Statistics of Imaging

Mats Rudemo

October 27, 2019

STATISTICS OF IMAGING

Copyright Mats Rudemo, Department of Mathematical Sciences, Chalmers University of Technology and Gothenburg University, 412 96 Gothenburg, Sweden

E-mail: rudemo@chalmers.se

Preface

The object of this book is to provide an introduction to several subjects connected to statistical inference from images. Image analysis is an extensive research field growing with considerable speed. Thus only some selected parts can be covered here and the choice of subjects is, of course, heavily influenced by my experience and interests. The book has grown from notes on a master level course on statistical image analysis that I have given at Chalmers technical university and Gothenburg university during a sequence of years.

Some specific features can be mentioned. The extensive MNIST database on handwritten integers is used in many examples on pattern recognition in the book. The MNIST database has been a proving ground for many machine learning methods, such as neural nets and support vector machines, and the book contains a brief chapter on such methods. An substantial part of the book, chapters 9-13, consists of material from about 10 research papers written for researchers in applied fields and where image analysis forms a crucial role. In my masters course mentioned above I asked the students to read these papers. But as the papers were written mainly with application audiences in mind and contained much discussion from the specific application areas, the papers turned out to be difficult for the image analysis students to read. In this book I have therefore written considerably condensed versions of the problems discussed in the papers but now with image analysis students and researchers in mind. Hopefully readers of the book that find some of these applications particularly interesting will go to the original papers for more details.

The first part "Images" includes a very brief introduction to basic digital image processing, including image acquisition, image filtering and object feature measurements. After that pattern recognition, typically based on features obtained from objects identified in images, is treated at some length. Both the case with known classes, called *discrimination* or *supervised learning* and the case with unknown classes, called *clustering* or *unsupervised learning* are covered. A chapter on machine learning gives a brief introduction to neural nets and support vector machines with image analysis and pattern recognition in mind. The first part is concluded by a chapter on statistical models for images. Markov models in two dimensions and Markov chain Monte Carlo methods are introduced.

The second part "Spatial Statistics" starts with some basic properties of spatial random processes: covariance properties and prediction (kriging). Spatial point processes are treated in some detail including image models constructed from point processes. The second part is concluded by a brief introduction to shape analysis and the related problems of image warping and image matching.

The third part "Applications" contains examples of image analysis applied to problems in biology, bioinformatics, remote sensing and microscopy. The examples cover analysis of data from microarray (DNA chip) images, two-dimensional electrophoresis, aerial photographs of forests, analysis of diffusion based on sequences of images, and finally image analysis of transmission electron micrographs.

The book concludes with a chapter with mixed exercises, a few of them with detailed solutions, and an appendix with mathematical, computational and statistical beckground.

In preparing this book I have benefited a lot from discussions with current and previous colleagues and students. [Then follows a long list of names]

In particular I am deeply indebted to Marco Longfils for detailed discussions on the subjects in the book and many computations, including all computations on the MNIST data set.

PART 1. IMAGES

[Here should follow about one page preamble]

Chapter 1

Digital images

A digital image may be regarded as a matrix of pixels (picture elements), $f = (f_{ij}) = (f_{ij}, i = 1, \ldots, m, j = 1, \ldots, n)$. Here $f_{ij} \in V$ where V is the set of possible pixel values, e.g. $V = \{0, 1\}$ for a binary image, $V = \{0, \ldots, 255\}$ for a grey level image with 256 grey levels, conveniently coded as bytes, and $V = \{0, \ldots, 255\}^3$ for a colour image with 256 levels in each of the three colours Red, Green and Blue. Thus each pixel is specified both by a location (i, j) and a pixel value f_{ij}. The first location index i specifies the row and the second index j the column. Rows are counted either from above (most common in the image processing literature) or from below, while columns are counted from the left.

1.1 Examples of images

Example 1.1. *Aerial photographs of a thinning experiment.*

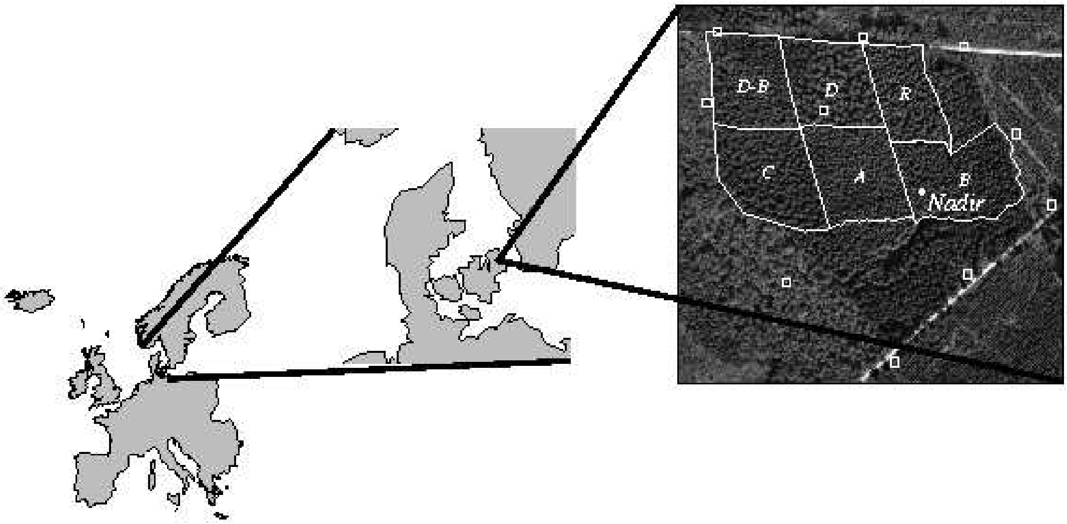


Figure 1.1: Aerial photograph of the thinning experiment KU in northern Sealand with Norway spruce trees. The position of the aeroplane at image acqusition was 560 m above “Nadir”.

Figure 1.1 shows an aerial photograph of the thinning experiment KU, in northern Sealand, with six subplots which were subject to different thinning treatments (Dralle & Rudemo, 1996). The six treatments were

A	No thinning
B	Light thinning
C	Medium-heavy thinning
D	Very heavy thinning
D–B	In the youth very heavy thinning, later light thinning
R	Heavy row thinning

The photograph was acquired from an aeroplane at the altitude 560 m above the point ”Nadir” in Figure 1.1. An enlargement of the subplot D is shown Figure 1.2.

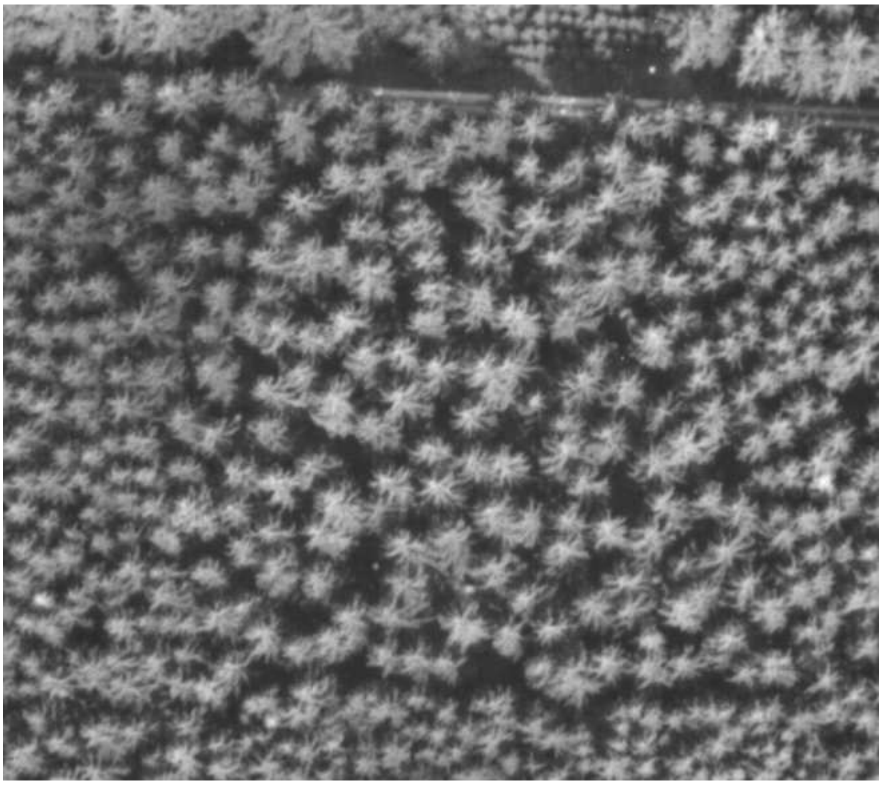


Figure 1.2: Detail of the aerial photograph in Figure 1.1 covering the subplot D with very heavy thinning.

A further enlargement of the southeast corner of subplot D is shown in Figure 1.3. Here the individual pixels, each corresponding to a square of about 15 cm × 15 cm at ground level, are visible.

In Figure 1.4 we see subplot D from a photograph acquired with the aeroplane in a position to the northwest of the experimental area. The time of acquisition was August 4 at 10:08 AM, which implies that the sun was in the direction southeast, and the trees were thus backlighted in Figure 1.4.

One object of the image analysis of the photographs obtained in this experiment was to estimate the number of trees in the different subplots and to estimate the positions of the tree tops. This application is further discussed in Chapter 11 in Part 3.

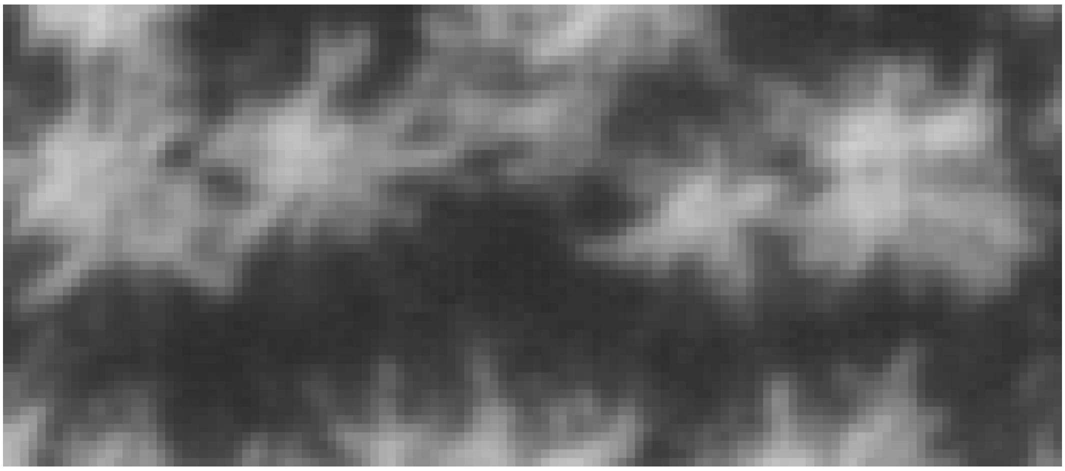


Figure 1.3: Detail of the aerial photograph in Figure 1.2 showing part of the southeastern corner of subplot D.

□

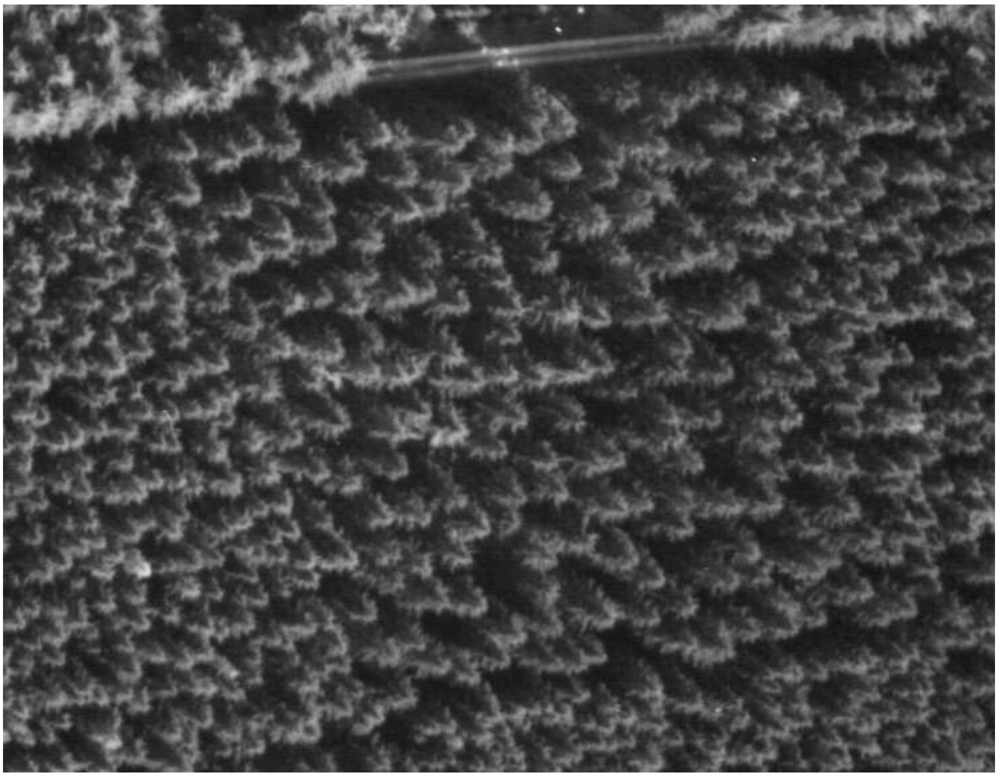


Figure 1.4: Detail of aerial photograph of subplot D of backlighted Norway spruce trees acquired from an oblique angle with the aeroplane located to the northwest of the experimental area shown in Figure 1.1.

Example 1.2. *Weed seeds.*

Figures 5 and 6 show images of 25 seeds of each of two weed species: curly dock, *Rumex crispus*, and thyrse sorrel, *Rumex thyrsiflorus.* The images were obtained in the study (Petersen, 1992), where seeds from 40 weed species were studied. The object was to find features from images of the weed seeds which enable recognition of the individual species. Problems of this type will be discussed in Chapter 2 on pattern recognition.

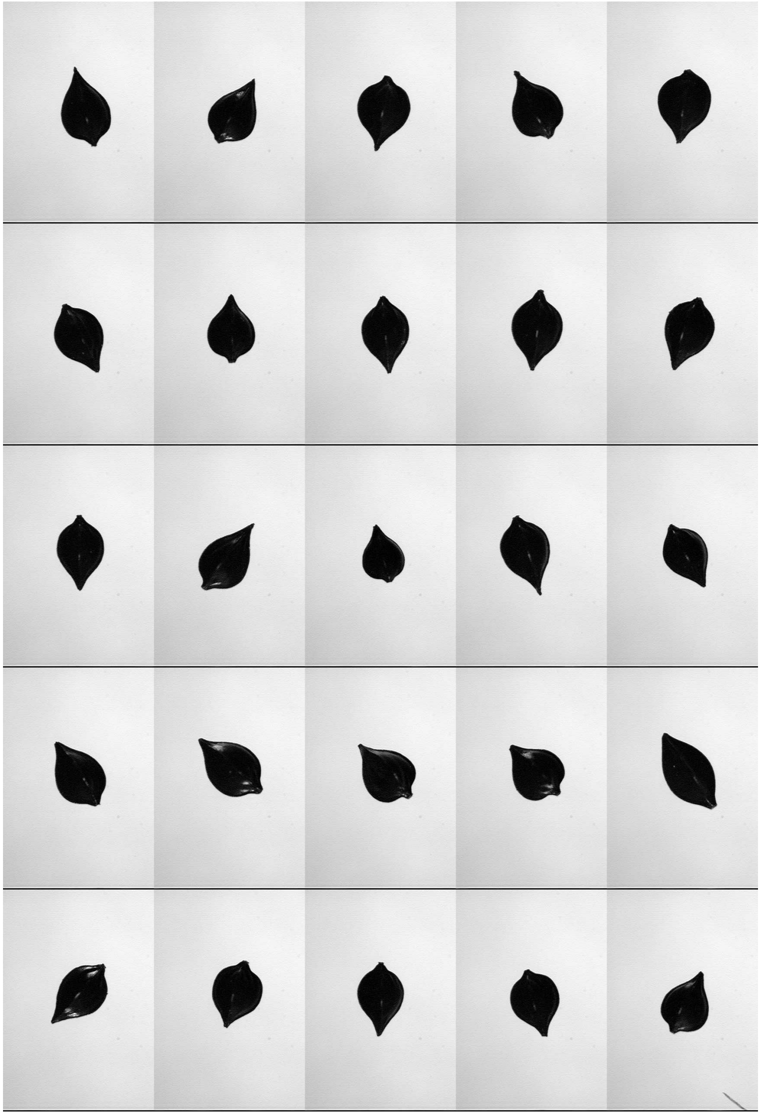


Figure 1.5: Images of seeds of *Rumex crispus.*

In Figures 1.5 and 1.6 we see varying orientations and sizes of the seeds but also some additional variation in the form of the contours. An important problem for series of images of this type, in addition to the previously mentioned pattern recognition, is to estimate some kind of average shape of a seed from a given species, and also to quantify in terms of statistical distributions the probable deviations from this average shape. In Chapter 8 on image warping and image matching such problems will be treated.

□

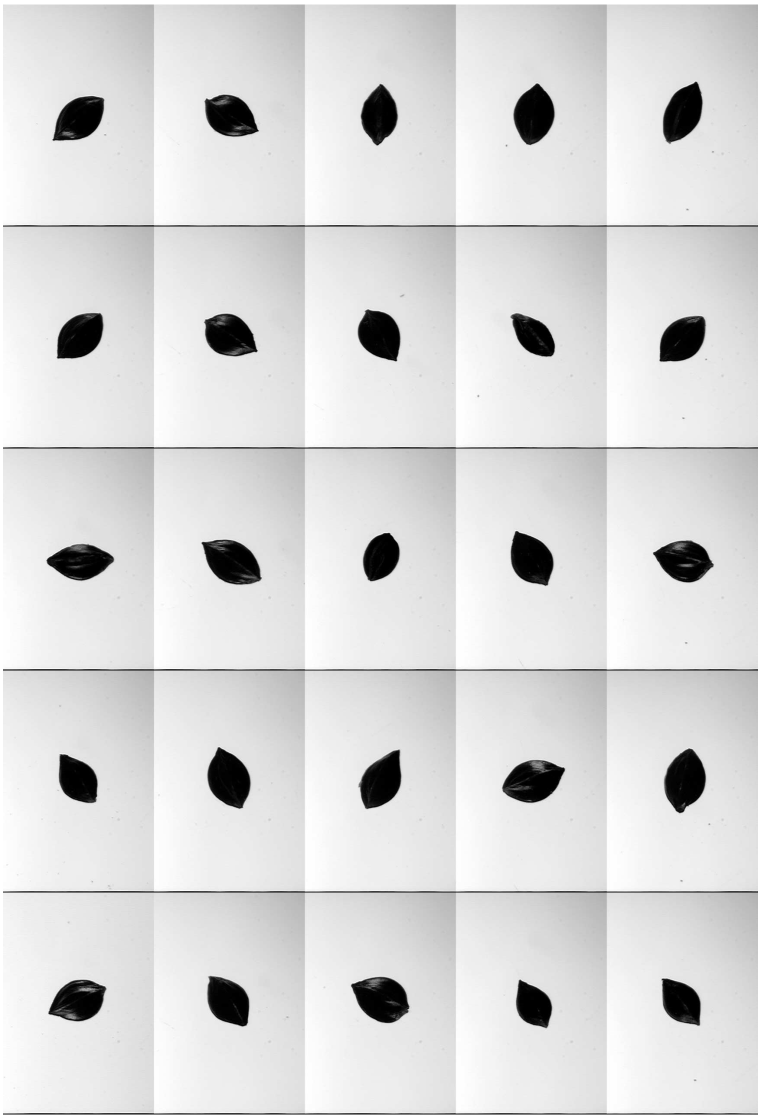


Figure 1.6: Images of seeds of *Rumex thyrsiflorus*.

Example 1.3. *Weed plants at an early stage.*

Weed and crop classification was studied by (Andersson, 1998) using a dataset with 27 images from each of 8 plant species: carrot, *Daucus carota*, which was the crop, and 7 weed species. Figure 1.7 shows photographs of two carrot plants and two ladythumb smartweed plants. Similarly, Figure 1.8 shows photographs of two fumitory plants and two corn spurry plants.

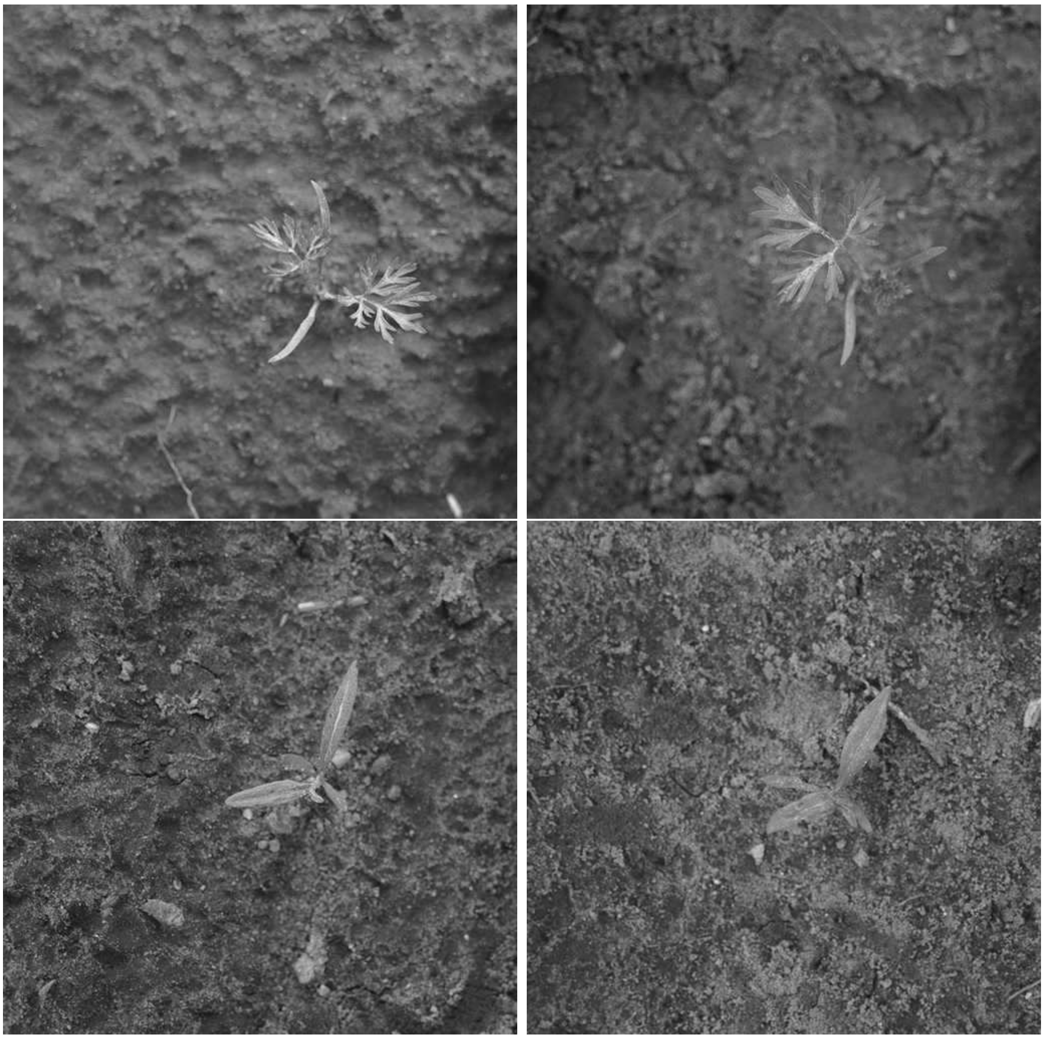


Figure 1.7: Above two images of plants of carrot, *Daucus carota*, L., and below two images of plants of ladythumb smartweed *Polygonum persicaria*, L.

The images were obtained with a Canon EOS500N still camera with a 80 mm zoom lens and mounted on a tri-pod pointing directly towards ground. The images obtained were in colour, although they are shown as grey-level images in Figures 1.7 and 1.8. The corresponding colour images may be obtained from
`http://www.math.chalmers.se/~rudemo/Images/WeedPlants/WeedPlants.html`
The number of pixels of the images was originally 512×768 but was reduced to 512×512 by cutting. The pixel width corresponds to 0.195 mm at ground level.

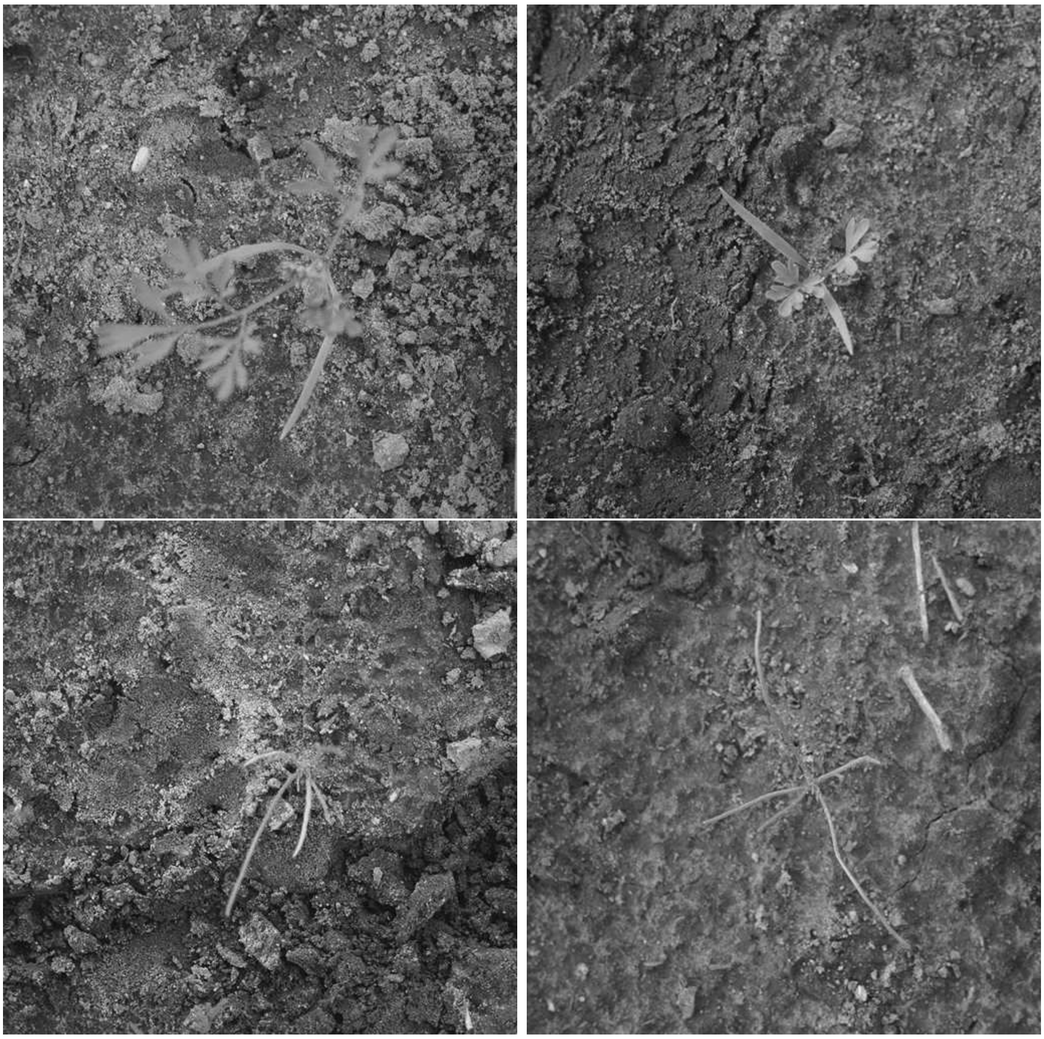


Figure 1.8: Above two images of plants of fumitory, *Fumaria officinalis*, L., and below two images of plants of corn spurry, *Spergula arvensis*, L.

□

Example 1.4. *Two-dimensional electrophoresis images.*

Yeasts are uni-cellular fungi which reproduce rapidly and thus are highly suitable as model systems for more complicated eucaryotic species such as mammals. In particular, the genome of baker's yeast, *Saccharomyces cervisiae*, was fully sequenced by (Goffeau *et al.*, 1996).

Figures 1.9 and 1.10 show four images from an experiment with baker's yeast and two treatments corresponding to growth under normal conditions and growth under stress with salt added to the nutrition solution, see (Gustafsson *et al.*, 2002). In the experiment there were five repetitions both for the standard treatment, corresponding to growth in a standard solution, and the treatment with growth under salt stress, which in this experiment corresponds to growth in a 1 M sodium chloride solution. Figure 1.9 shows the images obtained from two repetitions with the standard treatment and Figure 1.10 shows images from two repetitions with salt added.

Each spot in a 2D electrophoresis image such as in Figures 1.9 and 1.10 corresponds to a protein with a specific isoelectric point (pI) determined by isoelectric focusing in the horizontal direction as a first step and a specific molecular weight determined by vertical separation in a second step. For instance, under ideal conditions the protein molecules perform in the second

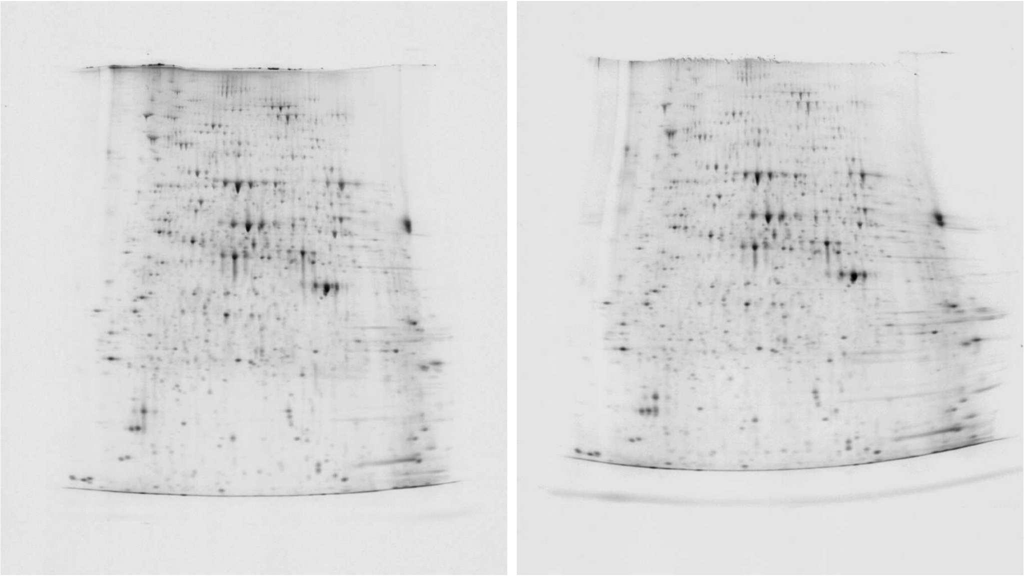


Figure 1.9: Images from 2D gel electrophoresis of baker's yeast grown in a standard solution.

step a vertical Brownian motion with drift from a starting position at the top such that small molecules travel a longer way than large molecules. Typically one can separate proteins in the pH range, or more precisely the pI range, 4–7 and with molecular weights in the range 5–200 kDa. Under favourable conditions thousands of proteins may thus be resolved, and the size of a spot in the electrophoresis image is a measure of the level of the corresponding protein.

The basic problem in an experiment such as the one described with yeast grown under standard conditions and under salt stress is to find those proteins that are upregulated and those that are downregulated under stress. As a first step we need to find those spots in the four images in Figures 1.9 and 1.10 that correspond to each other, that is, which measure the same protein. This is called matching of the images and may be performed by a warping of images onto each other. It is clear from an inspection of the two images in Figure 1.9, and similarly the two images in Figure 1.10, that also for experimental units that have received the same treatment the locations of spots corresponding to one protein can vary considerably due to random variation. And this random variation seems to be more complicated than the variation corresponding to a Brownian motion as referred to above.

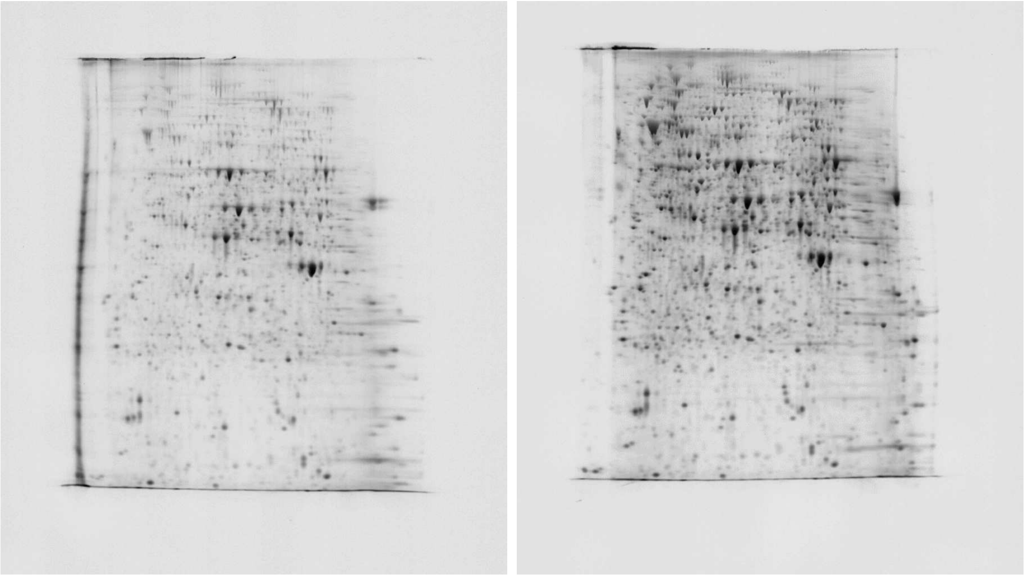


Figure 1.10: Images from 2D gel electrophoresis of baker's yeast grown under stress in a solution with salt added.

□

Example 1.5. *Two-colour spotted microarrays.*

In microarray analyses the expression level of thousands of genes can be estimated simultaneously. In two-colour spotted microarray analysis DNA fragments corresponding to different genes are typically arrayed on glass slides in spots with a diameter of the order 100 μm.

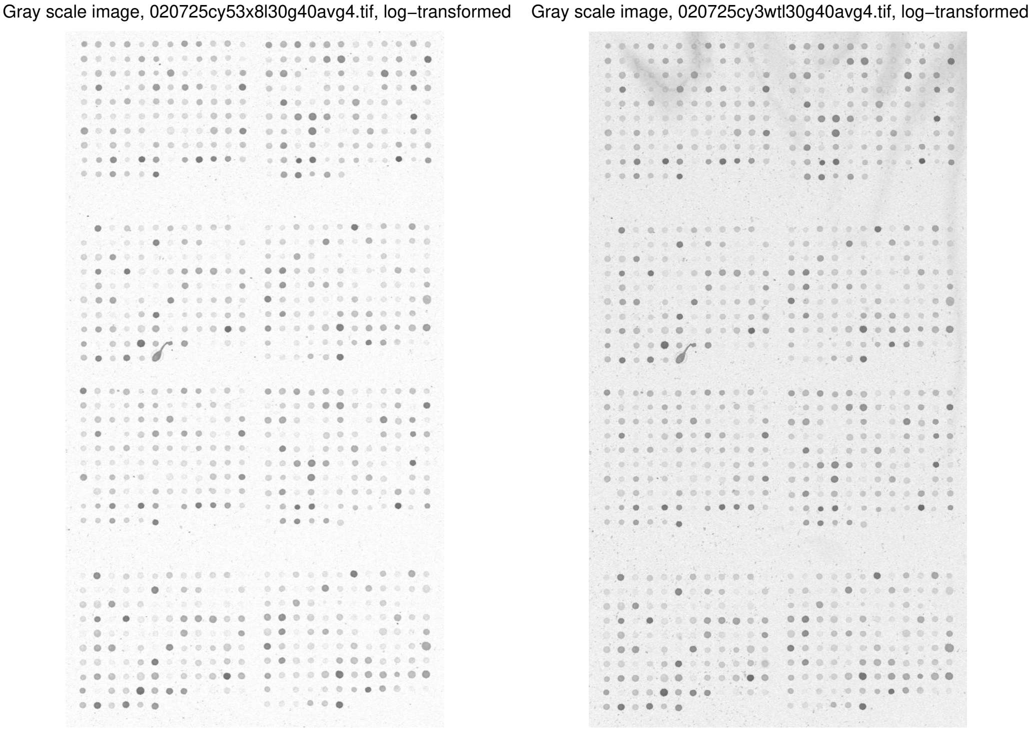

Gray scale image, 020725cy53x8l30g40avg4.tif, log−transformed
Gray scale image, 020725cy3wtl30g40avg4.tif, log−transformed

Figure 1.11: Images from an experiment with two varieties of *Arabidopsis*, Cy5 channel (left) for a transgenic line and Cy3 channel (right) for the wild-type in a two-colour spotted microarray experiment with 452 genes. The upper half with 20 rows contains all the 452 genes and the lower half is a repetition of the upper half. The images are shown inverted (high intensity shown as black) and a logarithmic scale transformation of intensities is also used.

Complementary DNA (cDNA) is synthesized from two sources of RNA of different origins and labeled with different fluorescent dyes, for instance, one with the green dye Cy3 and the other with the red dye Cy5. The pools of labeled cDNA are mixed together and allowed to hybridize with the DNA fragments in the different spots on the glass slide. The slide is illuminated with two laser light sources exciting the two fluorescent dyes and the intensity of emitted fluorescent light is measured at two suitably chosen wavelengths.

Figure 1.11 shows grey-level images for the two channels of one array in an experiment comparing RNA from two varieties of *Arabidopsis* plants, transgenic line 3x8 and wild-type wt (Kristensen *et al.*, 2005). For clarity of display the images are shown inverted, that is black corresponds to high intensity levels and before inversion a logarithmic transformation is also used. Data transformations and spot shape models for spotted microarrays are discussed in (Ekstrøm *et al.*, 2004) and applied to data from this experiment.

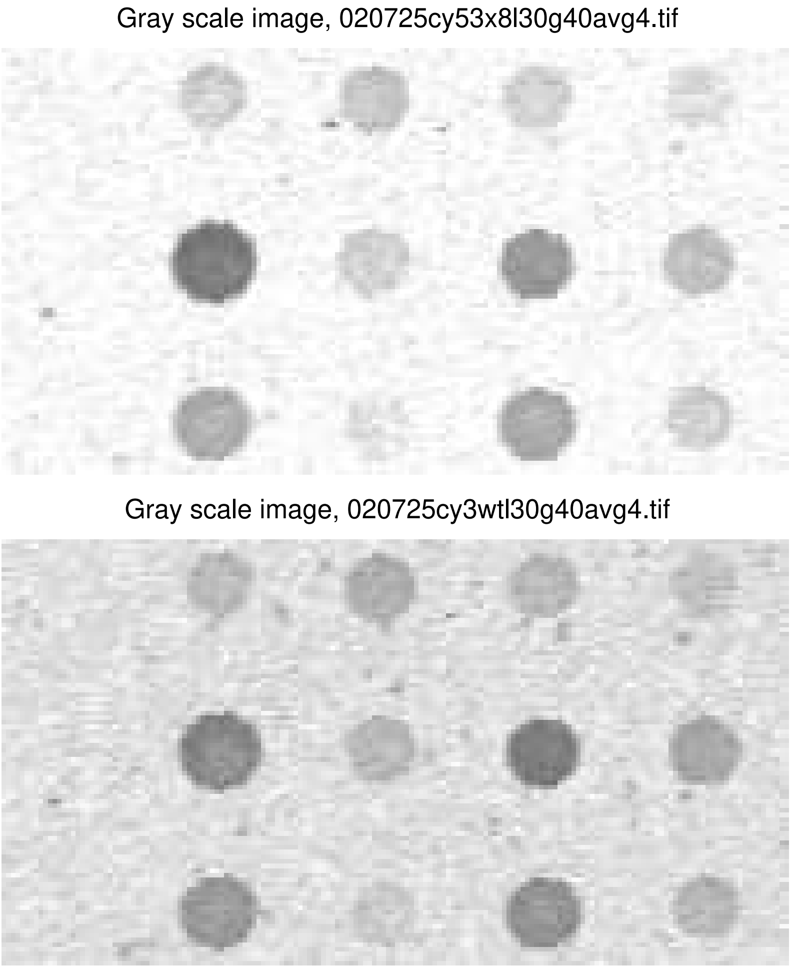

Gray scale image, 020725cy53x8l30g40avg4.tif
Gray scale image, 020725cy3wtl30g40avg4.tif

Figure 1.12: Blow-up of rows 6–8 and columns 1–4 in Figure 1.11 with the Cy5 channel for the transgenic line above and the Cy3 channel for the wild-type below.

Figure 1.12 shows a blow-up with 3 rows and 4 columns for both channels. One crucial question analysed in experiments of this type is to find out which genes that are differentiably expressed, that is show significantly higher or lower intensities. In this experiment it turned out that remarkably few genes in the transgenic line were affected in the comparison with the wild-type. One of the few genes affected was the gene that corresponds to the first spot in the middle row in Figure 1.12. As indicated in the figure it was upregulated in the transgenic line. However, random errors are large in this type of experiments and typically one needs to repeat the experiment for several slides and make a subsequent statistical analysis of the results, cf. Chapter 9. □

Example 1.6. *Diffusing particles.*

Colloidal particles in a suspension perform random motion essentially as a three-dimensional Brownian motion with the diffusion coefficient as a crucial parameter. However, as the particles come close they interact and this interaction may be described by an interaction potential.

A series of images were obtained by video microscopy, see (Kvarnström, 2005), in a joint project with Lennart Lindfors, AstraZeneca, Mölndal. The object in this project was to estimate the diffusion coefficient and, if possible, also the particle interaction potential.

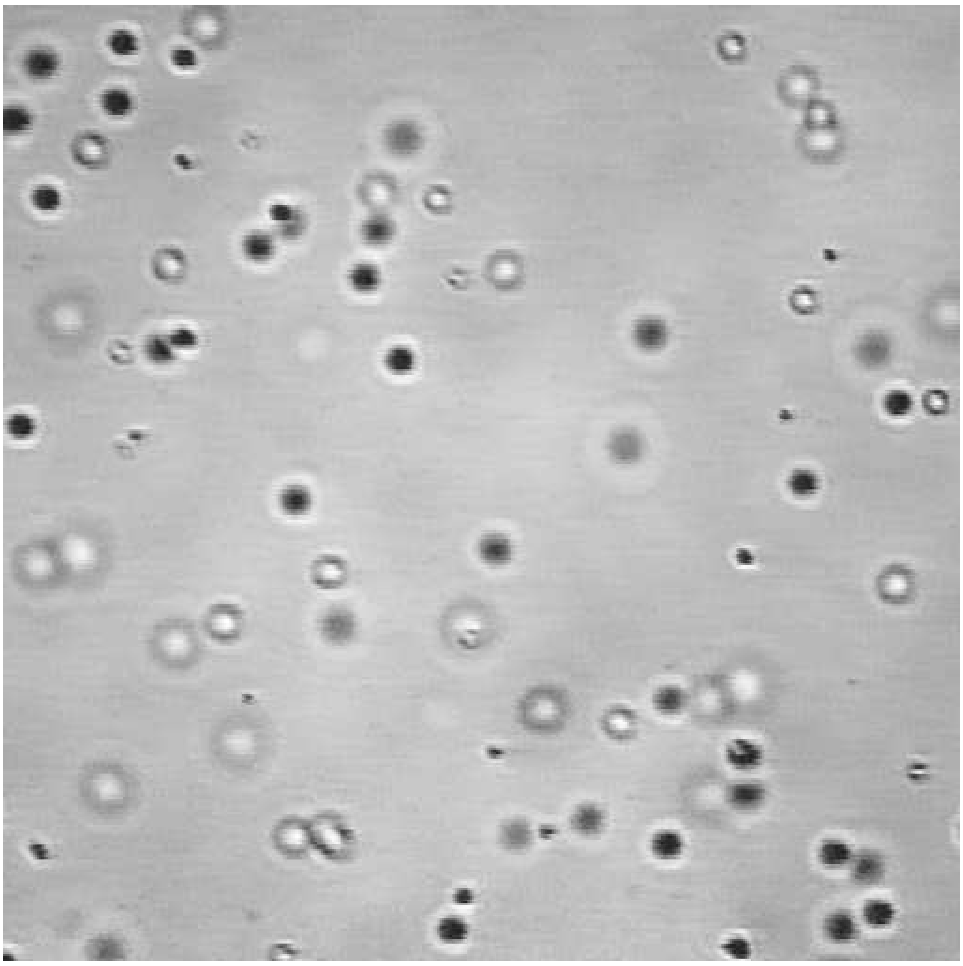


Figure 1.13: Image obtained by video microscopy showing diffusing particles. Particles in phocus are shown as small distinct black objects.

Images of the diffusing particles were obtained with a time interval of 0.02 seconds between images, and two consecutive images are shown in Figure 1.13 and Figure 1.14. Particles in focus are shown as small distinct black objects, while particles out of phocus are extended, the degree of extension depending on the distance to the phocal plane. An object corresponding to a particle out of phocus is further either white or black in its central part corresponding to the particle being above or below phocus, respectively.

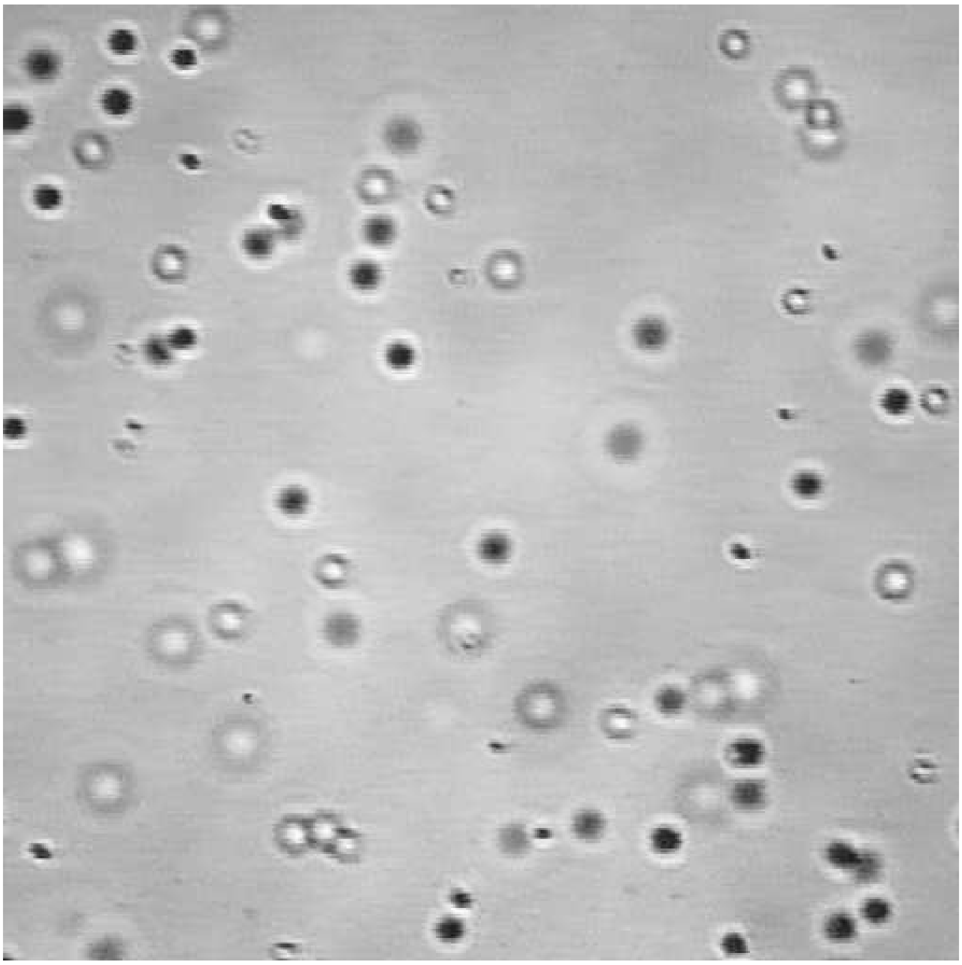


Figure 1.14: Image obtained by video microscopy showing diffusing particles. This image was obtained 0.2 seconds after the image in Figure 1.13.

□

Example 1.7. *Handwritten digits.*

The MNIST database of handwritten images consists of a training set with 60 000 digits and an evaluation set of 10 000 digits, see (LeCun *et al.*, 1998) and

http://yann.lecun.com/exdb/mnist/

Examples of images from this set is given in Figure 1.15, actually the first 100 digits from the training set. The digit images are 28×28 pixel grey level images obtained from 20x20 pixel binary black and white images. The MNIST dataset has been used extensively as a proving ground for pattern recognition methods and it will also be used substantially in this book in Chapters 2, 3 and 8.

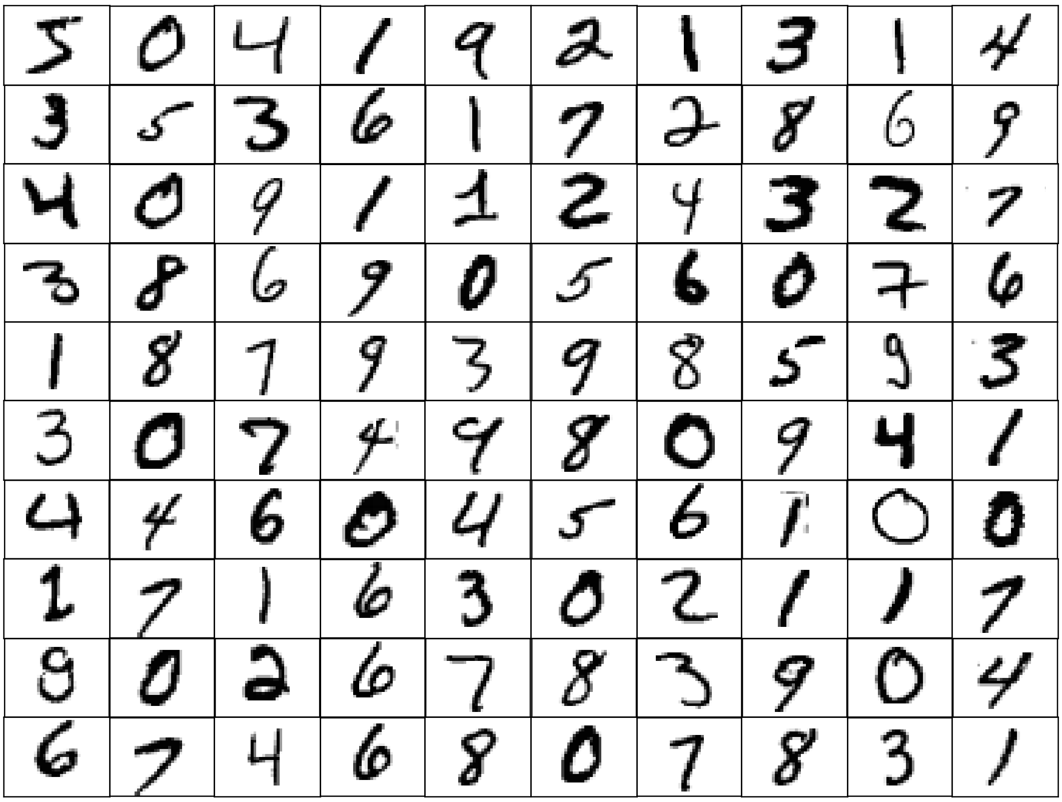


Figure 1.15: Examples of 100 handwritten digits from the MNIST database.

