathbf{u},\mathbf{u}}$,

$$x_{\mathbf{u}}|x_{\mathcal{N}(\mathbf{u})} \in N\left(\mu_{\mathbf{u}} + \sum_{\mathbf{v} \in \mathcal{N}(\mathbf{u})} C_{\mathbf{u},\mathbf{v}} \cdot (x_{\mathbf{v}} - \mu_{\mathbf{v}}), \tau_{\mathbf{u}}^2\right)$$

Normal density

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

Gibbs distribution

$$p(\mathbf{x}) = \frac{1}{Z} \exp(-W(\mathbf{x})) = \frac{1}{Z} \exp\left(\sum_c V_c(\mathbf{x})\right)$$

The Metropolis-Hastings algorithm Target distribution $\pi(\mathbf{x})$, proposal distribution $p(\mathbf{x}^{t+1}|\mathbf{x}^t)$, acceptance probability

$$\alpha(\mathbf{x}^t, \mathbf{x}^{t+1}) = \min \left\{ 1, \frac{\pi(\mathbf{x}^{t+1})}{\pi(\mathbf{x}^t)} \cdot \frac{p(\mathbf{x}^t|\mathbf{x}^{t+1})}{p(\mathbf{x}^{t+1}|\mathbf{x}^t)} \right\}$$

A popular alternative notation: Proposal kernel q , $q(\mathbf{x}^t, \mathbf{x}^{t+1}) = p(\mathbf{x}^{t+1}|\mathbf{x}^t)$, so that

$$\alpha(\mathbf{x}^t, \mathbf{x}^{t+1}) = \min \left\{ 1, \frac{\pi(\mathbf{x}^{t+1})}{\pi(\mathbf{x}^t)} \cdot \frac{q(\mathbf{x}^{t+1}, \mathbf{x}^t)}{q(\mathbf{x}^t, \mathbf{x}^{t+1})} \right\}$$

Third edition, 2005-10-18, Chapter 1–5

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

<http://www.maths.lth.se/>