□

1.2 Image filtering

Let $w = (w_{k,l}) = (w_{k,l}, k = -p, -p+1, \ldots p, l = -p, -p+1, \ldots, p)$ be a matrix of real numbers. A new image g may be constructed from a given image f by linear filtering,

$$g_{ij} = \sum_{k=-p}^{p} \sum_{l=-p}^{p} w_{k,l} f_{i+k,j+l}. \tag{1.1}$$

A simple filter example is a 3×3 averaging filter

$$w = \begin{bmatrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{1.2}$$

A more smooth averaging filter is obtained by use of circular 2D Gaussian filter with a variance parameter σ^2,

$$w_{k,l} = c \exp(-\frac{1}{2\sigma^2}(k^2 + l^2)), \tag{1.3}$$

where c is chosen such that

$$\sum_{k=-p}^{p} \sum_{l=-p}^{p} w_{k,l} = 1, \tag{1.4}$$

and p is chosen so that $w_{k,l}$ is small outside the region determined by $|k| \leq p$ and $|l| \leq p$. Chose, for instance, p to be the smallest integer which is at least as large as 3σ.

Care has to be taken in (1.1) when the indices in the summation fall outside the original image. One possibility is to restrict the filtering to those pairs (i, j) for which all indices $i+k$ and $j+l$ in (1.1) fall inside the image f, another possibility is to extend the original image in a suitable way, and a third possibility is to modify the filter close to the image edges.

The averaging filter (1.2) is relatively vulnerable to large errors in individual pixels. A more robust filter is the nonlinear *median* filter which for 3×3 neighbourhood is given by

$$g_{ij} = median\{f_{i+k,j+l} : |k| \leq p, |l| \leq p\} \tag{1.5}$$

with $p = 1$. Here $median(A)$ denotes the median for a finite set A of real numbers.

Image filtering can also be used to emphasize edges. Thus a linear filter with

$$w = \begin{bmatrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}. \tag{1.6}$$

will tend to emphasize vertical edges, and similarly the filter

$$w = \begin{bmatrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{1.7}$$

will tend to emphasize horisontal edges.

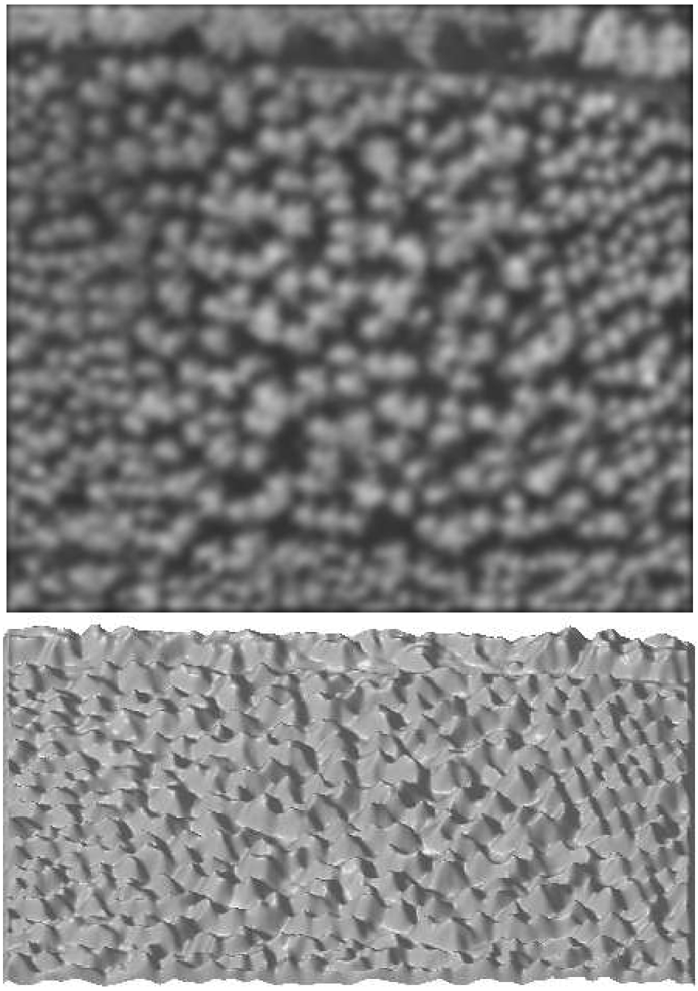


Figure 1.16: Upper part: Smoothed version of the image in Figure 1.2 by use of circular 2D Gaussian filter with $\sigma = 4.5$ pixel-widths. Lower part: The same image viewn in perspective as a 3D surface with light intensity as the vertical coordinate.

Example 1.8. *Aerial photographs of a thinning experiment. Continuation.*

Let us smooth the image in Figure 1.2 by use of a circular 2D Gaussian filter with a suitably chosen parameter σ to see if we can estimate the locations of the trees as 'whiteness' maxima in the smoothed image. With $\sigma = 4.5$ we find the image in Figure 1.16.

From Figure 1.16 and Figure 1.2 we see that maxima in the smoothed image seem to correspond well to the location of the trees. This is also indicated by Figure 1.17 which shows the locations of the maxima of the smoothed image (Here we have only included maxima which have a distance from the nearest edge which exceeds 3σ.)

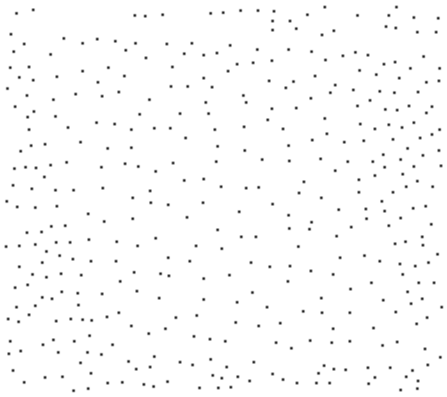


Figure 1.17: Location of maxima in Figure 1.16.

□

1.3 Histograms, thresholding and segmentation

An important characteristic of an image is its histogram. For a grey scale image, $f = (f_{ij}) = (f_{ij}, i = 1, \ldots, m, j = 1, \ldots, n)$, where $f_{ij} \in V$ with V as a set of real numbers, the histogram is defined as

$$h_k = \text{card}(\{(i,j) : f_{ij} \in I_k\}), \quad k = 1, \ldots, K, \tag{1.8}$$

where card(A) denotes the number of elements in the set A and $\{I_1, \ldots, I_K\}$ is a set of disjoint intervals with V as there union.

If an image consists of two parts with grey levels that do not overlap too much the histogram can be used to find a threshold level t which enables us to divide the image into two segments corresponding to these parts. Thus we can define a binary image $b = (b_{ij})$ with two levels, 0 and 1, by putting

$$b_{ij} = \begin{cases} 0 & \text{if } f_{ij} \leq t \\ 1 & \text{if } f_{ij} > t. \end{cases} \tag{1.9}$$

Segmentation by use of a threshold level found by inspection of the histogram of an image is illustrated in the following example.

Example 1.9. *Weed seeds. Continuation.*

In the upper part Figure 1.18 we see one of the seeds from Figure 1.5, actually the seed in the lower left corner rotated 90 degrees. In the lower part of the figure we see the corresponding histogram.

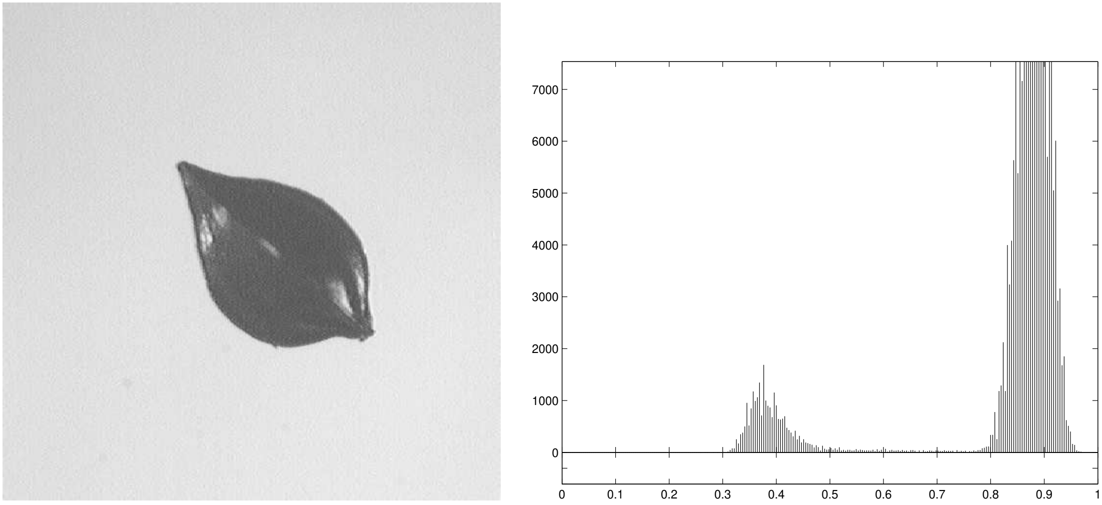


Figure 1.18: To the left an image of a *Rumex crispus* seed and to the right the corresponding histogram.

It seems clear that a threshold level somewhere between $t = 0.5$ and $t = 0.8$ would be suitable. In Figure 1.19 we see segmentations with the levels $t = 0.5$, upper left, $t = 0.8$, upper right, and $t = 0.65$, lower left. In the lower right part of the image we see a segmentation obtained from the lower left image by filling out the white "holes", an operation that can be performed in several ways.

□

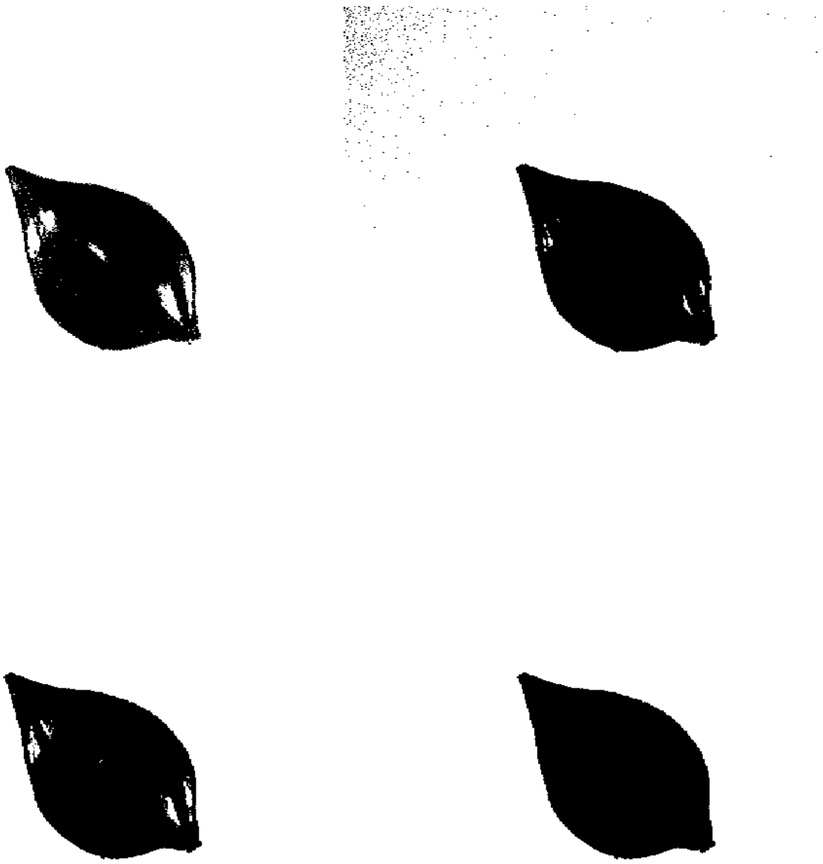


Figure 1.19: Binary images obtained by thresholding of the image in Figure 1.18 with the levels $t = 0.5$ (upper left), $t = 0.8$ (upper right), and $t = 0.65$ (lower left). The lower right image is obtained from the lower left image by filling out holes.

1.3.1 Segmentation by a normal mixture model

In many cases, as in Example 1.9 with a bimodal histogram it is fairly easy to separate components in a mixture. We will now describe a normal mixture model which can be used to get a precise threshold value and which also can be used in cases where there are not two modes in the histogram but one component only shows up as a prolonged tail. We suppose that the sets I_k in (1.8) consist of consecutive intervals with midpoints x_k and equal lengths Δ. Let $\phi(x) = (1/\sqrt{2\pi}) \exp(-x^2/2)$ and put

$$f(x; p_1, \mu_1, \sigma_1, \mu_2, \sigma_2) = \frac{p_1}{\sigma_1}\phi((x-\mu_1)/\sigma_1) + \frac{(1-p_1)}{\sigma_2}\phi((x-\mu_2)/\sigma_2). \tag{1.10}$$

We note that $f(\cdot; p_1, \mu_1, \sigma_1, \mu_2, \sigma_2)$ integrates to one, and if the interval length Δ is small we should have

$$\Delta \sum_k f(x_k; p_1, \mu_1, \sigma_1, \mu_2, \sigma_2) \approx 1. \tag{1.11}$$

Let $n = \sum_k h_k$ denote the total number of pixels and assume that

$$h_k \approx n\Delta f(x_k; p_1, \mu_1, \sigma_1, \mu_2, \sigma_2). \tag{1.12}$$

To estimate the parameters $p_1, \mu_1, \sigma_1, \mu_2, \sigma_2$ we minimize

$$Q(p_1, \mu_1, \sigma_1, \mu_2, \sigma_2) = \sum_k (h_k - n\Delta f(x_k; p_1, \mu_1, \sigma_1, \mu_2, \sigma_2))^2. \tag{1.13}$$

Example 1.10. *Weed plants at an early stage. Continuation*

In the upper left part of Figure 1.20 we see the grey level image of a weed plant. The original a image is colour a image with three channels, blue, green and red. To separate plant pixels from soil pixels we first regard the green channel which is shown in the upper right part of Figure 1.20. To improve the separation of plant and soil pixels we consider the normalized green colour, which for pixel (i, j) has the pixel value

$$g_{ij} = \text{Round}(\,255 G_{ij} \,/\, (B_{ij} + G_{ij} + R_{ij}) \;+\; 1), \tag{1.14}$$

where B_{ij}, G_{ij} and R_{ij} are the blue, green and red channel values for the colour image, and Round$(\cdot)$ denotes rounding to the nearest integer. The normalized green image is shown in the lower left part of Figure 1.20. The histogram for the normalized green channel is shown in the left part of Figure 1.21. Can you suggest why it is useful to normalize the green channel before computing the histogram? Now we fit the normal mixture model given by (1.10) and (1.12) for the normalized green channel by minimizing Q in (1.13) with the restriction $\mu_1 > \mu_2$. Thus the first component should correspond to plant pixels. Let $\hat{p}_1, \hat{\mu}_1, \hat{\sigma}_1, \hat{\mu}_2, \hat{\sigma}_2$ denote the estimated parameters. In Figure 1.21 we show the histogram and the fitted normal components.

To segment an images we could then choose to consider a pixel (i, j) as a plant pixel if $g_{ij} > T$, where the threshold $\hat{T}$ is obtained by solving the equation

$$\frac{\hat{p}_1}{\hat{\sigma}_1}\phi((\hat{T}-\hat{\mu}_1)/\hat{\sigma}_1) = \frac{(1-\hat{p}_1)}{\hat{\sigma}_2}\phi((\hat{T}-\hat{\mu}_2)/\hat{\sigma}_2) \tag{1.15}$$

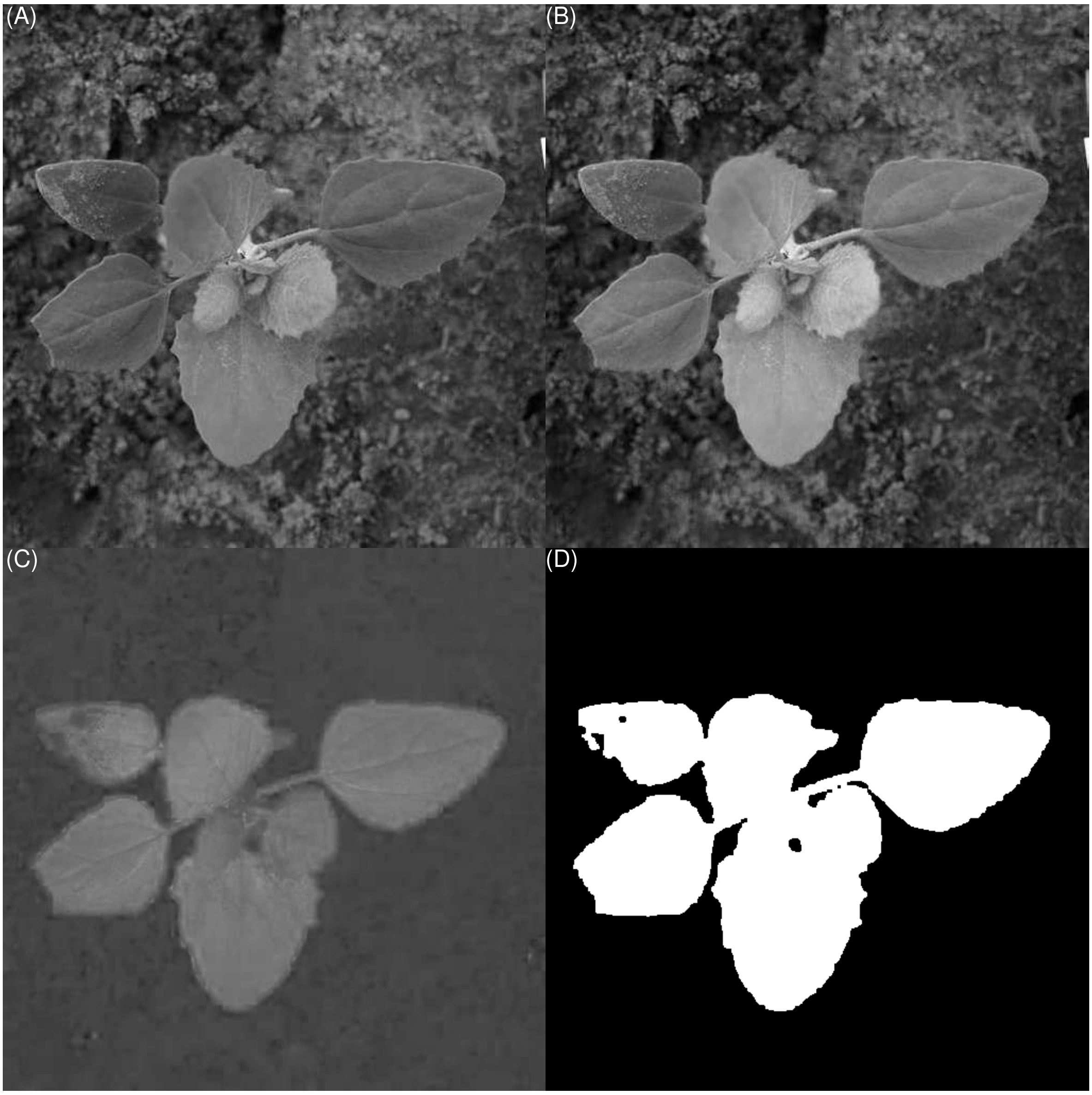


Figure 1.20: Images of a weed plant, lamb's quarter *Chenopodium album*, L., (A) grey scale image, (B) green channel image, (C) normalized green channel image, and (D) binary black and white image after thresholding.

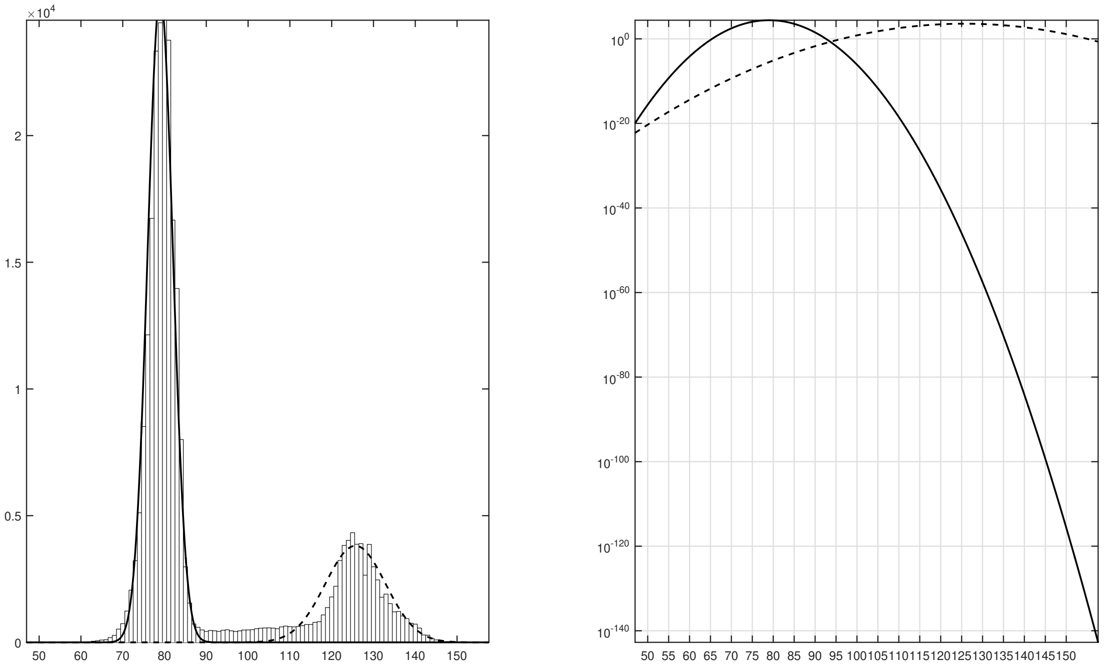


Figure 1.21: Left: histogram for the normalized green channel shown in the lower left part of Figure 1.20 and the two components shown as fully drawn and dashed curves. Right: the two components shown with a log scale on the vertical axis; here the threshold where the two curves cross can be seen.

and otherwise as a soil pixel. In the lower right part of Figure 1.20 we show the resulting binary black and white image obtained by thresholding the normalized green channnel. For the image shown in Figure 1.20 we find the following parameter estimates for the two component normal mixture model

$$\hat{p}_1 = 0.263, \;\; \hat{\mu}_1 = 126, \;\; \hat{\sigma}_1 = 7.22, \;\; \hat{\mu}_2 = 79.0, \;\; \hat{\sigma}_2 = 3.02, \;\; \hat{T} = 93.6. \tag{1.16}$$

□

1.4 The Hough transform

Often one tries to find curves of specific types in images, for instance circles, ellipses or lines. A useful method to find such curves is the Hough transform (Hough, 1959; Duda & Hart, 1972). We shall here only look at the use of the Hough transform to find straight lines.

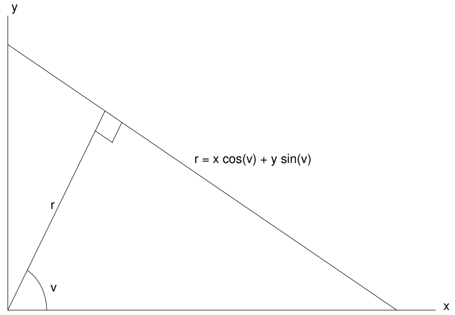


Figure 1.22: Representation of line in terms of angle and distance to origo.

Suppose that we have found a set S of points in an image, such as the set of tree tops in Figure 1.17. We are interested in finding out whether some of these points lie on lines. It is here convenient to use a representation of a line in terms of the distance r to the origin and the angle that the normal from the origin to the line forms with the horizontal axis,

$$r = xcos(v) + ysin(v), \tag{1.17}$$

see Figure 1.22. A point (x, y) in the original image corresponds now to a curve in the (r, v)-plane obtained by regarding r as a function of v in (1.17) for fixed (x, y). In practice we discretize the (r, v)-plane into pixels regarding it as an image H and start by assigning zero to all the pixels in H. Then for each point $(x, y) \in S$ we add one to all pixels in H which the curve (1.17) passes through.

For the set S of maxima in Figure 1.17 the corresponding Hough transform for finding lines is shown in Figure 1.23. In particular one finds in Figure 1.23 three maxima in the upper left part all corresponding to the angle v equal to 16 degrees (a corresponding tick mark is placed on the horizontal axis) and three distances r (marked with three tick marks on the vertical axis close to the maximal distance r_{max}. The corresponding three lines are shown in Figure 1.24.

The three lines found in Figure 1.24 correspond actually to three lines in plot R in Figure 1.1 with “Heavy row thinning”, that is from the original planting in rows thinning is performed by eliminating totally some rows keeping, say, only every third row. See also Figure 1.2 where the rows are clearly seen in the right part of the image.

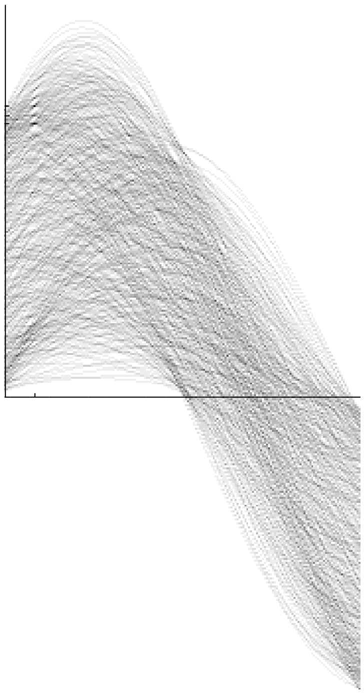


Figure 1.23: Hough transform for Figure 1.17 with angle v on the horizontal axis extending from 0 to 180 degrees and distance r on the vertical axis extending from $-r_{max}$ to r_{max}, where r_{max} is the length of the diagonal in Figure 1.2.

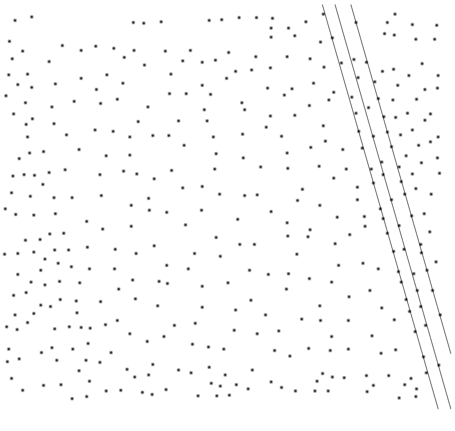


Figure 1.24: Location of maxima in Figure 1.16 together with three lines found by the Hough transform.

1.5 Morphological operations

Morphological operations can be used to regularize or clean binary images. Here we will only describe some of the most basic operations such as erosion, dilation, opening and closing. These operations are defined by a structure element S consisting of a small number of pixels with one specific pixel called reference pixel. We can, for instance, choose S as a 3×3 set of pixels with the centre pixel as reference. Let $S_{i,j}$ denote the structure element moved with reference pixel to (i,j). Let A be a set of pixels such as the set consisting of black pixels in one of the four images in Figure 1.19.

The *erosion* of A, denoted $A \ominus S$, is defined by

$$A \ominus S = \{(i,j) : S_{i,j} \subseteq A\} \tag{1.18}$$

The *dilation* of A, denoted $A \oplus S$, is defined by

$$A \oplus S = (A^c \ominus S)^c, \tag{1.19}$$

where A^c is the complement af A, that is the set of pixels outside A.

The operations *opening* and *closing*, denoted $\psi_S(A)$ and $\phi_S(A)$, are defined by

$$\psi_S(A) = (A \ominus S) \oplus S', \tag{1.20}$$

where S' denotes the structure element rotated 180^o around the reference pixel, and

$$\phi_S(A) = (A \oplus S) \ominus S'. \tag{1.21}$$

Thus an opening consists of an erosion followed by a dilation.

1.6 Object feature measurements

In connection with pattern recognition as mentioned in examples 1.2 and 1.3 we seek features of the objects, in the examples seeds and plants, which would enable us to distinguish between different classes of objects. Examples of such features are areas and perimeters of objects. Consider a set A of pixels as in the previous section on morphological operations. The area of A is typically defined as the number of pixels in A, possibly with some regularization operation first applied to A.

To define the perimeter we need the concept of neighbouring pixels. Typically we consider neighbourhoods consisting of either four or eight neighbours. The 4-neighbourhood of pixel (i,j) consists of the four pixels $(i-1,j)$, $(i+1,j)$, $(i,j-1)$ and $(i,j+1)$. The 8-neighbourhood of pixel (i,j) consists of the aforementioned pixels and in addition the pixels $(i-1,j-1)$, $(i-1,j+1)$, $(i+1,j-1)$ and $(i+1,j+1)$.

Edge pixels of a set A may be defined as those pixels of A that have at least one neighbour from A^c, the complement of A. Let $N(A)$ denote the number of edge pixels of A with at least one 4-neighbour in A^c. Then one can show that

$$\text{perimeter}(A) = N(A)/k_4, \tag{1.22}$$

where

$$k_4 = \frac{4}{\pi} \int_0^{\pi/4} \cos\theta \, d\theta = \frac{4}{\pi/\sqrt{2}} \approx 0.900, \tag{1.23}$$

is an approximately unbiased estimate of the perimeter of A provided that all orientations of the perimeter are approximately equally common. The unit of the perimeter estimate (1.22) is pixel width. As with the area, it may be useful to regularize A in some way before evaluating the perimeter. For more accurate perimeter estimates, see (Glasbey & Horgan, 1995), pp 165–168, and further references given there.

A feature often used is the compactness of an object defined to be

$$\text{compactness}(A) = 4\pi \frac{\text{area}(A)}{(\text{perimeter}(A))^2}. \tag{1.24}$$

Sometimes it is useful to compare a set A of pixels with the convex hull of A, that is the smallest convex set containing A. Some care has to taken in defining convexity for a set of pixels; one possibility is to define convexity for the point set of pixel centres. The convex perimeter of a set A is then defined to be the perimeter of the convex hull of A. One useful feature is the convexity of A defined by

$$\text{convexity}(A) = \frac{\text{convex\_perimeter}(A)}{\text{perimeter}(A)}. \tag{1.25}$$

1.6.1 Moment features

Consider a grey level or binary image $f = (f_{ij}) = (f_{ij})$, and let $A \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ be a subset of pixels, typically corresponding to an object but sometimes the whole image. The moment of order (p, q) in A is defined as

$$m_{pq} = m_{pq}(A) = \sum_{(i,j) \in A} i^p j^q f_{ij}, \;\; p = 0, 1. \ldots, \;\; q = 0, 1, \ldots, \tag{1.26}$$

and the *centroid* is defined as

$$\text{centroid} = \text{centroid}(A) = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}). \tag{1.27}$$

We also consider central moments (with respect to the centroid)

$$\mu_{pq} = \mu_{pq}(A) = \sum_{(i,j) \in A} (i - \frac{m_{10}}{m_{00}})^p (j - \frac{m_{01}}{m_{00}})^q f_{ij}, \;\; p + q > 1. \tag{1.28}$$

One could note that central moments are invariant with respect to translation of objects. It is possible to construct moments that are also invariant with respect to rotations. Two such second order moments are

$$\mu_{20} + \mu_{02} \;\; \text{and} \;\; (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2. \tag{1.29}$$

An informative discussion of different types of moments with literature references can be found in (Glasbey & Horgan, 1995), pages 156–161.

In Example 1.10 we saw how we could discriminate between plant and soil pixels quite well by use of a suitable feature, the normalized green colour. To discriminate between classes of objects we can as will be seen in detail in the next chapter on pattern recognition use a number of suitable chosen feature variables. In the following example we will consider two feature variables and a suitable plotting technique.

Example 1.11. *Handwritten digits. Continuation*

In this example we will consider discrimination between digits "one" and "two" by use of two second order moments. We use digits "one" and "two" among the first 400 digits in MNIST. Plotting moment μ_{11} on the vertical axis versus moment μ_{20} on the horizontal axis we get the plot shown in Figure 1.25. Try to draw by free hand first a straight line and then an ellipse that gives as good a discrimination as possible betweens the "one" and "two" digits. In the next chapter we shall describe systematic methods to draw such boundaries. □

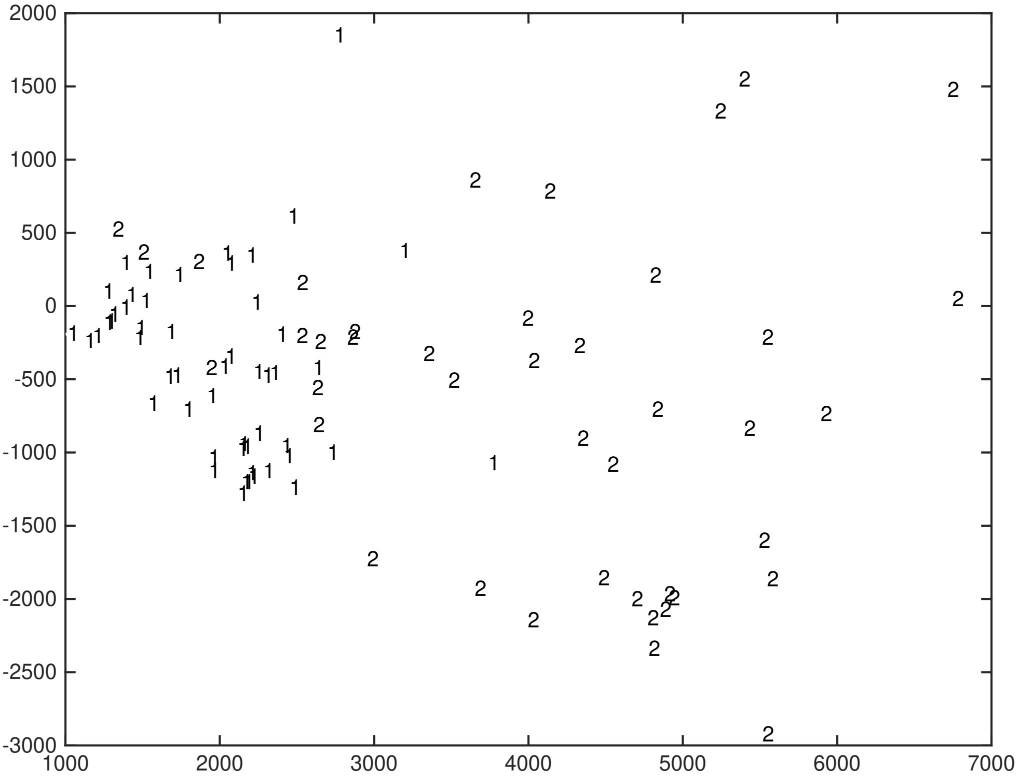


Figure 1.25: Plot of μ_{11} versus moment μ_{20} for handwritten digits digits 1 and 2 among the first 400 digits in the MNIST data base.

1.6.2 Exercises

The images used in the exercises below may be found at
`http://www.math.chalmers.se/~rudemo/images.html`

Exercise 1.1. Let R, G and B denote the values in the red, green and blue channels for one of the images from Example 1.3. Get the grey-level image corresponding to normalized green,

$$g = \frac{G}{R+G+B}.$$

Exercise 1.2. Find the histogram for the image of Exercise 1.1. Try to segment the image by use of the histogram.

Exercise 1.3. Compute area, perimeter and compactness for the green segment for the image of the two previous exercises.

Exercise 1.4. Get one of the seed images from Example 1.2. Note that one has to resample the image to get the correct form of the seed. How can that be done? After resampling, reduce the number of columns to get a square image.

Exercise 1.5. Apply the averaging filter (1.2), the median filter (1.5) and the edge emphasizing filters (1.6) and (1.7) to the image of the previous exercise.

Exercise 1.6. Consider the image from Exercise 1.4. Compute the histogram and transform to a binary image. Zoom in to see the individual pixels at the object edge. Apply the operations erosion, dilation, opening and closing. What is the effect of theses operations? What happens when one iterates these operations?

1.6.3 Literature on image analysis

There is a wealth of books on digital image processing. An excellent treatment from a statistical point of view focussing on examples from biology is given in (Glasbey & Horgan, 1995). A mathematically oriented text is (Rosenfeld & Kak, 1982), which is now a bit old but still quite useful. A comprehensive treatment of image processing, analysis and machine vision may be found in (Sonka *et al.*, 2015).

Chapter 2

Pattern recognition

Humans are particularly good at recognizing many patterns such as faces and voices of other individuals. A possibly harmful behaviour of another person or the appearance of a possibly dangerous animal may also be quickly identified. Obviously such pattern recognition abilities have implied a survival advantage during the evolution of humans.

By training humans can also be astonishingly good at tasks such as recognizing the species of a bird at a long distance, perhaps by using a combination of features such as the bird's shape and colours, its vocalization and its mode of flight. The human observer's previous knowledge of how common possible bird species are in the current environment at the given time of the year may also be highly useful in identifying the species.

One important task in pattern recognition based on digital images is to try to mimic human pattern recognition by choice of suitable features for recognizing and classifying observed objects. We can divide the field of pattern classification into two disciplines depending on the our previous knowledge of the possible classes. The most well developed discipline is *discriminant analysis* where we assume that we have a given number of classes and that we have a new object that we want to assign to one of these classes. Typically we also assume here that we have a set of objects for which we know the classes. Such a data set, often called a training set, will help us to choose the relevant features of the objects and to design the algorithm for recognizing the class by use of the chosen features. Therefore discriminant analysis is often called *supervised pattern recognition* or *learning with a teacher.*

In the second discipline, called *cluster analysis* we do not assume any prior knowledge of possible classes. However, we will typically assume that we also here have a given data set but without any classification. The data set will be used to find clusters, and the discipline is often referred to as *unsupervised pattern recognition* or *learning without a teacher.*

We will start by discussing discriminant analysis. Several of the sets of images in the previous chapter, the weed seeds in Example 1.2, the weed plants in Example 1.3 and the handwritten digits in Example 1.7 describe problems that call for discriminant analysis.

2.1 Optimal discrimination with two classes and a one feature variable

Suppose that we have two classes ω_1 and ω_2 and a real-valued feature variable X for each object to be classified. Assume that we know how common the two classes are, that is, we know the prior probabilities of the two classes. Assume also that we know the distributions of the feature variable corresponding to the two classes.

For $i = 1, 2$, let π_i denote the prior probability of class ω_i and let f_i be the probability density of X for an observation from class ω_i, or the probability function, $f_i(x) = P(X = x)$, if X is a discrete random variable.

The problem of deciding if an object comes from class ω_1 or ω_2 is to be based on observation of the corresponding feature variable X. Thus we need to specify two disjoint sets A_1 and A_2 with $A_1 \cup A_2 = \mathbb{R}$ and choose class ω_i if $X \in A_i$. To find optimal sets we need further specification corresponding to how costly it is to make different kinds of errors, that is the cost of choosing class ω_1 when ω_2 is true and vice versa. Let us first assume that these cost are equal, and more specifically, that we want to minimize the probability of misclassification.

It turns out that the probability of misclassification is minimized if we use the following rule:

$$\text{choose class } \omega_1 \text{ if } \pi_1 f_1(x) > \pi_2 f_2(x), \tag{2.1}$$

$$\text{choose class } \omega_2 \text{ if } \pi_1 f_1(x) < \pi_2 f_2(x). \tag{2.2}$$

To show that a decision rule satisfying (2.1) and (2.2) is optimal we note that the probability of misclassification is generally given by

$$\begin{aligned}\Pr(\text{misclassification}) &= \Pr(\omega_1 \text{ true and misclassification}) + \Pr(\omega_2 \text{ true and misclassification})\\ &= \Pr(\omega_1)\Pr(\text{misclassification}|\omega_1) + \Pr(\omega_2)\Pr(\text{misclassification}|\omega_2)\\ &= \pi_1 \int_{A_2} f_1(x)dx + \pi_2 \int_{A_1} f_2(x)dx.\end{aligned}$$

In Figure 2.1 the set A_1 extends up to a threshold t while A_2 is chosen above t. The probability of misclassification is equal to the area of the coloured region, and it follows that it is minimized precisely when the threshold is the horisontal location of the crossing point of the two curves. Thus the misclassification probability is minimized if A_1 and A_2 are chosen as in (2.1) and (2.2). (We note that x-values such that $\pi_1 f_1(x) = \pi_2 f_2(x)$ may be brought to either A_1 or A_2 without affecting the misclassification probability.)

Example 2.12. *Two-class discriminant analysis with estimated normal densities.*

Suppose that we have a training set with n_1 objects from class ω_1 and n_2 objects from class ω_2. We assume that we have obtained random samples from both classes and that the two samples are independent. We assume further that the variable X is normally distributed with expectation μ_i and variance σ_i^2 in class ω_i, $i = 1, 2$, where we assume that expectations are different in the two classes while the variances may either be assumed

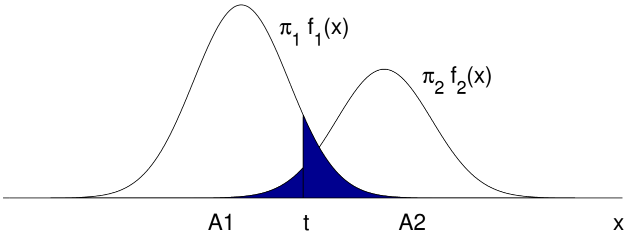


Figure 2.1: Probability of misclassification is given by the coloured area. The set A_1 where class ω_1 is chosen extends here up to the threshold t, while A_2 is chosen above t.

to be equal or unequal. Let the observations be denoted X_{im}, $m = 1, \ldots, n_i$, $i = 1, 2$. Then it is natural to estimate the expectation in class ω_i by

$$\hat{\mu}_i = \frac{1}{n_i} \sum_{m=1}^{n_i} X_{im}, \quad i = 1, 2. \tag{2.3}$$

If we make no assumption on equality of the variances we use the variance estimates

$$s_i^2 = \frac{1}{n_i - 1} \sum_{m=1}^{n_i} (X_{im} - \hat{\mu}_i)^2, \quad i = 1, 2, \tag{2.4}$$

but if we assume variance equality we use the estimate

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \tag{2.5}$$

for the common variance. □ □

We note that compared to Example 2.12 we have in Example 1.10, where we have classified pixels into soil or plant pixels, a similar but more complicated situation as we here do not have training sets for soil and plant pixels but use the model specified by (1.10) and (1.12) for all pixels. Also the proportions of soil and plant pixels are estimated.

2.2 Optimal discrimination with k classes and a d-dimensional feature vector

Suppose now that we have k classes $\omega_i, i = 1, \ldots, k$, and a d-dimensional feature vector X for each object to be classified. Let π_i be the prior probability of class ω_i and let f_i be the probability density of X for an observation from class ω_i, $i = 1, \ldots, k$. Let us further assume that the cost of assigning an object to class ω_i is $c(i|j)$ when the true class is ω_j. Rather than minimizing the misclassification probability we now want to *minimize the expected cost.*

A decision function for our problem is now specified by a partition of d-dimensional space $\mathbb{R}^d$ into k disjoint sets $A_1, \ldots, A_k$ with $\cup_{i=1}^k A_i = \mathbb{R}^d$. If $X \in A_i$ we assign our object to class $\omega_i, i = 1, \ldots, k$.

Now it turns out that the expected cost is minimized if the sets A_i satisfy the following condition

$$x \in A_i \;\Rightarrow\; \text{subscript } i \text{ minimizes } \sum_{j=1}^{k} c(i|j)\pi_j f_j(x). \qquad (2.6)$$

If the sum is minimized by several i-values for a certain x-value, then this x-value may be allocated to A_i for any of these i-values.

To show that a decision rule which satisfies (2.6) is optimal let us consider an arbitrary decision function specified by a partition $A_1, \ldots, A_k$ of $\mathbb{R}^d$. The expected cost for this decision rule may be written

$$\sum_{i=1}^{k} \int_{A_i} \sum_{j=1}^{k} c(i|j)\pi_j f_j(x)dx,$$

from which it follows that a decision rule satisfying the condition (2.6) is optimal.

Let us now assume that all misclassifications have the same cost, and that the cost of a correct decision is zero. Our criterion then implies that we shall *minimize the probability of misclassification*, and it is not difficult to see that we shall prefer class ω_i to class ω_j if

$$\pi_i f_i(x) > \pi_j f_j(x) \qquad (2.7)$$

similar to what we found previously for the case with two classes and one feature variable.

2.3 Normally distributed feature vectors, linear and quadratic discrimination

A d-dimensional random (column) vector X is said to be $\mathrm{N}(\mu,C)$, that is have a d-dimensional normal distribution with expectation vector μ and covariance matrix C, if X has the d-dimensional density function

$$f_X(x) = \frac{1}{(2\pi)^{d/2}(\det C)^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)), \qquad (2.8)$$

where $\det C$ denotes the determinant of the matrix C.

An important special case in discrimination is to assume that the d-dimensional feature vector X has a multivariate normal distribution $\mathrm{N}(\mu_i,C_i)$ in class ω_i, $i = 1, \ldots, k$. Sometimes the covariance matrices are assumed to be equal, that is

$$C_i = C, \quad i = 1, \ldots, k. \qquad (2.9)$$

Let us first assume that the covariance matrices are all equal to C and that we want to minimize the probability of misclassification. A computation from (2.7) and (2.8) shows that if $X = x$ is observed we shall prefer class ω_i to ω_j if

$$(\mu_i - \mu_j)^T C^{-1}(x - \frac{1}{2}(\mu_i + \mu_j)) > \ln \frac{\pi_j}{\pi_i}. \qquad (2.10)$$

We note that (2.10) is linear in x and this case is therefore often called *linear discrimination.*

Let us now find a corresponding rule without the assumption (2.9). It follows from (2.7) and (2.8) that we shall prefer class ω_i to ω_j if

$$\frac{1}{2}x^T(C_j^{-1} - C_i^{-1})x + (\mu_i^T C_i^{-1} - \mu_j^T C_j^{-1})x + \frac{1}{2}(\mu_j^T C_j^{-1}\mu_j - \mu_i^T C_i^{-1}\mu_i) > \ln \frac{\pi_j(\det C_i)^{1/2}}{\pi_i(\det C_j)^{1/2}}. \qquad (2.11)$$

We see that the border between the two regions in d-dimensional space where we should or should not prefer ω_i to ω_j is given by a quadratic surface. When we allow the covariance matrices for the classes to vary we therefore talk about *quadratic discrimination* compared to the linear discrimination referred to above.

Example 2.13. *k-class discriminant analysis with estimated normal densities.*

Suppose that we have a training set with n_i objects from class ω_i, $i = 1, \ldots, k$. From all the classes we assume that we have obtained independent random samples of objects. We assume further that the vector X is normally distributed with expectation vector μ_i and covariance matrix C_i in class ω_i. Let the observations vectors be denoted X_{im}, $m = 1, \ldots, n_i$, $i = 1, \ldots, k$. Then it is natural to estimate the expectation vector in class ω_i by

$$\hat{\mu}_i = \frac{1}{n_i}\sum_{m=1}^{n_i} X_{im}, \quad i = 1, \ldots, k. \qquad (2.12)$$

If we make no assumption on equality of the covariance matrices we use the covariance matrix estimates

$$\hat{C}_i = \frac{1}{n_i - 1}\sum_{m=1}^{n_i}(X_{im} - \hat{\mu}_i)(X_{im} - \hat{\mu}_i)^T, \quad i = 1, \ldots, k. \qquad (2.13)$$

If we assume equality of the covariance matrices we use instead the estimate

$$\hat{C} = \frac{1}{\sum_{i=1}^k (n_i - 1)}\sum_{i=1}^k (n_i - 1)\hat{C}_i \qquad (2.14)$$

for the common covariance matrix C. □ □

2.4 Error rate estimation. Resubstitution and cross-validation

An important issue in discriminant analysis is to estimate the rates of misclassification errors. One simple type of error estimates, often called *resubstitution error-rate estimates*, is obtained by directly computing the observed error rates in the training set for the chosen allocation rule.

However, the resubstition error-rates are typically too optimistic as the objects used to evaluate the error rates are also used in the choice of the discriminator including estimation of parameters in the discriminator. Particularly if the discriminator is complicated, for instance if it contains many parameters, we can grossly underestimate the error-rate corresponding to classification of a new object.

One way of avoiding the bias of resubstitution error rates is to divide the available data into one training set and one evaluation set, for instance, by using half of the data for estimation and half of it for evaluation. One critisism of this procedure is that it may seem wasteful if data are scarce.

Nowadays one often uses resampling methods for evaluation of error rates. One such method is *k-fold cross-validation*. Then we divide the data set consisting of n objects into k equal or approximately equal groups, often by random choice of which objects that should go into group $j, j = 1, \ldots, k$. Then we fix j temporarily and use all objects except those in group j to estimate parameters and compute error average rates for all objects in group j. This procedure is repeated for all groups and we finally average error rates also over groups to get overall error rate estimates. One can show that a small k increases the bias but decreases the variance of the error rate estimate. Originally one often used $k = n$, which is called *leave-one-out cross-validation*. Currently $k = 5$ or $k = 10$ is often recommended.

Example 2.14. *Handwritten digits. Digits 1 and 2*

We use the same data as in Example 1.11 with one small modification consisting of standardization of the two moment features by linear transformations so that they get average zero and varince one. We now use both liner and quadratic discrimination and get, respectively, the linear and elliptic boundaries shown in Figure 2.2. We also computed the resubstitution and 5-fold cross-validation errors for the liner and quadratic discrimination models. It turned out that all four error rate estimates were identical and equal to 15 %. □

Example 2.15. *Handwritten digits. Moment features. Digits* $0, 1, \ldots, 9$.

We use the first 8000 digits in the MNIST database, see Example 1.7, and consider discrimination between the 10 digits $0, 1, \ldots, 9$ by use of all central moment features μ_{pq} in (1.28) with $p + q \leq K$. We computed the resubstitution and the 10-fold cross-validation error estimates for all $K \leq 13$, see Figure 2.3. Note that both for the linear discrimination with full drawn curves and for the quadratic discrimination with dashed curves the resubstitution errors are smaller than the cross-validation errors. For the linear discrimination the cross-validation minimum error is 12.3 % for order 12 and for the quadratic discrimination the cross-validation minimum error is 9.6 % for order 7.

□

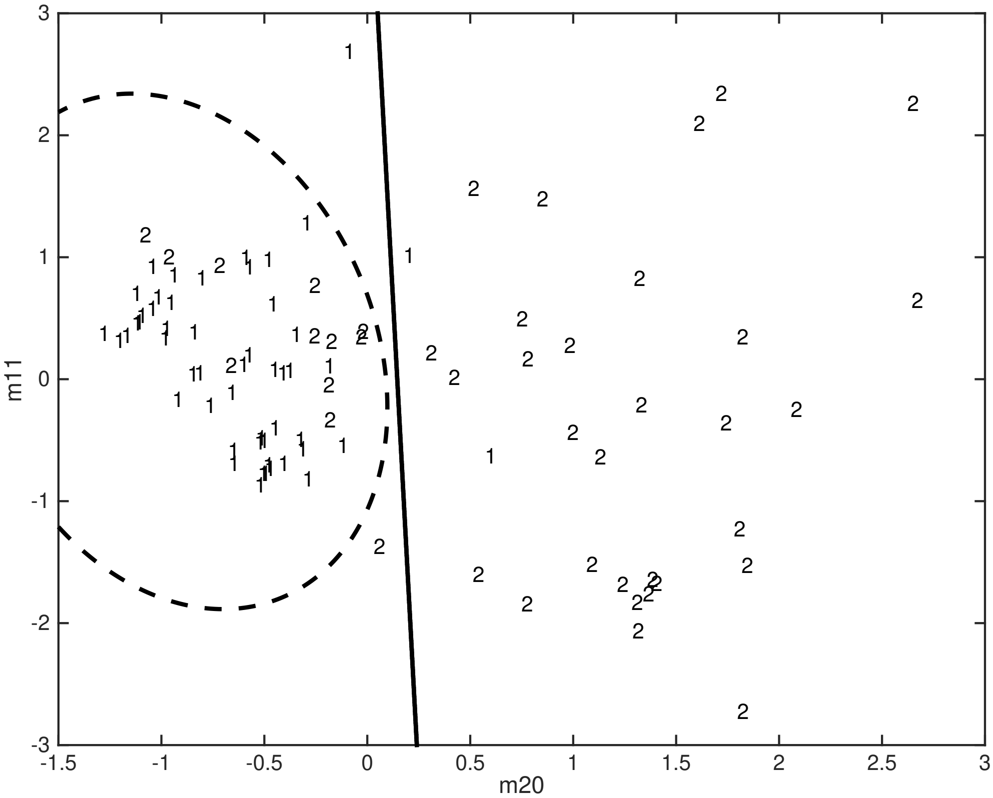


Figure 2.2: Plot of standardized moments μ_{11} versus μ_{20} for handwritten digits 1 and 2 among the first 400 digits in the MNIST data base together with the class boundaries corresponding to linear and quadratic discrimination.

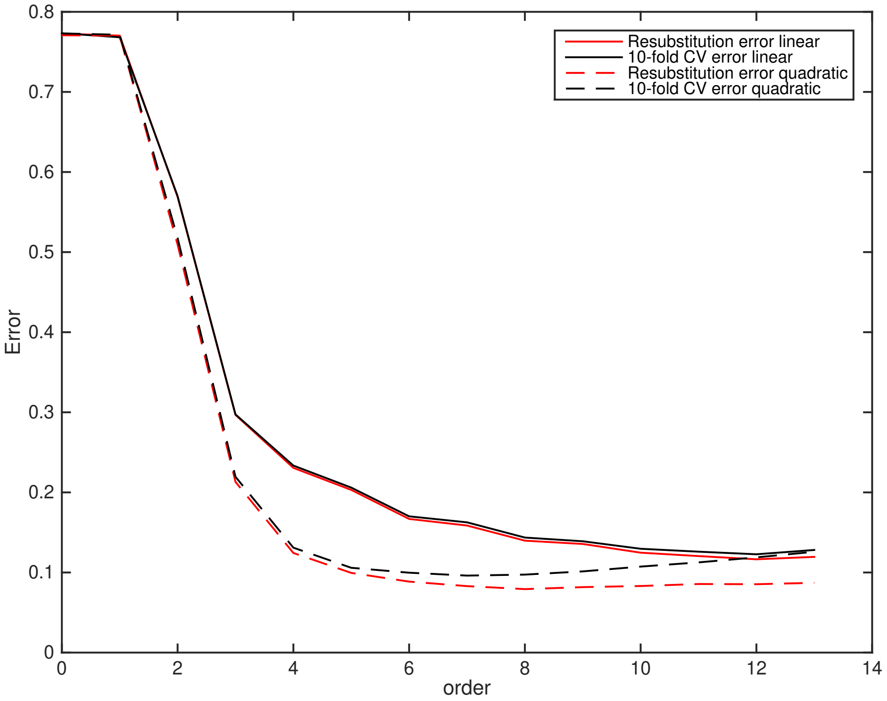


Figure 2.3: Plot of error probabilities for linear discrimination, full drawn curves, and quadratic discrimination, dashed curves, for discrimination between the ten digits $0, 1, \ldots, 9$. Resubstitution error curves are in red and cross-validation error curves are in black. Order K on the horizontal axis means that all moments μ_{pq} with $p+q \leq K$ are used as features to discriminate between the digits.

2.5 Nearest neighbour classifaction

Suppose that we have a distance function $\delta(x, x')$ between feature vectors x and x'. Examples of distance functions for d-dimensional feature vectors are the Euclidean distance

$$\delta(x, x') = (\sum_{i=1}^{d} (x_i - x'_i)^2)^{1/2} \tag{2.15}$$

and $\delta = 1 - r$, where r are is the correlation

$$r(x, x') = \frac{\sum_{i=1}^{d} (x_i - \bar{x})(x'_i - \bar{x}')}{(\sum_{i=1}^{d} (x_i - \bar{x})^2)^{1/2} \; (\sum_{i=1}^{d} (x'_i - \bar{x}')^2)^{1/2}} \tag{2.16}$$

where $\bar{x}$ and $\bar{x}'$ are the arithmetic means of the vectors x and x'.

A useful discrimination method is the *m-nearest neighbour* rule, which proceeds as follows. Suppose we have a training set for which we know the correct classification. For a new observation we find the m nearest neighbours in the training set, and we classify the new observation by majority voting among these nearest neighbours.

Example 2.16. *Handwritten digits. Nearest neighbour discrimination*

We use the same data as in Example 2.14. The m-nearest neighbour classications with m=3 and 5 are shown in Figure 2.4. We also computed the resubstitution and 5-fold cross-validation errors for m-nearest neighbour methods with m ranging from 1 to 10. the result is shown in Figure 2.5. The minimum cross-validated error is obtained for $m = 5$ and equals 12 %. □

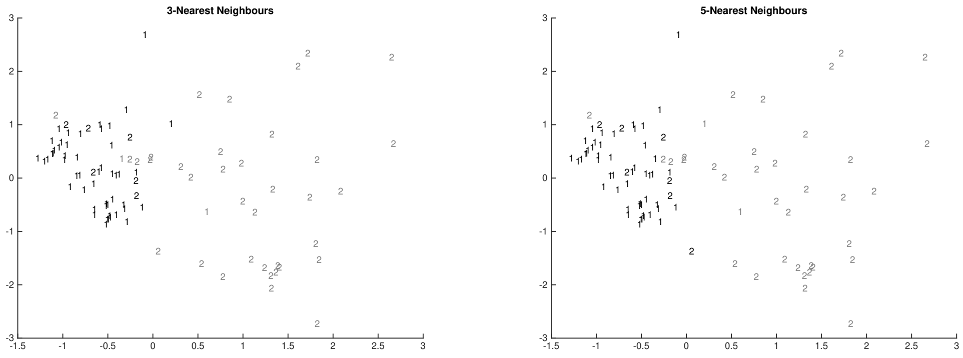


Figure 2.4: Plot of standardized moments μ_{11} versus μ_{20} for handwritten digits digits 1 and 2 among the first 400 digits in the MNIST data base together classifications from m-nearest neighbour classification for $m = 3$ and $m = 5$. Digit colours indicate classification: black digits are classified as 1 and grey digits are classified as 2.

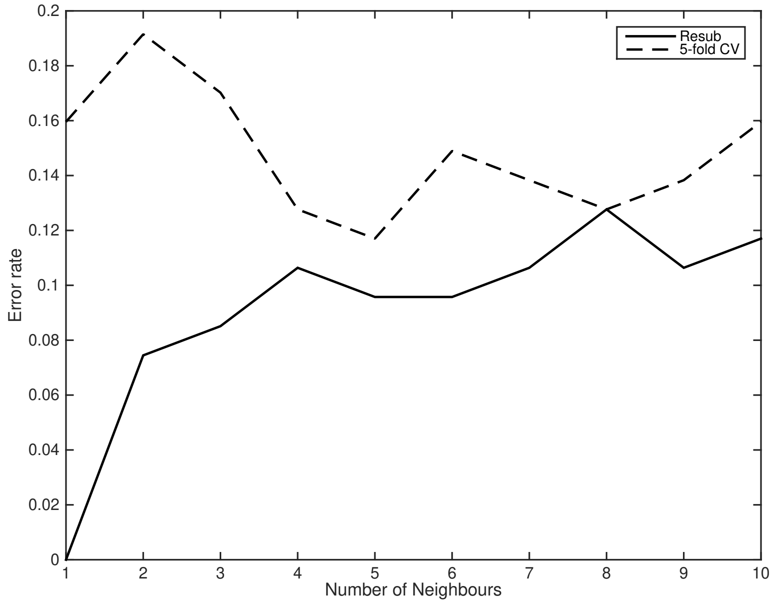


Figure 2.5: Plot of resubstitution and 5-fold cross-validation error estimates for m-nearest neighbour classications for $m = 1, \dots, 10$.

2.6 Multinomial logistic regression

Logistic regression with two classes is briefly described in Section 14.9. Here we will generalize to k classes $\omega_1, \dots, \omega_k$. Let Y denote the class number of an observation with associated explaining vector x, which we here will suppose consists of an image. Assume that

$$\Pr(Y = i) = \frac{\mathrm{e}^{\beta_i \cdot x}}{1 + \sum_{j=1}^{k-1} \mathrm{e}^{\beta_j \cdot x}}, \quad i = 1, \dots, k-1, \tag{2.17}$$

and

$$\Pr(Y = k) = \frac{1}{1 + \sum_{j=1}^{k-1} \mathrm{e}^{\beta_j \cdot x}}, \quad i = k, \tag{2.18}$$

where β_i denotes a parameter vector of the same dimension as x and $\beta_i \cdot x$ denotes the scalar product of β_i and x, obtained by multiplying componentwise the elements of β_i and x, and adding the corresponding products. For given data with observations of pairs (x, Y) we can then estimate the parameter vectors β_i by maximum likelihood.

Example 2.17. *Handwritten digits. Logistic regression, confusion matrix, display by t-SNE.*

The computations and figures in this example are taken from (Longfils, 2018). In Figure 2.6 we see parameter vectors β_i estimated from a multinomial logistic model by use of 10000 digits from MNIST. In this figure we can rather clearly identify the digit zero to the left in the upper row, and perhaps also the digit one next to it. A convenient way

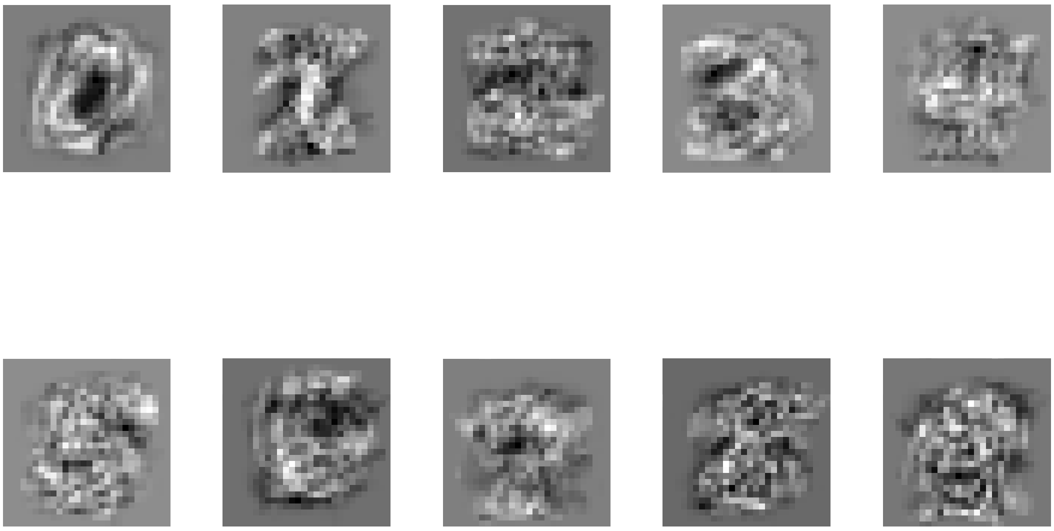


Figure 2.6: Parameter vectors β_i for digits $0, \ldots, 4$ in the upper row and digits $5, \ldots, 9$ in the lower row estimated from 10000 digits in the MNIST database.

of illustrating the results of a discrimination analysis is to compute a *confusion matrix* giving the resulting classifications for each class in the data used. In Table 2.1 we see the confusion matrix corresponding to the logistic model analysis in Example 2.17 with estimated error and identification probabilities. From the confusion matrix we see that the digit zero seems to be most easy to identify with an estimated identification probability of 97.6%. The overall estimated identification probability is $(1108 + 922 + \ldots 948)/10000 = 92.2\%$.

In Figure 2.7 we use the method *t-SNE*, compare Section 14.6 and (Longfils, 2018), to visualize how the 28×28-dimensional x-vector may be used to discriminate between hand-written digits.

□

2.7 Selection of features

If we have a large number of possible features it is useful to make a selection of features. One often used method is *forward selection* where we start by choosing the single feature which gives the smallest error rate. Then we add that feature of the remaining ones which together with the first chosen feature gives the best performance. The procedure is continued a suitable number of steps. If one uses cross-validation error rate estimates, we typically find that the error rates first decrease when we add new variables but then a minimum is obtained and after that the error rate increases due to overfitting.

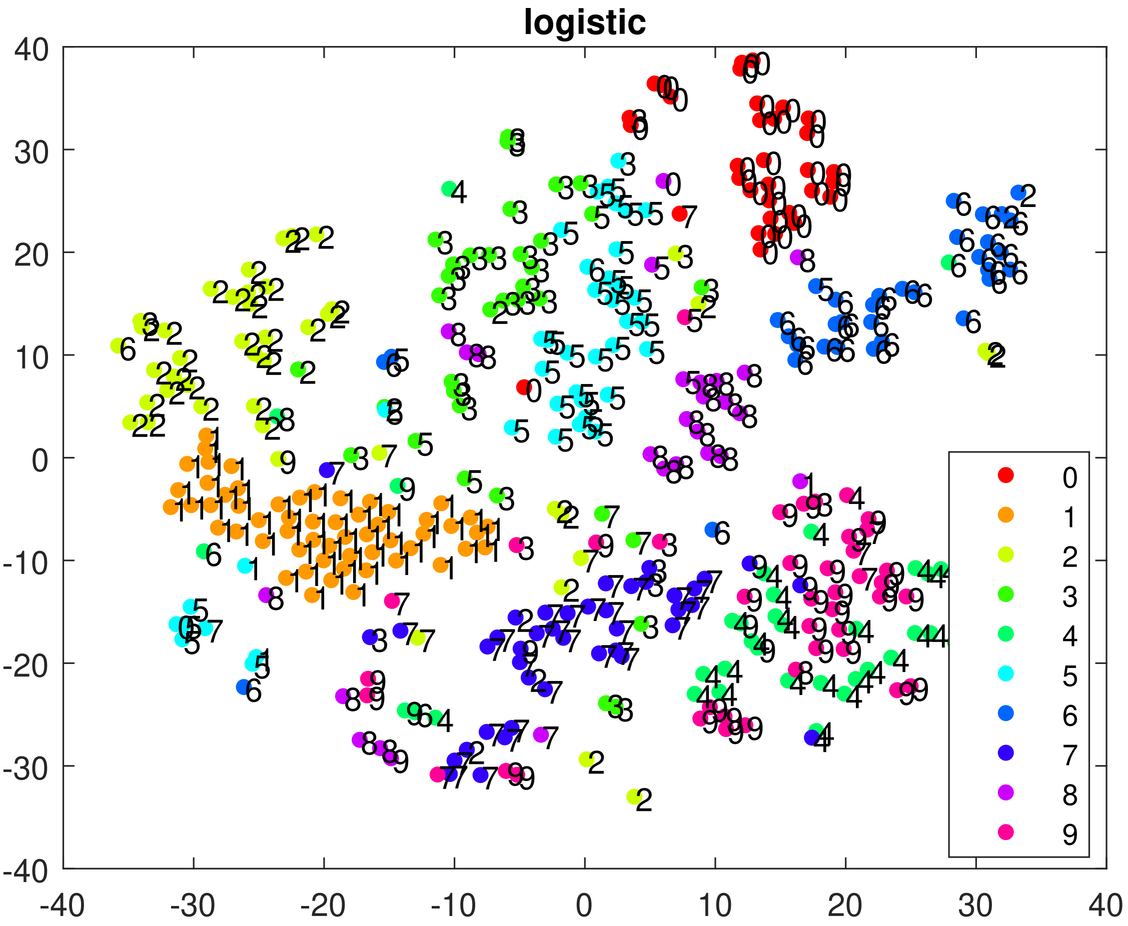


Figure 2.7: Visualization by use of t-SNE for the first 400 digits in the test set used in Example 2.17. The numbers close to points are the labels predicted by the logistic regression method, and the colours of points correspond to the true labels as given in the box in the lower right part of the image.

True		Estimated class											
class		0	1	2	3	4	5	6	7	8	9	Sum	Percent
0	Number	1108	8	2	0	2	3	1	11	0	0	1135	11.4
	Percent	97.6	0.7	0.2	0.0	0.2	0.3	0.1	1.0	0.0	0.0	100	
1	Number	9	922	19	11	4	12	11	32	4	8	1032	10.3
	Percent	0.9	89.3	1.9	1.0	0.4	1.2	1.1	3.2	0.4	0.8	100	
2	Number	2	18	921	2	22	3	10	21	7	4	1010	10.1
	Percent	0.2	1.8	91.2	0.2	2.2	0.3	1.0	2.1	0.7	0.4	100	
3	Number	4	6	4	918	1	9	5	6	27	2	982	9.8
	Percent	0.4	0.6	0.4	93.5	0.1	0.9	0.5	0.6	2.7	0.2	100	
4	Number	5	2	35	9	775	14	6	32	4	10	892	8.9
	Percent	0.6	0.2	3.9	1.0	86.9	1.6	0.7	3.6	0.4	1.1	100	
5	Number	3	8	2	6	17	907	1	2	1	11	958	9.6
	Percent	0.3	0.8	0.2	0.6	1.8	94.7	0.1	0.2	0.1	1.1	100	
6	Number	9	22	8	5	1	0	946	4	31	2	1028	10.3
	Percent	0.9	2.1	0.8	0.5	0.1	0.0	92.0	0.4	3.0	0.2	100	
7	Number	12	7	23	9	24	10	11	857	14	7	974	9.7
	Percent	1.2	0.7	2.4	0.9	2.5	1.0	1.1	88.0	1.4	0.7	100	
8	Number	6	2	9	23	8	0	22	10	922	7	1009	10.1
	Percent	0.6	0.2	0.9	2.3	0.8	0.0	2.2	1.0	91.4	0.7	100	
9	Number	0	2	4	1	13	5	3	3	1	948	980	9.8
	Percent	0.0	0.2	0.4	0.1	1.3	0.5	0.3	0.3	0.1	96.7	100	
Sum	Number	1158	997	1027	984	867	963	1007	978	1011	999	10000	100
	Percent	11.6	10.0	10.3	9.8	8.7	9.6	10.1	9.8	10.1	10.0	100	

Table 2.1: Confusion matrix for the logistic model analysis of the MNIST data set in Example 2.17.

In *backward selection* we start by including all features. Then we eliminate one feature so that the resulting error rate is as small as possible. The procedure is iterated a suitable number of steps.

2.8 Cluster analysis, k-means clustering

Suppose that we have collected a number of colonies of bacteria of a type that has not been studied before but which we want to order in classes corresponding species or subspecies. That is, we want to construct a taxonomy for these bacteria. Instead of an individual bacterial particle the natural unit here is a homogeneous colony of bacteria.

One possible procedure would be to measure a number of variables, say d for each individual colony and to see if these variables tend produce clusters in d-space. Let X denote the d-dimensional vector of measurements, and let $f(x)$ denotes the corresponding probability density (or probability function if X is discrete). Corresponding to k classes we would then expect that f could be written as a mixture,

$$f(x) = \sum_{i=1}^{k} p_i f_i(x), \tag{2.19}$$

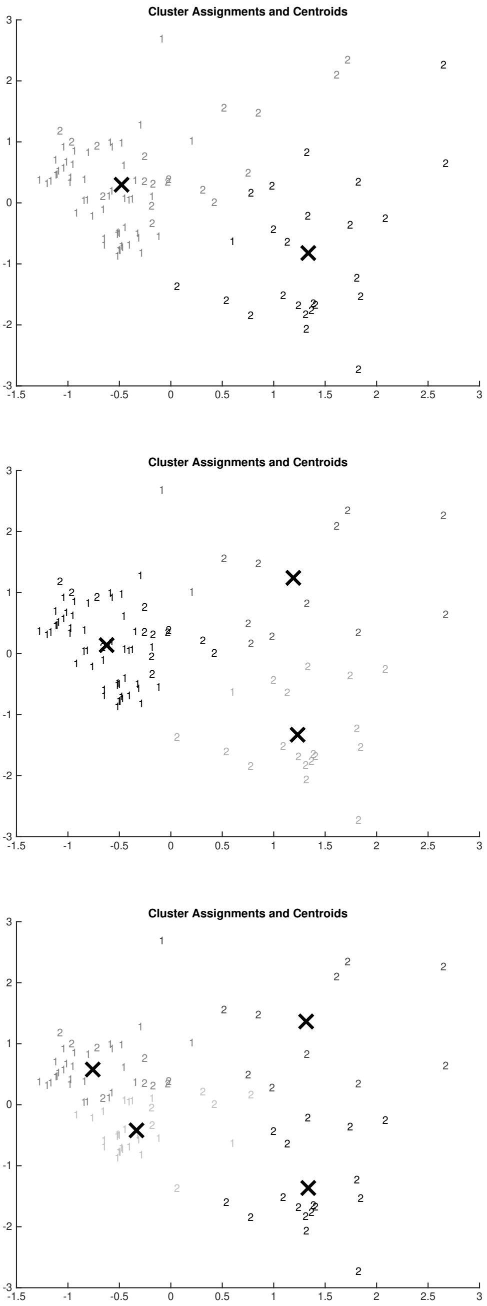
where f_i denotes the probability density in the ith class, and p_i the proportion of the ith class.

Let n denote the number of colonies observed, and let X_j, $j = 1, \dots, n$, denote our observed d-dimensional vectors. The basic problem in cluster analysis can then be formulated as estimation of the number k of classes and also the functions f_i, $i = 1, \dots, k$, on the basis of our observations $X_1, \dots, X_n$. Note that this problem is much more complicated than the problems previously discussed in this chapter as we neither know the number of classes, nor which observations (in a test set) that belong to the different classes.

One procedure that is often used is k-means clustering. Consider d-dimensional observations and let us for simplicity regard Euclidean distances between observations. We assume that there are k classes and choose first randomly k cluster centers among the observations X_j, $j = 1, \dots, n$. Then we alternate between two types of steps. In the *observation allocation step* we suppose that we have cluster centers $C_i, i = 1, \dots, k$, and allocate each observation to the closest cluster center. In the *cluster center recomputation step* we compute new cluster centers as averages of all observations allocated to each cluster. We alternate between the two types of steps until there are no changes. Typically we will also repeat the procedure a number of times with different (randomly chosen) starting cluster centres and finally choose the clustering which has the minimal total sum of within cluster square distances to cluster centres.

Example 2.18. *Handwritten digits. Cluster analysis*

We use the same data as in Example 2.14 but now we cluster them by k-means clustering with $k = 2$, 3 and 4. The results are shown in Figure 2.8. □



Cluster Assignments and Centroids

Cluster Assignments and Centroids

Cluster Assignments and Centroids

Figure 2.8: Results from k-means clustering with $k = 2$, 3 and 4 of the same data as used in Example 2.14. Crosses mark estimated cluster centers.

2.9 Case studies

Weed seed identification

In (Petersen, 1992) weed seed identification was studied with 25 images of seeds for each of 40 species.

A large number of possible features were investigated and with 25 features an optimal cross-validation error rate of 2.3% was found.

Weed plant identification

(Andersson, 1998) studied identification of plants at an early stage of carrot and seven weed species. With 27 images for each of the eight plant species a cross-validation error rate of about 16% was found with 7 or 8 features.

Comparison of discrimination methods for microarray data

In (Dudoit *et al.*, 2002) different discrimation methods are compared for classification of tumors based on gene expression data from three datasets available on the Internet. In particular, the nearest neighbour method is found to perform well in these examples. The number of neighbours is here determined by cross-validation.

2.10 Exercises

Images and data sets for the exercises below may be found from the course home pages.

Exercise 2.1. Fisher's Iris data, a classical data set. One of the famous data sets in statistics is Fisher's Iris data, used in (Fisher, 1936), where discriminant analysis was introduced. Consider the data in Table 2.2 with four variables measured for 50 plants of each of three *Iris* species. The data were assembled by E. Anderson, see (Anderson, 1935), and analysed in detail by (Fisher, 1936).
(a). Draw scatter plots for all 150 observations and all six pairs of variables. Alternatively, if you do not have access to a computer, draw scatter plots for subsets with, say, 5 plants from each species, and for, say, two pairs of variables.
(b). Find the best linear discriminators using all four variables for discrimination between all pairs of the three species. Alternatively, without a computer, describe with formulas how the computations are made. Under what assumptions is this discrimination method optimal.
(c). Find the best quadratic discriminators using all four variables for discrimination between all pairs of the three species. Alternatively, without a computer, describe with formulas how the computations are made. Under what assumptions is this discrimination method optimal.
(d). Find the optimal combination of two variables for discriminating between the three species. Alternatively, without a computer, describe with formulas how the computations are made.

Exercise 2.2. Weed seeds. Consider the weed seed images of *Rumex crispus* and *Rumex thyrsiflorus* from Figures 1.5 and 1.6 in Example 1.2 or a subset of these 25 plus 25 images.
(a). Compute the areas of the seeds and the convexity of them for the images considered.
(b). How well can you discriminate between the two species by use of the feature convexity and linear discrimination?
(c). How well can you discriminate between the two species by use of the feature convexity and quadratic discrimination?
(d). How well can you discriminate between the two species by use of the features convexity and area and linear discrimination?
(e). How well can you discriminate between the two species by use of the features convexity and area and quadratic discrimination?

Exercise 2.3. Weed plants. Consider images of carrot and weed plants such as those described in Example 1.3. Choose two or more species and see well you can discriminate between them by suitably chosen featuers. Compare with the results found by Andersson (1998).

Exercise 2.4. Handwritten digits. Resubstitution error. Consider the data in Example 2.14. Show by use of Figure 2.2 that the resubstitution error is equal to 14/93 both for linear and quadratic discrimination.

2.11 Literature on pattern recognition

A good introductory text on statistical pattern recognition is (Fukunaga, 1990). Many algorithms are described in (Ripley, 1996) which also contains an extensive list of references for the period up to 1996. A highly useful review of clustering methods with particular emphasis on applications with image data is given in (Jain *et al.*, 1999).

Table 2.2: Four flower features (in cm) for 50 plants of three *Iris* species, from (Fisher, 1936).

Iris setosa				*Iris versicolor*				*Iris virginica*			
Sepal	Sepal	Petal	Petal	Sepal	Sepal	Petal	Petal	Sepal	Sepal	Petal	Petal
length	width	length	width	length	width	length	width	length	width	length	width
5.1	3.5	1.4	0.2	7	3.2	4.7	1.4	6.3	3.3	6	2.5
4.9	3	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4	1.3	6.3	2.9	5.6	1.8
5	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5	3.4	1.5	0.2	4.9	2.4	3.3	1	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5	2	3.5	1	6.5	3.2	5.1	2
4.8	3.4	1.6	0.2	5.9	3	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3	1.4	0.1	6	2.2	4	1	6.8	3	5.5	2.1
4.3	3	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5	2
5.8	4	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3	4.5	1.5	6.5	3	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6	2.2	5	1.5
5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
5.1	3.7	1.5	0.4	6.1	2.8	4	1.3	5.6	2.8	4.9	2
4.6	3.6	1	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2
5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
5	3	1.6	0.2	6.6	3	4.4	1.4	7.2	3.2	6	1.8
5	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
5.2	3.5	1.5	0.2	6.7	3	5	1.7	6.1	3	4.9	1.8
5.2	3.4	1.4	0.2	6	2.9	4.5	1.5	6.4	2.8	5.6	2.1
4.7	3.2	1.6	0.2	5.7	2.6	3.5	1	7.2	3	5.8	1.6
4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
5.4	3.4	1.5	0.4	5.5	2.4	3.7	1	7.9	3.8	6.4	2
5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
5.5	4.2	1.4	0.2	6	2.7	5.1	1.6	6.3	2.8	5.1	1.5
4.9	3.1	1.5	0.1	5.4	3	4.5	1.5	6.1	2.6	5.6	1.4
5	3.2	1.2	0.2	6	3.4	4.5	1.6	7.7	3	6.1	2.3
5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
4.9	3.1	1.5	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
4.4	3	1.3	0.2	5.6	3	4.1	1.3	6	3	4.8	1.8
5.1	3.4	1.5	0.2	5.5	2.5	4	1.3	6.9	3.1	5.4	2.1
5	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
4.5	2.3	1.3	0.3	6.1	3	4.6	1.4	6.9	3.1	5.1	2.3
4.4	3.2	1.3	0.2	5.8	2.6	4	1.2	5.8	2.7	5.1	1.9
5	3.5	1.6	0.6	5	2.3	3.3	1	6.8	3.2	5.9	2.3
5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
4.8	3	1.4	0.3	5.7	3	4.2	1.2	6.7	3	5.2	2.3
5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5	1.9
4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3	5.2	2
5.3	3.7	1.5	0.2	5.1	2.5	3	1.1	6.2	3.4	5.4	2.3
5	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3	5.1	1.8

Chapter 3

Machine learning, neural nets, support vector machines

In recent decades a number of machine learning methods for patter recognition have been launched such as neural nets and support vector machines which will be briefly discussed in this chapter. To evaluate these methods a number of large datasets have also been brought forth, compare Table 3.1 and
`https://en.wikipedia.org/wiki/List_of_datasets_for_machine_learning_research`
for more details.

Table 3.1: Datasets of images and videos for tasks such as classification, object detection and face recognition

Dataset name	Brief description	Instances	Format	Default task	Created
MNIST	Handwritten digits	60 000 + 10 000	Images, text	Classifcation	1998
CIFAR-10	Images of 10 classes of objects	60 000	Images	Classification	2009
CIFAR-100	Images of 100 classes of objects	60 000	Images	Classification	2009
KITTI	Images and videos obtained from cars	>100GB of data	Images, text	Classification, object detection	2012
SVHN	Street View House Numbers	73 257 + 26 032	Images	Classification	2011
FERET	Face Recognition Technology	11 338 from 1 199 individuals	Images	Classification, face recognition	2003

3.1 Neural nets

Let us start by considering a neural net consisting of one input layer with n_1 units corresponding to input variables $x_i, i = 1, \ldots, n_1$, an intermediate (hidden) layer with n_2

units and an output layer with K units. For unit j in the intermediate layer we compute the so-called activation value $a_j, j = 1, \dots, n_2$, by

$$z_j = \sum_{i=1}^{n_1} w_{ji}^{(1)} x_i + b_j^{(1)}, \tag{3.1}$$

$$a_j = \frac{e^{z_j}}{\sum_{j'=1}^{n_2} e^{z_{j'}}}, \tag{3.2}$$

for weights $w_{ji}^{(1)}$ and biases $b_j^{(1)}$. With some abuse of notation we will write

$$a_j = \sigma(z_j), \quad j = 1, \dots, n_2, \tag{3.3}$$

and we call σ given by (3.2) and (3.3) the *softmax* function. From the hidden layer we proceed to the output in a similar way and we obtain neural net output variables $f_k(k), k = 1, \dots, K$, as

$$f_k(x) = f_k(x, \theta) = \sigma\left(\sum_{j=1}^{n_2} w_{kj}^{(2)} \sigma\left(\sum_{i=1}^{n_1} w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b_k^{(2)}\right), \quad k = 1, \dots, K, \tag{3.4}$$

where $x = (x_1, \dots, x_{n_1})$ is the vector of input variables, and θ is the parameter vector of all weights, $w_{ji}^{(1)}$ and $w_{kj}^{(2)}$, and biases $b_j^{(1)}$ and $b_k^{(2)}$.

We can add now add one more hidden layer which gives a neural net with two hidden layers and output

$$f_k(x) = \sigma\left(\sum_{\ell=1}^{n_3} w_{k\ell}^{(3)} \sigma\left(\sum_{j=1}^{n_2} w_{\ell j}^{(2)} \sigma\left(\sum_{i=1}^{n_1} w_{ji}^{(1)} x_i + b_j^{(1)}\right) + b_\ell^{(2)}\right) + b_k^{(3)}\right), \quad k = 1, \dots, K, \tag{3.5}$$

and it should be clear how we can extend the neural net with an arbitrary number of hidden layers.

If we for instance consider a neural net for the MNIST database it is natural to consider $n_1 = 28^2 = 784$ units in the input layer, each input unit corresponding to one pixel value, and $K = 10$ corresponding to the 10 possible digits. We note that the output variables $f_k(x)$ sum to one and we can interpret $f_k(x, \theta)$ as the probability of digit k. To classify images we can first in some way estimate the parameter θ by use of a training set. Let $\hat{\theta}$ denote the estimate of θ. To classify an image x we can then put

$$\hat{k}(x) = \text{argmax}_k f_k(x, \hat{\theta}). \tag{3.6}$$

The crucial step here is to obtain the estimate $\hat{\theta}$. In practice the parameter vector θ may contain several thousand components and the estimation procedure is thus quite delicate. We will now discuss possible estimation methods.

Parameter estimation for neural nets, regularization

Suppose that we have a training set $\mathcal{T}$ of $|\mathcal{T}|$ pairs (x, y) and that the neural net output $f(x, \theta)$ should approximate y. Then we introduce a suitable loss function. Let us first consider a simple case where y and $f(x,\theta)$ are real-valued. Then we may choose the loss function

$$L(\theta,\mathcal{T}) = \frac{1}{|\mathcal{T}|}\sum_{(x,y)\in\mathcal{T}} (y - f(x,\theta))^2. \tag{3.7}$$

Let us then consider a classification setting with K classes, for instance for MNIST classification with $K = 10$. As described above we then get as output from a neural net a probability distribution $f_k(x,\theta), k = 1,\dots,K$, for the possible class values. For a pair (x,y) where k_c is the correct class we can define $y_k, k = 1,\dots,K$, as

$$y_k = \begin{cases} 1 & \text{if } k = k_c \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

and choose the cross-entropy loss function

$$L(\theta,\mathcal{T}) = -\frac{1}{|\mathcal{T}|}\sum_{(x,y)\in\mathcal{T}}\sum_k y_k \log f_k(x,\theta). \tag{3.9}$$

We can minimize $L(\theta,\mathcal{T})$ and obtain an estimate $\hat{\theta} = \hat{\theta}(\mathcal{T})$. The result is then that we often get a good fit to the observations in $\mathcal{T}$, but if we go to a new data set the fit is typically not so good. We say then that we get an overfit. To compensate for overfitting we can introduce a regularization term $R(\theta)$, for instance

$$R(\theta) = \sum_{i=1}^{|\theta|} |\theta_i|^2, \tag{3.10}$$

where we sum over all components of $\theta = (\theta_1,\dots,\theta_{|\theta|})$. Then we estimate θ by minimizing the regularized loss function

$$\mathcal{L}(\theta;\mathcal{T},L,\lambda,R) = L(\theta,\mathcal{T}) + \lambda R(\theta), \tag{3.11}$$

where $\lambda \geq 0$ is a tuning parameter. Note that $\lambda = 0$ corresponds to no regularization which typically gives overfitting, while a very large λ corresponds to underfitting. To choose a proper value of the tuning parameter we can evaluate the regularized loss function for a separate validation set $\mathcal{T}'$ of pairs (x,y) or use cross-validation.

Let us also note that instead of the softmax function given by $a = \sigma(z)$, see (3.2) and (3.3), as activation function, one often uses a *rectified linear unit* given by

$$a = \max(0,z). \tag{3.12}$$

Convolutional neural nets

Let $w = (w_{k\ell})$ and $g = (g_{ij})$ be matrices. The convolution $w * g$ is then defined by

$$(w * g)_{ij} = \sum_k \sum_\ell w_{k\ell}\, g_{i-k,j-\ell}, \tag{3.13}$$

compare Section 1.2 on image filtering.

Convolutional neural nets are particularly useful for analysis of images. Such neural nets contain layers with layer transitions of the following convolution type

$$a_{ij}^{(r+1)} = \sigma\left(\sum_{k=-p}^{p}\sum_{\ell=-p}^{p} w_{k\ell}^{(r)}\, a_{i-k,j-\ell}^{(r)}\right), \tag{3.14}$$

where p usually is a small positive number. We note that we use here only $(2p+1)^2$ different weights and that there is the same filter operation applied in different parts of $a^{(r)}$ here regarded as an image. The filter operation could for instance consist of finding edges in an image.

A convolution layer is often followed by a pooling layer reducing the layer size. We can for instance use a maxpool operation where a layer of pixels is divided into adjacent and non-overlapping rectangles and each rectangle is replaced in the following layer by one pixel with pixel value equal to the maximal pixel value in the rectangle.

Example 3.19. *Handwritten digits. Analysis with a convolutional neural net.* The computations in this example are taken from (Longfils, 2018). In Table 3.2 we see the confusion matrix corresponding to a convolutional neural net trained on 50 000 digits and evaluated on 10 000 digits from the MNIST data set. The neural network used consisted of six layers:

1. An input layer (28×28 pixel images)
2. A convolution layer with 20 filters of size 5×5
3. A rectified linear unit layer
4. A max pooling layer with size 2×2 pixels
5. A fully connected layer
6. A softmax layer (outputting the probability for each of the 10 classes)

From the second item in the list above we see that the convolution layer actually exists of 20 different filters (working in parallel). The resulting 20 filters are given in Figure 3.1. It is easy to visualize that combinations of convolutions with these filters can be advantageous in identifying digits.

From the confusion matrix we see that the digit zero seems to be most easy to identify with an estimated identification probability of 99.6%. The overall estimated identification probability is $(1130 + 1016 + \ldots 975)/10000 = 98.5\%$. □

3.2 Support vector machines

The following description is inspired by the more complete description in Chapter 19 of (Efron & Hastie, 2016). Suppose that we have a training set $\mathcal{T}$ consisting of pairs (x, y), where x is an n-dimensional column vector and $y \in \{-1, +1\}$ is a two-class indicator. To

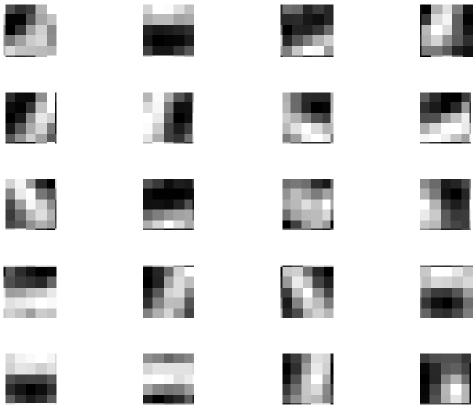


Figure 3.1: The 20 filters in the convolutional neural net used for identifying MNIST integers.

True		Estimated class											
class		0	1	2	3	4	5	6	7	8	9	Sum	Percent
0	Number	1130	1	1	1	0	1	0	1	0	0	1135	11.4
	Percent	99.6	0.1	0.1	0.1	0.0	0.1	0.0	0.1	0.0	0.0	100	
1	Number	1	1016	2	1	0	1	5	4	1	1	1032	10.3
	Percent	0.1	98.4	0.2	0.1	0.0	0.1	0.5	0.4	0.1	0.1	100	
2	Number	0	3	999	0	1	0	2	4	1	0	1010	10.1
	Percent	0.0	0.3	98.9	0.0	0.1	0.0	0.2	0.4	0.1	0.0	100	
3	Number	0	1	0	974	0	1	0	0	5	1	982	9.8
	Percent	0.0	0.1	0.0	99.2	0.0	0.1	0.0	0.0	0.5	0.1	100	
4	Number	0	1	9	0	874	3	0	3	0	2	892	8.9
	Percent	0.0	0.1	1.0	0.0	98.0	0.3	0.0	0.3	0.0	0.2	100	
5	Number	3	0	0	3	1	942	0	2	0	7	958	9.6
	Percent	0.3	0.0	0.0	0.3	0.1	98.3	0.0	0.2	0.0	0.7	100	
6	Number	3	8	2	0	0	0	1012	2	1	0	1028	10.3
	Percent	0.3	0.8	0.2	0.0	0.0	0.0	98.4	0.2	0.1	0.0	100	
7	Number	0	3	2	1	2	0	5	953	2	6	974	9.7
	Percent	0.0	0.3	0.2	0.1	0.2	0.0	0.5	97.8	0.2	0.6	100	
8	Number	2	0	3	10	3	0	8	2	977	4	1009	10.1
	Percent	0.2	0.0	0.3	1.0	0.3	0.0	0.8	0.2	96.8	0.4	100	
9	Number	0	0	0	0	1	1	2	1	0	975	980	9.8
	Percent	0.0	0.0	0.0	0.0	0.1	0.1	0.2	0.1	0.0	99.5	100	
Sum	Number	1139	997	1027	984	867	963	1007	978	1011	999	10000	100
	Percent	11.4	10.0	10.3	9.8	8.7	9.6	10.1	9.8	10.1	10.0	100	

Table 3.2: Confusion matrix for a convolutional neural net analysis of the MNIST data set in Example 2.17.

begin with we will suppose that the two classes are linearly separable in the sense that there exists a real parameter β_0 and an n-dimensional parameter vector β such that with $f(x) = \beta_0 + x^T\beta$

$$yf(x) > 0 \quad \text{for all} \quad (x, y) \in \mathcal{T}. \tag{3.15}$$

We can then classify a new x-vector and predict the corresponding y-value as $\text{sign}(f(x))$. A natural question is then if we can choose β_0 and β in an optimal way. The suggested solution here is to maximize the minimal distance (margin) to the separating hyperplane $f(x) = 0$ in n-space. The solution to this problem turns out to be to find

$$\max_{\beta_0,\beta} \left\{ M : \text{subject to } \frac{1}{||\beta||} y(\beta_0 + x^T\beta) \geq M \text{ for all } (x, y) \in \mathcal{T} \right\} , \tag{3.16}$$

where $||\beta||$ is the Euclidean (quadratic) norm in n-space. An equivalent somewhat simpler formulation is to find

$$\min_{\beta_0,\beta} \left\{ ||\beta|| : \text{subject to } y(\beta_0 + x^T\beta) \geq 1 \text{ for all } (x, y) \in \mathcal{T} \right\} . \tag{3.17}$$

In general we can not expect to find a hyperplane giving complete separation between the two classes. Then we can instead find a minimum with a regularized loss function

$$\min_{\beta_0,\beta} \left\{ \sum_{(x,y)\in\mathcal{T}} [1 - y(\beta_0 + x^T\beta)]_+ + \lambda||\beta||^2 \right\} , \tag{3.18}$$

where $[a]_+$ denotes the positive part of a real number a. For linearly separable classes one can show that $\lambda = 0$ gives the previously described solution which is determined by a few points close to the separating boundary. Increasing λ corresponds to taking account of more and more data points. Similarly as for neural nets one can find an optimal tuning parameter λ by use of a separate validation set or by cross-validation.

For a multiclass classification problem we can for instance for each class make a two-class classification versus the union of all other classes and then for a new observed x-vector choose the class giving the largest margin. Another possibility is to consider voting for all pairwise comparisons and for a new observation to choose the class that gets that the maximal number of votes.

Support vector machines with kernel functions

One can show that for a new vector x to be classified one can write the classifier on the form

$$f(x) = \beta_0 + x^T\beta = \beta_0 + \sum_{i=1}^{|\mathcal{T}|} \alpha^i x^T x^i, \tag{3.19}$$

where $x^1, \ldots, x^{|\mathcal{T}|}$ are x-vectors in a training set $\mathcal{T}$ and $\alpha^1, \ldots, \alpha^{|\mathcal{T}|}$ are real parameters. This representation allows us to use a modified classifier of the form

$$f(x) = \beta_0 + x^T\beta = \beta_0 + \sum_{i=1}^{|\mathcal{T}|} \alpha^i k(x, x^i), \tag{3.20}$$

where $k(u,v)$ is a positive-definite kernel function, for instance the Gaussian kernel

$$k(u,v) = \mathrm{e}^{-||u-v[[^2}. \tag{3.21}$$

Use of kernel functions implies possibilities of nonlinear transformations of the x-vectors and adds considerable flexibility to support vector machines.

For an application of support vector machine methods for identifying MNIST integers, see (Longfils, 2018). Figure 5 there gives a confusion matrix similar to the confusion matrices in Tables 2.1 and 3.2 in the present text.

3.3 Literature on machine learning, neural nets and support vector machines

Highly useful texts from a statistical point of view are (Efron & Hastie, 2016) and (Bishop, 2006). Two recent references, both with the title 'Deep Learning' which is a current term for advanced neural nets, are (LeCun *et al.*, 2015) giving an overview and (Goodfellow *et al.*, 2016) giving a thorough and up-to-date coverage of the field.

3.4 Exercises

Exercise 3.1. Draw a figure illustrating the neural net in (3.5). Choose for instance $n_1 = 6$, $n_2 = 5$, $n_3 = 4$ and $K = 2$.

Exercise 3.2. Think of guessing wrong for handwritten integers, that is guessing integer y when x is correct, $x = 0, \ldots, 9$, $y = 0, \ldots, 9$, $y \neq x$. Which of the possible 90 errors do you think is the largest one? Look in Table 3.2 and compare with the largest error probability there.

Chapter 4

Statistical image modelling

In Figure 4.1 we see two examples of images obtained by simulation from simple models with independent pixel values. To the left we have a 'pepper-and-salt' pattern corresponding to equal probababilities for black and white. To the right we have a grey-level image from a normal distribution (μ, σ^2) with $\mu = 0.5$, $\sigma = 0.2$ and truncated to the interval $[0, 1]$, that is, if a value less than 0 was generated it was replace by 0 and if a value larger than 1 was generated it was replaced by 1.

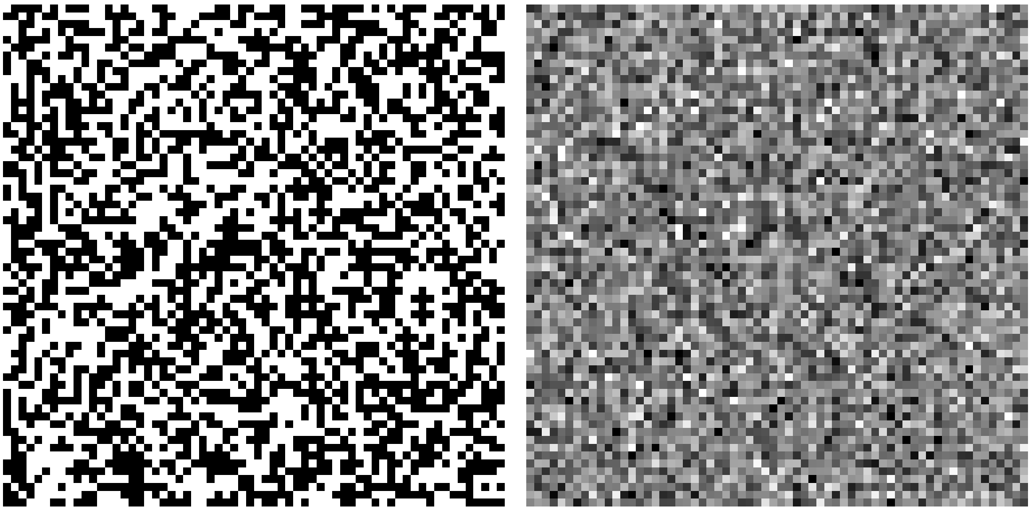


Figure 4.1: Images of size 64×64 obtained by simulation from models with independent pixel values: to the left a black-and-white image with equal probabilities for the two colours, and to the right a grey-level image with values from a normal distribution with expectation $\mu = 0.5$, a standard deviation $\sigma = 0.2$ and truncated to the interval $[0, 1]$.

In the following sections we will generalize to models with dependence between pixel values. We will consider Markov random field models defined by a neighbourhood for each pixel and a corresponding conditional distribution for the pixel value given the pixel values in the neighbourhood. But first we will take a look at Markov chains in one dimension.

4.1 One-dimensional Markov chains

A random sequence X_t with values in a finite or countable set V is a Markov chain if

$$\Pr(X_{t+1} = x | X_s, s \leq t) = \Pr(X_{t+1} = x | X_t), \quad x \in V. \tag{4.1}$$

It is not easy to see how this can be generalized to processes in the plain. However, one can prove that the condition (4.1) is equivalent to the condition

$$\Pr(X_t = x | X_s, s \neq t) = \Pr(X_t = x | X_{t-1}, X_{t+1}), \quad x \in V, \tag{4.2}$$

that is, if we want to predict X_t from all values $X_s, s \neq t$, it is enough to know X_s in the two neighbouring sites with $s = t - 1$ and $s = t + 1$. And the condition (4.2) can be generalized in a straightforward way to several dimensions as will be seen in the next section.

4.2 Markov random field models

Let us regard a random image $X = (X_s, s \in S)$, where S denotes the set of sites (pixel locations). We suppose that to each site $s \in S$ there is defined a set $N_s \subset S$ of neighbour sites such that the following two conditions are satisfied:

(i) $s \notin N_s$,
(ii) $t \in N_s$ if and only if $s \in N_t$.

Two often used neighbourhood systems are shown in Figure 4.2. To the left we see the system where the site $s = (i, j)$ has the neighbourhood

$$N_s = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}. \tag{4.3}$$

In the system shown in the right part of the figure there are four additional neighbours so that N_s then consists of eight sites.

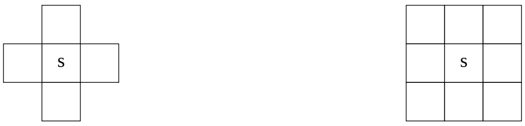


Figure 4.2: Two often used neighbourhood systems: to the left the site s has four neighbours and to the right it has eight neighbours.

Suppose that $X = (X_s, s \in S)$ is a set of discrete random variables taking values in the set V. We say that X is a *Markov random field* with respect to the system $(N_s, s \in S)$ of neighbourhoods if

$$\Pr(X_s = x | X_t, t \neq s) = \Pr(X_s = x | X_t, t \in N_s), \quad x \in V, s \in S. \tag{4.4}$$

This means that if we want to predict the pixel value X_s at s knowing all other pixel values we get the same prediction as when we only know the pixel values in the neighbourhood

N_s. This will turn out to be highly useful in an iterative sampling method called Gibbs sampling, which may be used for simulation of a Markov random field.

Neighbourhoods of border sites have to be considered separately. Suppose that the set of sites is

$$S = \{(i, j) : i = 1, \ldots, m, j = 1, \ldots, n\}. \tag{4.5}$$

One possibility is to use *periodic boundary conditions* which means that sites in the leftmost column are considered as neighbours of sites in the rightmost column, and, similarly, that sites in the top row are considered as neighbours of the bottom row. Specifically, if (4.3) gives neighbourhoods for non-border sites, we define for $s = (i, n)$ with $1 < i < m$

$$N_s = \{(i-1, n), (i+1, n), (i, n-1), (i, 1)\}, \tag{4.6}$$

with similar definitions for other border sites. We can think of periodic boundary conditions as corresponding to a folding of S like a torus (a doughnut).

Example 4.1. The Ising model. Let S be given by (4.5) with periodic boundary conditions. In physical applications to be discussed below we are interested in large values of m and n. Suppose that X_s can take two possible values, -1 and $+1$. Let X_s^+ and X_s^- denote the number of neighbours of s that take positive and negative values, respectively. Thus $X_s^+ + X_s^- = 4$. In the basic two-dimensional model we assume that

$$\Pr(X_s = +1 | X_t, t \in N_s) = \frac{\exp(2\beta(X_s^+ - X_s^-))}{1 + \exp(2\beta(X_s^+ - X_s^-))}. \tag{4.7}$$

We assume that $\beta > 0$. Note that if $X_s^+ > X_s^-$, that is, if the number of neighbours of s with positive values is larger than the number of neighbours with negative values, then the probability that s shall also have a positive value is greater than 1/2.

An alternative way of specifying the probability distribution of X is as a Gibbs distribution,

$$\Pr(X = x) = \frac{1}{Z} \exp(\beta \sum_{s \sim t} x_s x_t), \tag{4.8}$$

where Z is a normalizing constant, which is notoriously difficult to compute in models of this type, and where $s \sim t$ denotes that s and t are neighbours. Thus we sum in the right member of (4.8) over all pairs (s, t) of sites that are neighbours. In physics the Ising model is used as a model for ferromagnetism and β may be interpreted as inverse temperature. It turns out that for temperature below a critical value, that is for $\beta > \beta_c$, there are long range dependencies and possible phase transitions, that is a clear majority of the X_s-values will either be equal to $+1$ or a clear majority will be equal to -1. But for $\beta < \beta_c$ there are no phase transitions and the value of X_s averaged over large sets of sites is close to zero. A famous computation by (Onsager, 1944) gives

$$\beta_c = \frac{1}{2} \log(1 + \sqrt{2}) = 0.44069 \tag{4.9}$$

A review of Gibbs distributions and their use in mathematical physics may be found in (Georgii *et al.*, 2001). □

4.3 Autonormal random field models

Let us now also regard Markov random field models, where $X_s, s \in S$ are continuous real-valued random variables. The condition (4.4) needs then a modification to

$$\Pr(X_s \in A | X_t, t \neq s) = \Pr(X_s \in A | X_t, t \in N_s), \quad A \subseteq \mathbb{R}, s \in S, \tag{4.10}$$

for all considered subsets A of $\mathbb{R}$. We here only consider some simple *autonormal* models where we assume that the conditional distribution of X_s given its neighbours is normal with a constant variance σ^2 and an expectation that is a linear combination of the neighbour values. Specifically, let us consider the neighbourhood system given by the left part of Figure 4.2 and denote the neighbours of s in the West, North, East and South directions $W(s)$, $N(s)$, $E(s)$, and $S(s)$, and assume that

$$\mathbf{E}(X_s | X_t, t \in N_s) = \mu + \beta_W (X_{W(s)} - \mu) + \beta_N (X_{N(s)} - \mu) + \beta_E (X_{E(s)} - \mu) + \beta_S (X_{S(s)} - \mu). \tag{4.11}$$

4.4 Simulation of Markov random fields

There are several ways of simulating images from Markov random field models. We will describe one of the most used methods, Gibbs sampling.

In *Gibbs sampling* we visit the sites $s \in S$ in a specified way which may be random or deterministic. An often used random method is to choose successive sites to be visited independently and in a purely random way from the set of all sites. And an often used deterministic visiting scheme for a set of sites such as (4.5) is to choose sites to be visited row-wise from left to right starting with the first row and proceeding until all sites have been visited. Such a set of visits is called a sweep. The procedure is iterated a given number of of sweeps.

Example 4.2. The Ising model. Continuation. Consider Gibbs sampling for the Ising model by use of (4.7). As start configuration we use a purely random configuration as in the left part of Figure 4.1. For a set of β-values we see in Figure 4.3 binary images obtained by deterministic row-wise sweeps as described above. The upper two rows correspond to β values under the critical value (4.9), that is to high temperature, while the two lower rows correspond to low temperature. In the middle row we have β very close to the critical value, actually slightly above.

It may be noted that for large β-values (the two lower rows) the number of iterations used in Figure 4.3 is far too small to arrive at a stationary distribution for the Markov chain formed by the successive iterations. □

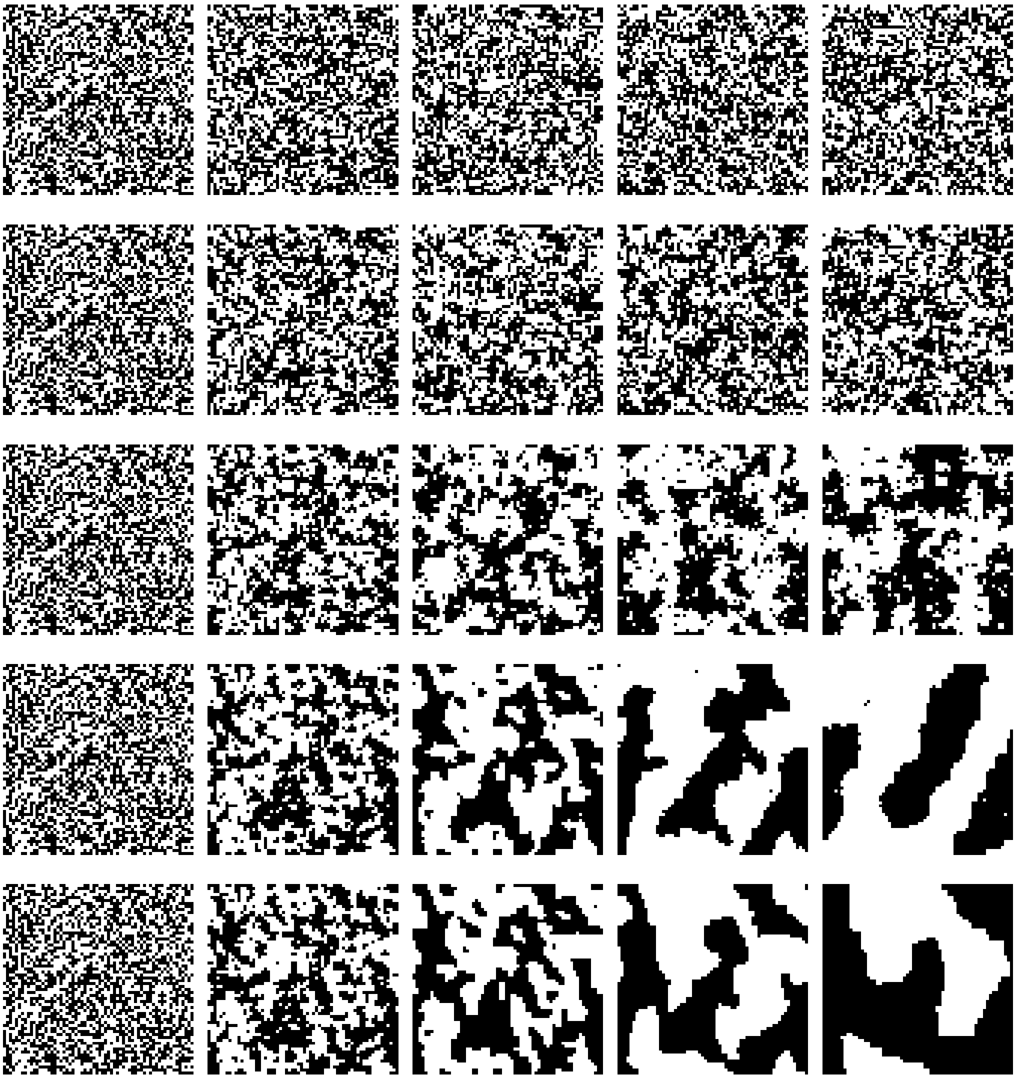


Figure 4.3: Binary images obtained by simulation for the Ising model with $\beta = 0.11$, 0.22, 0.4407, 0.88 and 1.76 in rows 1 to 5, respectively. In the columns we have to the left a purely random start configuration and then the result after 1 sweep, after 4 sweeps, after 16 sweeps and after 64 sweeps, respectively.

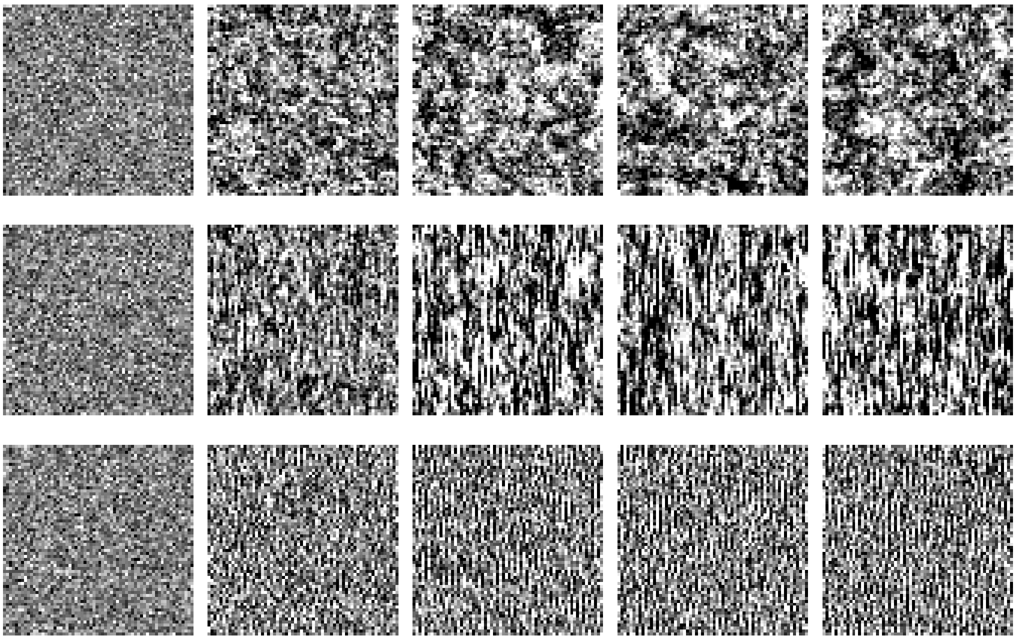


Figure 4.4: Grey-scale images obtained by simulation for autonormal models. In the columns we have to the left a purely random start configuration and then the result after 1 sweep, after 16 sweeps, after 128 sweeps and after 256 sweeps, respectively. The parameters in (4.11) are in the upper row $\beta_W = \beta_E = \beta_N = \beta_S = 0.24$, in the second row $\beta_W = \beta_E = 0$ and $\beta_N = \beta_S = 0.48$, and in the third row $\beta_W = \beta_E = -0.24$ and $\beta_N = \beta_S = 0.24$. In all three rows we have $\mu = 0.5$ and the residual standard deviation $\sigma = 0.3$.

Example 4.3. Simulation of an autonormal model. Consider Gibbs sampling for the autonormal model with conditional expectations (4.11) and constant conditonal variance given the neighbour values. For three sets of parameters we obtain results shown in Figure 4.4. □

4.5 Bayesian analysis of images

A common approach in Bayesian image analysis, is to assume that we start with a random image X given by a Markov random field. Then we observe a distorted image Y and one basic problem is to reconstruct X from Y. A simple model for the observed image $Y = (Y_s, s \in S)$ is to assume that given X the Y_s-variables are independent and furthermore that the distribution of Y_s only depends on X_s, that is we assume that

$$\Pr(Y = y|X) = \prod_{s \in S} \Pr(Y_s = y_s|X_s). \tag{4.12}$$

The reconstruction of X from Y is a difficult computional problem, and a series of iterative algorithms have been developed for this type of problems, most of them based on Markov chain Monte Carlo algorithms.

The use of Bayesian models for image reconstruction by use of Markov random field models as priors for the unobserved image X has generally suffered from the problem that it seems difficult to specify realistic priors for images typically found in applications. An interesting approach developed in particular by David Mumford and Song Chun Zhu is based on the following type of models, see for instance (Zhu & Mumford, 1997) for details and examples of which images that might be generated. Briefly the structure of the model for the prior is a Gibbs distribution, cf. (4.8) above, with

$$\Pr(X = x) = \frac{1}{Z}\exp(-U(x; \Lambda, F)), \tag{4.13}$$

where

$$U(x; \Lambda, F) = \sum_{\alpha=1}^{K}\sum_{s\in S}\lambda^{(\alpha)}((F^{(\alpha)} * x)(s)). \tag{4.14}$$

Here $F = \{F^{(1)}, \ldots, F^{(K)}\}$ is a set of linear filters and $\Lambda = \{\lambda^{(1)}, \ldots, \lambda^{(K)}\}$ is a set of functions, called potential functions, acting on the features extracted by the filter bank F.

4.6 Exercises

Exercise 4.1. Simulate images with independent pixel values as in Figure 4.1 but with k equi-distributed levels. Choose $k = 3$ and $k = 256$. (Note that the left image in Figure 4.1 corresponds to $k = 2$.)

Exercise 4.2. Regard the Ising model with negative β-values. (In physics this model is used as a model for anti-ferromagnetism.) Use Gibbs sampling to simulate images as in Figure 4.3 with β = -0.11, -0.22, -0.44, -0.88 and -1.76. Try also to guess what the images will look like before making the simulations.

Exercise 4.3. Regard an autonormal model with a neighbourhood system as in the right part of Figure 4.1. Choose suitable notation and write a model corresponding to (4.11). Use Gibbs sampling to simulate images as in Figure 4.4 and suggest parameter combinations to obtain different types of random textures.

Exercise 4.4. Show that if the distribution of X is given by (4.8), then (4.7) holds. Hint: one can use that

$$\Pr(X_s = +1|X_t = x_t, t \in N_s) = \frac{\Pr(X_s = +1, X_t = x_t, t \in N_s)}{\Pr(X_s = +1, X_t = x_t, t \in N_s) + \Pr(X_s = -1, X_t = x_t, t \in N_s)}.$$

4.7 Markov Chain Monte Carlo methods

Let us briefly describe Markov Chain Monte Carlo methods. We start with the Metropolis-Hastings algoritm. Suppose that we want to estimate the expectation

$$\mathbf{E}(g(X)) = \int g(x)f(x)\,dx\ , \tag{4.15}$$

where X is a random variable in d-dimensional Euclidean space with probability density f. Suppose further that we only know the density f except for a multiplicative constant, that is we know an unnormalized density

$$f^\star(x) = cf(x) \tag{4.16}$$

but not the normalization constant

$$c = \int f^\star(x)\, dx\,. \tag{4.17}$$

In the Metropopolis-Hastings algorithm we generate a sequence of random variables $X_1, \ldots, X_n$ forming a Markov chain with a distribution converging to the distribution of X. To generate X_{t+1} from X_t use a proposal distribution $q(\cdot|X_t)$ and generate a d-dimensional random variable Y_t. An often used proposal distribution is obtained by a random walk model, that is

$$Y_t = X_t + \epsilon_t\,, \tag{4.18}$$

where ϵ_t has d independent zero mean normal components with variance σ^2. The proposed variable Y_t is accepted as X_{t+1} with probability

$$\alpha(Y_t|X_t) = \min\left\{1, \frac{f^\star(Y_t)\, q(X_t|Y_t)}{f^\star(X_t)\, q(Y_t|X_t)}\right\}\,. \tag{4.19}$$

If Y_t is not accepted we put $X_{t+1} = X_t$. To control the acceptance or rejection of Y we generate an independent random variable U_t with a uniform distribution on the interval $(0, 1)$ independent of Y_s and U_s for $s < t$. Then we put

$$X_{t+1} = \begin{cases} Y_t & \text{if } U_t < \alpha(Y_t|X_t) \\ X_t & \text{otherwise}\,. \end{cases} \tag{4.20}$$

An excellent self-contained introduction to Markov chain Monte-Carlo methods with focus on the Metropolis-Hastings algorithm is given in (Robert, 2016).

4.8 Literature on statistical image modelling

Bayesian models for images became popular in the eighties following work by (Grenander, 1983) and (Geman & Geman, 1984). Markov chain Monte Carlo methods play an important role in reconstruction of images observed with noise. Important algorithms are simulated annealing, the Metropolis algorithm and Gibbs sampling, which all are examples of randomized algorithms. A simple iterative method, iterated conditional modes, was introduced by (Besag, 1986). (Winkler, 2003) gives a thorough treatment of these methods from a mathematical point of view. For an introduction to randomized algorithms viewed as Markov chains, see (Häggström, 2002), including a description of *exact* or *perfect* simulation algorithms.

PART 2 SPATIAL STATISTICS

Chapter 5

Spatial random processes

5.1 Spatial covariance functions

Let $X = (X_s, s \in S)$ be a spatial random process also called random field, where s is a spatial coordinate. In this chapter S may either be a discrete set, as when X is a digital image, or a continuous set, e.g. a rectangle $S = \{(s_1, s_2) \in \mathbb{R}^2 : a_1 \leq s_1 \leq b_1, a_2 \leq s_2 \leq b_2\}$. In this book we limit ourselves to spatial processes in two dimensions, but generalizations to d dimensions are fairly straightforward.

A spatial random process may be characterized by its mean value function,

$$m_s = \mathbf{E}X_s \tag{5.1}$$

and its covariance function

$$C(s,t) = \mathbf{E}(X_s - m_s)(X_t - m_t). \tag{5.2}$$

A spatial random process $X = (X_s, s \in S)$ is Gaussian if the joint distribution of $(X_{s_1}, \ldots, X_{s_n})$ is an n-dimensional normal distribution for any choice of coordinates $s_1, \ldots, s_n$ in S. A Gaussian random process is completely specified by its mean value and covariance functions.

It should be noted that not all functions of two variables are possible covariance functions. In fact, a necessary and sufficient condition that C is a valid covariance functions is that C is symmetric, that is $C(s,t) = C(t,s)$, and that it is positive-definite, that is satisfies

$$\sum_i \sum_j a_i a_j C(s_i, s_j) \geq 0 \tag{5.3}$$

for all n, $a_1, \ldots, a_n$, and $s_1, \ldots, s_n$. Note that the necessity of the condition (5.3) follows directly from the fact that

$$\mathbf{E}(\sum_{i=1}^{n} a_i(X_{s_i} - m_{s_i}))^2 = \sum_i \sum_j a_i a_j C(s_i, s_j). \tag{5.4}$$

A spatial random process $(X_s), s \in S$ is said to be stationary if its distribution is invariant under a translation $t \in \mathbb{R}^2$, more precisely if for each choice of $n \geq 1$ and

$(s_1, \ldots, s_n)$ the distribution of the n-dimensional random vector $(X_{s_1+t}, \ldots, X_{s_n+t})$ does not depend on t, as long as $s_i + t \in S, i = 1, \ldots, n$. The mean value for a stationary spatial process is a constant $m = \mathbf{E}X_t$ and the covariance function $C(s,t)$ depends only on $s - t$. A stationary spatial process is further called *isotropic* if its distribution is invariant under rotation of S. For an isotropic spatial process the covariance function $C(s,t)$ only depends on $|s - t|$, the Euclidean distance between s and t. The covariance function can then be written on the form

$$C(s,t) = \sigma^2 \rho(|s-t|), \tag{5.5}$$

where $\rho = \rho(r), r \geq 0$, is called the correlation function. Examples of correlation functions ρ that give valid (positive-definite) covariance functions are:

the *exponential correlation function*

$$\rho(r, \theta) = \exp(-r/\theta), \tag{5.6}$$

the *Gaussian correlation function*

$$\rho(r, \theta) = \exp(-(r/\theta)^2), \tag{5.7}$$

the *linear correlation function*

$$\rho(r, \theta) = (1 - r/\theta)1(r < \theta), \tag{5.8}$$

the *rational quadratic correlation function*

$$\rho(r, \theta) = \frac{1}{1 + (r/\theta)^2} \tag{5.9}$$

and the *spherical correlation function*

$$\rho(r, \theta) = (1 - \frac{2}{3}(r/\theta) + \frac{1}{2}(r/\theta)^3)1(r < \theta). \tag{5.10}$$

In Section 9.4, see in particular Figure 9.4, we show an example with comparison of five fitted correlation functions (5.6) - (5.10).

Suppose that we have a valid covariance function $C(s,t)$, and that $\sigma_0^2 > 0$. Then we can construct a new valid covariance function $C_0(s,t)$ by putting

$$C_0(s,t) = \begin{cases} \sigma_0^2 + C(s,t) & \text{if } s = t \\ C(s,t) & \text{if } s \neq t. \end{cases} \tag{5.11}$$

The constant σ_0^2 in (5.11) is sometimes called a *nugget* effect with regard to applications in mining. Another interpretation of the added quantity σ_0^2 in (5.11) is that it just corresponds to adding independent noise with variance σ_0^2 to all our original observations.

In the geostatistics literature one often uses instead of the covariance function the *semivariogram* introduced by Matheron, cf. (Matheron, 1971) and defined as

$$\gamma(s,t) = \frac{1}{2}\mathrm{var}(X(s) - X(t))\,,$$

or the *variogram* $\mathrm{var}(X(s) - X(t))$. For an isotropic process with variance σ^2 and correlation function $\rho(r)$ one finds

$$\gamma(s,t) = \sigma^2(1 - \rho(r)) \tag{5.12}$$

when $|s - t| = r$.

Let us now look at how one can simulate a Gaussian spatial random process $X_s, s \in S$, on a rectangular lattice S with given mean and covariance functions. Let us regard the process on a spatial lattice with n_1 rows and n_2 columns, and let X denote the matrix of random variables that we want to simulate. Let (here) m and C denote the mean matrix and covariance matrix of X.

Let us reorder the X-values into a column vector called $\tilde{X}$. (This can be done in several ways, for instance by starting with the first column of X, then take the second column et cetera.) The transformation from X to $\tilde{X}$ we call $\mathcal{T}$ and it's inverse we call $\mathcal{T}^{-1}$. Thus

$$\tilde{X} = \mathcal{T}X \quad \text{and} \quad X = \mathcal{T}^{-1}\tilde{X}\,.$$

Put $n = n_1 n_2$ and let $\tilde{C}$ denote the $n \times n$ covariance matrix of $\tilde{X}$. Let $R = \mathrm{chol}(\tilde{C})$ be the *cholesky factor* of $\tilde{C}$, here defined as the unique upper-triangular matrix with non-negative diagonal elements such that

$$\tilde{C} = R^T R\,. \tag{5.13}$$

Let further Z be a column vector with n independent standard normal random variables and put

$$\tilde{X} = R^T Z\,. \tag{5.14}$$

It follows that the covariance matrix of $\tilde{X}$ is

$$\mathbf{E}(R^T Z Z^T R) = R^T R = \tilde{C}$$

and that

$$X = m + \mathcal{T}^{-1}\tilde{X} \tag{5.15}$$

has mean matrix m and covariance matric C as desired. In the next section we will see how such a simulation functions.

5.2 Matérn's covariance function

A flexible and much used correlation function is Matérn's correlation function suggested 1960

$$\rho(r) = \rho(r;\nu,\theta) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{r}{\theta}\right)^{\nu} K_\nu\left(\frac{r}{\theta}\right), \tag{5.16}$$

where $\nu > 0$ and $\theta > 0$ are smoothness and scale parameters, and K_ν is a modified Bessel function of the second kind, which may be expressed as an integral

$$K_\nu(x) = \frac{2^\nu \Gamma(\nu + 1/2)}{\sqrt{\pi} x^\nu} \int_0^\infty \frac{\cos xt}{(t^2+1)^{\nu+1/2}}\, dt\,, \tag{5.17}$$

see (Matérn, 1986) and (Gelfand *et al.*, 2010) for further details. Some special cases, where Matérn's correlation function can be expressed in terms of elementary functions are given in Table 5.1.

Table 5.1: Special cases of Matérn's correlation function

Smoothness parameter ν	Matérn's correlation function $\rho(r)$ for scale parameter $\theta = 1$
$\nu = 1/2$	$\rho(r) = \exp(-r)$
$\nu = 3/2$	$\rho(r) = (1 + r)\exp(-r)$
$\nu = 5/2$	$\rho(r) = (1 + r + r^2/3)\exp(-r)$

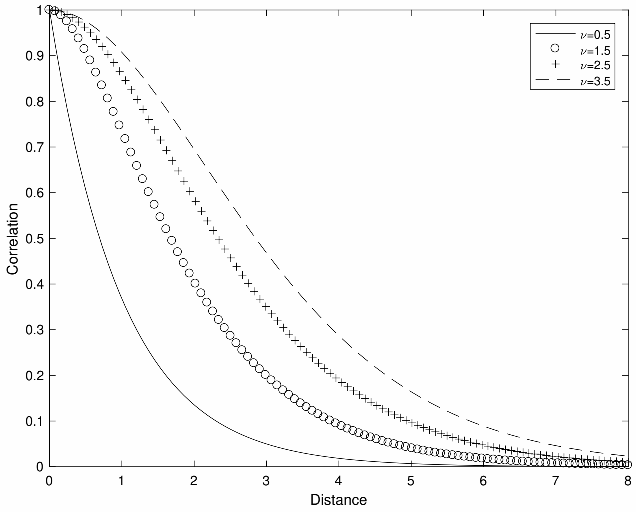


Figure 5.1: Four examples of Matérn correlation functions $\rho(r;\nu,\theta)$ from (5.16), plotted against distance r, with varying smoothness parameters ν and with constant scale parameter $\theta = 1$.

In Figure 5.1 some Matérn correlation functions, including those in Table 5.1, are plotted. One can show that for increasing shape parameter ν the Matérn correlation function gets close to the Gaussian correlation function in (5.7), which seems plausible from Figure 5.1. However, to get a proper limit result one has to normalize with a proper scale parameter, which also seems plausible from Figure 5.1. More precisely we can put

$$\theta = 1/(2\sqrt{\nu})\,. \tag{5.18}$$

One can show that with $\rho(r;\nu,\theta)$ given by (5.16) we have

$$\lim_{\nu\to\infty} \rho(r;\nu,1/(2\sqrt{\nu})) = \exp\left(-r^2\right). \tag{5.19}$$

Related to the scaling (5.18) is the observation that the practical correlation range d_{range} for Matérn's correlation function is

$$d_{\text{range}} \approx \theta\sqrt{(8\nu)}\,. \tag{5.20}$$

Check in Figure 5.1 if the relation (5.20) seems reasonable.

Let us now see how we can simulate Gaussian processes with Matérn correlation functions. We will use the method described in the previous section, see (5.15), for two different correlation functions (5.16) with $\nu = 0.5$ and $\nu = 1.5$. To get essentially the same correlation range we also use the scale normalization (5.18), compare Figure 5.3, and to get covariance functions we multiply the correlation functions with $\sigma^2 = 1$. Resulting realizations (sample surfaces) are shown in Figure 5.2. Let us note that the for both the left and the right part of Figure 5.2 the practical correlation range $d_{\text{range}} \approx \sqrt{2}$ corresponding to the length of diagonals.

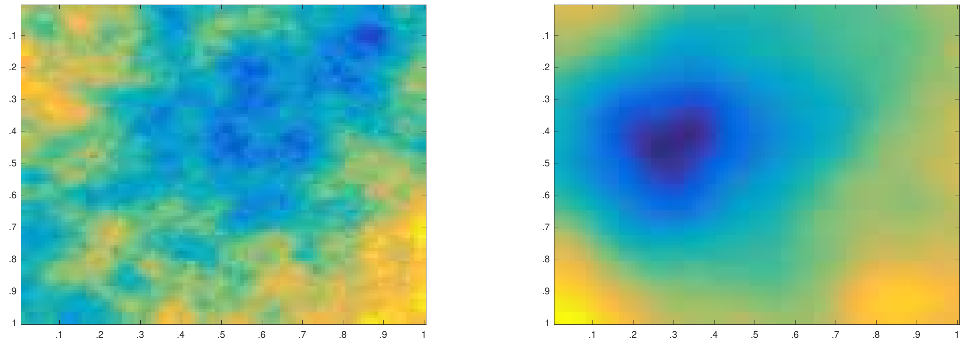


Figure 5.2: Two two-dimensional realizations with Matérn correlation functions $\rho(r;\nu,\theta)$ from (5.16) with $\nu = 0.5$ (left) and $\nu = 1.5$ (right). The simulation method with realizations obtained with (5.15) was used on a square two-dimensional set $S = [0,1] \times [0,1]$ with 100 pixels in both the horizontal and vertical directions. In both cases the scale parameter θ is given by (5.18) and to get covariance functions we multiply the correlation functions with $\sigma^2 = 1$.

One can show that realizations from a Matérn process with $\nu > 0$ are continuous. They are m times differentiable if and only if $\nu > m$. Thus the left realization in Figure 5.2 is continuous but not differentiable, and the right realization is differentiable once but not twice. To illustrate continuity and differentiability better we show in Figure 5.4 one-dimensional realizations of Matérn processes with ν equal to 1/2, 3/2 and 5/2, respectively, which thus are zero, one and two times differentiable. With $v = 1/2$ we get in one dimension the well-known *Ornstein-Uhlenbeck process*, see . It is a Markov process with correlation function

$$\rho(t) = \exp(-t)\,,$$

which similar to the Wiener process is continuous but nowhere differentiable.

5.3 Euclid's hat covariance function

Matérn's covariance function $\rho(r;\nu,\theta)$ in (5.16) is positive for all $r > 0$. Sometimes it is useful to have a covariance function that is zero from a finite range and onwards. A family of such covariance functions is Euclid's hat covariance function

$$\rho(r) = \rho(r;n,\theta) = \sigma^2 I_{\frac{n+1}{2},\frac{1}{2}}(1 - r^2/\theta^2)\,1(r < \theta)\,, \tag{5.21}$$

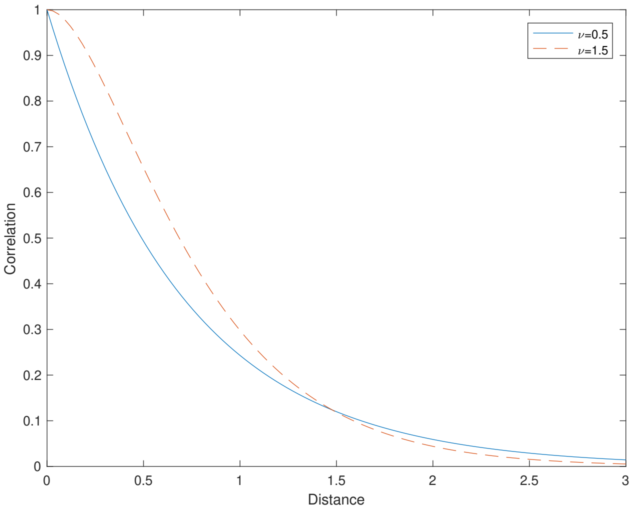


Figure 5.3: The two Matérn correlation functions $\rho(r;\nu,\theta)$ from (5.16) with $\nu = 0.5$ and $\nu = 1.5$ used in Figure 5.2, plotted against distance r. In both cases the scale parameter θ is given by (5.18).

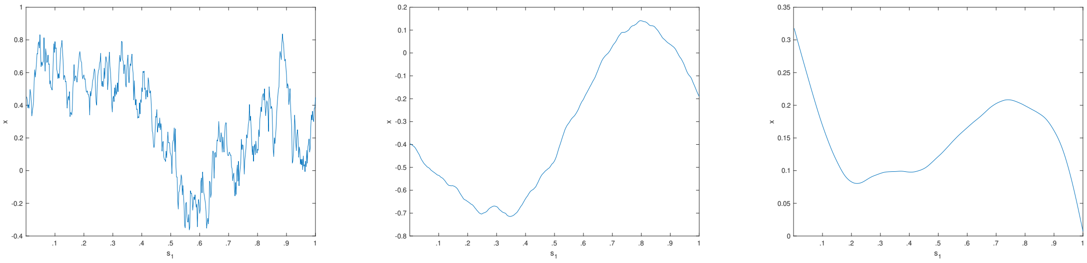


Figure 5.4: Three one-dimensional realizations with Matérn correlation functions $\rho(r;\nu,\theta)$ from (5.16) with $\nu = 0.5$ (left), $\nu = 1.5$ (center) and $\nu = 2.5$ (right). In all three cases the scale parameter θ is given by (5.18) and to get covariance functions we multiply the correlation functions with $\sigma^2 = 1$. The processes are simulated on the interval $[0, 1]$ which is divided into 500 pixels.

where

$$I_{\frac{n+1}{2},\frac{1}{2}}(x) = \frac{\int_0^x \sqrt{t^{n-1}(1-t)^{-1}}\,dt}{\int_0^1 \sqrt{t^{n-1}(1-t)^{-1}}\,dt} \tag{5.22}$$

is the regularized incomplete beta function.

Let us note that $n = 1$ corresponds to the linear correlation function (5.8) and $n = 3$ corresponds to the spherical correlation function (5.10). Some Euclid's hat correlation functions are shown in Figure 5.5.

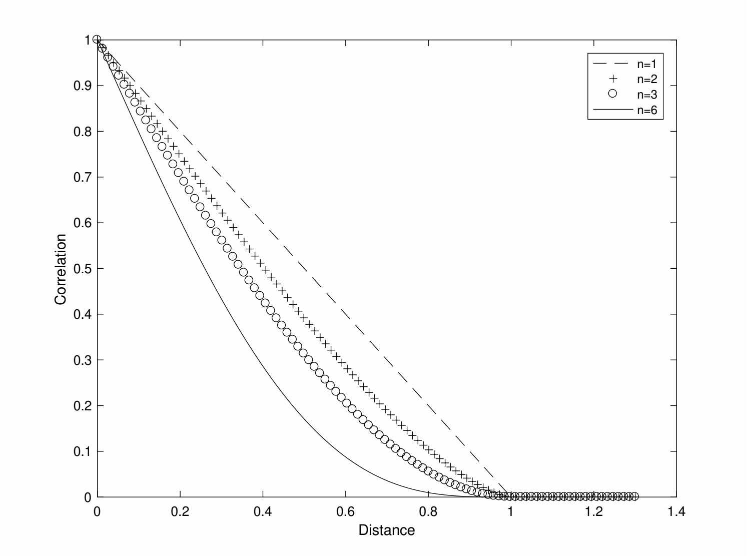


Figure 5.5: Four examples of Euclid's hat correlation functions $\rho(r; n, \theta)$ from (5.21), plotted against distance r, with varying parameter n and constant scale parameter $\theta = 1$.

5.4 Statistical models for observations of random fields

Suppose that we have measurements $Y_i, i = 1, \ldots, n$, taken at spatial locations $s_1, \ldots, s_n$. Let $B_1, \ldots, B_K$ be explanatory variables and assume that

$$Y_i = \sum_{k=1}^{K} B_k(s_i)\beta_k + X(s_i) + \epsilon_i \,, \tag{5.23}$$

where $X = (X(s), s \in S)$ is a Gaussian random field and $\epsilon_1, \ldots, \epsilon_n$ are zero mean normal random variables with variance σ_ϵ^2 independent mutually and of X. The questions we shall look at are:

(i) How can we estimate parameters in the model (5.23)?

(ii) How can we predict an observation at an unobserved location s_0 ?

As an example we shall look at mean summer time (June – August) temperatures in continental US recorded at 250 weather stations 1997. The temperatures and a number of possible explanatory variables can be obtained from
http://www.image.ucar.edu/GSP/Data/US.monthly.met/
and some further information including how missing data were estimated can be found in (Johns *et al.*, 2003). Figure 5.6 shows the mean summer temperatures.

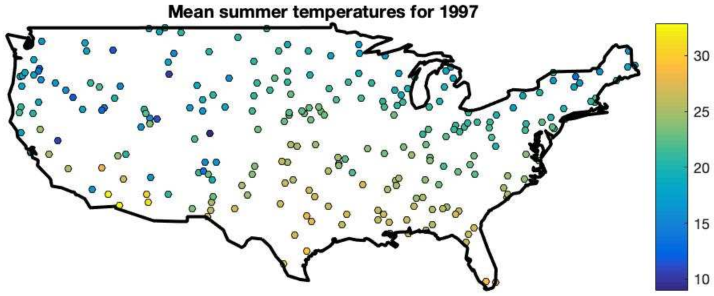


Figure 5.6: Mean summer temperatures for 1997 recorded at 250 weather stations in contintental US.

Our first approach will be to use ordinary least squares with a number (here five) covariates but without the random field X, that is to use the model

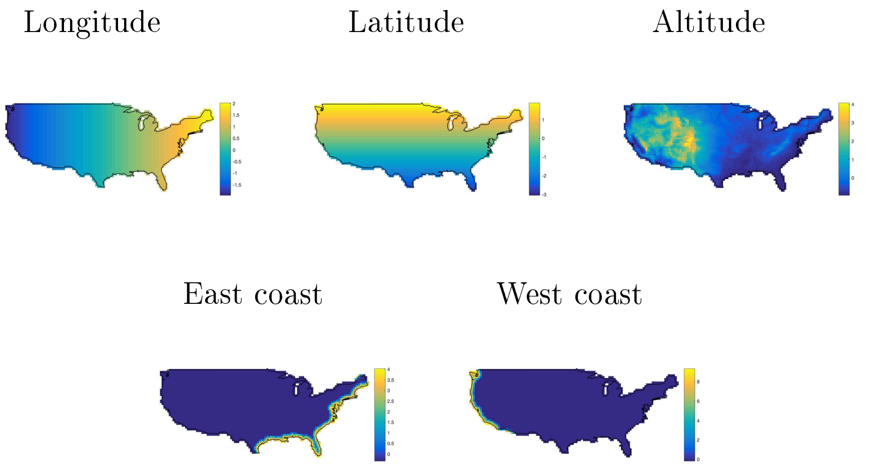
$$Y_i = \sum_{k=0}^{K} B_k(s_i)\beta_k + \epsilon_i \,, \tag{5.24}$$

where we also have included an intercept β_0 and correspondingly we put $B_0(s_i) = 1$. The model can also be written

$$Y = B\beta + \epsilon \,, \quad \epsilon \sim N(0, \sigma_\epsilon^2 I) \,. \tag{5.25}$$

Table 5.2: OLS (Ordinary Least Squares) analysis of US continental summer temperatures 1997. Residual standard deviation estimate $\hat{\sigma_\epsilon} = 1.10$.

Explaining variable	Estimate $\hat{\beta}_k$	Corresponding t-value
Intercept	21.63	189.17
Longitude	-1.29	-8.15
Latitude	-2.70	-22.72
Altitude	-2.67	-18.33
East coast	-0.10	-0.74
West coast	-1.31	-10.24



Longitude

Latitude

Altitude

East coast

West coast

Figure 5.7: Five covariates used in the analysis of summer temperature in contintental US.

The covariates we use are Longitude, Latitude, Altitude, East coast and West coast, see Figure 5.7. Table 5.2 shows the parameter estimates

$$\hat{\beta}_{\text{OLS}} = (B^T B)^{-1} B^T Y \tag{5.26}$$

of the OLS analysis of the data. The residual degrees of freedom is 250-6= 244. From the column of t-values we see that all the parameter estimates except one in Table 5.2 are higly significantly different from zero. The OLS regression surface estimate

$$\hat{Y}_{\text{OLS}} = B\hat{\beta}_{\text{OLS}} \tag{5.27}$$

of the temperature surface is shown in Figure 5.8 and the OLS regression residuals

$$\text{res}_{\text{OLS}} = Y - B\hat{\beta}_{\text{OLS}} \tag{5.28}$$

are shown in Figure 5.9. From Figure 5.9 we see that residuals close in location seem highly correlated, which indicates that the model could be improved.

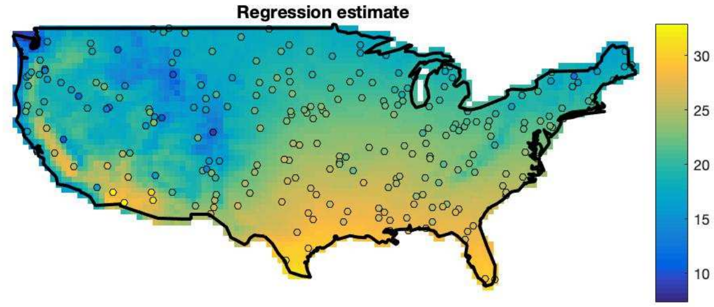


Figure 5.8: OLS regression temperature surface estimate

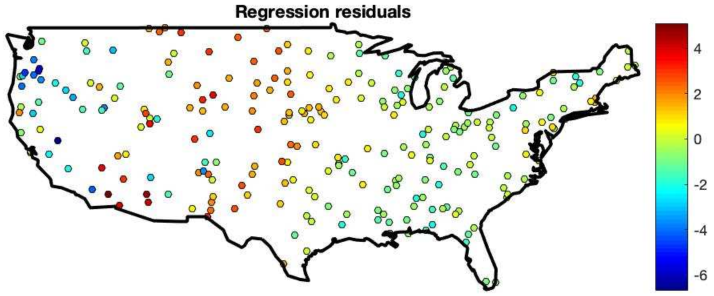


Figure 5.9: OLS regression temperature residuals

To improve the model (5.25) we will assume that

$$Y = B\beta + \epsilon\,, \quad \epsilon \sim N(0, \Sigma)\,, \tag{5.29}$$

where Σ is a general positive-definite covariance matrix. One can show that with this model the least squares estimate of β called the GLS (Generalized Least Squares) estimate is

$$\hat{\beta}_{\text{GLS}} = (B^T\Sigma^{-1}B)^{-1}B^T\Sigma^{-1}Y \tag{5.30}$$

with corresponding GLS regression surface estimate

$$\hat{Y}_{\text{GLS}} = B\hat{\beta}_{\text{GLS}} \tag{5.31}$$

and GLS regression residuals

$$\text{res}_{\text{GLS}} = Y - B\hat{\beta}_{\text{GLS}}\,. \tag{5.32}$$

One problem with GLS is that typically the covariance matrix Σ in (5.30) is unknown and has to be estimated. One possible estimation method is to start with OLS residuals

Table 5.3: OLS (Ordinary Least Squares) and GLS (Generalized Least Square) parameter estimates US continental summer temperatures 1997. Stars indicate that the corresponding parameter is significantly different from zero

Explaining variable	OLS estimate	GLS estimate
Intercept	21.63*	20.47*
Longitude	-1.29*	-1.00*
Latitude	-2.70*	-2.68*
Altitude	-2.67*	-4.22*
East coast	-0.10	-0.01
West coast	-1.31*	-1.01*

and bin them in a suitable way [NOTE this has to be explained]. For our summer temperature data this results in the parameter estimates in the third column in Table 5.3

The GLS regression surface estimate and residuals are shown in Figures 5.10 and 5.11.

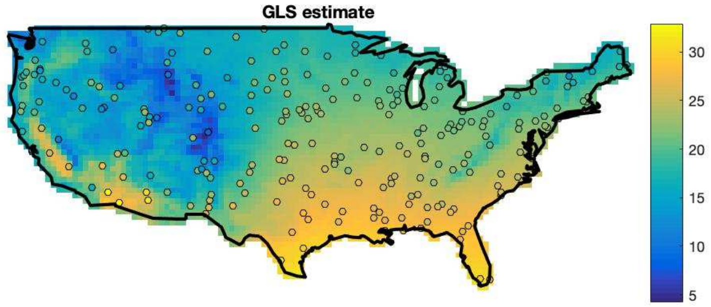


Figure 5.10: GLS regression temperature surface estimate

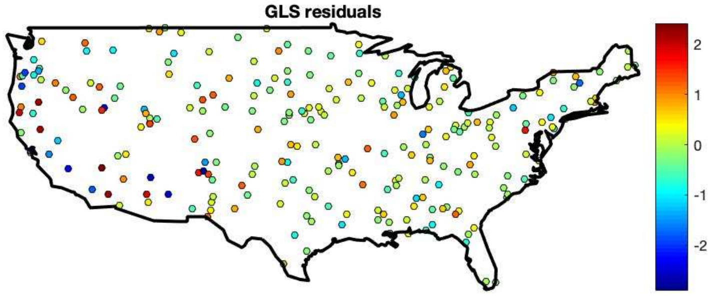


Figure 5.11: GLS regression temperature residuals

5.5 Literature on spatial random processes

A classical, still readable, monograph on spatial statistics is Bertil Matérn's doctoral dissertation from 1960, reproduced 1986 as (Matérn, 1986). Two other classical books are (Ripley, 1981) and (Cressie, 1993). A more modern, highly useful book is (Gelfand *et al.*, 2010).

Chapter 6

Point processes. Poisson processes.

Let A be a subset of $\mathbb{R}^2$ with finite and positive area $|A|$. We will consider a random subset X of A consisting of finitely many points, and call X a point process on A. If $B \subseteq A$ we let $X(B)$ denote the number of points in X that belong to B.

The point process X is said to be *stationary* if the probability distribution of X is invariant under any translation of the sets B where we regard the point process, and we say that X is *isotropic* if the process is stationary and if, additionally, the distribution of X is invariant under any rotation of such sets B.

Consider a stationary point process X on A such that $X(A)$ has finite expectation. One can then show that

$$\mathbf{E}(X(B)) = \lambda|B| \tag{6.1}$$

for some constant λ which we call the intensity of the point process.

Example 6.20. *Poisson process with constant intensity.*

A point process X is called a Poisson process with constant intensity $\lambda \geq 0$ on A if $X(B_1)$ and $X(B_2)$ are independent for disjoint subsets B_1 and B_2 of A and if $X(B)$ is Poisson distributed with expectation $\lambda|B|$ for a subset $B \subseteq A$ with area $|B|$, that is

$$\Pr(X(B) = n) = \frac{(\lambda|B|)^n}{n!}\exp(-\lambda|B|). \tag{6.2}$$

A Poisson process with constant intensity is stationary and isotropic.

A Poisson process on A with intensity λ can be generated in the following way. Let first N be Poisson distributed with expectation $\lambda|A|$. Given that $N = n$, generate $X_1, \dots, X_n$ as independent and identically distributed variables, each with a uniform distribution over A. (See Section 14.13 for a description of how to generate random numbers with a uniform distribution on a given bounded set in two dimensions.) Then we let X consist of the points $X_1, \dots, X_n$, that is $X = \{X_1, \dots, X_n\}$.

In Figure 6.1 we see two examples of such generation of a Poisson process in the unit square with the constant intensity $\lambda = 50$.

□

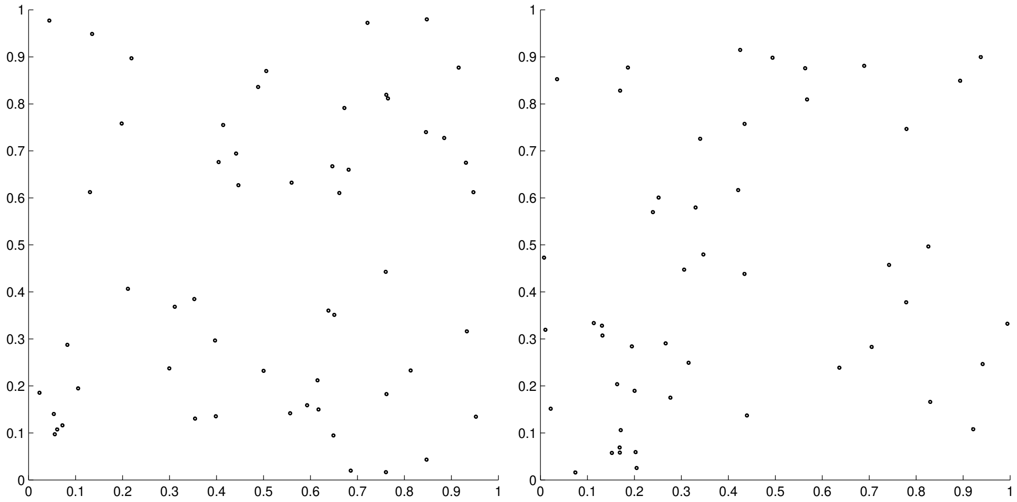


Figure 6.1: Two examples of Poisson point processes generated in the unit square with $\lambda = 50$. The generated number of points is to the left $N = 55$ and to the right $N = 49$.

Example 6.21. *Poisson process with varying intensity.*

A point process X is called a Poisson process with intensity function $\lambda(s), s \in A$, if $X(B_1)$ and $X(B_2)$ are independent for disjoint subsets B_1 and B_2 of A and if $X(B)$ is Poisson distributed with expectation $\int_B \lambda(s)\, ds$ for $B \subseteq A$.

A Poisson process with intensity function $\lambda(s)$, $s \in A$, can be generated in the following way. Let first N be Poisson distributed with expectation $\int_A \lambda(s)\, ds$. Given that $N = n$, generate $X_1, \ldots, X_n$ as independent and identically distributed variables, each with a distribution specified by

$$\Pr(X_i \in B) = \frac{\int_B \lambda(s)\, ds}{\int_A \lambda(s)\, ds} \quad \text{for} \quad B \subseteq A. \tag{6.3}$$

Then we put $X = \{X_1, \ldots, X_n\}$. □

6.1 The Neyman-Scott process, a point processes with clustering

Consider a Poisson process with constant intensity λ, and regard the points of this process as mother points. From each mother point we generate daughter points such that the number of daughter points from the mother points are all independent and identically distributed. Further, the two-dimensional vectors from a mother point to the daughter points are all independent and identically distributed. This distribution we call the scattering distribution. The process of daughter points is called a Neyman-Scott process.

Suppose that we want to generate a Neyman-Scott process. If the daughter process is regarded on a set A we need to start by generating the mother point process on a set larger than A, in fact so large that (essentially) all points from which daughters can get scattered into A are included. With this observation it is straightforward to generate a Neyman-Scott process from the definition above.

Example 6.22. *A Neyman-Scott plant process with 2D normal scattering.*

Suppose that we want simulate a Neyman-Scott process of mother and daughter plants within the unit square $[0,1]\times[0,1]$ with intensity $\lambda = 10$ for the Poisson process of mother points, with a number of daughter points that is binomial (n, p) with $n = 8$ and $p = 0.5$ and with a 2D scattering distribution that is $N(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$ with $\mu_1 = \mu_2 = \sigma_1 = \sigma_2 = 0.1$ and $\rho = 0.5$ corresponding to wind spread of seeds with a main wind direction from south-west. We start by simulating the Poisson mother plant point process in the axis-parallell quadrat with south-west and north-east corners in $(-0.5, -0.5)$ and $(1.3, 1.3)$, respectively. The result of the simulation is shown in Figure 6.2.

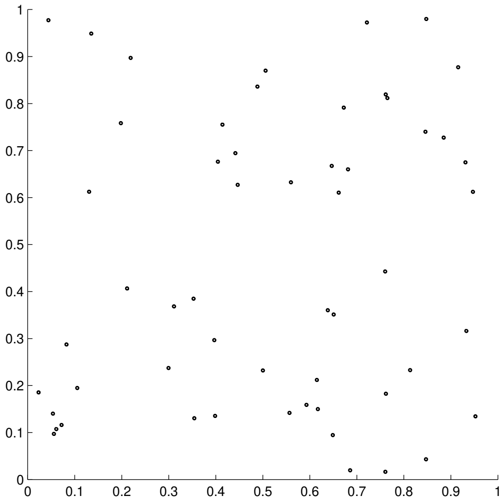


Figure 6.2: A simulation of a Neyman-Scott process with mother points as circles and daughter points as dots. OBS OBS a new figure must be generated.

□

6.2 A hard-core inhibition point process

In the cluster point process in the previous section the occurrence of a point typically increases the intensity of points in a neighborhood of this point. We will now describe a point processes with inhibition, suggested 1960 by Matérn, see (Matérn, 1986), which has the opposite property: the occurrence of a point inhibits other points within a certain distance.

Start by generating a Poisson point process with intensity λ on a bounded set A. To each point $X_i, i = 1, \dots, N$, we associate a random mark consisting of random variable U_i, which is uniformly distributed on the interval $(0, 1)$ and such that the U_i's are indendent, mutually and of the X_i's. We can think of U_i as the birth time of the point X_i.

Then we thin the X-process by deleting each point X_i for which there exists an older point X_j of the original point process closer than a distance d, that is a point X_j satisfying $|X_i - X_j| < d$ and $U_j < U_i$. The distance d is called the hard core distance.

6.3 The K-function, a diagnostic tool for detecting clustering and inhibition

Consider an isotropic point process with intensity λ and suppose that x is a point of the point process X. Let $\|y - z\|$ denote the distance between two points y and z in $\mathbb{R}^2$, and define the K-function of X as follows,

$$K(r) = \frac{1}{\lambda}\mathbf{E}(\text{number of further points of } X \text{ within distance } r \text{ from } x|x \in X) \tag{6.4}$$

or more precisely

$$K(r) = \frac{1}{\lambda}\mathbf{E}(X(C_x(r)|x \in X), \tag{6.5}$$

where $C_x(r) = \{y : 0 < \|y - x\|) \leq r\}$ denotes a circular disk with radius r around x with the point x excluded.

For a stationary Poisson process it follows that

$$K(r) = \pi r^2. \tag{6.6}$$

Sometimes one chooses to regard $L(r) = (K(r))^{1/2}$ as this function is linear in r for a Poisson process, for which

$$L(r) = \sqrt{\pi} r. \tag{6.7}$$

If we have a point process with clustering as for example the Neyman-Scott process we can expect that the K-function will lie above the K-function for a Poisson process for r-values where we have clustering, while for a point process with inhibition such as the Matérn hard-core process it should lie below for those r-values for which we have inhibition.

6.4 Point processes operations such as thinning, displacement and superposition

Consider a point process X on a set A. Suppose that the points of X are deleted independently with a probability $1-p$, and retained with retention probablity p, $1 \leq p \leq 1$. The resulting point process of retained points is called a *p-thinned point process.* If X is a Poisson process with constant intensity λ one can show that the p-thinned point process is a Poisson process with intensity $p\lambda$. Note that the hard-core inhibition point process described in Section 6.2 is obtained from a Poisson process by a more complicated thinning than independent thinning.

In Section 6.1 we described a daughter point process obtained by a clustering operation on a mother Poisson point process. The same clustering operation with independent and identically distributed daughter points can be obtained starting from an arbitrary mother point process. A useful special case is that each mother point gives birth to one exactly daughter point with a given scattering distribution. The resulting daughter point process then gives a *point process with displacements* with the original points independently displaced according to the scattering distribution.

A third useful point process operation is superposition $X \cup Z$ of two point processes X and Z on a given set A. For instance, if X is the basic point process that we consider, then Z can be an independent Poisson process of "ghost" points. In (Dralle & Rudemo, 1997) and (Lund & Rudemo, 2000) a point process X of tree positions measured on ground is studied together with positions Y obtained from an aerial photograph such as in Figure 1.2 or 1.4. The points of Y are modeled as obtained from X by the mechanisms of thinning, displacement and superposition of independent "ghost" points. The analysis of these mechanisms as discussed by (Dralle & Rudemo, 1997) is desribed in some detail Chapter 11 below.

6.5 Estimation of characteristics for point processes

Suppose that we have observed a stationary point process X on a set $A \subset \mathbb{R}^2$. The intensity of X we estimate by

$$\hat{\lambda} = \frac{X(A)}{|A|}. \tag{6.8}$$

It follows generally that for a stationary point process with finite intensity λ the estimator (6.8) is an unbiased estimator of the intensity, that is, $\mathbf{E}(\hat{\lambda}) = \lambda$.

For a Poisson process we can also compute the variance of the estimator (6.8). We find

$$\mathrm{var}(\hat{\lambda}) = \frac{\lambda}{|A|}. \tag{6.9}$$

Let us now regard estimation of the K-function of a point process X observed in the region A. The basic problem in estimating $K(r)$ is that for a point $x \in X$ we want to consider all neighbouring X-points within distance r. But some of these neighbours may be located outside A.

For our first estimator of $K(r)$ we consider pairs of X-points x and y such that $x \in A_r^-$, where A_r^- denotes the subset of A of points with a distance at least r to the border of A. Let $1\{P\}$ denote the function which is 1 when P is true and zero else. From the definition (6.4) it follows

$$\sum_{x \in X \cap A_r^-} \sum_{y \in X} 1\{0 < \|y - x\| < r\} \tag{6.10}$$

is an unbiased estimator of $\lambda^2 |A_r^-| K(r)$. The procedure of restricting to points within a certain distance to the border is called *minus-sampling*, and the corresponding estimator of $K(r)$ is therefore called $\hat{K}_{\text{minus}}(r)$, and it is obtained from the unbiased estimator (6.10) of $\lambda^2|A_r^-|K(r)$ by replacing λ with its estimator (6.8). We get

$$\hat{K}_{\text{minus}}(r) = \frac{1}{\hat{\lambda}^2|A_r^-|} \sum_{x \in X \cap A_r^-} \sum_{y \in X} 1\{0 < \|y - x\| < r\}. \tag{6.11}$$

Let us now give another estimator of the K-function which utilizes our observations more effectively. Regard two points x and y in the region A and a circle with centre at x and radius $\|y - x\|$. Let $w(x, y)$ denote the proportion of the perimeter of this circle that lies within A. If, for instance A is the unit square $[0, 1] \times [0, 1]$, $x = (1/2, 1/2)$ and $y = (1/2, -1/2 + 1/\sqrt{(}2)$, then a straightforward compution shows that $w(x, y) = 1$ and $w(y, x) = 3/4$. One can now show that

$$\sum_{x \in X} \sum_{y \in X} \frac{1\{0 < \|y - x\| < r\}}{w(x, y)} \tag{6.12}$$

is an unbiased estimator of $\lambda^2|A|K(r)$. The corresponding estimator of the K-function is

$$\hat{K}(r) = \frac{1}{\hat{\lambda}^2|A|} \sum_{x \in X} \sum_{y \in X} \frac{1\{0 < \|y - x\| < r\}}{w(x, y)}. \tag{6.13}$$

There is one minor restriction in the use of (6.13) which means that we cannot consider r so large that $w(x, y)$ become close to zero. In practice this is not important as we are usually interested in reasonably small r-values. Thus, for observations in the unit square an upper limit for r is $1/\sqrt{2}$.

6.6 Simulation-based envelope tests for point processes

Suppose that we have an estimate $\hat{K}(r)$ of the K-function of a point process X on the set A with, say, the estimator (6.13). As indicated in the end of Section 6.3 we should then be able to detect clustering or inhibition by comparing the estimated K-function with the K-function (6.6) valid for a stationary Poisson process. But how large deviation could we expect to find by pure randomness?

Useful simulation-based envelope-techniques have been introduced to tackle this problem, compare (Diggle, 2013). Let us start with describing a technique which is useful as an exploratory tool. Put $n = X(A)$ and generate M independent copies $X_1, \dots, X_M$ of a Poisson process on A conditioned on $X_m(A) = n, m = 1, \dots, M$. Thus the points of each X_m can be obtained by independent random sampling of n points in A. Let $\hat{K}_m(r)$ denote the K-function estimate corresponding to X_m, $m = 1, \dots, M$. We are interested in evaluating the probability that $\hat{K}(r)$ lies between the envelopes $\text{min}_m\hat{K}_m(r)$ and $\text{max}_m\hat{K}_m(r)$.

Assume for simplicity that $M = 39$. Then we have provided that X is a Poisson process, and for fixed r,

$$\begin{aligned} &\Pr\left(\text{min}_{1\le m\le M}\hat{K}_m(r) \le \hat{K}(r) \le \text{max}_{1\le m\le M}\hat{K}_m(r\right) = \\ &1 - \Pr\left(\text{min}_{1\le m\le M}\hat{K}_m(r) > \hat{K}(r)\right) - \Pr\left(\hat{K}(r) > \text{max}_{1\le m\le M}\hat{K}_m(r\right) = \\ &1 - 0.025 - 0.025 = 0.95. \end{aligned} \tag{6.14}$$

A tempting strategy is then to plot $\hat{K}(r)$ together with the envelopes $\min_m \hat{K}_m(r)$ and $\max_m \hat{K}_m(r)$, and to conclude that the Poisson hypothesis is rejected if $\hat{K}(r)$ somewhere falls outside the envelopes. However this procedure does not give a valid test at the level $p = 0.05$ as the calculation above is only valid for a fixed r-value. However, it may still be used as an exploratory technique indicating for which r-values the Poisson hypothesis may not be valid. There have been developed valid tests with envelope bounds, see for instance (Myllymäki *et al.*, 2017).

6.7 Exercises

Exercise 6.1. Generate a Poisson process on the unit square $[0,1]\times[0,1] \subset \mathbb{R}^2$ with constant intensity 100. Show the result in a figure.

Exercise 6.2. Generate a Poisson process on the unit square $A = [0,1]\times[0,1]$ with varying intensity $\lambda(s) = 200s_1, s = (s_1, s_2) \in A$. Show the result in a figure.

Exercise 6.3. Generate a Neyman-Scott process on the unit square $A = [0,1]\times[0,1] \subset \mathbb{R}^2$ in the following way. Assume that (*i*) the mother process is a Poisson process with constant intensity 50, (*ii*) each mother point generates two daughter points, and (*iii*) the scattering distribution (from mother to daughter) is an isotropic two-dimensional normal distribution with zero means and standard deviation 0.01 in both horizontal and vertical directions. (Truncate here the normal distributions at, say, plus and minus three standard deviations.) Show the result in a figure.

Exercise 6.4. Compute the expected distance from one mother point to its nearest neighbour mother point for the point process of the previous exercise, and also the expected distance between the two daughter points from one mother point (disregard in these computations edge effects, that is the limited size of the set A). Instead of the two expected distances you may choose to compute root-mean square distances, that is the square root of the expected squared distances, which are a bit easier to compute.

Exercise 6.5. Generate a hard core Matérn point process on the unit square $[0,1]\times[0,1] \subset \mathbb{R}^2$ with $\lambda = 100$ and $d = 0.1$. Show the result in a figure.

Exercise 6.6. Estimate the intensity and the K-function for the point processes considered in (a) Exercise 6.1, (b) Exercise 6.3, and (c) Exercise 6.5. Compare the three K-function estimates.

Exercise 6.7. Generate copies of Poisson processes $X_1, \ldots, X_M$ with $M = 39$ and corresponding K-function estimates as described in Section 6.6 for the point processes considered in (a) Exercise 6.1, (b) Exercise 6.3, and (c) Exercise 6.5. For each of these three examples plot both the K-function estimates (as in Exercise 6.6) and the envelopes $\min_m \hat{K}_m(r)$ and $\max_m \hat{K}_m(r)$.

6.8 Extensions and literature on point processes

Highly readable general introductions to spatial point processes are given in (Diggle, 2013) now in its third edition, (Baddeley *et al.*, 2015) which also provides R programmes for point process analysis, (Daley & Vere-Jones, 2003),(Daley & Vere-Jones, 2008), and (Illyan *et al.*, 2008). The important class of Markov point processes, which are related to the Markov image models discussed in Chapter 4, are treated in (van Lieshout, 2000) and (Møller & Waagepetersen, 2003). In (Chiu *et al.*, 2013) point processes are discussed in detail but also more general random spatial objects such as, for instance, random closed sets generated by placing closed discs with centers at points in a point process and taking the union these discs. Such objects are also briefly discussed in the following Chapter 7.

Chapter 7

Marked point processes and patterns of randomly placed objects

Point processes are natural building blocks for more complicated spatial processes such as patterns of random objects, for instance disks of random sizes. Let us consider a point process X and associate with each point X_i of X a random mark M_i, which could be the radius of a disk centered at X_i. By letting the mark be a vector with several components we could model more complex objects.

For the 2D gel electrophoresis images in Figures 1.9 and 1.10 we could associate with a protein at position $X_i = [X_{1i} X_{2i}]^T$ the mark $M_i = (S_i, C_i)$, where S_i is the expression level of the corresponding protein and C_i could describe the shape of the spot at X_i. A straightforward model would be to assume that protein molecules are in the first step transported horisontally to a position with mean X_{1i} depending on the molecules pI-value (see example 1.4), and in the second step transported vertically (downwards) by 2D Brownian motion with drift to a position with mean X_{2i} with long transports for small molecules. A simple model would thus be to assume that the spot shape is a two-dimensional normal distribution with 2×2 covariance matrix C_i with means and correlation coefficient zero. The observed pixel grey level Y_x at a pixel with location x could then modeled by

$$Y_x = \sum_i S_i f(x, X_i, C_i) + \epsilon_x, \tag{7.1}$$

where ϵ_x is the observation noise at pixel x and

$$f(x, X_i, C_i) = \frac{1}{2\pi(\det C_i)^{1/2}} \exp(-\frac{1}{2}(x - X_i)^T C_i^{-1}(x - X_i)). \tag{7.2}$$

Looking at Figures 1.9 and 1.10 it is evident that the 2D-normal assumption is clearly not perfect, but anyhow this simple model turns out to be useful s a first step.

For the diffusing particles in Figures 1.13 and 1.14 we could consider a model

$$Y_x = \sum_i f(x, X_i, z_i) + \epsilon_x, \tag{7.3}$$

where again ϵ_x is the observation noise at pixel x, but the mark consists of the scalar z_i representing the vertical position of a particle relative to the focal plain. The function f

may be estimated from data obtained by a special arrangement where one lets particles absorb on a glass surface and the glass surface is then moved step-wise vertically with known distances to the focal plane, see (Kvarnström & Glasbey, 2007) for details.

Similar models could be considered for the aerial photographs in Figures 1.2 and 1.4 where we could assume a similar shape for trees in a given view. This shape function could then be estimated from data combined with a simulation model based on the geometry and illumination of the trees from the sun (Larsen & Rudemo, 1998).

A specific problem is interaction between objects that overlap partly. In 2D gel electrophoresis it is natural to assume an additive model as in (7.1), but in the aerial photographs, and particularly for the diffusing particles, objects may occlude each other and then an additive model may be an untenable approximation. In some applications such as the one shown in Figure 7.1 objects do (essentially) not overlap.

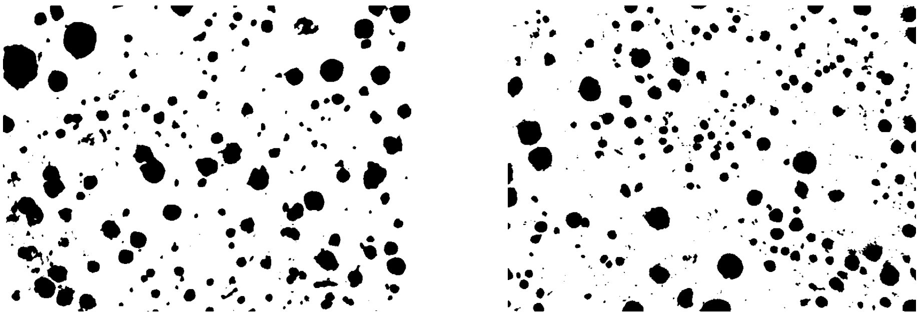


Figure 7.1: Binary images of two cuts in cast iron showing approximately disk-shaped defects. Data from Beretta (2000) and Månsson and Rudemo (2002).

Let us regard models for random placed disks. For disks of constant size we can then use the inhibition point process of Section 6.2 by placing disks of diameter d centered at the points of the thinned point process. In the following section we shall regard two modifications of this model.

7.1 Two processes of varying-sized disks

Let us regard marked point processes constructed in two steps as follows.

In the first step we generate a Poisson point process with constant intensity λ in the plane, and to each point in this point process we generate identically distributed radii with a *proposal* distribution function F_{pr}. The radii are independent mutually and of the point process.

In the second step we thin the generated point process by letting all pairs of points whose associated disks intersect 'compete'. A point is kept if it has higher weight in all pairwise comparisons, where the, possibly random, weights are assigned to the points according to two different approaches:
1) *Pairwise assignment of weights*: For each comparison, weights are assigned to the involved pair of points, and assignments are independent both within and between pairs.

2) *Global assignment of weights*: Weights are assigned once and for all to all points, and assignments to different points are independent. These weights are then used in all comparisons.

In both cases the weight of a point may depend on the associated radius. (When the weights are constant or deterministic functions of the radii, the two approaches coincide.)

It is possible to compute both the intensity of the point process after thinning and the radius distribution function after thinning. Details are given in Månsson and Rudemo (2002). Let us here only show a simulation example of disks before and after thinning with three different thinning procedure, see Figure 7.2.

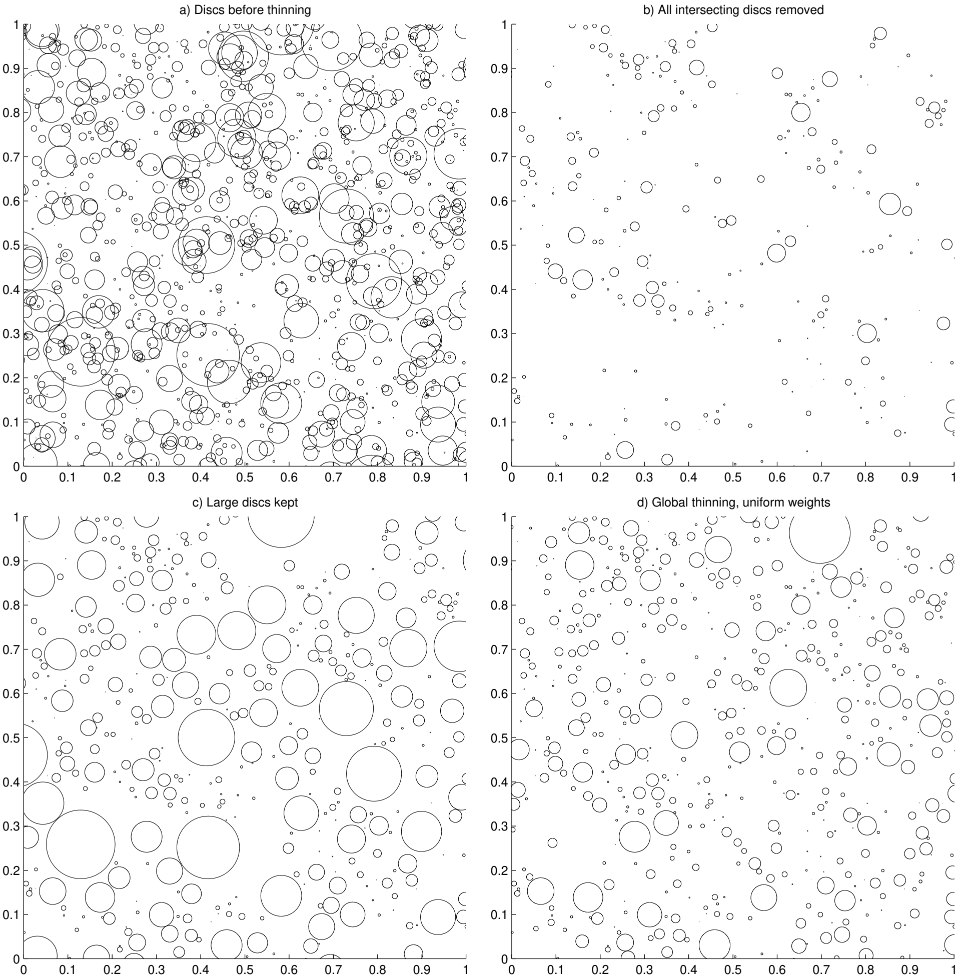


Figure 7.2: Simulation of a disk process before and after three different thinning procedures. In the first step a Poisson process with intensity 1000 in the unit square is generated with exponentially distributed disk radii with expectation 0.01.

Chapter 8

Warping and matching

An important problem in analysis of multiple images is to match objects in different images. Thus we would like to know which spots in the 2D gel electrophoresis images in Figures 1.9 and 1.10 that correspond to each other in order to compare the expression levels of the proteins. Similarly we want to match objects in Figures 1.13 and 1.14 in to order to be able to follow the diffusing particles and to estimate the diffusion coefficient of their motion. There is, however, a fundamental difference between these two problems. The diffusing particles move independently of each other except for the rare occasions when they come very close in all three dimensions. Thus displacements of particles that are close in the two-dimensional images are essentially independent of each other. In contrast, displacements of nearby spots in the electrophoresis images are highly correlated. The matching of objects in these two situations therefore demand quite different methods. In the present section we shall study warping methods which are useful for matching of objects in images such as the 2D gel images.

Suppose that we have a reference image $Y = Y(x)$ and another image Y' that we want to warp (transform) into Y as closely as possible according to some criterion by transforming locations such that $Y(x')$ is close to $Y(x)$. Here we regard x and x' as 2-dimensional column vectors and put

$$x' = f(x) \tag{8.1}$$

for some *warping function* f. The general affine warping function is

$$x' = Ax + b = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{8.2}$$

A special case of the affine transformation is the Procrustes transformation for which

$$x' = \begin{bmatrix} c\cos\theta & c\sin\theta \\ -c\sin\theta & c\cos\theta \end{bmatrix} x + b. \tag{8.3}$$

A special case of the Procrustes transformation consists of a dilation (scale change with a fixed factor c) and a translation

$$x' = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix} x + b = cx + b, \tag{8.4}$$

and another special case of the Procrustes transformation consists of a rotation and a translation,

$$x' = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} x + b. \tag{8.5}$$

A simple nonlinear warping is the bilinear transformation

$$\begin{aligned} x'_1 &= a_{11}x_1 + a_{12}x_2 + c_1x_1x_2 + b_1 \\ x'_2 &= a_{21}x_1 + a_{22}x_2 + c_2x_1x_2 + b_2. \end{aligned} \tag{8.6}$$

We note that for fixed x_2 the bilinear transformation (8.6) is linear in x_1 (with slope and intercept depending on x_2) and, similarly, for fixed x_1 the transformation (8.6) is linear in x_2. This means that an axes-parallell rectangle in the x_1x_2-plane is transformed into a polygon with four sides and four corners in the $x'_1x'_2$-plane (but generally not with pairwise parallell sides).

Another nonlinear warping function is the perspective transformation

$$\begin{aligned} x'_1 &= (a_{11}x_1 + a_{12}x_2 + b_1)/(c_{11}x_1 + c_{12}x_2 + 1) \\ x'_2 &= (a_{21}x_1 + a_{22}x_2 + b_2)/(c_{21}x_1 + c_{22}x_2 + 1). \end{aligned} \tag{8.7}$$

The perspective transformation may be used for matching the tree tops in Figures 1.2 and 1.4. Note that both the bilinear and the perspective transformations are generalisations of the affine transformation (8.2).

To choose parameters of a warping transformation $x' = f(x) = (f_1(x_1, x_2), f_2(x_1, x_2))$ we may consider minimization of a distortion-weighted least squares criterion function such as

$$L(Y', Y, f) = \sum_x (Y'(x') - Y(x))^2 + \lambda D(f), \tag{8.8}$$

where $D(f)$ is a distortion measure of the warping function f, and λ is a non-negative weighting constant determining the balance between closeness of matching and distortion. Let us also note that with normally distributed variables least squares minimization corresponds to log-likelihood maximization, and a method where we use a distortion measure as in (8.8) is often called a *penalized log-likelihood method.* The distortion measure could for instance measure the deviation from linearity of the warping function, and could be a sum of squared second derivatives of f integrated over the region regarded,

$$D(f) = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} \int \left(\frac{\partial f_i}{\partial x_j \partial x_k} \right)^2 dx_1 dx_2, \tag{8.9}$$

where the partial derivatives in computations are approximated by finite differences. The integrals are also approximated by sums over pixels.

A useful type of warping consists af a grid of local bilinear transformations. This method is used in (Glasbey & Mardia, 2001) to warp images of fish, haddock and whiting, into each other. Similarly it is used in Gustafsson et al. (2002) to match 2D gels electrophoresis images such as those in Figures 1.9 and 1.10 into each other, see Chapter 10 below for details. Here we will now describe how handwritten digits can be warped into each other, which will also be used for averaging of the handwritten digit images. Note that simple direct averaging of digits such as those shown in Figure 8.3 will not produce a useful end-result, although such averaging, as we will see, can be used as an initial step.

Example 8.23. *Handwritten digits. Warping and averaging. Classification by minimal warping effort.*

Consider 28×28 images from MNIST and warping of the handwritten digit "2" to the left in the upper row of Figure 8.1 to the digit to the right of it by use of a grid of bilinear transformations shown in Figure 8.2. The grid has 7×7 cells and the weighting constant in (8.8) is $\lambda = 1$. Computations and figures are from (Longfils, 2018), where more details are given, including a discussion of the choice of the grid size and the weighting constant.

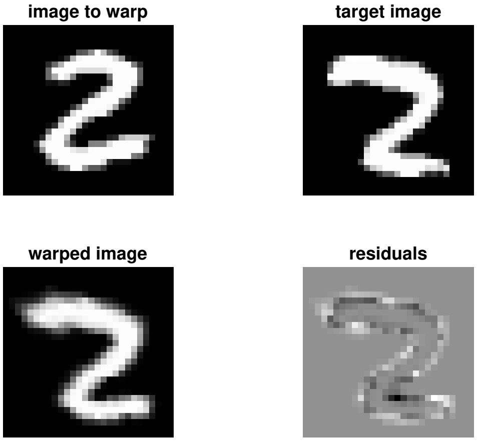


Figure 8.1: Warping of the digit "2" left in upper row to the digit "2" right in the same row. The lower row shows the warped image and the residuals relative to the target upper right.

Let us now consider averaging of handwritten digits of the same type by use of data from MNIST as used earlier in Example 2.17. Thus we have for instance 958 digits "5", compare Table 2.1, of which 100 are shown in Figure 8.3. To find the average handwritten 5-digit we first average all the 958 5-digits. Then we warp all 958 digits separately with the average as target. Then we average the warped 5-digits, warp into the new average and proceed iteratively until changes are sufficiently small. After a few iterations we obtain the average shown in Figure 8.4.

Let us describe how we can use warping techniques to classify images. The method was suggested in (Glasbey & Mardia, 2001) and there used to identify fish species. Consider as before a set of MNIST images, and let $\mu_j, j = 0, \dots, 9$, denote average iteratively warped image for digits j as described above, and where μ_5 is shown in Figure 8.4. To

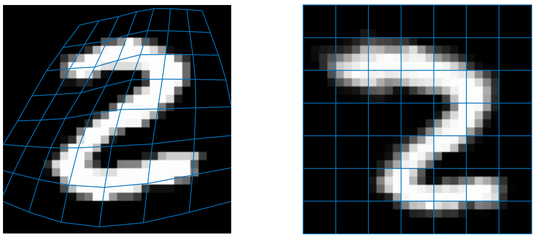


Figure 8.2: Original and warped handwritten digits also shown in Figure 8.1, upper left and lower left, here with the 7×7-grid for the bilinear transformations. The target is the upper right digit in Figure 8.1.

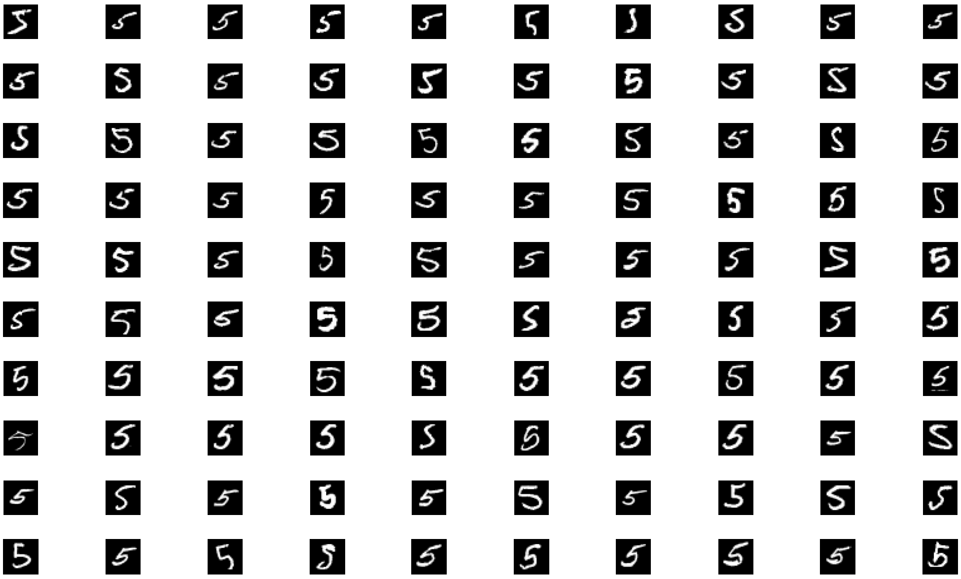


Figure 8.3: First 100 digits “5” in the MNIST database.

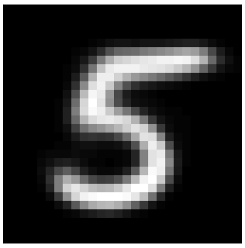


Figure 8.4: Average handwritten digit “5” obtained by sequential warping and averaging.

classify a new image Y, let Y_f denote the image Y warped by the transformation f. Put

$$Q_j = \min_f \left\{ \sum_x (Y_f(x) - \mu_j(x))^2 + \lambda \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \int \left(\frac{\partial f_i}{\partial x_j \partial x_k} \right)^2 dx_1 dx_2 \right\}, \tag{8.10}$$

and classify Y as the digit j for which Q_j is minimal. In Figure 8.5 classification of 197 digits are shown with two fours and four fives miss-classified.

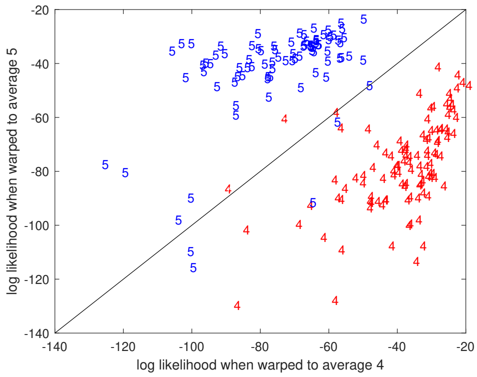


Figure 8.5: Classification of 110 handwritten digits “4” and 87 digits “5” by warping classification. Penalized log-likelihoods for the two types of digits are shown on the axes. Six digits are miss-clasified.

□

For reviews of image warping methods, see (Glasbey & Mardia, 1998, 2001).

PART 3 APPLICATONS

Chapter 9

Analysis of two-coloured DNA microarray images

There are several types of DNA microarrays used to analyze expression levels of genes. We shall here look at a specific type of two-coloured spotted microarrays briefly described in Example 1.5, and look at spot shape modelling and data transformation of microarray data as described in (Ekstrøm *et al.*, 2004). As seen in Figures 1.11 and 1.12 spots are approximately circular with a diameter of about 18 pixels. Let S denote the set of spots, and for each spot $s \in S$ we associate a set A_s of pixels containing the spot approximately in the centre. We can for instance let A_s be a square with side length 24 pixels. The sets A_s and $A_{s'}$ should be disjoint for different spots s and s'.

From Figures 1.11 and 1.12 it is seen that the signal intensity of spots varies from weak to strong. To see details in weakly expressed spots it is useful to increase the photometric gain in the scanning. However, if we increase the gain we can get some pixels in the strongly expressed spots to get saturated, also called censored. One aim in (Ekstrøm *et al.*, 2004) was to to see if one can reconstruct the pixel valued in satured pixels by use of suitable spot shape modelling.

9.1 Data transformations

Let $Z = Z(x)$ denote the intensity of pixel x. For the data in (Ekstrøm *et al.*, 2004) the intensity Z is a 16-bit integer, $0 \leq Z \leq 2^{16} - 1 = 65535$. Let Y denote a transformation of Z. We consider three types of transformations. Firstly, a logarithmic transformation

$$Y = k \log(Z + \lambda_1), \tag{9.1}$$

where λ_1 is a positive parameter; secondly, a Box-Cox transformation

$$Y = \begin{cases} k((Z+\lambda_1)^{\lambda_2} - 1)/\lambda_2 & \text{if } \lambda_2 \neq 0 \\ k \log(Z + \lambda_1) & \text{if } \lambda_2 = 0, \end{cases} \tag{9.2}$$

where $\lambda_1 > 0$; and thirdly, an inverse hyperbolic sine transformation

$$Y = k \text{ arsinh}\left(\frac{Z + \lambda_1}{\lambda_2}\right), \quad \lambda_2 > 0. \tag{9.3}$$

The logarithmic transformation is a special case of the Box-Cox transformation (for $\lambda_2 = 0$). One can show that $\operatorname{arsinh}(z) = \log(z + \sqrt{z^2+1})$ for $z > 0$, and thus for large z we have arsinh $\approx \log(2z)$. We see that for large values of z the logarithmic transformation is thus essentially also a special case of the hyperbolic sine transformation (for $\lambda_2 = 2$).

9.2 Spot shape models

Let us consider a spot s and pixels $x \in A_s$. Let $c_s = (c_{s1}, c_{s2})$ denote the spot centre of spot s, and let $r_s(x) = \| x - c_s \|$ denote the Euclidean distance from the spot centre to the pixel x. Assume that

$$Y(x) = B_s h_s(r_s(x)) + b_s + \epsilon(x), \quad x \in A_s. \tag{9.4}$$

Here B_s measures the intensity of spot s, and this intensity is typically the most important parameter to be estimated for spot s. Further b_s is a background intensity, $h_s(r)$ is a spot shape function assumed to be symmetric around the spot centre, and $\epsilon(x)$ corresponds to zero-mean noise at pixel x. We will assume that noise contributions are normally distributed with constant variance σ_ϵ^2, and to begin with we will also assume that noise from different pixels are independent. Thus we assume that $(Y(x), x \in A_s)$ has a multivariate normal distribution with means

$$\mu_s(x) = B_s h_s(r_s(x)) + b_s, \quad x \in A_s, \tag{9.5}$$

and covariance matrix $\sigma_\epsilon^2 I$, where I is an identity matrix. We consider four different choices of the spot shape function $h_s(r)$:

The cylindrical shape model. Put

$$h_s(x) = \frac{1}{\pi\sigma_s^2} 1(r \leq \sigma_s), \tag{9.6}$$

where $1(P) = 1$ if P is true and $1(P) = 0$ if P is false. The parameter σ_s can be interpreted as the radius of the spot.

The Gaussian shape model. Here

$$h_s(x) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \phi(r/\sigma_s), \tag{9.7}$$

where ϕ is the standardized one-dimensional normal density $\phi(r) = (1/\sqrt{2\pi}) \exp(-r^2/2)$.

The Gaussian difference shape model. Put

$$h_s(x) = \frac{1+\alpha_s}{\sqrt{2\pi\sigma_s^2}} \phi(\frac{r}{\sigma_s}) - \frac{\alpha_s}{\sqrt{2\pi(\beta_s\sigma_s)^2}} \phi(\frac{r}{\beta_s\sigma_s}), \tag{9.8}$$

where $\sigma_s > 0$, $\alpha_s \geq 0$ and $0 < \beta < 1$.

The polynomial-hyperbolic shape model. Here

$$h_s(r) = \begin{cases} \frac{K_s}{\sigma_s^2} \exp(g_s(r/\sigma_s)) & \text{if } 0 \leq r < \gamma_s\sigma_s \\ 0 & \text{if } r \geq \gamma_s\sigma_s, \end{cases} \tag{9.9}$$

with

$$g_s(r) = \sum_{i=1}^{2} b_{si} r^i - \frac{a_s}{\gamma_s - r}, \;\; 0 \le r < \gamma_s, \tag{9.10}$$

where $a_s > 0$ and $\gamma_s > 1$, σ_s represents the radius of the spot, K_s is a normalizing constant and

$$\begin{aligned} b_{s1} &= a_s/\gamma_s^2 \\ b_{s2} &= \tfrac{a_s}{2}\left\{\tfrac{1}{(\gamma_s-1)^2} - \tfrac{1}{\gamma_s^2}\right\}. \end{aligned}$$

Some spot-shape parameters may be common for all spots and some may be spot-specific.

9.3 Maximum likelihood estimation

To estimate parameters in the spot shapes and the transformations we use the maximum likelihood method. Let us first assume that there are no saturated pixels, that is all pixel-values are below the maximum level, which is $2^{16} - 1$ before data transformation. Then the log-likelihood for the Y-values in the neighbourhood A_s of spot s is

$$L_Y = \sum_{x \in A_s} \log \left\{ \frac{1}{\sigma_\epsilon} \phi \left(\frac{Y(x) - B_s h_s(r_s(x)) - b_s}{\sigma_\epsilon} \right) \right\}. \tag{9.11}$$

Let us now assume that there are some saturated pixel-values, and let ℓ_c denote the saturation level for the Y-values. Thus if $Y(x) < \ell_c$ we know the value $Y(x)$ but otherwise we only know that $Y(x) \ge \ell_c$. Let $A'_s = \{x \in A_s : Y(x) < \ell_c\}$ and $A''_s = \{x \in A_s : Y(x) \ge \ell_c\}$ denote the set of pixels that are unsaturated and saturated, respectively. Then we find that the log-likelihood becomes

$$L_Y = L_1 + L_2, \tag{9.12}$$

where

$$L_1 = \sum_{x \in A'_s} \log \left\{ \frac{1}{\sigma_\epsilon} \phi \left(\frac{Y(x) - B_s h_s(r_s(x)) - b_s}{\sigma_\epsilon} \right) \right\} \tag{9.13}$$

and

$$L_2 = \sum_{x \in A''_s} \log \left\{ 1 - \Phi \left(\frac{\ell_c - B_s h_s(r_s(x)) - b_s}{\sigma_\epsilon} \right) \right\}, \tag{9.14}$$

where Φ denotes the distribution function of the standardized one-dimensional normal distribution.

In Figure 9.1 original data (one-dimensional profiles through spot middle) and model fits for one specific spot and the four spot shape models are shown. It is seen that the first and particularly the fourth model seem to give considerably better fits compared to the second and the third models. The original data and the fit for the polynomial-hyperbolic model (9.9) are shown in more detail in Figure 9.2 for the same spot as in Figure 9.1.

Let us now look at a simultaneous comparison of transformations and spot shape models by use of maximum likelihood estimation. Results are shown as median differences of log-likelihoods relative to the best model fit in Table 9.1 for 25 spots and four different

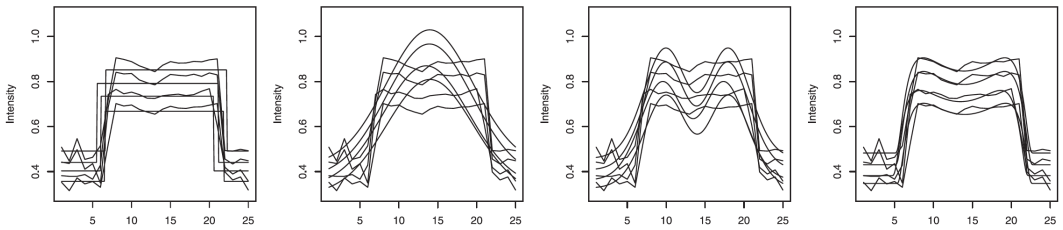


Figure 9.1: One-dimensional intensity profiles (through spot center) for observed intensities of one spot, four photometric gains and maximum likelihood fits for the four spot shape models (9.6), (9.7), (9.8) and (9.9).

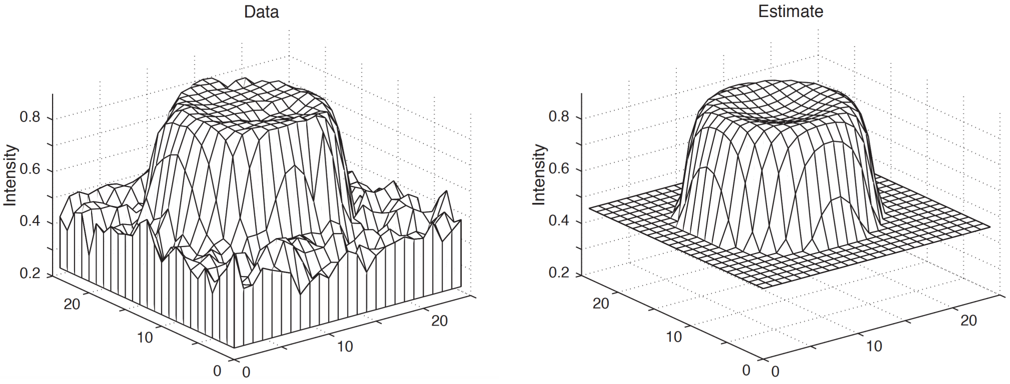


Figure 9.2: Three-dimensional plot (for one photometeric gain) of observed intensities (left surface) for the same spot as in Figure 9.1 and the corresponding estimated spot shape for the polynomial-hyperbolic shape model (right surface).

photometric gains in the scanning. The 25 spots were selected to represent both low, median and high intensity levels. We see that the polynomial-hyperbolic model is the best spot shape model followed in order by the cylindrical, the Gaussian difference and the Gaussian model, which is also clearly indicated in Figure 9.1. The best combination is the Box-Cox transformation together with the polynomial-hyperbolic spot shape model.

Table 9.1: Median decrease in log-likelihood for 25 spots and four gains relative to the polynomial-hyperbolic spot shape model with the Box-Cox transformation

	Spot shape model			
Transformation	Cylindrical	Gaussian	Gaussian difference	Polynomial-hyperbolic
Logarithm	136.3	329.6	185.4	17.0
Arsinh	127.2	258.7	144.4	13.9
Box-Cox	134.3	320.3	178.2	0.0

As mentioned in the second paragraph of this chapter one of the aims of (Ekstrøm *et al.*, 2004) was to reconstruct values in saturated pixels. In Figure 9.3 we show how artificially saturated levels can be reconstructed for one spot.

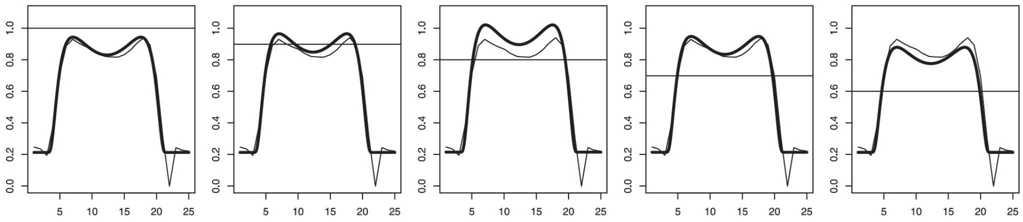


Figure 9.3: One-dimensional intensity profiles through the center of one spot together with reconstructions by use of the polynomial-hyperbolic spot shape model for different levels of artificial saturation indicated by horizontal lines. Both data (thin curves) and reconstructions (heavy curves) are shown for each saturation level.

9.4 Models with dependent pixel residuals

Up till now we have regarded residuals $\epsilon(x), x \in A_s$, in (9.4) as independent. However, a closer look at the left part of Figure 9.2 indicates that residuals at least for adjacent pixels seem positively correlated.

Following (Ekstrøm *et al.*, 2005) let us assume that the vector Y with components $Y(x), x \in A_s$, has a multivariate normal distribution, $Y \sim N(\mu, \sigma_\epsilon^2 R)$, where μ as before has components $\mu(x) = B_s h_s(r_s(x)) + b_s, x \in A_s$, but R, instead of being an identity matrix, corresponds to an isotropic correlation function. Thus we assume that

$$\mathrm{cov}(Y(x), Y(x')) = \sigma_\epsilon^2 \rho(r, c), \tag{9.15}$$

where $r = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2}$ is the Euclidean distance between $x = (x_1, x_2)$ and $x' = (x_1', x_2')$ and c is a real (positive) parameter. We consider five different correlation functions:

Let us further choose the Box-Cox transformation and the polynomial-hyperbolic spot shape model. To estimate parameters including the parameter c for the different correlation function by maximum likelihood we have to maximize the log-likelihood

$$\log L = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log(\det C) - \frac{1}{2}(Y-\mu)^T C^{-1}(Y-\mu), \tag{9.16}$$

where n is the number of pixels, μ contains parameters for the spot shape and $C = \sigma_\epsilon^2 R$ contains the correlation function parameter c for the different correlation functions considered. The computations turn out to be considerably more complicated compared to the independent residuals model, see (Ekstrøm *et al.*, 2005) for details.

The resulting log-likelihood improvements compared to the independent residuals model are shown in Table 9.2. The fit of the different correlation functions are further illustrated in Figure 9.4. We see that the two correlation structures that give the best fit in Table 9.2, that is the Gaussian and the spherical correlation, also give the best agreement with the empirical correlation coefficients in Figure 9.4.

Table 9.2: Median improvement in log-likelihood for 25 spots and four gains relative to the model with independent residuals for five models with residual correlation

Correlation structure	Exponential	Gaussian	Linear	Rational quadratic	Spherical
	69	82	73.5	75	78

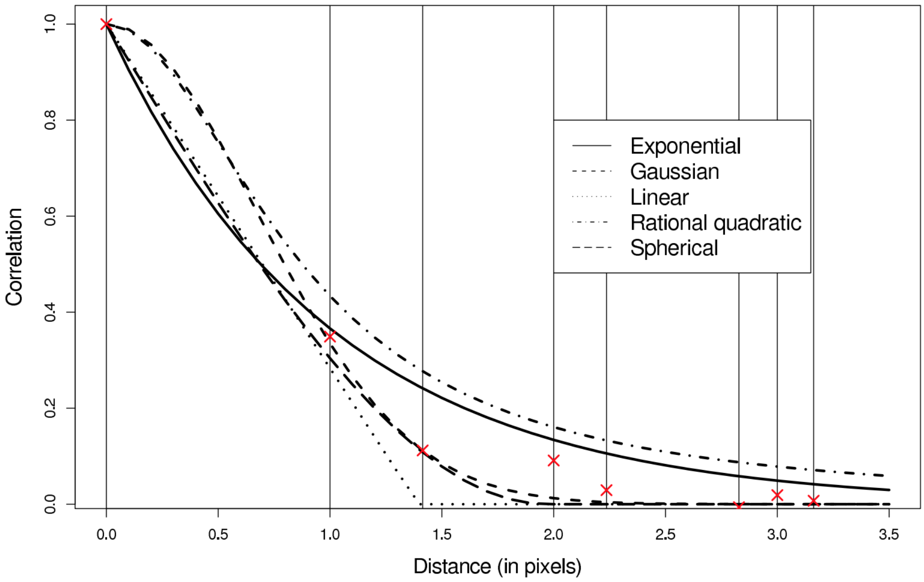


Figure 9.4: Median estimated correlation functions for the five studied correlation structures. The possible observable distances between pixel centres are shown by vertical lines and the crosses on these lines show the median empirical correlation coefficients.

9.5 Exercises

Exercise 9.1. Check that the spot shape functions (9.6), (9.7) and (9.8) satisfy $\iint h(x)\,dx_1\,dx_2 = 1$, where $x = (x_1, x_2)$ and the integral is taken over the entire two-dimensional space. (The same relation holds for (9.9), but that is a bit more complicated to show.)

Exercise 9.2. Describe how the reconstructions (heavy curves) in Figure 9.3 can be computed.

Exercise 9.3. What details in Figure 9.2 should one look at to get an indication of that residuals for adjacent pixels are positively correlated?

Exercise 9.4. In Figure 9.4 there are computations for the seven smallest inter-pixel distances (marked by crosses). Describe how pairs of pixels are located to achieve these distances. One distance corresponds to a knight move in chess; which distance is that?

Chapter 10

Two-dimensional electrophoresis

Two-dimensional electrophoresis is an experimental technique that can be used to measure the expression of up to several thousands of proteins, compare Example 1.4 with Figures 1.9 and 1.10. In this chapter we shall describe techniques from (Gustafsson *et al.*, 2002) based on warping and matching of such images. The image data in (Gustafsson *et al.*, 2002) consist of five images similar to Figure 1.9 from 2D gel electrophoresis of baker's yeast grown in a standard solution and five images similar to Figure 1.10 from 2D gel electrophoresis of baker's yeast grown under stress in a solution with salt added.

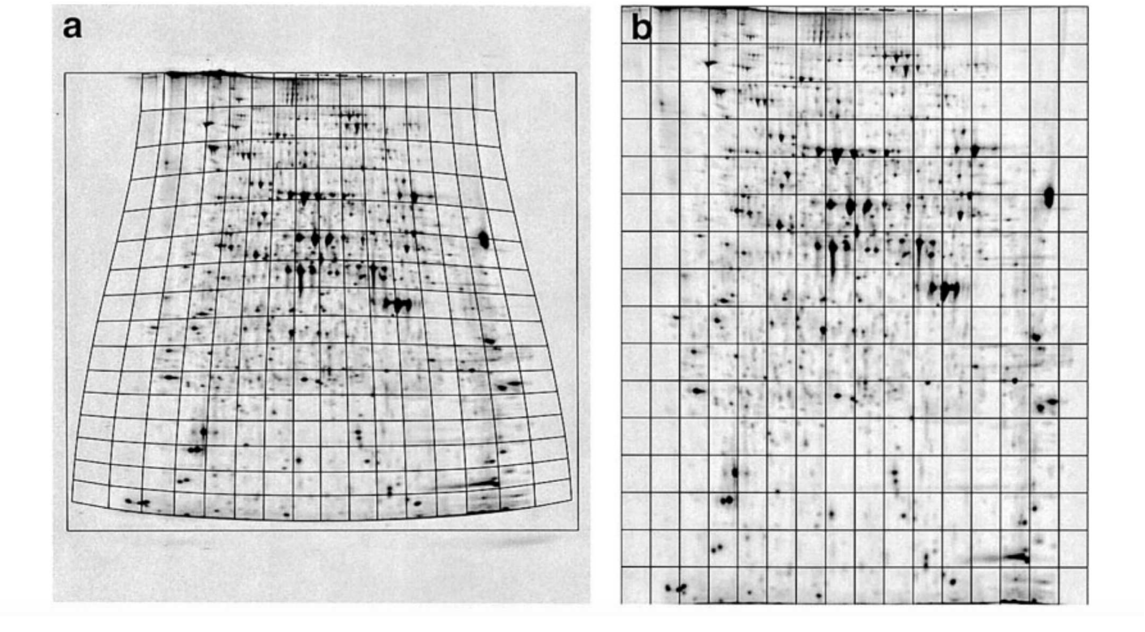


Figure 10.1: Illustration of warping step I with correction for current leakage sideways through the left and right boundaries during the second-dimensional gel electrphresis. Part **a** of the figure shows the original image and part **b** shows the warped current-leakage corrected image.

The warping in (Gustafsson *et al.*, 2002) consists of two steps. As described in Example 1.4 images are obtained by first letting protein molecules move horizontally along a string to a position determined (except for random noise) by the protein isoelectric point pI. In the next step, the second-dimensional gel electrophoresis, a polyacrylamide gel is cast between two glass plates separated from each other by thin plastic spacers and placed vertically in a bath. The protein string is placed horizontally on the top of the polyacrylamide gel. A voltage is applied between the upper and the lower boundaries of the plates and the proteins perform a Brownian motion with downwards vertical drift in the bath. The vertical distances traveled by the protein molecules are determined (except for random noise) by the protein mass. During this second step there may be current leakage sideways, and the first warping step in (Gustafsson *et al.*, 2002) models this by solving a partial differential equation with suitable boundary conditions taking care of current leakage. The result of the warping is illustrated in Figure 10.1, and we refer to (Gustafsson *et al.*, 2002) for further details of this warping step. After the first warping step two image transformations are applied. Firstly, to compensate for large scale trends in the background level, a top-hat transformation is applied, see (Glasbey & Horgan, 1995) for a description of the top-hat transformation and (Gustafsson *et al.*, 2002) for parameter values used in the transformation. Secondly, a logarithmic transformation of pixel values is applied.

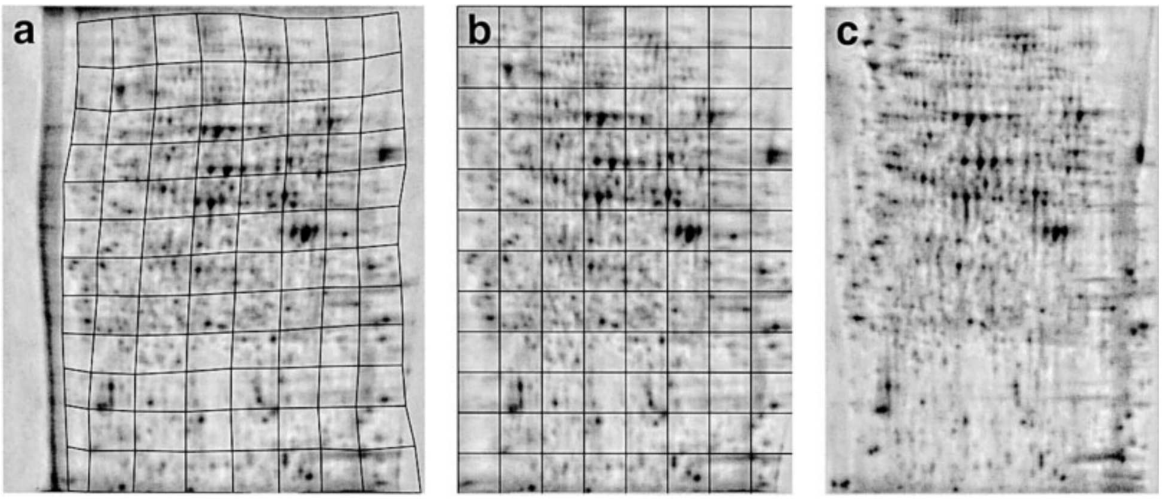


Figure 10.2: Illustration of warping step II. The image in **a** is warped onto the reference image in **c** by use of the grid shown in **a** warped to the grid in **b**.

In the second warping step images are transformed by use of a grid of bilinear transformations similar to the warping of handwritten digits shown in Figure 8.2. The result of such a warping is shown in Figure 10.2. One of the five images for yeast grown under standard conditions is used as a reference image, and the other nine images are warped onto this reference image. We use a penalized log-likelihood method and minimize a criterion function such as (8.8) with $D(f)$ given by (8.9). Thus we minimize with respect to f the criterion function

$$L(Y', Y, f) = \sum_x (Y'(x') - Y(x))^2 + \lambda \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} \int \left(\frac{\partial f_i}{\partial x_j \partial x_k} \right)^2 dx_1 dx_2, \qquad (10.1)$$

with $x' = f(x)$ and where we sum over pixels x. The partial derivatives in computations are approximated by finite differences, and the integrals are approximated by sums over pixels.

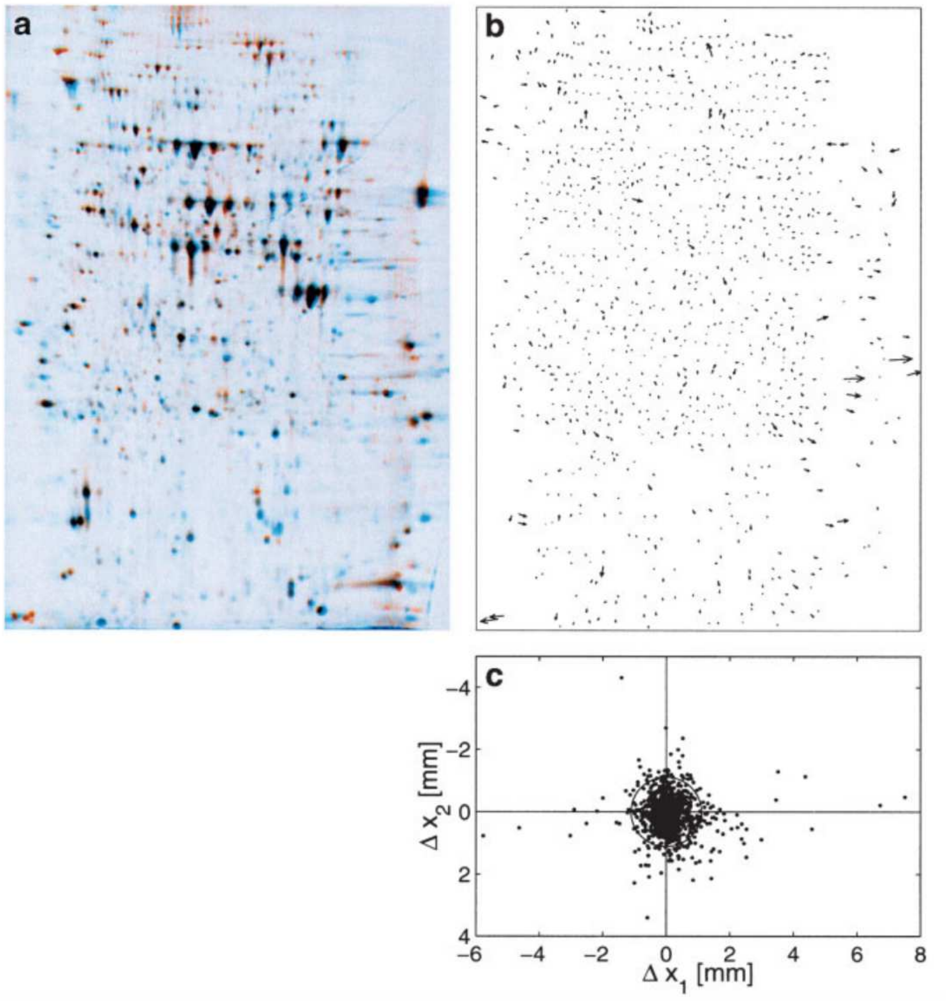


Figure 10.3: Further illustration of warping step II. In part **a** the reference image coloured red and the warped image coloured blue are superimposed. Displacement vectors for spots are shown in part **b**, and also in part **c**, here as relocated vectors starting at the origin and ending at dots. In **c** we also show a criterion for adjacency of spot pairs: adjacent spot pairs have dots within the circle shown.

The second warping step is further illustrated in Figure 10.3. Here we show in part **a** of the figure a superposition of the reference image coloured red and the warped image coloured blue. For protein spots that are equally expressed in both images we should then ideally get black spots. However if the warping is less perfect we expect adjacent spots coloured red and blue. (Further even if the warping is perfect we can get spots that are predominantly blue or predominantly red for a protein that is differently expressed in the two images.) In part **b** of Figure 10.3 spot displacement vectors are shown, and for more clear illustration arrow heads are large for large displacements. We see that large displacements mainly occur close to the boarders. Spot displacement vectors are also shown in part **c** of the figure, and here all the displacement vectors are relocated so that they start in the origin and end in positions shown as dots.

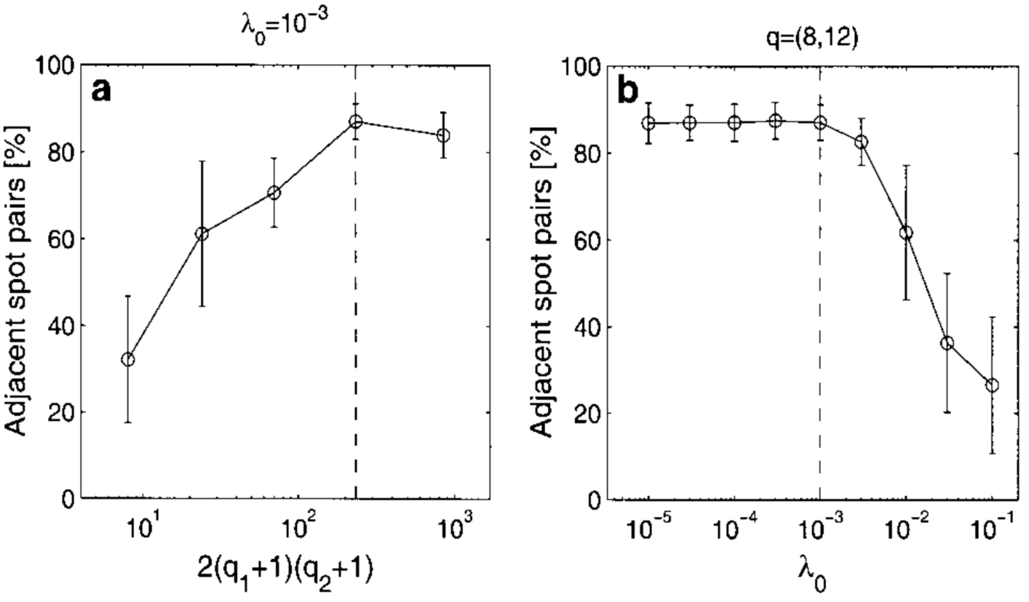


Figure 10.4: Illustration of spot pattern similarity in aligned images. The left part **a** shows the effect of changing grid size for the particular λ-value 10^{-3}. The graph shows the percentage of adjacent spot pairs as a function of the number of grid size parameters. The right part **b** shows the effect of changing the log-likelihood penalizing parameter λ for the particular grid $q = (8, 12)$, and the graph shows the percentage of adjacent spot pairs as a function of λ. Circles show mean values and error bars show standard deviations for the nine images aligned to the reference image. Vertical dashed lines show the finally chosen grid size and likelihood penalty weight.

Two crucial issues are choice of how fine the grid in the bilinear transformation net should be and the size of the non-negative parameter λ in the penalization of the likelihood in (10.1). If we start with a course net and steadily refine it we can expect the fit to improve but to level off at a certain degree of fineness. Similarly if we start with a large λ-value and then decrease λ we can expect an improvement in fit but similarly a leveling of at some point. As a measure of fit we use the percentage of spot pairs with dots inside the circle in **c** of Figure 10.3. We specify the net grid by $q = (q_1, q_2)$, where q_1 and q_2 are the number of rectangles in the horizontal and the vertical directions. We note that in Figure 10.2 we have $q = (8, 12)$. It turns out that the number of parameters in a grid specified by $q = (q_1, q_2)$ is $2(q_1 + 1)(q_2 + 1)$. We use a sequence of grids with q equal to: $(1, 1)$, $(2, 3)$, $(4, 6)$, $(8, 12)$ and $(16, 24)$. Similarly we use the following sequence of λ-values: $30\lambda_0$, $10\lambda_0$, $3\lambda_0$, λ_0 and $0.3\lambda_0$, with $\lambda_0 = 10^{-3}$. Results from some computations with different grid sizes and different λ parameters are shown in Figure 10.4. The chosen grid size is $q = (8, 12)$, and the chosen λ-value is $\lambda_0 = 10^{-3}$.

The two warping steps are compared in Figure 10.5, which shows the length distribution of spot displacement vectors for three sets of images: the original images, the

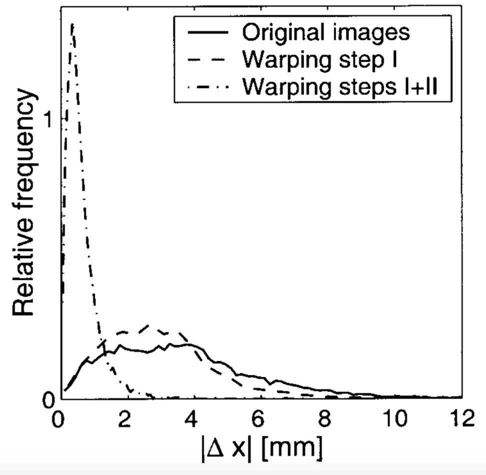


Figure 10.5: Length distribution of spot displacement vectors for the original data (solid line), after the current leakage warping step (dashed line) and after both warping steps (dash-dot line).

current leakage corrected images (only warping step I) and the current leakage corrected and aligned images (warping steps I and II). From the figure it is clear that warping step I gives some improvement, but the large improvement is obtained with the combination of both warping steps. In (Gustafsson *et al.*, 2002) there is also a comparison of warping I+II with the use of only warping step II. It turns out that beside a slight improvement in the percentage of adjacent spot pairs, an effect of warping step I is a considerable reduction of the total computation time.

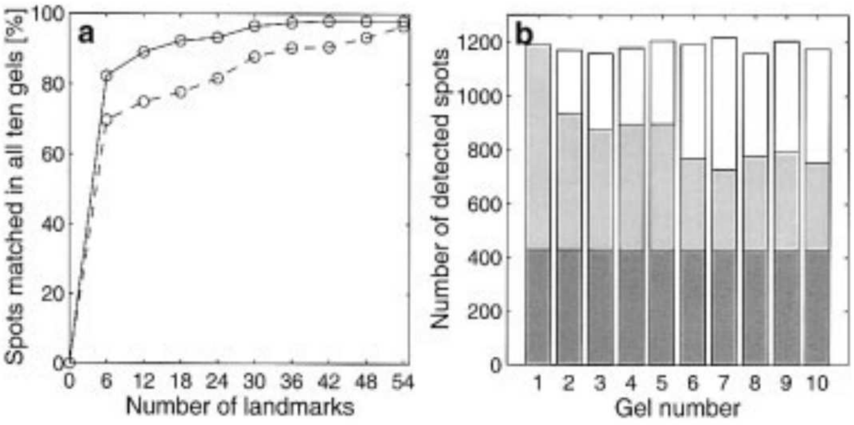


Figure 10.6: Efficiency profiles in the left part **a** showing the number of automatically matched spots in all ten gels (with gel images two-step warped) by the software PDQuest as a function of an initial manual matching of a number of spots (in the image called landmarks) both for the original set of images (dashed line) and for the set of warped images (solid line). The right part **b** of the figure shows the number of detected spots in the ten gels for the warped gel images. The spots detected in all gels are shown dark grey, the spots found additionally in common with the reference gel 1 is shown for each gel in light grey, while detected spots not in common with the reference gel are shown in white.

Figure 10.6 illustrates the improvement in matching efficiency when the warped images are used together with the PDQuest software (Garrels, 1989). In the method illustrated in the figure the reference image is divided into 54 subrectangles and in each subrectangle the most intense spot is chosen. The chosen spots are ordered according to intensity and an increasing number of theses spots are manually matched. Based on this manual matching the software PDQuest then automatically matches other spots. The left part **a** of the figure shows the global matching efficiency as the number of automatically matched spots found in all ten gel images as a function of the number of manually found spot pairs. The dashed line shows the efficiency profile for the original images and the solid line shows the efficency profile with warped images (using two-step warping). A clear improvement using warping can be seen (compare Exercise 10.2 below).

In part **b** of Figure 10.6 we see bars showing the number of spots detected in the ten gels. Here gels 1–5 are gels with yeast grown in standard solution (including the reference gel 1) and gels 6–10 are gels grown with salt added. The mean number of gels detected in all ten gels is 1194, and the average number of detected spots in common with the reference gel (for gels 2–9) is 826, while the number of spots detected in all ten gels is 430.

10.1 Exercises

Exercise 10.1. As mentioned above a top-hat transformation was used after the first warping step to compensate for large-scale trends in the background level. Describe briefly how alternatively a low pass filtering technique could be used for that purpose.

Exercise 10.2. Determine approximately (both for the original image set and for the set of warped images) from Figure 10.6 the number of manually matched spots needed to achieve subsequently in the automatic step a 90% spot number matching in all ten gels.

Exercise 10.3 In part **b** of Figure 10.6 gels 1–5 correspond to yeast grown in standard conditions (including the reference gel 1) and gels 6–10 correspond to yeast grown in a salt solution. What are the general features of the fluctuations of the light grey bars? Give also an explanation of these general features.

Chapter 11

Point processes observed with noise, two examples with aerial photographs of forests

11.1 Estimation of tree top and tree base positions from aerial photos

Here we consider how we can estimate tree base positions from images such as the image, below called Image 148, shown in the right part of Figure 1.1. A detail of Image 148 is shown Figure 1.2. We follow the exposition (Dralle & Rudemo, 1997) and start by estimating the tree top positions using a Gaussian filter as described in Chapter 1, see Figures 1.16 and 1.17. Let $X = (x_1, \ldots, x_n)$ denote the tree base positions, and let $Y = (y_1, \ldots, y_m)$ denote the positions of maxima after the Gaussian filtering. The object is to estimate X from Y and also to find the correspondence between y- and x-points. We will start by going the other way and estimate Y from X. If we know the positions of tree stems at ground level we can estimate the positions of tree tops as indicated in Figure 11.1. The white linear segments in this figure show the expected positions of tree trunks projected on ground as seen from the aeroplane with tree ground position at the segment end point closest to the nadir point (the point vertically below the aeroplane) and the tree top position at the segment end point furthest away from the nadir point. The model used here for the tree top positions is based on field measurements of the tree ground positions and the diameter at breast height (1.3 m), and additionally a regression of tree height from breast height diameter, which is a well-known method of estimating tree height in forestry.

In practice it is tedious to measure tree ground positions and the object of (Dralle & Rudemo, 1997) is to use aeroplane photographs to estimate first tree top positions and then tree ground positions. The model used is based on the assumption of three sources of distortion:

1. some trees are lost (errors of omission),

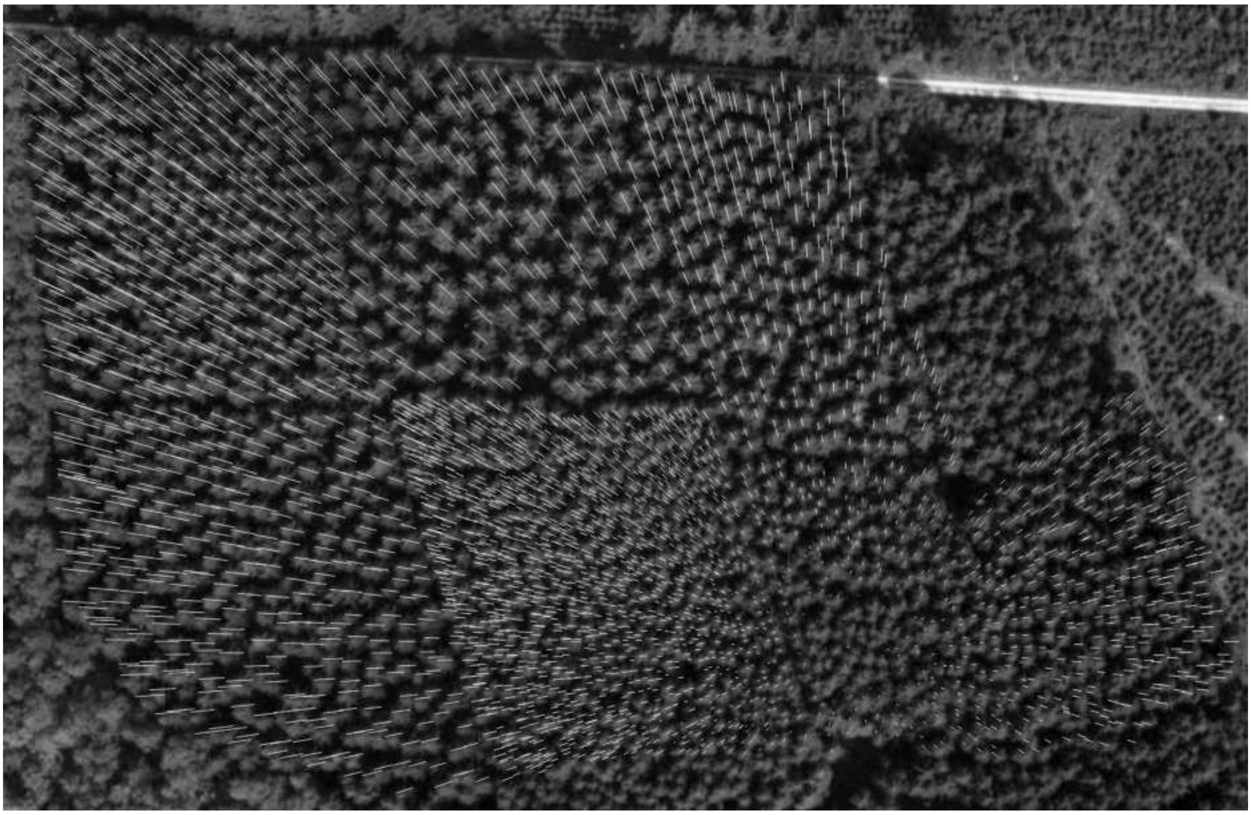


Figure 11.1: The Image 148 with tree projections superimposed, as seen from the camera position. The tree projections were computed from the tree ground level location measurements and the tree heights estimated from height-diameter regression and breast diameter field measurements, see the text for further details.

2. the remaining trees become displaced as a consequence of image geometry and lighting conditions; the displacement of a point $x_i = (x_{i1}, x_{i2})$ is composed of a systematic displacement from x_i to x'_i and a random displacement from x'_i to $x'_i + z_i$,
3. some spurious maxima that do not correspond to real treas are generated (ghost treas, errors of commission).

In the model we will make the simplifying assumption that these three mechanisms are mutually independent, and further that within each of these three categories the trees behave independently of each other. More specically, we assume:

1. For each tree there is a probability θ_0, depending on the thinning treatment, that the tree gives rise to a maximum. Thus the probability of an error of omission is $1 - \theta_0$ for each tree, and the events that different trees are omitted are independent.
2. The systematic displacement to x'_i, see Figure 11.2, of the base location x_i of a tree is obtained by two displacements in the horizontal plane, or, eqivalently, in the image plane. Move first along the projection of the tree a distance $\theta_1 p_i$, where p_i is the projection length, and move then orthogonally in the horizontal plane (to the same side of the tree projection as the sun) a distance $\theta_2 h_i \sin \alpha_i$. Here h_i is the height of a tree and α_i is the angle between the horizontal projection of the tree and a line which is the intersection of the horizontal plane and a vertical plane containing both x_i and the sun. The subsequent random motion $z_i = (z_{i1}, z_{i2})$ in the local coordinate system with one axis parallel to the tree projection and one axis orthogonal to it has a two-dimensional normal distribution with means zero, standard deviations σ_1 and σ_2 and correlation coefficient ρ.
3. Spurious maxima are generated by a Poisson process with the intensity λ maxima per hectare.

For the systematic displacement from x_i to x'_i and for a corresponding maximum $y_{j(i)}$ in the smoothed image we thus assume

$$x'_i = x_i + \theta_1 p_i e_{i1} + \theta_2 h_i \sin \alpha_i \, e_{i2} \tag{11.1}$$

and

$$y_{j(i)} = x'_i + z_i = x'_i + z_{i1} e_{i1} + z_{i2} e_{i2}, \tag{11.2}$$

where z_{i1} and z_{i2} are random errors and e_{i1} and e_{i2} are unit vectors, see Figure 11.2.

We assume that the parameters θ_0, θ_1, θ_2, σ_1, σ_2, ρ and λ are constant within subplots. In the discussion in the text close to Table 11.1 below we will see that some of the parameters vary between subplots in a way that may be interpreted in terms of thinning treatments and the geometry at image acquisition.

The parameters are estimated iteratively, and for more details including choice of initial values for parameters see (Dralle & Rudemo, 1997). For trees in a polygonal area A we compute a displaced area A_d, compare Figure 11.3, by moving each border polygon corner point according to the transformation (11.1) as if the border point was the ground location of a tree with a height computed from the height-diameter regression when

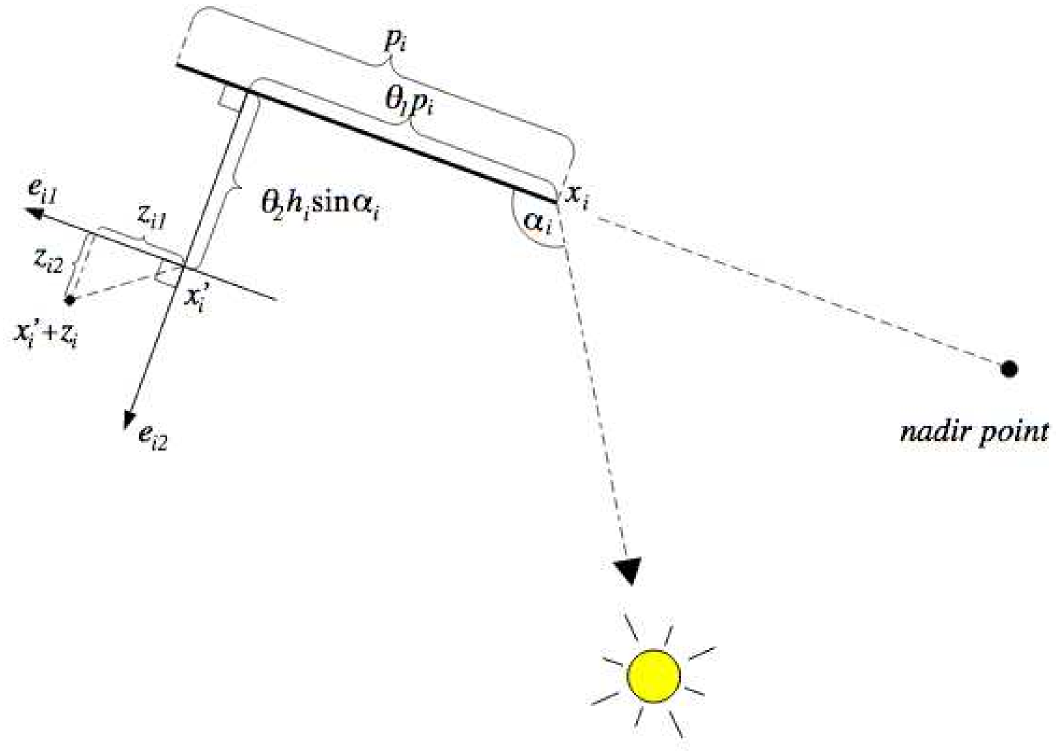


Figure 11.2: Displacement model for the positioning of trees. The full-drawn thick line represents the ith tree stem projected, as seen from the camera, onto the image with the base position x_i nearest to the nadir point. The length (in pixel units) of the projection is denoted p_i, and the height of the tree (also in pixel units) is h_i. The systematic displacement takes x_i to the expected position x_i' for the grey-level maximum and an additional random displacement gives the observed location $x_i' + z_i$ of a corresponding maximum. The coordinates of z_i are assumed to have a two-dimensional normal distribution with zero means.

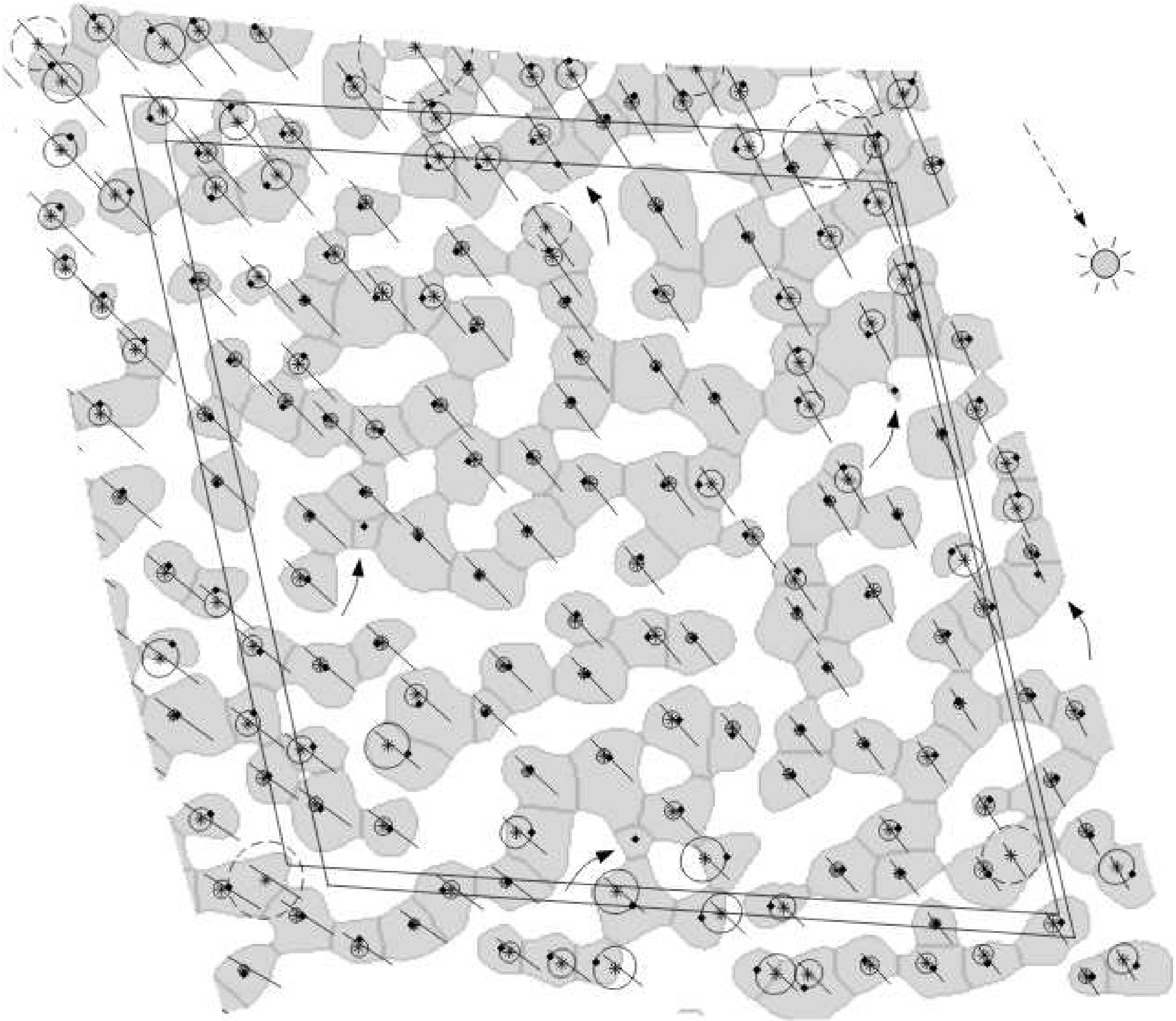


Figure 11.3: Subplot D in Image 148 with the net subplot borders (lower right quadrilateral) and the corresponding displaced area where maxima are expected (upper left quadrilateral). The local maxima after smoothing with the optimal bandwidth are shown as small black squares (diamonds), and for each local maximum the corresponding "watershed" segment above median grey level is shown in light grey colour with borders between segments in slightly darker grey colour. Tree projections, as seen from the camera, based on ground measurements are shown as line segments and expected positions for local maxima, according to the model indicated in Figure 11.2, as stars. From each star an ellipse is grown until it hits a local maximum. The ellipse is dashed if this maximum has already been hit by a smaller ellipse from another star. Thus stars with a dashed ellipse represent errors of omission, while small squares not hit by an ellipse (these squares have pointers to them) represent errors of commission. The sun azimuth is marked in the upper right corner of the image.

the tree diameter corresponds to mean trea diameter of the subplot. Here the current estimates of θ_1 and θ_2 are used.

For a tree with base x_i we compute the position x'_i according to (11.1) and around this point we let an ellipse with constant probability density according to the model (11.2) grow. Here the current estimates of σ_1 and σ_2 are used but with $\rho = 0$. Ellipses grow simultanously at the same rate around all points $x'_i, i = 1, \ldots, n$, for the trees in the regarded area, or, actually, in a slightly larger area as shown in Figure 11.3. As soon as an ellipse starting from x'_i catches a maximum in the smoothed image the growth of that ellipse is stopped. If this maximum has not been catched from another point earlier the maximum (at $y_{j(i)}$) is associated with the ith tree with base location x_i. (If the maximum has already been catched from another tree, no maximum is associated with the ith tree; it is considered lost.)

Let $x_i, i \in I'$, be the set of base locations in A for trees that catch maxima, and let $n' \leq n$ be the number of elements in this set. As an estimate for the probability θ_0 that a tree gives rise to a maximum (one minus the probability of omission), put

$$\hat{\theta}_0 = n'/n. \tag{11.3}$$

Using (11.1) and (11.2) estimates for θ_1 and θ_2 are obtained by coordinate-wise linear regression analyses along the e_{i1}- and e_{i2}-axes, respectively, for $i \in I'$. Corresponding estimates for σ_1^2, σ_2^2 and ρ are obtained as the sample variances and the sample correlation for the set of the n' two-dimensional residuals $y_{j(i)} - x'_i, i \in I'$.

An estimate for the density of spurious maxima (errors of commission) is

$$\hat{\lambda} = (m - m')/|A|, \tag{11.4}$$

where m is the number of maxima in A_d, $m' = n'$ is the number of maxima in A_d that are caught by trees, and $|A|$ is the area of A.

The estimation procedure is performed for each of the subplots D, R, DB and B and also in groups 'all except B' and 'all', and the results are shown in Table 11.1, where also the estimate of the root-mean-square random displacement in metres

$$\sigma = 0.15(\sigma_1^2 + \sigma_2^2)^{1/2} \tag{11.5}$$

is given.

From Table 11.1 we see that for medium and heavy thinning, around 95% of the trees are found with a root-mean-square residual error in the displacement model of about 60 cm or less, and for light thinning around 85% of trees are found and positioned with an error of about 75 cm. The unthinned control was not investigated here because this treatment gives an exceptionally dense population, and a large number of trees are suppressed which are not possible to see from above.

One could try to use a maximum likelihood method corresponding to our statistical model (11.1) - (11.2) for the present data set with both the image and the ground truth available. However, a straightforward computation that takes all possible correspondences between the set of maxima and the set of trees is prohibitive, as the number of such correspondences is astronomical. One may concentrate on a small number of 'probable'

Table 11.1: Parameter estimates for subplots in Image 148 with five different treatments. N is the true stem number per hectare; θ_0 is the probability that a tree gives rise to a maximum (and $\hat{\theta}_0$ the corresponding parameter estimate) ; θ_1 and θ_2 specify the systematic displacement from the base location x_i to x_i' at which the corresponding intensity peak is expected (Figure 11.2); σ_1 and σ_2 (in pixel units corresponding to 15 cm at ground level) and ρ are parameters in a two-dimensional normal distribution for the random displacement z_i from the expected to the observed location (Figure 11.2); λ is the expected number of spurious maxima per hectare; σ is the root-mean-square random displacement in metres.

Subplot	N	$\hat{\theta}_0$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\sigma}_1$	$\hat{\sigma}_2$	$\hat{\rho}$	$\hat{\lambda}$	$\hat{\sigma}$
D	367	0.970	0.651	0.028	2.74	2.94	0.370	15	0.60
C	625	0.971	0.731	0.056	2.48	1.69	0.088	37	0.45
R	746	0.980	0.634	0.082	3.20	2.12	-0.313	15	0.58
DB	824	0.956	0.767	0.006	2.69	2.19	-0.219	40	0.52
B	1257	0.843	0.871	0.045	4.29	2.65	-0.035	168	0.76
All except B		0.969	0.730	0.046	3.23	2.76	-0.096	26	0.64
All		0.925	0.734	0.045	3.61	2.75	-0.071	55	0.68

correspondences. In (Dralle & Rudemo, 1997) we have been even more reductionistic, considering only one such correspondence. After establishing the correspondence, the subsequent parameter estimation is straightforward, particularly if we assume that the correlation between errors along the tree projection and orthogonal to it is zero. A likelihood-based analysis with a limited number of 'most probable' correspondences is given in (Lund & Rudemo, 2000).

11.2 Optimal templates for finding tree tops in aerial photos from different angles

In the previous section we analyzed aerial photos acquired essentially vertically above the area studied. Following (Larsen & Rudemo, 1998) we shall in this section study photos obtained from varying angles, in particular three specific cases where the trees are sidelighted, backlighted and frontlighted.

We shall use a tree model from (Larsen, 1997), which extends a model in (Pollock, 1994). The tree is modelled as a generalised ellipsoid that in (x, y, z) coordinates has the surface

$$\frac{(z^2)^{n/2}}{a^n} + \frac{(x^2+y^2)^{n/2}}{b^n} = 1, \tag{11.6}$$

where z is the vertical coordinate, the "centre" of the tree crown is at the origin, a is half the length of the ellipsoid, b is half the width and n is a shape parameter; here we use $a = 17.7$ m, $b = 2.84$ m and $n = 1.6$ as in (Larsen, 1997). The tree model is shown in Figure 11.4, where we also show how light from the sun is scattered by single reflection into the camera on board the aeroplane.

We will study an algorithm for automatically selecting tree tops from images and compare them with manually obtained "ground truth" data. Let us first consider manually estimated "ground truth" tree top positions. As described in Dralle & Rudemo (1997), see Section 11.1, we can from known tree base positions estimate the tree top positions according to (11.1). The tree top positions thus obtained were then manually corrected by inspection of the photos to compensate for errors in tree height estimates, variations due to wind, and imprecision in image rectification.

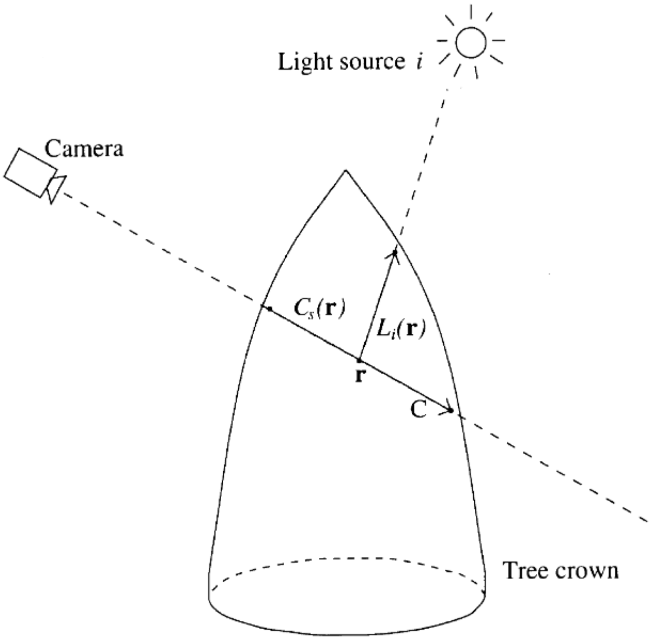


Figure 11.4: Tree crown model from Equation (11.6). The figure also shows single reflection of light from the sun into the airborne camera.

In the upper right part of Figure 11.5 we see an ellipse template placed close to the top of the optical model for sidelighted trees. There are three template parameters, size r, which is the radius of a circle with the same area as the ellipse, shape s, the width to length ratio of the ellipse, and t, the translation in r-units of the ellipse centre along the tree trunk such that tr is the downwards translation.

The following procedure was used to match the set tree top candidates with "ground truth" positions. Pairs of positions from the two sets were found in order of increasing error distance such that each position in each set was used at most once. The procedure was stopped when the error distance in the next match exceeded $d_{max} = 1$ m, and trees not matched at this stage were declared "unmatched". As penalty measure used for comparing a set of tree top candidates with "ground truth" the following modified standard error measure was used

$$SE^{\star} = \sqrt{\frac{\sum_{i \in \text{matched}} |\mathbf{x}_i - \bar{\mathbf{x}}|^2 + n_{\text{unmatched}} d_{max}^2}{n_{\text{matched}} + n_{\text{unmatched}}}}, \tag{11.7}$$

where the sum is taken over all matched tree tops, $\mathbf{x}_i$ is the error vector for tree top i, $|\cdot|$ is the Euclidean distance, $\bar{\mathbf{x}}$ is the average error vector for all matched trees, while n_{matched} and $n_{\text{unmatched}}$ are the number of matched and unmatched tree tops.

In the search for optimal parameters r, s and t with criterion function $SE^{\star}$ an iterated grid search was used, see (Larsen & Rudemo, 1998) for details. The search was performed separately for the three image with sidelighted, backlighted and frontlighted trees as seen in Figures 11.5 – 11.7. Similar parameter values were obtained with averages radius $r = 1.5$ m, width/length ration 0.9 and translation factor $t = 0.2$. The number of found and missed trees for the three images are shown in Table 11.2.

Table 11.2: Results for the three images with sidelighted trees, image 120, backlighted trees, image 124, and frontlighted trees 144.

	Image		
	120	124	144
Total number of trees	171	171	171
Number of missed trees	15	6	3
Percent of trees found	91	96	98
Standard error in cm (matched only)	27	24	28
Modified standard error (11.7) in cm	39	30	31

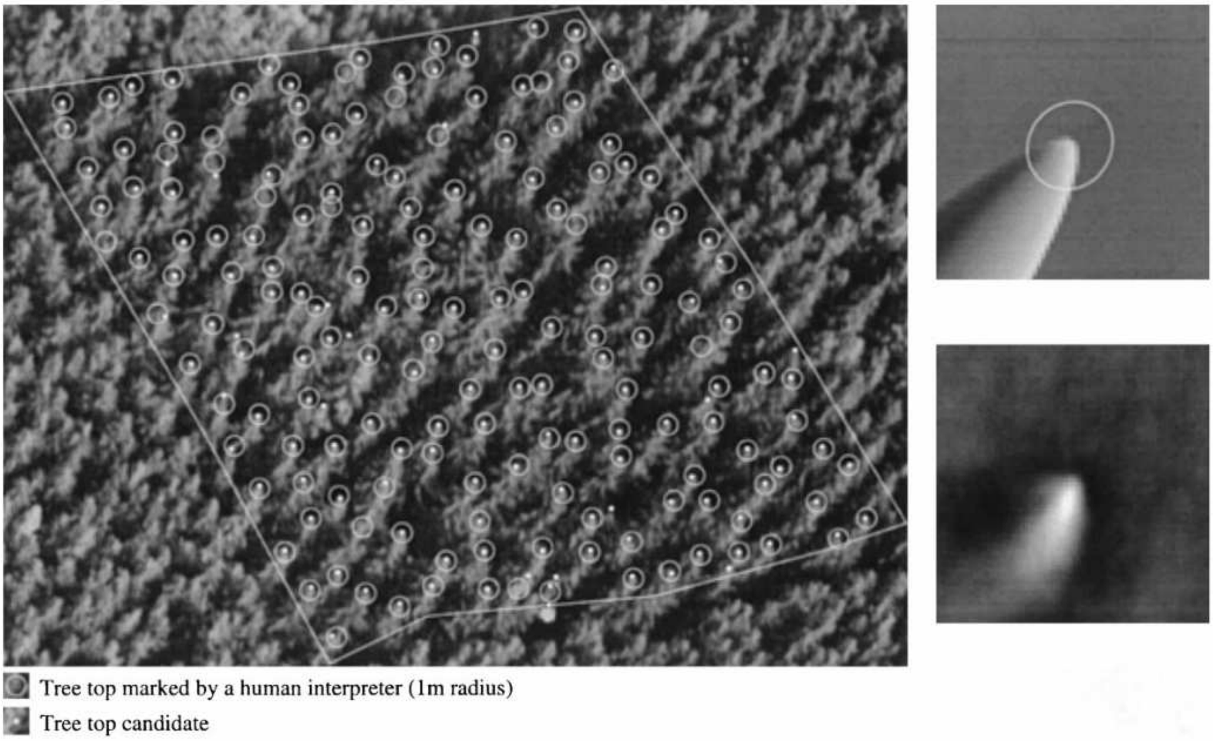


Figure 11.5: The sidelighted image “120” with 171 tree tops manually marked (circles) and automically estimated (dots), and to the right the corresponding single reflection optical model with optimal boundary (upper right) and the empirical average (lower right).

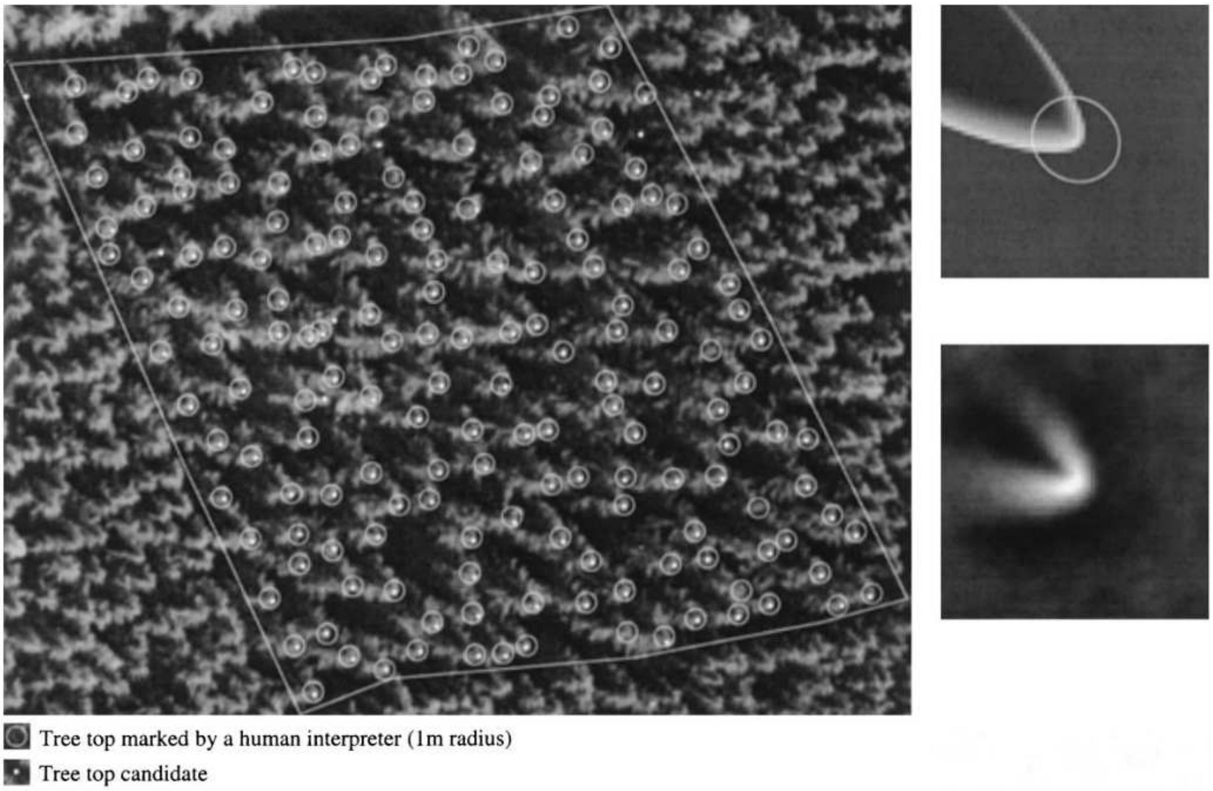


Figure 11.6: The backlighted image “124” with 171 tree tops manually marked (circles) and automically estimated (dots), and to the right the corresponding single reflection optical model with optimal boundary (upper right) and the empirical average (lower right).

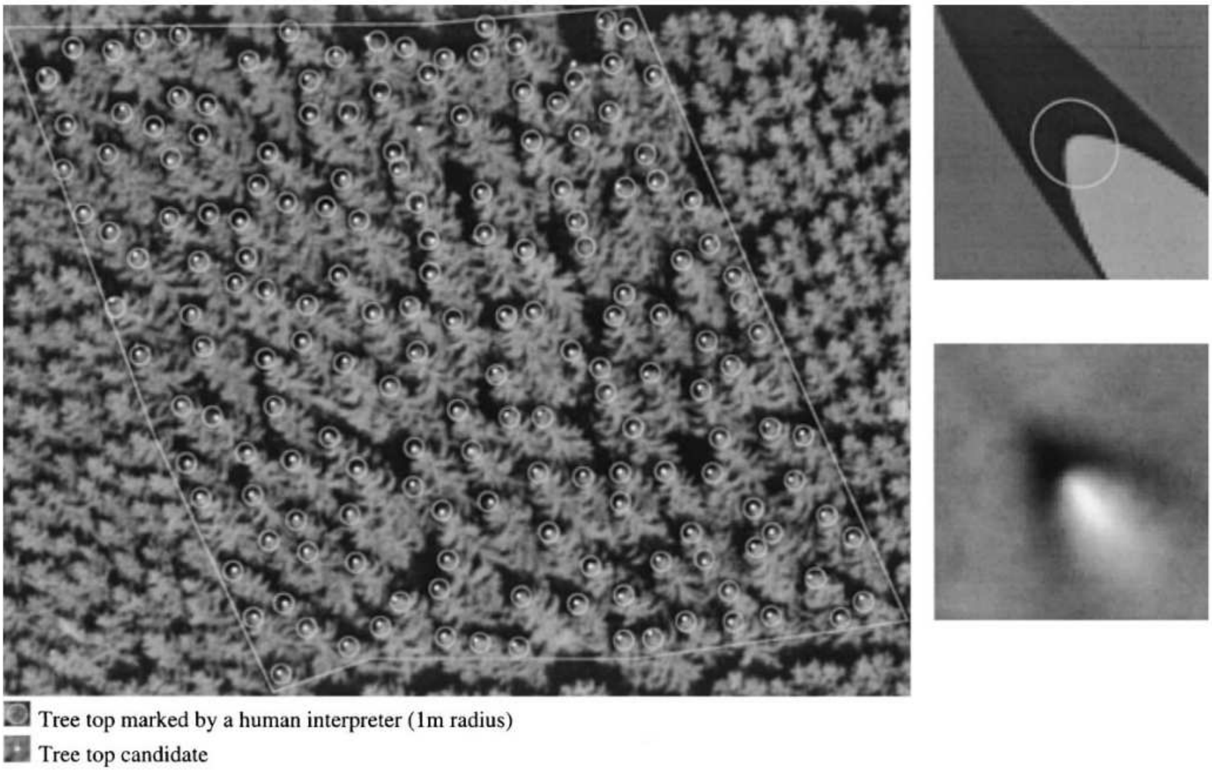


Figure 11.7: The frontlighted image “144” with 171 tree tops manually marked (circles) and automically estimated (dots), and to the right the corresponding single reflection optical model with optimal boundary (upper right) and the empirical average (lower right).

11.3 Exercises

Exercise 11.1. As a motivation for the model (11.1) for the systematic displacement from x_i to x_i' consider the following crude "tree" model: A tree consist of thin rod with a reflecting sphere on top. How is then the base location x_i in the image plane displaced to a point x_i' in the image plane corresponding to the point on the sphere where the sun is reflected as seen from the airplane? (This will not give exactly equation (11.1) but something quite close to it, at least if the angle α_i is small.)

Exercise 11.2. In the lower right parts of Figures 11.5 – 11.7 you can see empirical average shapes obtained from the use of the templates in the upper right parts in these figures. Discuss how these empirical average shapes can be computed, and discuss also if it might be possible to find improved templates by use of the empirical average shapes.

Chapter 12

Diffusion

12.1 Tracking a single diffusing particle

Let X_i denote the position at time $i\Delta t$, $i = 0, 1, \ldots, K$, of a diffusing particle in d-dimensional space, where $d = 1$, 2 or 3 in applications. We assume that

$$X_i = X_{i-1} + \Delta G_i, \tag{12.1}$$

where ΔG_i are independent d-dimensional normal vectors with a mean vector with all components zero and a covariance matrix

$$C(\Delta G_i) = 2D\Delta tI, \tag{12.2}$$

where D is the diffusion coefficient and I is the d-dimensional unit matrix. Thus in each dimension the diffusing particle has a normally distributed increment with mean zero and variance $2D\Delta t$, and the increments in different dimensions and at different time-points are all independent.

Let $||x||$ denote the Euclidean norm in d-dimensional space, that is $||x||^2 = \sum_j x_j^2$ if x has components $x_1, \ldots, x_d$. Then

$$\mathbf{E}(\sum_{i=1}^{K} ||\Delta G_i||^2) = 2dD\Delta tK \tag{12.3}$$

and it follows that

$$\hat{D} = \frac{1}{2d\Delta tK} \sum_{i=1}^{K} ||\Delta G_i||^2 \tag{12.4}$$

is an unbiased estimate of the diffusion coefficient D.

We can also obtain a confidence interval for D with, say, confidence degree 95%. The variable

$$\chi^2 = \frac{1}{2D\Delta t} \sum_{i=1}^{K} ||\Delta G_i||^2 \tag{12.5}$$

is chi-square distributed with dK degrees of freedom. Thus

$$\Pr(\chi^2_{.025} < \chi^2 < \chi^2_{.975}) = 0.95. \tag{12.6}$$

Straightforward computations give that (12.6) can be rewritten

$$\Pr(\frac{dK}{\chi^2_{.975}}\hat{D} < D < \frac{dK}{\chi^2_{.025}}\hat{D}) = 0.95. \tag{12.7}$$

and we see that

$$\frac{dK}{\chi^2_{.975}}\hat{D} < D < \frac{dK}{\chi^2_{.025}}\hat{D} \tag{12.8}$$

is a confidence interval for D with confidence degree 95 %.

12.2 Fluorescence recovery after photobleaching (FRAP)

Fluorescence recovery after photobleaching (FRAP) is a convenient method for analyzing diffusion which can be applied to the study of possibly heterogenous materials with locally varying diffusion coefficients. We shall here follow the exposition in (Jonasson *et al.*, 2008) which gives a pixel-based likelihood framework for FRAP. In FRAP the diffusion coefficient of fluorescent molecules is determined locally in a microscope. Fluorescent molecules are bleached and deactivated typically in a vertical cylinder by a high intensity laser pulse of short duration. This results in a decreased fluorescense in the bleached volume, see the upper left image in Figure 12.1 which shows fluorescence, observed in a confocal laser scanning microscope, shortly after the laser pulse. The sequence of images in Figure 12.1 shows the evolution of fluorescence in a horizontal 2D area corresponding to a thin volume extending a short distance in the vertical direction. From the image sequence we see how fluorescence is recovering due to the fact that unbleached molecules diffuse into and bleached molecules diffuse out of the deactivated volume. The recovery is clearly seen in Figure 12.1 and even more clearly in Figure 12.2.

The observed pixel intensity in the images will be modelled by a combination of a solution to the diffusion equation and an assumption of independent normally distributed errors. The diffusion of fluorochromes is supposed to follow the diffusion equation (similar to the heat equation)

$$\frac{\partial C}{\partial t} = D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2}\right), \tag{12.9}$$

where C is the concentration of unbleached fluorochromes and D is the diffusion coefficient. Let us regard a rotationally symmetric bleached region and assume that there is no net diffusion in the z-direction and further that the fluorochromes are initially (before the high intensity laser pulse) uniformly distributed.

With polar coordinates the diffusion equation can be written

$$\frac{\partial C}{\partial t} = D\left(\frac{1}{r}\frac{\partial C}{\partial r} + \frac{\partial^2 C}{\partial r^2}\right), \tag{12.10}$$

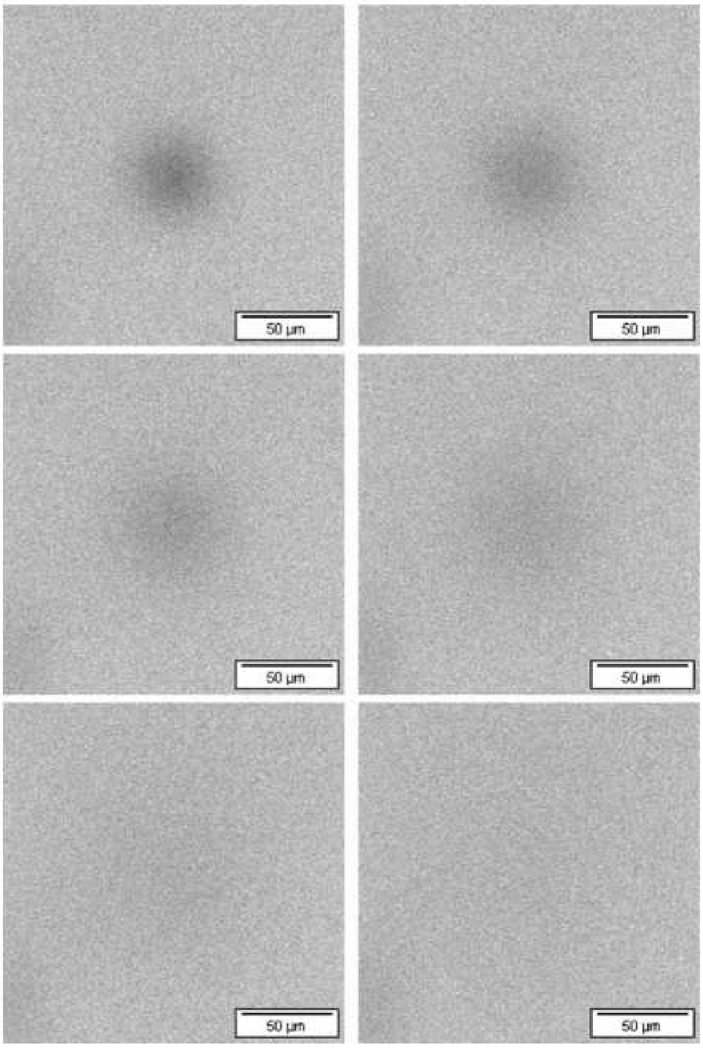


Figure 12.1: Plots of images from the first photobleaching series with 256 x 256 pixels described in Table 12.1. The left top image is the first after bleaching, then follows images about 1 s, 2 s, 4 s, 8 s and 16 s later.

where r is the distance from the centre of the bleached region. Let $C_0(r)$ denote the fluorochrome concentration at time zero (immediately after the high intensity pulse), and let $I_0(x) = (1/\pi)\int_0^\pi \exp(-x \cos t)\, dt$ denote the modified Bessel function of order zero. The solution of equation (12.10) can be written on the form

$$C(r,t) = \frac{1}{2Dt}\exp\left(-\frac{r^2}{4Dt}\right)\int_0^\infty uC_0(u)I_0\left(\frac{ru}{2Dt}\right)\exp\left(-\frac{u^2}{4Dt}\right)du. \tag{12.11}$$

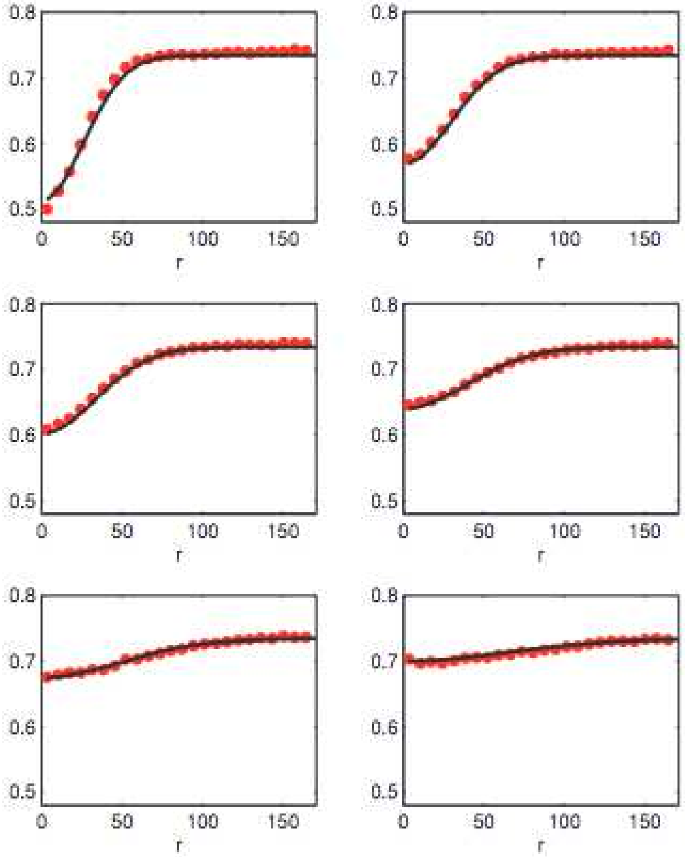


Figure 12.2: Plots of fitted concentration and pixel values, averaged over pixels with equal distances to the bleaching centre, as a function of distance r to the bleaching centre for the same series as shown in Figure 12.1. The left top image is the first after bleaching, then follows images about 1 s, 2 s, 4 s, 8 s and 16 s later.

If we would have complete bleaching the intensity profile immediately after bleaching would be described by an inverse top hat function. However, the bleaching is not complete and diffusion starts directly to blur this profile and in the upper left images in Figures 12.1 and 12.2 we see a profile rather different from a top hat. We will assume that the initial profile is an approximately Gaussian profile, and suppose that the initial concentration has the form

$$C_0(r) = a_0 - \frac{a_1}{r_0^2}\exp\left(-\frac{r^2}{r_0^2}\right)du. \tag{12.12}$$

Then the solution of equation (12.10) with the initial condition $C(0,r) = C_0(r)$ simplifies to

$$C(r,t) = a_0 - \frac{a_1}{4Dt + r_0^2} \exp\left(-\frac{r^2}{4Dt + r_0^2}\right) du. \tag{12.13}$$

Let $p(i,t)$ denote the observed intensity at time t at pixel i with distance r_i to the centre of the bleached region. We will assume that except for additive random noise the pixel intensity is proportional to the fluorochrome concentration $C(r_i,t)$. Let us further assume that pixel-wise the noise is normal with mean zero and variance σ^2 with independence between different pixels and different times. Let S denote the set of pixels and T the set of times regarded. Thus we assume that the pixel-values $p(i,t), i \in S, t \in T$, are independent with probability density

$$f(p(i,t); a_0, a_1, D, r_0, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(p(i,t) - C(r_i,t))^2}{2\sigma^2}\right). \tag{12.14}$$

The likelihood function is the joint probability density for all pixels and all times, and due to independence it is

$$L(\theta) = \prod_{t \in T} \prod_{i \in S} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(p(i,t) - C(r_i,t))^2}{2\sigma^2}\right), \tag{12.15}$$

where θ is the parameter vector $\theta = (a_0, a_1, D, r_0, \sigma^2)$.

The log-likelihood $\ell(\theta) = \log L(\theta)$ is then

$$\ell(\theta) = \frac{|T||S|}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{t \in T} \sum_{i \in S} (p(i,t) - C(r_i,t))^2, \tag{12.16}$$

and it is maximized with respect to the parameter vector θ to find the ml estimates, the most likely parameter values given the observed images. Likelihood theory allows computation of parameter estimates together with corresponding standard errors, compare Section 14.12. The parameter estimates are approximately multivariate normally distributed with a covariance matrix that is the inverse of the observed information matrix. The entry in row j and column k of the observed information matrix is

$$-\frac{\partial^2}{\partial\theta_j \partial\theta_k} \ell(\theta), \tag{12.17}$$

evaluated at $\theta = \hat{\theta}$, where $\hat{\theta}$ is the ml estimate of θ. If the coordinates of the centre of the bleached disk are unknown there will be two extra parameters in the likelihood.

In (Jonasson *et al.*, 2008) results from experiments with a Sodium Fluorescein probe in polyethylene glycol are reported. Two series of experiments with respectively 128×128 pixel images and 256 x 256 pixel images were performed, and in each series four replicates with differently placed bleaching centres were used. Results from the experiments are shown in Table 12.1 and for one of the replicates in more detail in Figures 12.1 and 12.2.

As a check of the FRAP results given in Table 12.1 a corresponding NMR diffusometry experiment was performed. It gave an estimated diffusion coefficient of 62.0 $\mu m^2/s$ with a standard error of 1.9 $\mu m^2/s$, which is well in line with the results in Table 12.1.

Table 12.1: Results from an experiment with two replicate series. For the first four replicates (with 128×128 pixels) 48 images were used and for the last four replicates (with 256 x 256 pixels) 18 images were used. The columns D and s show diffusion coefficients and standard errors estimated by maximum likelihood, while $\bar{D}$ and s_{repl} show averages and standard deviation from the replicate series.

Replicate	No of pixels	D ($\mu m^2/s$)	s ($\mu m^2/s$)	$\bar{D}(\mu m^2/s)$	s_{repl} ($\mu m^2/s$)
1	128×128	64.3	0.8		
2	128×128	60.1	0.8		
3	128×128	61.1	0.8		
4	128×128	59.6	0.8	61.3	2.1
1	256×256	61.0	0.5		
2	256×256	61.8	0.5		
3	256×256	60.8	0.4		
4	256×256	63.8	0.5	61.8	1.4

12.2.1 Exercises

Exercise 12.1. Verify that χ^2 in equation (12.5) is chi-square distributed with dK degrees of freedom.

Exercise 12.2. Verify that $C(r, t)$ given by equation (12.13) satisfies equation (12.10) with the initial condition (12.12).

Exercise 12.3. In the computation of the likelihood in equation (12.15) it is assumed that noise contributions in different pixels are independent. Take a close look at one of the images in Figure 12.1. Does it seem as the independence assumption is valid. Describe how you could check the independence assumption with access to the images in Figure 12.1.

12.3 Estimation of particle concentration from single-particle tracking

Nano-sized fluorescent particles observed in a microscope can typically be detected in a rather thin rectangular box such as shown in Figure 12.3. To determine particle concentration we need to know the dimensions of the detection region. The extension in the horizontal directions can ususally be determined in a straightforward way from the microscope field of view. However, the size in the vertical direction is much more difficult to measure as it depends on a number of factors such as the particle detection algorithm and the brightness of the observed particles. Such properties are not fixed but can vary considerably between experiments. In (Röding *et al.*, 2011) this problem is analyzed and the vertical dimension is estimated from the trajectory length distribution. We will assume that the detection region thickness is considerably smaller than the horizontal dimensions which means that particles typically enter and leave the detection region by moving upwards or downwards. The trajectory length distribution is then essentially determined by the detection region thickness. Roughly, short trajectory lengths indicate a small thickness.

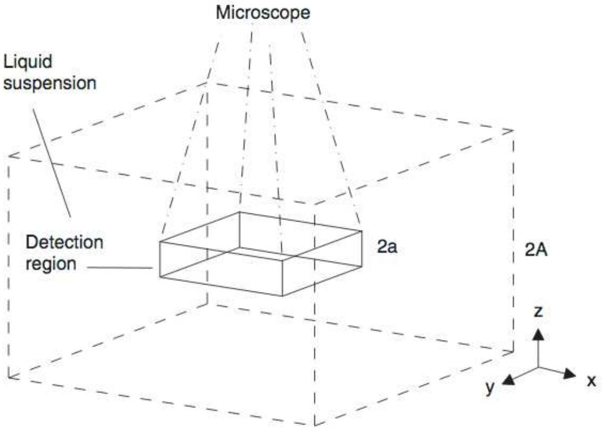


Figure 12.3: A microscope detection region modeled as a rectangular box centred in the liquid suspension where particles move. Particles outside the detection region cannot be observed. The tracking depth is $2a$ and the thickness of the suspension is $2A$.

Let us assume that we observe a particle at positions X_i at K equidistant time-points $t_i = i\Delta t, i = 1, \ldots, K$, typically corresponding to K consecutive frames in a video sequence. We use the same notations as in Section 12.1, just that we start here at time $t_1 = \Delta$ rather than time 0. Thus the particle enters the detection region at time t_1 and leaves it after K observed positions. Although the particle moves in 3D we will simplify and only consider the motion in 1D, namely the z-direction. Thus we assume that the particle enters and leaves the detection region from above or below, which should be a good approximation when the verticle dimension $2a$ of the detection region is much smaller than the horizontal dimensions. We also assume that the detection region thickness $2a$ is much smaller than the thickness $2A$ of the liquid suspension volume.

Considering only 1D diffusion in the z-direction we assume that initially the particle position is uniformly distributed in the interval $[-A, A]$ and a particle outside the detecion region is assumed to be uniformly distributed over $[-A, -a] \cup [a, A]$. Let $f(z)$ denote the probability density of the position of a particle that has just entered the detection region. One can then show that $f(z) = 0$ for $|z| > a$ and

$$f(z) = \frac{h(z)}{\int_{-a}^{a} h(z)dz}, \quad z \in [-a, a], \tag{12.18}$$

where

$$h(z) = \frac{1}{2(A-a)} \left[\Phi\left(\frac{z+A}{\sqrt{2D\Delta t}}\right) - \Phi\left(\frac{z+a}{\sqrt{2D\Delta t}}\right) + \Phi\left(\frac{z-a}{\sqrt{2D\Delta t}}\right) - \Phi\left(\frac{z-A}{\sqrt{2D\Delta t}}\right) \right] \tag{12.19}$$

and Φ denotes the standard normal cumulative distribution function. A proof of this is outlined in Exercise 12.5 below. Let Z_k denote the position of a particle and let f_k denote the non-normalized density of the particle position after k steps assuming that $K \geq k$, more precisely $f_k(z) = d/dz[P(Z_k \leq z \text{ and } K \geq k)]$, for $k \geq 1$. By definition $f_k(z)$ is zero outside $[-a, a]$. For the first position of the particle in the detection region we have $f_1 = f$ given by (12.18). To compute the probability density of the particle after step 2, f_1 is convolved with the Gaussian propagator

$$G(z) = \frac{1}{2D\Delta t}\phi\left(\frac{z}{2D\Delta t}\right), \tag{12.20}$$

where ϕ is the density of a standardized normal variable. Since we assume that the particle stays in the detection region K steps it cannot be outside the interval $[-a, a]$ and the density has to be truncated. Generally the density f_k can be recursively computed from f_{k-1} according to

$$f_k(z) = \begin{cases} \int_{-\infty}^{\infty} f_{k-1}(z_0)G(z-z_0)\, dz_0, & z \in [a, a], \\ 0, & z \notin [-a, a]. \end{cases} \tag{12.21}$$

Computation of f_k for $k > 1$ cannot be performed analytically, but a fast numerical scheme with probability densities approximated by translates of a Gaussian kernel is described in (Röding *et al.*, 2011). In Figure 12.4 the computation of the sequence of densities $f_k, k \geq 1$ is illustrated.

The probability that a particle stays in the detection region for at most k consecutive steps is

$$P_a(K \leq k) = 1 - \int_{-a}^{a} f_{k+1}(z)\, dz, \tag{12.22}$$

where the dependence on a is emphasized. The probability distribution for the trajectory length is then obtained from

$$P_a(K = k) = P_a(K \leq k) - P_a(K \leq k-1). \tag{12.23}$$

Suppose now that we have observed an ensemble of identical particles with known diffusion coefficient. The assumption of known (or well estimated) diffusion coefficient is reasonable as it can readily be estimated from the particle trajectories, compare (12.4).

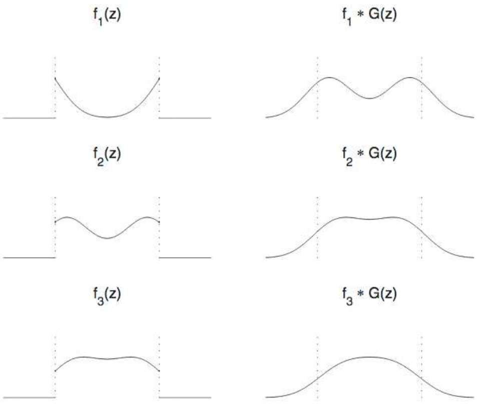


Figure 12.4: Illustration of the procedure for computing the trajectory length distribution. Here $f_1(z)$ is the probability density of a particle that has just entered the detection region according to equation (12.18). Truncation outside of $[-a, a]$ of the convolution $f_1 \star G(z)$ yields the non-normalized density $f_2(z)$ which integrates to the probability that the particle still remains in the detection region for a second sampling point, and so forth.

Let us consider trajectories with length $K \geq k_{min}$. As discussed in (Röding *et al.*, 2011) it is typical to impose a lower threshold like $K \geq 3$ or $K \geq 4$ for the trajectory length as shorter trajectories are more likely to be false positives. Let N_k denote the number of observed trajectories of length k. Then the log-likelihood function is

$$\ell(a) = \sum_{k \geq k_{min}} N_k \log P_a(K = k | K \geq k_{min}), \tag{12.24}$$

where

$$P_a(K = k | K \geq k_{min}) = \frac{P_a(K = k)}{P_a(K \geq k_{min})} \tag{12.25}$$

and $P_a(K \geq k_{min})$ is computed from (12.22). The maximum likelihood estimate $\hat{a}$ is the a-value that maximizes $\ell(a)$ in (12.24).

After having estimated the tracking depth a it is possible to estimate the particle concentration (also called the number concentration). Let $\bar{N}$ denote the mean number of particles per video frame. A suitable point estimator of the particle concentration c is

$$\hat{c} = \frac{\bar{N}}{8\hat{a}a_x a_y 10^{-12}} \text{ particles/ml}, \tag{12.26}$$

where $2a_x$ and $2a_y$ are the lateral sizes in μm of the detection region. We can estimate $\bar{N}$ by counting trajectories as follows. Let n be the number of frames, and let N_k as earlier be the number of observed trajectories of length k. The number of observed particle positions is the sum of all trajectory lengths. Dividing by the number of frames we get an estimate of the mean number of particles per frame, and we estimate $\bar{N}$ by

$$\bar{N} = \frac{1}{\hat{p}_{obs}} \frac{1}{n} \sum_{k \geq k_{min}} k N_k. \tag{12.27}$$

The factor $\hat{p}_{obs}$ corrects for underestimation of the concentration due to discarding trajectories with length $k < k_{min}$,

$$\hat{p}_{obs} = \frac{\sum_{k \geq k_{min}} k P_{\hat{a}}(K = k)}{\sum_{k \geq 1} k P_{\hat{a}}(K = k)}. \tag{12.28}$$

With this correction factor the estimate $\bar{N}$ in (12.27) becomes approximately unbiased. The standard error of the concentration estimate can be assessed by bootstrapping, compare Section 14.12. It is here suitable to perform the bootstrapping on video level, since videos are (approximately) independent. Thus B bootstrap samples are obtained by sampling B times with replacement from the set of videos, and from each sample the concentration estimates $\hat{c}_1, \ldots, \hat{c}_B$ are computed according to (12.26). This gives an approximate standard deviation estimate

$$\sigma_{\hat{c}} = \left(\frac{1}{B-1} \sum_{i=1}^{B} (\hat{c}_i - c_{mean})^2 \right)^{1/2}, \tag{12.29}$$

where c_{mean} is the mean of the bootstrap estimates. This method relies on that the videos are approximately equally long and independent. The simulation study briefly described below, and in more detail in (Röding *et al.*, 2011), leads to the conclusion that the tracking depth and the concentration estimates are approximately unbiased and that bootstrap errors for $B = 50$ are quite close to the actual standard errors.

In the simulation study particles were moving according to 3D random walk with time increments Δt and independent zero mean normally distributed increments with variance $2D\Delta t$ in all three dimensions. Particles moved in a cube with side length $2A = 40\ \mu m$, compare Figure 12.3, with periodic boundary conditions. Particle trajectories were recorded when particles entered the detection region. In the study three different diffusion coefficients, $D = 1\ \mu m^2/s$, $D = 2\ \mu m^2/s$, and $D = 5\ \mu m^2/s$, and a series of values for the detection region thickness from 0.1 to 2 μm were used. The concentration of particles was $c = 10^9$ particles/ml. For each combination of diffusion coefficient and detection region thickness 20 000 simulations were performed and the mean obtained estimates of a and c are shown in Figure 12.5.

In addition to the results from simulations, results from experiments with 0.19-μm and 0.52-μm particles are also reported in (Röding *et al.*, 2011). In Figure 12.6 we see concentration estimates for 5 dilutions with the 0.19-μm particles. Estimated 95% confidence intervals obtained by bootstrapping for each dilution are also shown. Ideally the concentration estimates should fall on the solid straight line shown. However, this line is not perfectly known as there are some uncertainties of the size of the particles. Mean particle diameter was estimated by use of light scattering and was found to be 0.207 μm with a standard deviation of 0.008 μm. From this a 95% confidence interval for the solid line is obtained and shown in Figure 12.6.

From Figures 12.5 and 12.6 we see that the method suggested in (Röding *et al.*, 2011) performs well both for simulated and experimental data.

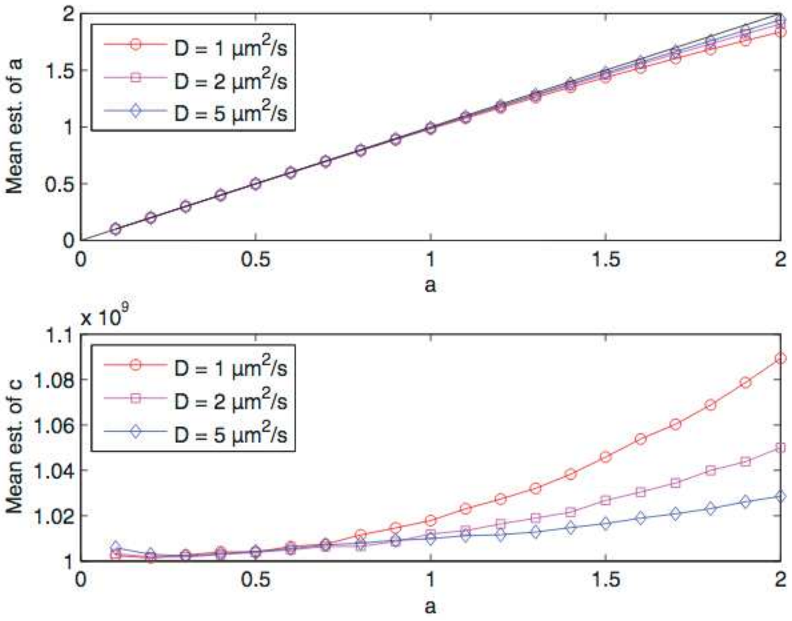


Figure 12.5: Simulation study of the tracking depth parameter a (upper) and the concentration c (lower). Mean estimates are shown for $D = 1$ $\mu m^2/s$ (red circles), for $D = 2$ $\mu m^2/s$ (magenta squares), and for $D = 5$ $\mu m^2/s$ (blue diamonds) as functions of the true value of a. The true value of a is given by the black solid line. The true concentration of particles was $c = 10^9$ particles/ml. The increasing bias (negative for a and positive for c) for increasing a is due to the 1D approximation in the model for the trajectory length distribution.

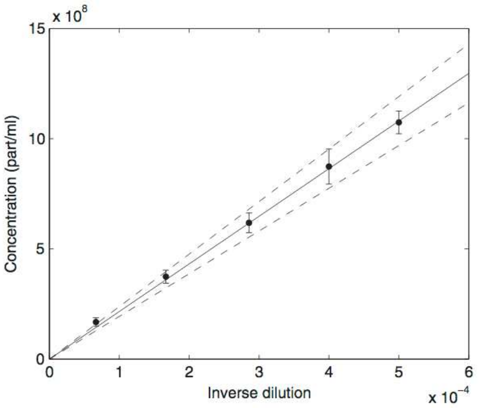


Figure 12.6: Experimental results with estimated concentrations for different dilutions of 0.19-μm particles with estimated 95% confidence intervals. The concentration as estimated from the stock-solution concentration (solid line) and estimated 95% confidence intervals (dashed lines) are also shown.

12.3.1 Exercises

Exercise 12.4. In Figure 12.5 we see that there is a negative bias for the estimated thickness a of the detection region and a positive bias for the estimated particle concentration. In both cases the size of the bias increases with increasing a and also with decreasing diffusion coefficient D. Give qualitative explanations of these effects.

Exercise 12.5. Try to derive (12.18). You can for instance start by finding the distribution of a particle for which you only know that it is outside the interval $[-a, a]$. Convolve then this distribution with the Gaussian propagator, compare with (12.21).

12.4 Estimation of particle concentration from particle count time series

The method for estimation of particle concentration discussed in the previous section requires particle tracking, that is pairing particles from one frame to the following frame. This may be difficult for fast particles and high concentrations. In this section we will follow (Röding *et al.*, 2013) and describe a method which only requires counting the number of particles in each frame but no tracking of the individual particles.

In Figure 12.7 we see an experimentally observed count process with the number of particles varying between about 18 and 37 particles. Such a process of particle counts we will call a Smoluchowski process in honour of the Polish physicist M. von Smoluchowski who in (von Smoluchowski, 1906) developed an alternative to Einstein's description (Einstein, 1905) of Brownian motion.

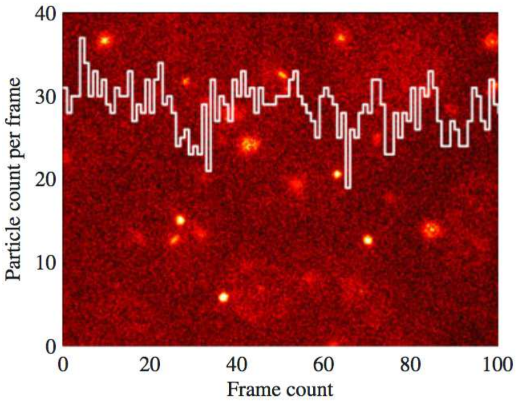


Figure 12.7: An example of an experimentally observed Smoluchowski process obtained by counting liposomes in whole blood, superimposed over a sample frame from the raw image data, compare (Braeckmans *et al.*, 2010).

We will assume that particles move in and out of a microscope detection region of the type shown in Figure 12.3. In this section we will call the lateral dimensions of the detection region $2a_x$ and $2a_y$ and the vertical dimension $2a_z$. Thus $2a$ in Figure

12.3 corresponds here to $2a_z$. The number of particles in a sequence of frames varies as illustrated in Figure 12.8.

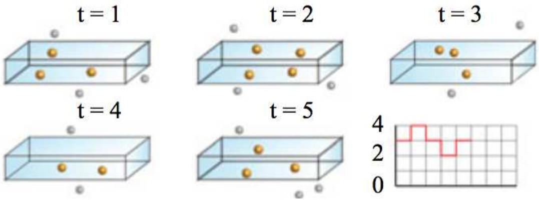


Figure 12.8: Illustration of a Smoluchowski process. Diffusing particles reside both inside (yellow) and outside (grey) the detection region. Particles moving in and out of the detection region and the number of detected particles is fluctuating, forming a random time series.

Let us now describe an approximate Markov statistical model for the Smoluchowski process. We assume that particles move independently of each other according to a Brownian motion with independent increments in all three dimensions with mean zero and variance $2D\Delta t$, where Δt is the interval between observations (frames). Let $X_n, n = 1, \ldots, N$, denote the number of particles observed in the nth frame. Then

$$X_{n+1} = X_n - O_n + I_n, \tag{12.30}$$

where O_n is the number of particles, out of the X_n particles initially present, exiting the detection region, and I_n is the number of particles entering that region, between the two observations X_n and X_{n+1}. We shall assume that regardless of observation up to (and including) X_n the random variable I_n is Poisson distributed with a parameter λ, that is,

$$\Pr(I_n = k | X_1, \ldots, X_n) = \frac{\lambda^k}{k!} e^{-\lambda}. \tag{12.31}$$

Another assumption, which we shall make, is that given observations up to (and including) X_n, the random variable O_n is binomially distributed with probability-parameter μ, more precisely, that

$$\Pr(O_n = j | X_1, \ldots, X_n) = \left(\begin{array}{c} X_n \\ j \end{array} \right) \mu^j (1-\mu)^{X_n - j}. \tag{12.32}$$

Based on these assumptions we approximate the distribution of the process of particle counts $(X_n, n \geq 1)$ with a Markov model with transition probabilities $p_{ij} = \Pr(X_{n+1} = j | X_n = i)$ given by

$$p_{ij}(\lambda, \mu) = e^{-\lambda} \sum_{k=\max(0, j-i)}^{j} \frac{\lambda^k}{k!} \left(\begin{array}{c} i \\ i-j+k \end{array} \right) \mu^{i-j+k} (1-\mu)^{j-k}. \tag{12.33}$$

One can show that a Markov chain with transition probabilities given by (12.33) has a stationary distribution which is a Poisson distribution with parameter λ/μ, that is

$$\Pr(X_n = k) = \pi_k = \frac{(\lambda/\mu)^k e^{-\lambda/\mu}}{k!}. \tag{12.34}$$

Given the Markov assumption the joint distribution of particle counts $X_1, \ldots, X_N$ can be written

$$\Pr(X_1 = x_1, \ldots, X_N = x_N) = \Pr(X_1 = x_1) \prod_{k=2}^{N} \Pr(X_k = x_k | X_{k-1} = x_{k-1}). \qquad (12.35)$$

For a realization $x_1, \ldots, x_N$ we obtain a log-likelihood function $\ell(\lambda, \mu) = \ell(\lambda, \mu | x_1, \ldots, x_N)$ given by

$$\ell(\lambda, \mu) = \log \frac{(\lambda/\mu)^{x_1} e^{-\lambda/\mu}}{x_1!} + \sum_{i,j} N_{ij} \log p_{ij}(\lambda, \mu), \qquad (12.36)$$

where N_{ij} is the number of transitions from state i to state j. We obtain the maximum likelihood estimates $\hat{\lambda}$ and $\hat{\mu}$ by maximizing the log-likelihood $\ell(\lambda, \mu)$. For estimation of the lateral dimension parameter a_z it turns out that the crucial parameter is μ. The parameter μ may be interpreted as the probability that a particle uniformly distributed in the detection region exits this region in a time interval of length Δt, compare (12.32). With this interpretation one can show that

$$\mu = \mu(a_z) = 1 - F(a_x, D)F(a_y, D)F(a_z, D), \qquad (12.37)$$

with

$$F(a, D) = \frac{\sqrt{2D\Delta t}}{2a} \left\{ \frac{2a}{\sqrt{2D\Delta t}} \left[2\Phi\left(\frac{2a}{\sqrt{2D\Delta t}}\right) - 1 \right] + 2\phi\left(\frac{2a}{\sqrt{2D\Delta t}}\right) - 2\phi(0) \right\}, \qquad (12.38)$$

where Φ and ϕ denote the distribution function and the probability density of a standardized normal variable. Note that in (12.37) we write $\mu = \mu(a_z)$ because here a_z is the important unknown parameter. The lateral dimension parameters a_x and a_y can be measured directly from the microscope geometry and D here needs to be estimated separately, for instance by separate particle tracking. Let us also note that in order to obtain valid standard errors and confidence intervals it is suitable, as in Section 12.3, to use bootstrapping on the 'video level'.

To validate the suggested method both simulations and experiments were used. In the simulations a predetermined number of particles were allowed to diffuse in three dimensions in rectangular box, as the large box in Figure 12.3, with periodic boundary conditions. Three different diffusion coefficients, $D = 1$, $D = 2$ and $D = 5\,\mu\text{m}^2\,\text{s}^{-1}$, and 20 different a_z-values ranging between 0.1 and 2 μm were used. The resulting estimates of the detection depth parameter a_z and the concentration are shown in Figure 12.9. From the figure we see that the method performs very well except for some minor bias for small a_z-values.

To experimentally verify the method suggested two experiments with fluorescent polymer nanospheres with diameter $0.2\,\mu$m and $0.5\,\mu$m were performed. We will here show the results for the smaller diameter. A water dispersion of the particles was diluted by a factor of 1900, 2400, 3400, 5800 and 14800. The theoretical concentration of particles in particles mL^{-1} can be estimated from

$$C_{\text{theoretical}} = \frac{6{\times}10^{10}{\times}S\rho_L}{\pi\rho_S d^3}, \qquad (12.39)$$

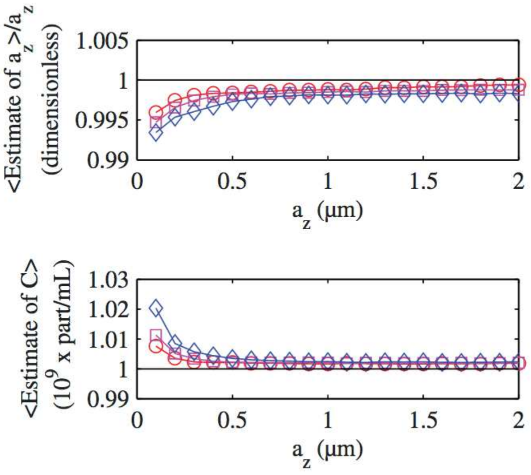


Figure 12.9: Simulation study of estimation of the detection depth parameter a_z and the concentration estimate C. For $D = 1\,\mu\text{m}^2\,\text{s}^{-1}$ (red circles), $D = 2\,\mu\text{m}^2\,\text{s}^{-1}$ (magenta squares) and $D = 5\,\mu\text{m}^2\,\text{s}^{-1}$ (blue diamonds) the mean estimates of a_z (divided by the true value of a_z) and C are shown as functions of the true value of a_z. The mean estimates were computed from 10^6 simulations for each data point, and the true concentration of particles was $C = 10^9$ particle mL^{-1}.

where $S = 1$ is the weight percent of solids, with a relativ standard deviation of 5%, $\rho_L = 1.00\,\text{g}\,\text{cm}^{-3}$ is the density of the suspension, $\rho_S = 1.05\,\text{g}\,\text{cm}^{-3}$ is the density of the solid particles (all values according to the manufacturer). Further, using dynamic light scattering the diameter of the particles was found to be $d = 0.207\,\mu\text{m}$ with a standard deviation of $0.008\,\mu\text{m}$ (in correspondence with the manufacturer results for the particular batch of nanospheres). Using the standard error-propagation equation the theoretical particle concentration with standard deviations were found for all dilutions and compared with the results from the method suggested. The results are shown in Figure 12.10 and it clear from the figure that an excellent agreement was found between the theoretically and experimentally obtained concentration values.

12.4.1 Exercises

Exercise 12.6. The Markov model used in the present section is based on (12.31) and (12.32) leading to the transition probabilities (12.33). Motivate why equations (12.31) and (12.32) are only approximately true.

Exercise 12.7. In Figure 12.9 we see that there is a negative bias for the estimated thickness a_z of the detection region and a positive bias for the estimated particle concentration. In both cases the size of the bias increases with decreasing a_z and also with increasing diffusion coefficient D. Give qualitative explanations of these effects.

Exercise 12.8. Show that (12.34) gives a stationary distribution for a Markov chain with transition probabilities (12.33).

Exercise 12.9. Try to derive (12.37) with $F(a, D)$ given by (12.38) by computing the

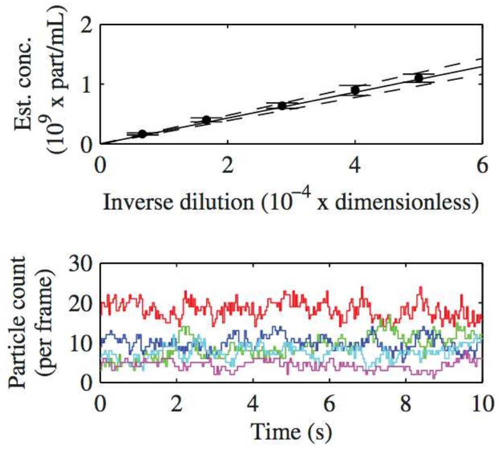


Figure 12.10: Estimated concentrations from an experiment with different dilutions of 0.2 μm particles with estimated 95% confidence intervals ('inverse dilution' is a 'relative concentration'). The concentration as estimated from the stock-solution concentration (solid line) with estimated 95% confidence intervals (dashed lines) is shown (upper). Further, typical examples of the underlying Smoluchowski processes are shown with colours red/green/blue/cyan/magenta in order of of decreasing concentration (lower).

probability that a diffusing particle stays within the detection region in all three dimensions.

12.5 Single particle raster image analysis of diffusion

As we have seen in the previous sections of this chapter there are several powerful methods for estimating diffusion coefficients from series of images. In this section we shall study a method that in principle allows us to estimate diffusion coefficients from single images. The method is based on using a clever raster scan pattern described in (Digman *et al.*, 2005). Images of diffusing molecules are collected with a confocal laser scanning microscope using a raster scan pattern with a specific time structure. During the scanning, adjacent pixels in the x-direction are visited within a short time interval whereas the time between visits to adjacent pixels in the y-direction is much longer, see Figure 12.11. The method RICS, Raster Image Correlation Spectroscopy, is suggested in (Digman *et al.*, 2005) to analyse raster images by use of correlation function estimation typically computed by the fast Fourier transform method. One can show, compare equation (24) in (Longfils *et al.*, 2018), that the theoretical correlation function $G(\xi, \psi)$ for the scanned image corresponding to two points (x, y) and $(x+\xi, y+\psi)$ is

$$G(\xi,\psi) = \frac{1}{\langle N\rangle}\left(1+\frac{4D\tau(\xi,\psi)}{\omega_0^2}\right)^{-1}\left(1+\frac{4D\tau(\xi,\psi)}{\omega_z^2}\right)^{-1/2}\exp\left[-\frac{(S\xi)^2+(S\psi)^2}{\omega_0^2+4D\tau(\xi.\psi)}\right], \tag{12.40}$$

where $\langle N\rangle$ is the average number of particles in the observation volume, S is the pixel size, and the function $\tau(\xi, \psi)$ takes the form

$$\tau(\xi, \psi) = |\tau_p \xi + \tau_l \psi| \tag{12.41}$$

corresponding to the time it takes to move between the points (x, y) and $(x + \xi, y + \psi)$. The parameters ω_0 and ω_z in (12.40), called lateral and vertical waists, correspond to the decay rate of the point spread function in the lateral and vertical directions and can be estimated as standard deviations of a 2- or 3-dimensional Gaussian function fitted to the laser point spread function.

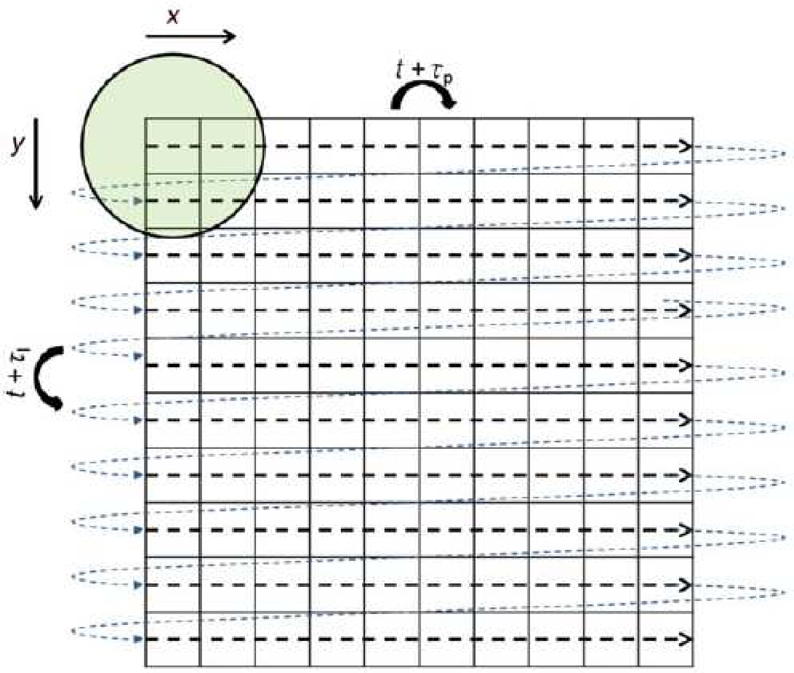


Figure 12.11: Movement of scanning beam according to the raster scan pattern used in RICS and SPRIA. The scanning time between adjacent pixels in the x- and y-directions are τ_p and τ_l, respectively, and $\tau_p \ll \tau_l$.

In (Longfils *et al.*, 2017) an alternative analysis method SPRIA, Single Particel Raster Image Analysis, is introduced based on properties of the individual diffusing particles. By analysing each particle separately possibilities are opened for analysis of systems of particles with a mixture of different diffusion coefficients and for heterogenous materials where the diffusion properties vary with location.

The sampling time structure provides information on the dynamics of the particles inside each image. In Figure 12.12 typical raster scan images are shown for different time scales demonstrating the effect of varying scan rates visually on experimental data with 175-nm beads scanned at decreasing speed. In (A) at scan rate 8000 Hz almost immobile round-shaped particles are observed. In (B) the particles scanned at 400 Hz move slowly and the horizontal particle lines look like sequences of shifted bright lines. In (C) the scan rate is further decreased to 100 Hz and the sequence of bright lines corresponding to one particle become even more shifted as the particle is moving significantly between aquisition of consecutive lines.

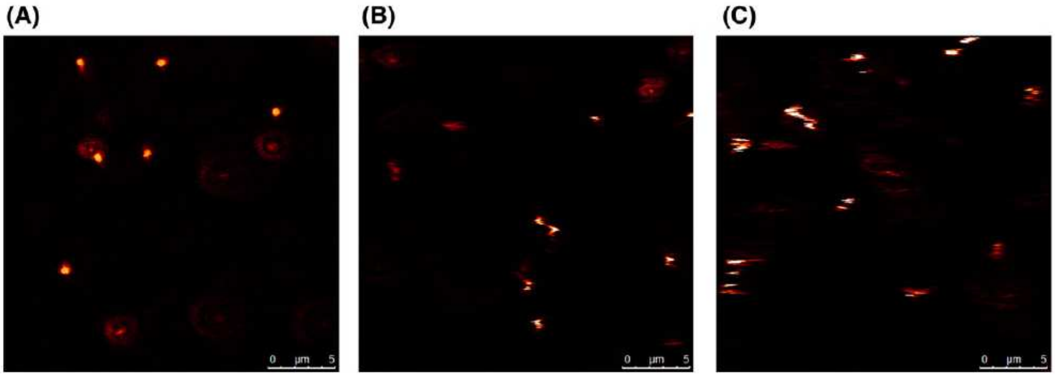


Figure 12.12: Raster scan images of 175 nm beads at a pixel size of 48.1 nm and varying scan rate: (A) 8000 Hz, (B) 400 Hz and (C) 100 Hz.

To be able to apply the single particle method, individual particles have to be extracted from an image as shown in Figure 12.13. The procedure is based on two threshold levels $T_1 > T_2$. First find local photon count maxima above the level T_1. Then find around each chosen maximum the smallest axis-parallell rectangle such that all observed photon count levels just outside the rectangle border are below T_2, see (Longfils *et al.*, 2017) for details. It turns out that the choice of levels is not critical, compare Exercise 12.11, which seems quite plausible from a look at Figure 12.13. In Figure 12.14 we find an axis-parallell rectangle chosen with $T_1 = 10$ and $T_2 = 5$.

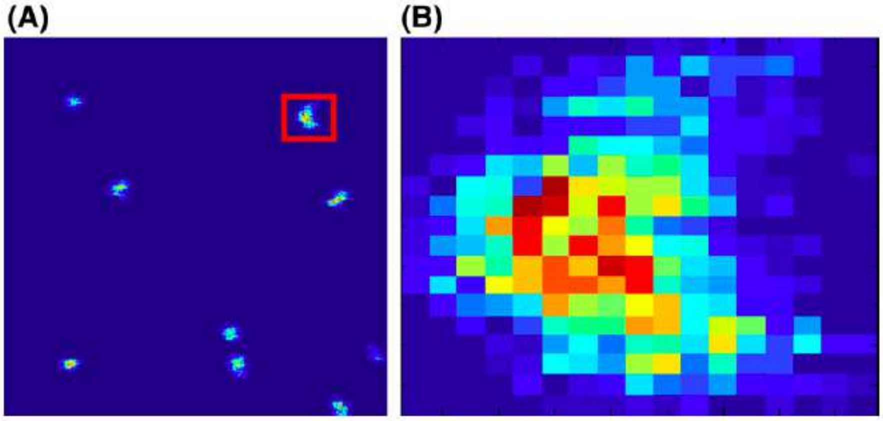
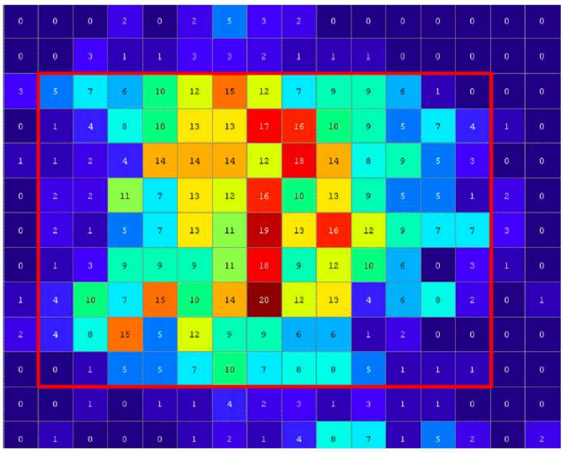


Figure 12.13: Freely diffusing fluorescent beads with 175 nm diameter. (A) A 256×256 pixel raster scan image. (B) The particle highlighted with a red square in (A) extracted from the image.

Let a particle P be defined by the axis-parallell rectangle

$$P = \{(x, y) : a < x < a + L, b < y < b + K\} \tag{12.42}$$

around a local maximum of photon counts. The trajectory of the particle can be estimated by use of the extracted image and used to estimate the diffusion coefficient D of the



0	0	0	2	0	2	5	3	2	0	0	0	0	0	0	0
0	0	3	1	1	3	3	2	1	1	1	0	0	0	0	0
3	5	7	6	10	12	15	12	7	9	9	6	1	0	0	0
0	1	4	8	10	13	13	17	16	10	9	5	7	4	1	0
1	1	2	4	14	14	14	12	18	14	8	9	5	3	0	0
0	2	2	11	7	13	12	16	10	13	9	5	5	1	2	0
0	2	1	5	7	13	11	19	13	16	12	9	7	7	3	0
0	1	3	9	9	9	11	18	9	12	10	6	0	3	1	0
1	4	10	7	15	10	14	20	12	13	4	6	8	2	0	1
2	4	8	15	5	12	9	9	6	6	1	2	0	0	0	0
0	0	1	5	5	7	10	7	8	8	5	1	1	1	0	0
0	0	1	0	1	1	4	2	3	1	3	1	1	0	0	0
0	1	0	0	0	1	2	1	4	8	7	1	5	2	0	2

Figure 12.14: Magnified neighbourhood of a 175-nm bead in a 2048×2048 raster scan image. The numbers correspond to photon counts in each pixel and the local maximum is 20. The levels chosen are $T_1 = 10$ and $T_2 = 5$, and the rectangle defining the particle is found within the contour defined by the red lines. The corresponding border consists of the pixels immediately outside the red contour.

particle. In (Longfils *et al.*, 2017) a maximum likelihood method for estimating the trajectory and the diffusion coefficient D is described, but we will here describe another more direct way to estimate the trajectory and D. Let $t(y)$ denote the time at which we scan the horizontal line at y, and let $N(x, y, t_k)$ denote the measured number of photons for a given particle at the pixel with centre (x, y) at time $t(y) = t_k$, where $k = 0, \ldots, K$. The x position at time $t(y) = t_k$ is estimated by the centroid

$$\psi_k = \frac{\sum_{\{(x,y)\in P: t(y)=t_k\}} N(x,y,t_k)\cdot x}{\sum_{\{(x,y)\in P: t(y)=t_k\}} N(x,y,t_k)}. \tag{12.43}$$

Then

$$\hat{D} = \frac{1}{2\Delta t K}\sum_{k=1}^{K}(\psi_k - \psi_{k-1})^2, \tag{12.44}$$

where $\Delta t = \tau_l$ denotes the time between two consecutive line scans, is a suitable estimator of D, compare (12.4). From the simulated image shown in Figure 12.15, where we know the true trajectory, it can be seen that both the maximum likelihood method (green) and the centroid method (black) give accurate estimates of the true trajectory (red).

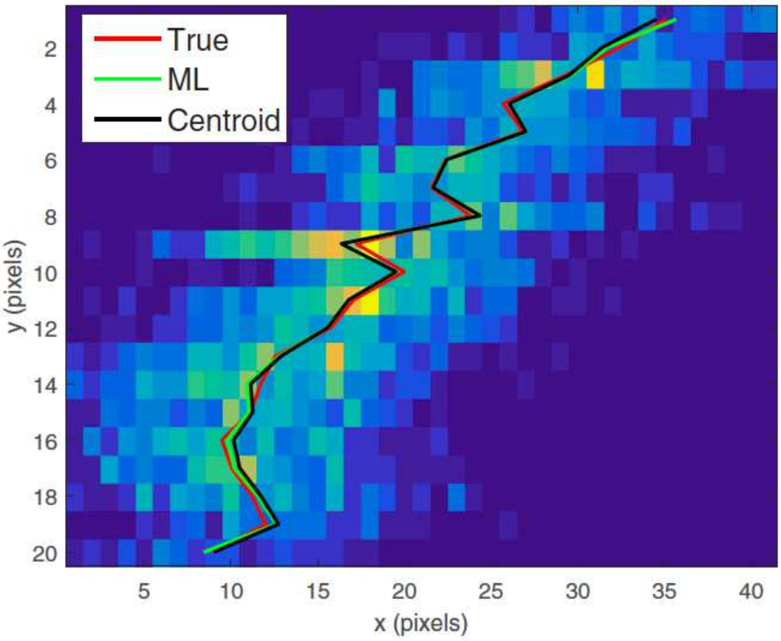


Figure 12.15: A simulated raster scan image of a particle with true trajectory (red), the corresponding estimated trajectory computed with the maximum likelihood method (green) from (Longfils *et al.*, 2017) and with the centroid method (black) from (Longfils *et al.*, 2018), see (12.43).

By use of analyses of images from both simulations and experiments the SPRIA method is in (Longfils *et al.*, 2017) evaluated and also compared with the traditional RICS method. In the simulation study, Gaussian random walk (discrete time Brownian motion) of spheres was generated in a box of with periodic boundary conditions. The sphere diameter ranged from 15 to 1000 nm. A minimum image size of 256×256 was chosen. The number of images simulated in each case varied and was chosen such that at least 300 particles and at least 100 images were collected. Lateral and axial waists of the point spread function were $\omega_0 = 248$ and $\omega_z = 1270$ nm corresponding to Gaussian fit to an average measured z-scan of immobile 175-nm fluorescent beads. The pixel dwell time was $\tau_p = 1.71\times10^{-7}$s, the line time $\tau_l = 1.4\times10^{-3}$ s and the pixel size $S = 0.03\mu$m. The results for 10 D-values between 0.0625 and 64 $\mu\text{m}^2\text{s}^{-1}$ are shown in Figure 12.16 including bootstrap standard deviation estimates obtained by $B = 40$ repetitions, compare (12.29). Results from experiments with particles of four different sizes: 100 nm, 175 nm, 490 nm and 1000 nm, are shown in Figure 12.17. Both Figure 12.16 and Figure 12.17 show that the SPRIA method performs well for the settings chosen.

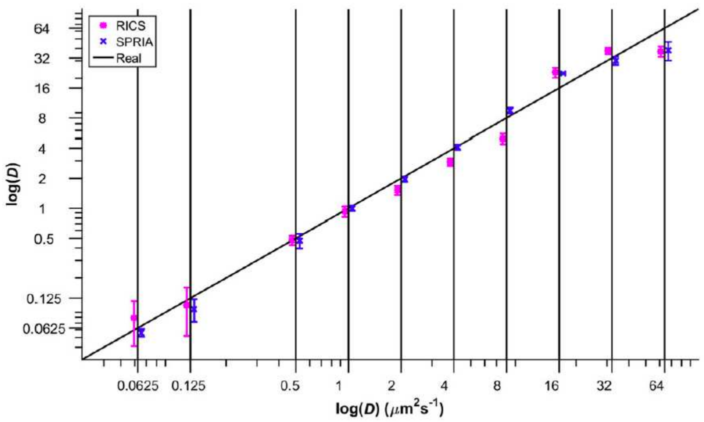


Figure 12.16: Illustration of simulation results with logarithmic scales on both axes. Vertical black lines correspond to the expected diffusion coefficient according to Stoke-Einstein's equation, and ideally the estimated D-values should be located at the crossings of the black identity line and the vertical black lines. Blue markers refer to SPRIA and magenta to RICS, and both estimates are presented as estimates ± standard errors. Markers have been moved horizontally relative to each true value of the diffusion coefficient to make the figure more legible.

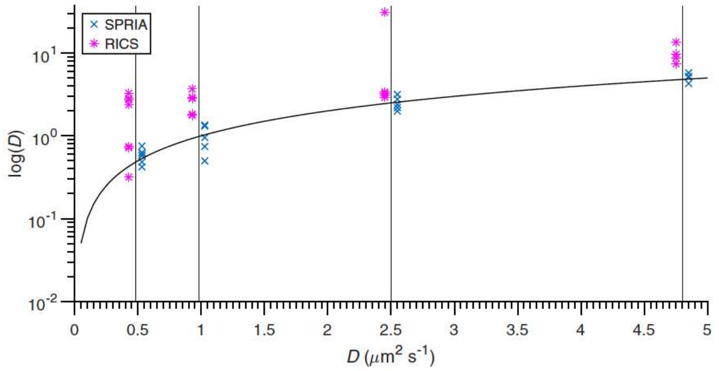


Figure 12.17: Illustration of experimental results for the SPRIA (blue markers) and the RICS (magenta markers) diffusion-coefficient estimations methods with fluorescent beads. A logarithmic scale is uesd on the y-axis. Vertical black lines correspond to the expected diffusion coefficient according to Stoke-Einsteins's equation, and ideally the measured D-values should be located at the crossings of the black logarithmic curve and the vertical black lines. Markers have been moved horizontally to make the figure more legible.

To indicate that the SPRIA method may be used to analyse diffusion in heterogenous media a simple simulation example was performed in (Longfils *et al.*, 2017) with results shown in Figure 12.18. Here the dynamics of particles is supposed to vary such that the diffusion coefficient is 0.8 μm^{2}s^{-1} inside a circle and 0.4 μm^{2}s^{-1} outside the circle. In the simulation 2142 particles in 300 images were found. The pixel size was 0.03 μm and the image resolution was 256×256 pixels. The mobility map shown in the right part of Figure 12.18 was obtained by smoothing with a Gaussian kernel with bandwidth $\sigma = 15$ pixels. The mobility map indicates that the SPRIA method allows recovery of medium heterogeneity to a certain extent.

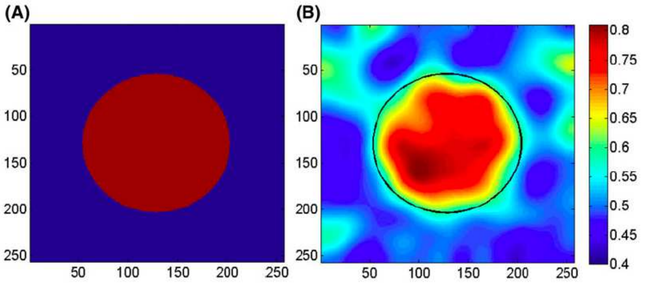


Figure 12.18: (A) Image of a heterogenous sample used in a simulation study with pixel colours corresponding to the expected diffusion coefficient in μm^{2}s^{-1}. (B) Mobility map, smoothed with a Gaussian kernel with bandwidth $\sigma = 15$ pixels, based on 2142 particles in 300 images with the boundary circle between the two media in black. The pixel size is 0.03 μm and the resolution of the images is 256×256 pixels.

What are the conditions for using the SPRIA method? This is discussed in detail in (Longfils *et al.*, 2017). Roughly two conditions should be satisfied. Firstly, the density of particles must not be too large if individual particles should be identifiable. Secondly, sampling time between lines should be such that adjacent horizontal particle lines should differ to some extent, but not too much as particles then become split into several parts. This second condition is illustrated in Figure 12.18. Particularly the lower right part (D) shows a situation where the detection algorithm will split the bead into several (presumably five) particles.

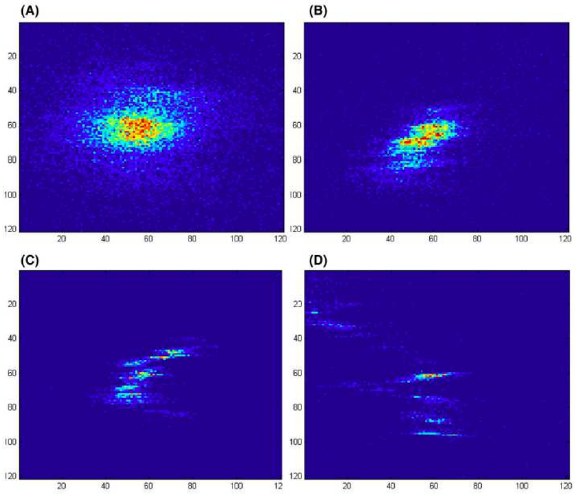


Figure 12.19: Typical particle images with settings: $S_x = 0.03\mu\text{m}$, $\tau_p = 7.5 \cdot 10^{-7}$ s, $\tau_l = 2.5 \cdot 10^{-3}$ s, and scan rate 400 Hz. (A) 1000-nm bead; (B) 490-nm bead; (C) 175-nm bead; (D) 100-nm bead. The extraction step will identify exactly one bead in (A) and in (B). In (C) the bead may be split into two particles, as the last four to six lines are not connected to the previous. In (D) the algorithm will (depending on the thresholds used) split the bead into presumably five particles (corresponding to the following maxima: one in the top left, and four in the middle lower part of the image separated by lines with low photon counts).

12.5.1 Exercises

Exercise 12.10. Figure 12.12 shows what happens when we scan with different scan rates. If we instead sample at constant rate but have particles (of the same kind and size) in three different media with high, medium and low viscosity we would get similar results. If we have low viscosity, which of the three cases (A), (B) and (C) would that correspond to?

Exercise 12.11. In Figure 12.14 we use thresholds $T_1 = 10$ and $T_2 = 5$. Suppose that we change thresholds to $T_1 = 15$ and $T_2 = 10$. How much will that change the estimated diffusion coefficient (in percent) for the particle shown?

Chapter 13

Image analysis of transmission electron micrographs

13.1 Identification of three-dimensional gel microstructures

Mass transport in gels depends crucially on local properties of the gel network. In (Nisslert *et al.*, 2007) a method for identifying the three-dimensional gel microstructure from statistical information in transmission electron micrographs (TEM) is suggested. The gel-strand network is modelled as a random graph with nodes and edges (branches). The distribution of edge length, the number of edges at nodes and the angles between edges at a node are estimated from transmission electron micrographs. The 3D gel network is simulated by Markov chain Monte Carlo (MCMC) methods based on statistical information found from the micrographs. The micrographs can be viewn as projections of stained gel-strands in slices, and a formula is derived for estimating the thickness of the stained gel slice based on the total projected gel-strand length and the number of times that gel-strands enter or exit the slice.

To find relevant features of a gel-strand network we aim at separating the gel-strands from the background, and to create a skeleton showing the network as thin lines. In Figure 13.1 we see in the left part a TEM micrograph of a Sepharose gel to be analysed. The image corresponds to an area of about 1700×1700 nm and shows the microstructure of the gel network at a magnification of 10 000 times. Images are saved as 8-bit grey-scale images of size 1024×1024 pixels. One could hope for directly finding the gel-strand network by thresholding, compare Figure 1.18, but the histogram in Figure 13.2 shows that that seems difficult.

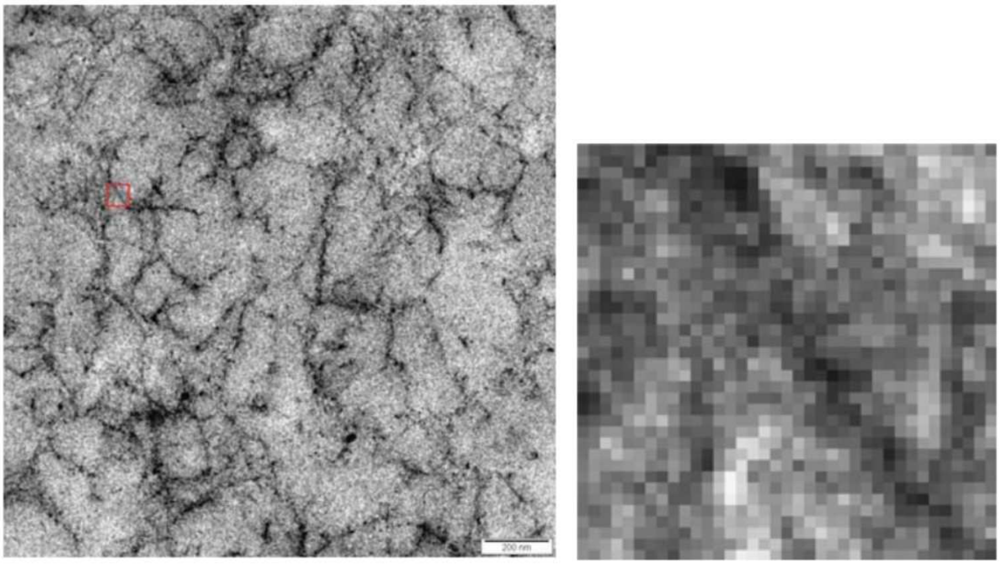


Figure 13.1: Left: one of the 1024×1024 pixel TEM micrographs showing a Sepharose gel at 10 000 times magnification. Right: magnification of the area within the red box in the upper left part of the left image.

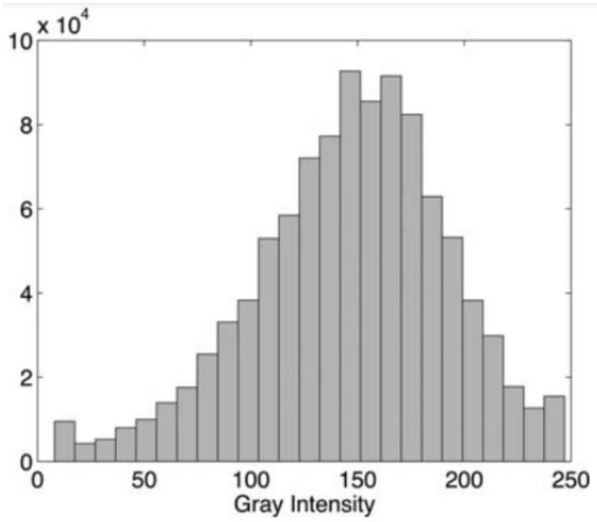


Figure 13.2: Histogram of the grey level intensity of the micrograph in the left part of Figure 13.1.

To find the gel-strand network a series of image operations are performed as shown in Figure 13.3. We will here give a brief description of the different steps illustrated in this figure, and refer to (Nisslert *et al.*, 2007) for more details and litterature references. The upper left image in Figure 13.3 shows the starting image. To decrease the level of noise a Wiener filter is used to smooth the image. The Wiener filter is a low-pass filter that uses prior information about the noise in the image to optimize the noise reduction. As the intensities of pixels that represent the gel-strands vary considerably, a minimum filter is then applied. This takes the minimum pixel value from a 3×3 neighbourhood, resulting in an overall darker image with a more uniform grey-level polymer structure as shown in the right part in the upper figure row. The minimum filter also produces some new dark spots in the background, and to suppress these a 5×5 median filter is applied.

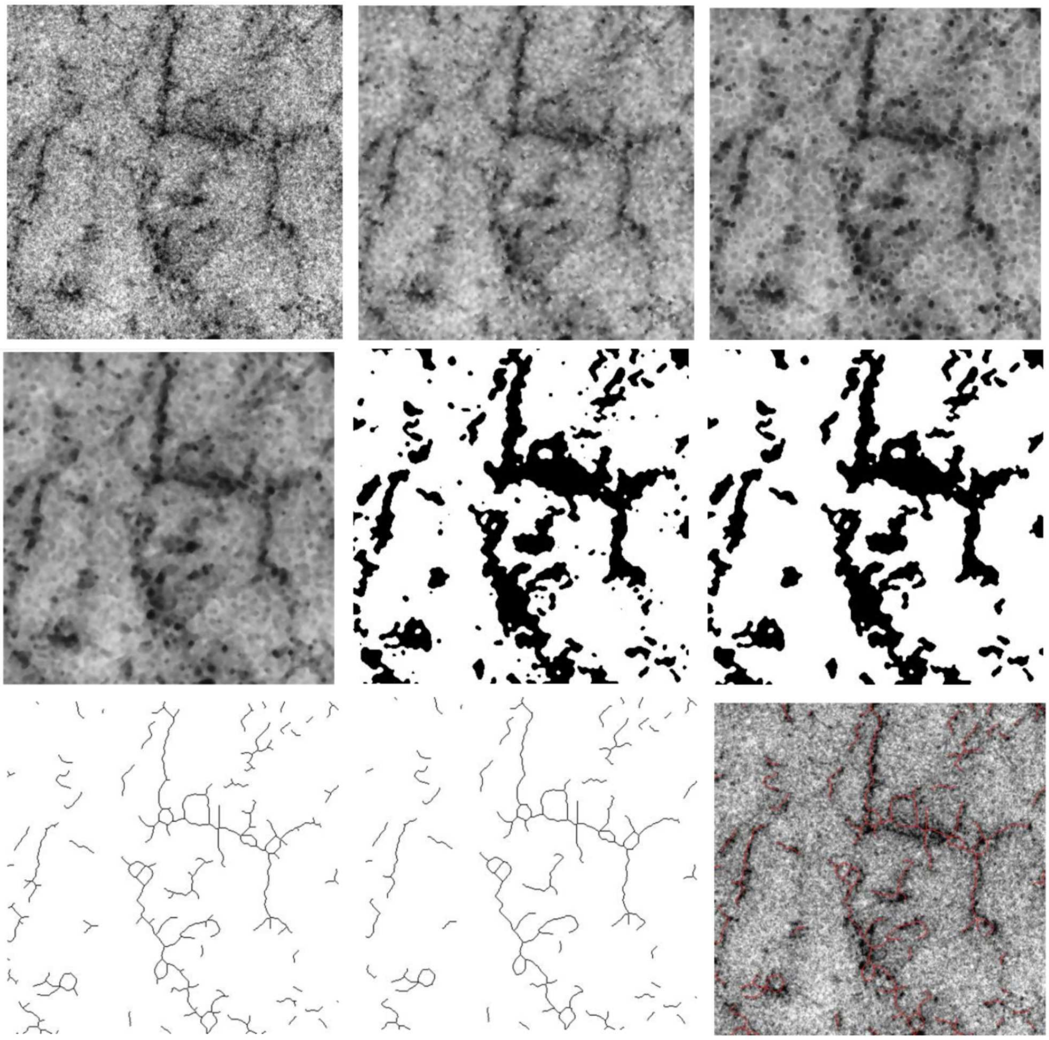


Figure 13.3: Illustration of the sequence of image analysis steps used to find the gel-strand network. In the three upper images we see (from left to right): a 400×400 pixel part of the image to left in Figure 13.1, the same image after application of a Wiener filter, and the result of applying a minimum filter. In the middle row we see similarly (from left to right): the result after application of a 5×5 median filter, the binary image after thresholding, and the result after removal of small black spots. The lower row shows (from left to right): the skeleton created from the binary image, a skeleton with branches shorter than 10 pixels removed, and finally to the lower right the resulting skeleton displayed in red onto the original image.

The gel-strand network is now somewhat better separated from the background. Thresholding is still problematic, but possible. The method of (Otsu, 1979) is used with the threwsholds determined by minimizing the intraclass variance between dark and light pixel values. In the thresholding the image was further divided in 25 different parts with individual thresholds to take care of varying intensity in different parts of the image. The threshold values from Otsu's method are also decreased with 4% to get the gel-strands thinner. Further black spots smaller than 80 pixels are removed as shown in the right image in the middle row of the Figure 13.3.

Having now a binary image which reasonably well represents the gel structure in the micrographs, the next step is to create a skeleton representation of the network. Two mehods were tried: *medial axis transformation* and *thinning*, (Sonka *et al.*, 2015). Both methods create skeletons that represent the microstructure of the gel network rather well, but a lot of small branches are created without correspondence in the original micrographs. Comparing skeletons produced by the two methods, thinning was found to produce less of these artefacts and was chosen. Branches shorter than 10 pixels are also removed from the skeletons. The resulting network is shown in the middle image of the lower row in Figure 13.3. Finally in the lower right part of the figure we see the resulting skeleton in red overlaid on the original image.

By looking at the locations in the gel-strand network where branches have been removed we can find a special type of nodes called bending points, as the removed short branches were often created where the gel strands change direction. In the left part of Figure 13.4 part of a skeleton is shown with nodes (with three branches) marked in red, end points in green and bending points in blue. The bending points are in the sequel treated as nodes having only two connected branches.

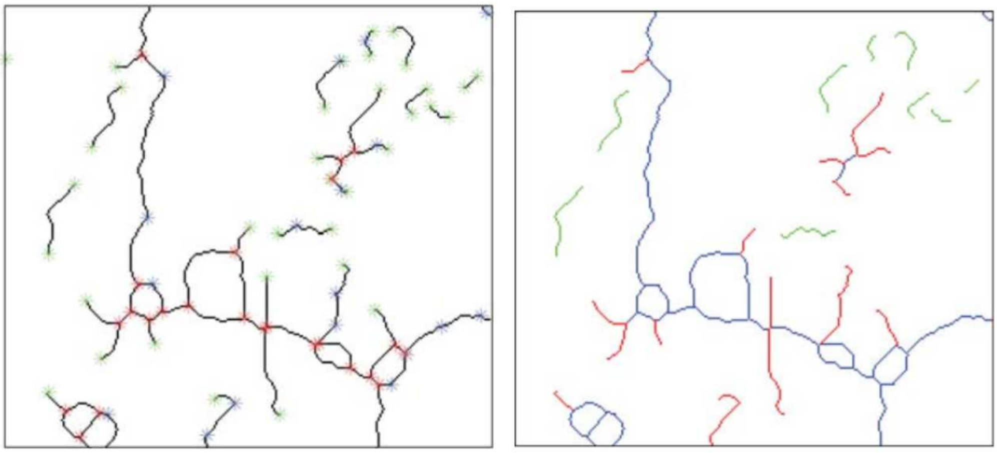


Figure 13.4: Left: Part of a micrograph skeleton with nodes (with three branches) marked in red, end points in green and bending points in blue. Right: Skeleton with branches between nodes marked in blue, branches between end-points in green and branches between an end point and a node in red.

By using the end points and nodes we can measure branch lengths. This is done by measuring the straight line distance between two end points, two nodes or one end point and one node. The branches are also measured more accurately by looking at the distance between the pixels in the branches, where the distance between horizontally or vertically adjacent pixels is one and between diagonally adjacent pixels is $\sqrt{2}$.

In a skeleton we see, compare the right part of Figure 13.4, three types of branches. Note that we look at projections of a thin section and that we assume full connectivity in the gel network. Branches between between two end points correspond to gel-strands that enter and exit the section. Branches between an end point and a node correspond to gel-strands that enter or exit the section, and branches between two nodes are fully contained in the section.

We want to find the distribution of angles between branches at nodes. As we have a discrete pattern of pixels we cannot just use angles between adjacent pixels as that would give only multiples of 45°. Choosing a point too far from a node increases the probability of an incorrect branch direction. As a comprise the direction eight pixels from a node was used. Further, to find the thickness of gel-strands a binary representation of the gel network and a distance map (Breu *et al.*, 1995) is constructed. The distance map gives the shortest distance for each pixel to the nearest non-gelstrand pixel. Combining this with the skeleton representation of the gel microstructure gives the radius distribution of gel-strands.

Let us now look at slice thickness estimation. To obtain contrast in the TEM images a thin slice of gel is put into a uranium and lead bath. Uranium and lead are allowed to diffuse into the gel for about one hour and are adsorbed on the gel-strands. In (Nisslert *et al.*, 2007) a method is proposed to estimate slice thickness, which corresponds to how far uranium and lead has diffused into the gel. The method as described below is based on the total projected gel-strand length and the number of gel-strand crossings into or out of the slice.

Assume that the gel-strand network consists of a collection of curves such that essentially all end points in the 2D projection correspond to a curve passing into or out of a slice of thickness D. Gel-strand curves are supposed to have locally a well-defined orientation. Figure 13.5 shows a short approximately linear curve segment ΔC of length ΔL that forms an angle α with a vertical line and has vertical distance y from segment midpoint to the bottom of the slice. Assuming that the line segment is randomly rotated in 3D around its center one finds that the angle α has probability density

$$p(\alpha) = \sin\alpha, \quad 0 \leq \alpha \leq \pi/2, \tag{13.1}$$

compare Exercise 13.1. The length of the horizontal projection of the corresponding curve segment ΔC is $L_{\text{hor}}(\Delta C) = \Delta L \sin\alpha$ with expectation

$$\mathbf{E}(L_{\text{hor}}(\Delta C)) = \int_0^{\pi/2} \Delta L \sin\alpha p(\alpha) d\alpha = \frac{\pi \Delta L}{4}, \tag{13.2}$$

compare (Baddeley & Jensen, 2005), p. 172.

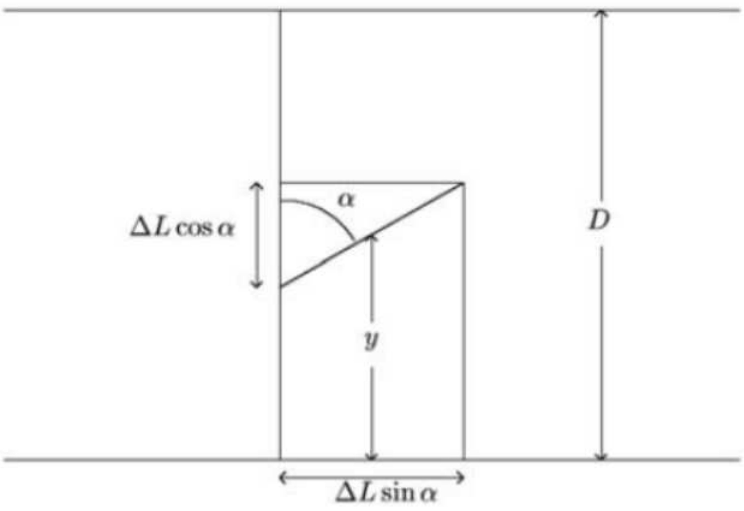


Figure 13.5: Short approximately linear curve segment ΔC of length ΔL with orientation α relative to a vertical line and with a vertical distance y from midpoint to bottom of slice. The slice is oriented such that it has horizontal upper and lower boundaries.

To evaluate the number of crossings we condition on the angle α. The length of the vertical projection of the curve segment is $\Delta L\cos\alpha$, and we get a crossing out of the slice if either $y < (\Delta L/2)\cos\alpha$ or $y > D - (\Delta L/2)\cos\alpha$. Thus the conditional expectation of the number of crossings of curves moving out of the slice is $(1/D)\Delta L\cos\alpha$. Since we should have equally many crossings from curves moving into the slice, the conditional expectation of the total number of crossings is $(2/D)\Delta L\cos\alpha$. Averaging over α we get the expected number of crossings

$$\mathbf{E}(N_{\text{cross}}(\Delta C)) = \frac{2}{D}\int_0^{\pi/2} \Delta L\cos\alpha\sin\alpha d\alpha = \frac{\Delta L}{D}. \tag{13.3}$$

Summing over all curve segments we find from (13.2) that the expected horizontal curve length is

$$\mathbf{E}(L_{\text{hor}}) = \frac{\pi L}{4}, \tag{13.4}$$

where L is the total 3D curve length, and similarly the expected number of crossings is

$$\mathbf{E}(N_{\text{cross}}) = \frac{L}{D}. \tag{13.5}$$

From the last two equations we find

$$\mathbf{E}(N_{\text{cross}}) = \frac{4}{\pi D}\mathbf{E}(L_{\text{hor}}). \tag{13.6}$$

This means that the slice thickness D can be estimated from the horizontal length L_{hor} in the projection and the number of crossings N_{cross} into or out the slice, and we get the estimate

$$\hat{D} = \frac{4L_{\text{hor}}}{\pi N_{\text{cross}}}. \tag{13.7}$$

Let us now see how the 3D microstructure can be modeled as a random graph $G = (V, E)$ and reconstructed by use of MCMC simulation. A random graph consists of a set $V = \{v_1, \ldots, v_N\}$ of nodes (vertices) and a set E of undirected edges between nodes. An edge between two nodes u and v is denoted $\langle u, v \rangle$. In our MCMC modelling of the gel-strand network we let nodes in the graph be either branching points or bending points. An edge $\langle u, v \rangle \in E$ if and only if there is a direct connection (that is a branch) between the nodes u and v.

Define a probability measure for the random graph $G = (V, E)$ by use of a real-valued energy function $f(g)$ such that the probability density at $G = g$ is

$$\pi_f(g) = \frac{1}{Z_f} \mathrm{e}^{-f(g)}, \tag{13.8}$$

where Z_f is a normalizing constant ensuring that the probability density $\pi_f(g)$ integrates to one. The chosen energy function is a sum of three components,

$$f(g) = f_1(g) + f_2(g) + f_3(g), \tag{13.9}$$

corresponding to lengths of edges, number of edges at nodes and angles between edges at nodes, respectively.

The first energy component is

$$f_1(g) = c_1 \sum_{u \sim v} (d_{uv} - d_0)^2, \tag{13.10}$$

where u and v are nodes of g, d_{uv} is the distance between them and $u \sim v$ means that there is an edge between u and v. The target distance d_0 is essentially the mean branch length (in pixels) from the micrographs multiplied with a factor $4/\pi$ to compensate for going from two to three dimensions, compare (13.2). The constant c_1 weighs the importance to the energy of the edge length component.

The second energy component corresponds to the number edges connected to nodes. In the micrographs almost all intersection points have three connected edges, but since bending points that have only two edges are also included, the target number n_0 of connections will be somewhat less than three (about 2.7). An energy component taking this into account is

$$f_2(g) = c_2 \sum_{v} (n_v - n_0)^2, \tag{13.11}$$

where we sum over all nodes v of g and n_v is the number of edges at v.

The third energy component is

$$f_3(g) = c_3 \sum_{v} \sum_{\alpha_v} (\alpha_v - \alpha_0)^2, \tag{13.12}$$

where we sum over all nodes v of g and over all angles α_v, less than π radians, between edges connected to v. With three edges at a node we expect by symmetry angles around 120° degrees, but for nodes with two edges we expect somewhat larger angles.

In the reconstruction of the gel network a sequence of graphs, $G_n = (V_n, E_n), n = 0, 1, \ldots, n_{\max}$, is updated with a Metropolis algorithm, compare Section 4.7. The updating from G_n to G_{n+1} is performed with two types of steps. In the first type of steps we add or remove an edge to E_n (or leave E_n unchanged). In the second type we move one of the nodes of V_n a random distance (or leave V_n unchanged). The number N of nodes in V_n is kept constant. The two types of steps are performed as follows.

Type 1 step. Pick a random pair $\{u, v\}$ of nodes. If $\langle u, v\rangle \in E_n$ we first remove it. To obtain E_{n+1} either add $\langle u, v\rangle$ or leave the set of edges unchanged according to the conditional π_f distribution given V_n and all other edges of E_n.

Type 2 step. Pick a node $v \in V_n$ and sample a random movement of the position of the node v. The random movement ΔR is sampled from a uniform distribution in a 3D sphere with radius 2 pixels. Consider the conditional π_f distribution given all other nodes of V_n and the set of edges E_{n+1}, and choose according to this distribution V_{n+1} either equal to V_n or equal to V_n with the chosen node moved by ΔR.

For further details of the MCMC simulation, including choice of the weights (c_1, c_2, c_3) and the target values (d_0, n_0, α_0) in the energy function, the reader is referred to (Nisslert *et al.*, 2007). As always in this type of simulations a large number of updates is needed.

Let us now show some results obtained from the image analysis. In Figure 13.6 we see distributions of branch lengths, of angles between branches and of gel-strand thickness.

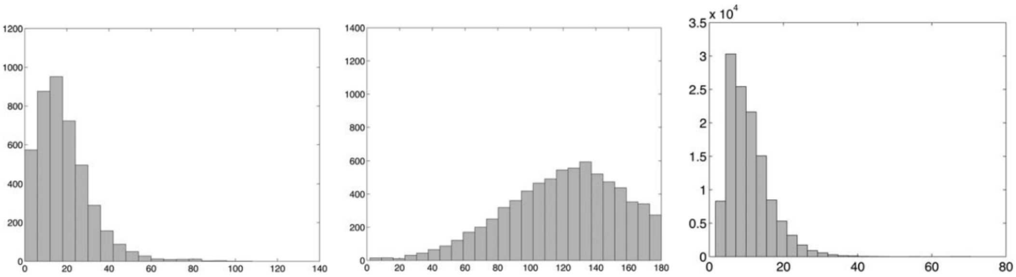


Figure 13.6: Left: Distribution of branch lengths (in nm) between nodes, one pixel corresponds to 1.66 nm. Middle: Distribution of angles (in degrees) between branches at nodes. Right: Distribution of gels strand thickness (gel-strand radius in nm).

The slice thickness of the section, from which the micrographs are projections, was estimated by use of (13.7). The total projected length was obtained as the sum of all branch lengths, and the total number of crossings was obtained as the total number of end points in the skeleton (excluding end points at borders). From six images this resulted in a thickness estimate of 90.6 nm with a standard error of 3.5 nm. A minor reduction of the thickness estimate was used, for details see (Nisslert *et al.*, 2007), where it was also concluded that uranium and lead had penetrated most of the slice.

Let us now briefly look at the results from the MCMC simulations which were performed in an approximately 800×800×800 nm cubic box with periodic boundary conditions. With a gel slice thickness of 90 nm, the node intensity per volume unit was found, corresponding to $N = 1604$ in the cubic box and this N value was used in the simulations. To achieve approximate stationarity in the MCMC simulations 200 hundred iterations were used. In each of these iterations first 10 000 iterations were performed with connections between nodes created or removed (see *Type 1 step* above), and then 10 000 iterations were performed with movement of nodes (see *Type 2 step* above). For further details of the MCMC simulations, see (Nisslert *et al.*, 2007). After obtaining the skeleton, the branches were thickened to obtain a volume percent of 4%. A result with a 3D rendering of the simulated gel network is shown in Figure 13.7.

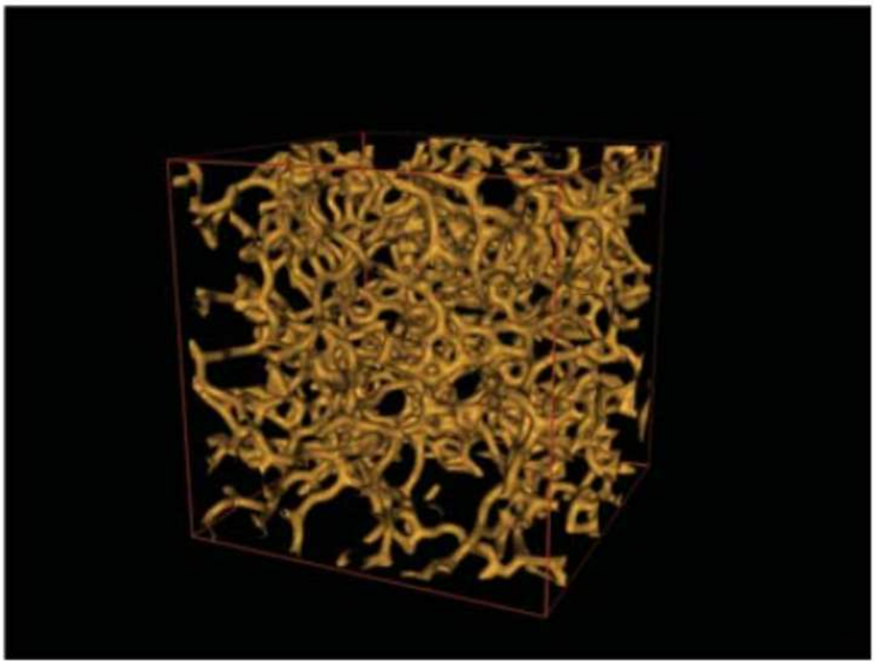


Figure 13.7: Screen shot of a 3D rendering of a simulated gel network with constant gel-strand radius of 9.2 nm, in a cube with 500 pixels side corresponding to 830 nm.

To evaluate the image processing algoritms and the MCMC simulation algorithms we compare in Figure 13.8 a micrograph with computed skeleton (red) to the left with a simulated "micrograph" also with a computed skeleton (red) to the right. Visually the left and right part look rather similar.

Finally, in Figure 13.9 we show (in black) the projection of an actual simulated 3D skeleton and (in red) the corresponding skeleton obtained from image processing. The skeletons look quite similar, indicating that the image processing actually succeeds in creating a skeleton representation from the micrographs.

13.1.1 Exercise

Exercise 13.1. Consider the approximately linear curve segment ΔC in Figure 13.5. Show that if the curve segment is randomly rotated in 3D the angle α has probability density given by (13.1).

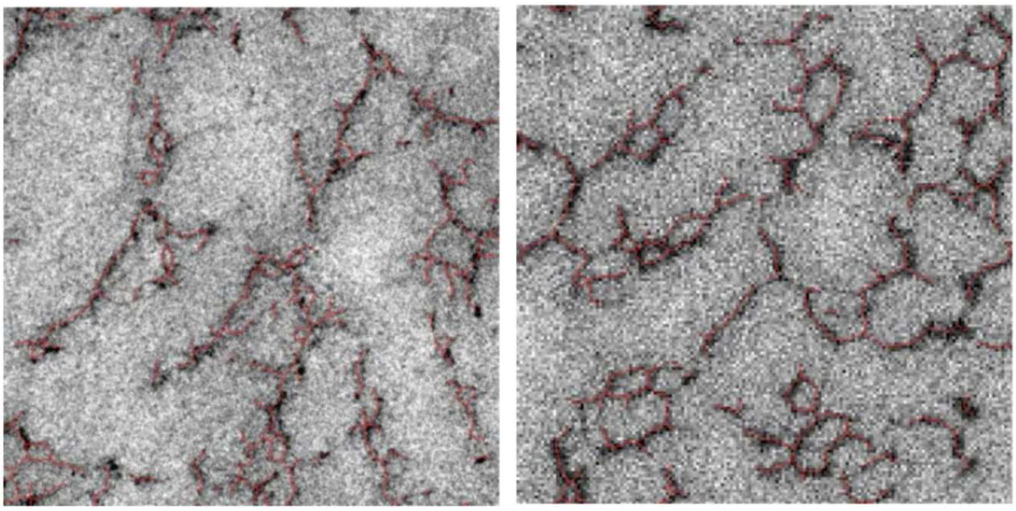


Figure 13.8: Left: Part of a micrograph, 500×500 pixels, with corresponding skeleton marked in red. Right: Simulated ”micrograph”, 500×500 pixels, with corresponding skeleton marked in red.

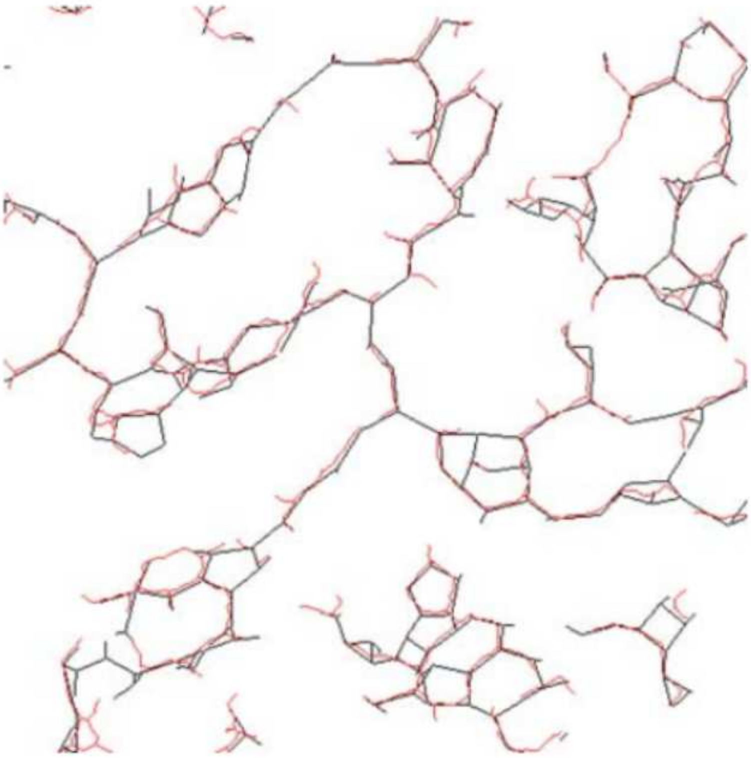


Figure 13.9: Projection of the actual simulated 3D skeleton (displayed in black), size 500×500 pixels, compared with the corresponding skeleton obtained from the image processing (displayed in red).

13.2 Structural characterization from scanning transmission electron micrographs

In (Nordin *et al.*, 2014) material characterization from high angle annular dark field scanning transmission electron micrograph (HAADF-STEM) images is studied by use of maximum likelihood methods. The upper left part of Figure 13.10 shows the intensity response in 2D from a model material consisting of 20nm silica particles that have aggregated to form a stable particle gel. The approximately square image is obtained by scanning a gel slice of thickness about 90 nm. The mass thickness $\alpha(x, y)$ of the silica at a 2D position (x, y) can be written as fraction of the total sample thickness

$$\alpha(x, y) = \frac{1}{z_{\max}} \int_0^{z_{\max}} \rho_S(x, y, z)\, dz, \tag{13.13}$$

where $\rho_S(x, y, z)$ denotes the silica density (depending on whether there is a silica particle at the point (x, y, z) or not). A simple model for the observed intensity $I(x, y)$ is

$$I(x, y) = b + cg(\alpha(x, y)) + \epsilon(x, y), \tag{13.14}$$

where b and c are constants, g is a power function

$$g(\alpha) = \alpha^{\beta}, \tag{13.15}$$

and the noise $\epsilon(x, y)$ is supposed to be $N(0, \sigma^2)$ and independent for different pixels (x, y).

Let us give a maximum likelhood function for an image such as the one shown in the upper left part of Figure 13.10. Suppose that in the corresponding 3D rectangular box, compare Figure 12.3, there are N silica particles with 3D centres $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Then the following log-likelihood function corresponds to the model (13.14)

$$\ell(\theta) = -|M| \log\left(\sqrt{2\pi}\sigma\right) - \frac{1}{2\sigma^2} \sum_{(x,y) \in M} \left[I_0(x, y) - b - cg(\alpha(x, y))\right]^2, \tag{13.16}$$

where M is the set of pixels, $|M|$ is the number of elements in M and the parameter vector is $\theta = (b, c, \beta, N, \mathbf{x}_1, \ldots, \mathbf{x}_N)$. A complication in finding maximum likelihood estimates is the large number particles, see Figure 13.10, but it can be done (at least approximately), for instance by use of simulated annealing, compare (Nordin *et al.*, 2014).

The maximum likelihood estimate of the power parameter β in (13.15) was $\hat{\beta} = 0.69$, which gave a considerably better fit compared to the linear response with $\beta = 1$ as shown in Figure 13.11. The models with $\beta = 1$ and a general β are nested and the hypothesis $\beta = 1$ can be tested for instance by use of repeated images. In the present case there was only one image available. To obtain approximately independent repetitions the image was divided into 16 subimages of equal size by three vertical and three horizontal lines. With the corresponding 16 estimates of β a t-test showed rejection of the null hypothesis $H_0 : \beta = 1$ with a p-value much less than 0.001.

As a check of the model (13.14) histograms of the pixel intensity in the observed and estimated micrograph images are shown in Figure 13.12. In general there is a good agreement between the observed and the estimated image intensities. However, the histograms indicate that a feature not taken into account of the model is that the edges of the observed spheres are somewhat blurry.

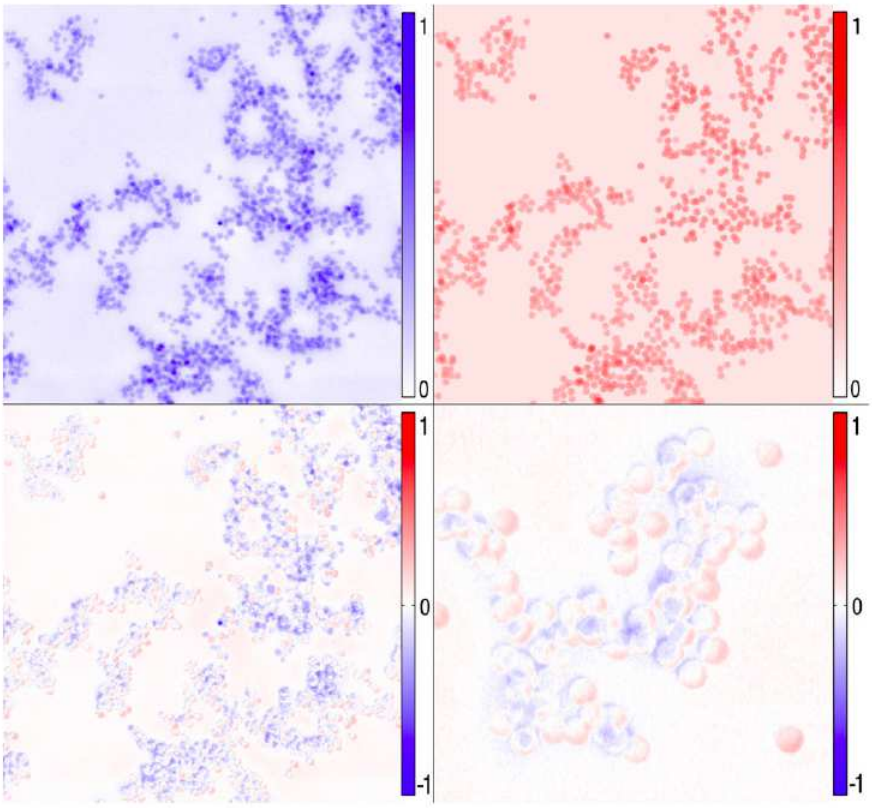


Figure 13.10: Upper left: Micrograph intensity image $I_0(x, y)$ of a 90 nm slice of 5 wt% aggregated nano silica obtained with HAADF-STEM. Upper right: Regenerated micrograph image $I(x, y)$ obtained by maximizing the log-likelihood function (13.16). Lower left: Residual image $I(x, y) - I_0(x, y)$ of the original micrograph and the regenerated micrograph. Lower right: Blow-up of the top left cluster in the residual image.

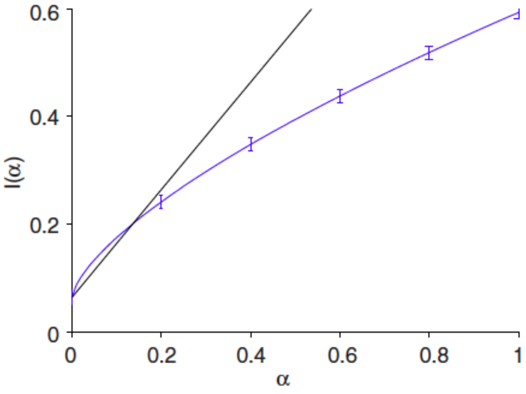


Figure 13.11: The intensity response I versus mass thickness α of silica as estimated by maximizing the likelihood function (13.16). The micrographs are STEM projections of 90 nm thick microtomed sample sections, where the individual silica spheres are 20 nm in diameter, compare Figure 13.10. One single sphere corresponds to $\alpha = 0.22$. The blue line shows the estimate with the estimated power $\beta = 0.69$ in (13.15), while the black line corresponds to $\beta = 1$. The error bars show the estimated standard deviation of the background noise.

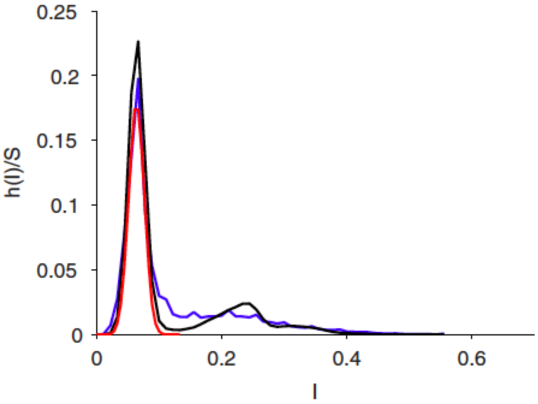


Figure 13.12: Histogram of the pixel intensity for the observed micrograph (blue) and the estimated micrograph intensity (black). The estimated background noise is also shown (red). Note that the observed micrograph image is blurry at the edges of the spheres, compare the lower right part of Figure 13.10. This is not taken into account in the model, which can explain why the estimated intensity shows an underestimate just between $I = 0.1$ and 0.2.

13.2.1 Exercises

Exercise 13.2. Verify that (13.16) gives the log-likelihood corresponding to the model (13.14).

Exercise 13.3. How well can the coordinates of the 3D centre positions $\mathbf{x}_1, \ldots, \mathbf{x}_N$ be estimated?

Exercise 13.4. Try to count approximately the number of particles in the left and the right upper parts of Figure 13.10.

Exercise 13.5. Try to count approximately the number of overlapping particles in the left and the right upper parts of Figure 13.10.

Chapter 14

Appendix. Mathematical, computational and statistical background

Below you can find condensed descriptions of concepts and methods used in this book. If you have a basic knowledge of some area these descriptions can serve as a repetition, but if some concepts are new to you, you presumably need to go to textbooks for more complete information. Nowadays quite useful information can also be obtained from the internet, for example from the Wikipedia pages.

14.1 Some matrix algebra

A matrix with m rows and n columns, or briefly a matrix of type $m \times n$, is a rectangular array

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \tag{14.1}$$

of numbers $a_{i,j}$, sometimes written a_{ij}, called matrix elements. If the type is understood we can write $A = [a_{i,j}]$. Row and column vectors are thin matrices with $m = 1$ and $n = 1$, respectively. If $m = n = 1$ the matrix is just a number. A square matrix has $m = n$.

Let A be an $m \times n$ matrix. The transpose A^T of A is an $n \times m$ matrix obtained by making rows in A into columns, that is the (i, j) element in A^T is the (j, i) element in A. A matrix is symmetric if it equal to its transpose.

Matrices of the same type can be added by element-wise addition. If A and B are matrices of types $m \times n$ and $n \times k$, respectively, the product $C = AB$ is a matrix type $m \times k$ with elements $c_{i,j} = \sum_r a_{i,r} b_{r,j}$. A square $n \times n$ matrix A is called invertible (or non-singular) if there exists an inverse denoted A^{-1} such that

$$AA^{-1} = A^{-1}A = I \tag{14.2}$$

where I is the unit $n \times n$ matrix with diagonal elements $i_{j,j} = 1$ and off-diagonal elements $i_{j,k} = 0, j \neq k$.

Let us now define recursively the determinant $\det A$ of a square $n \times n$ matrix $A = [a_{i,j}]$. For $n = 1$ we define $\det A = a$ for the matrix $A = [a]$. Suppose that we have defined determinants for matrices of type $(n-1) \times (n-1)$ and let A be a matrix of type $n \times n$. Let the minor $A_{i,j}$ be the determinant of the matrix obtained from A by deleting row number i and column number j. Then we put

$$\det A = \sum_{j=1}^{n} (-1)^{1+j} a_{1,j} A_{1,j}. \tag{14.3}$$

One can show that a square matrix A is non-singular if and only if $\det A \neq 0$.

Let A be a square matrix. We say that a real number λ is an eigenvalue of A and that a column vector x is an eigenvector of a if

$$Ax = \lambda x. \tag{14.4}$$

A symmetric real $n \times n$ matrix A is said to be positive-definite or positive-semidefinite if $x^T A x > 0$ or $x^T A x \geq 0$, respectively, for each non-zero n-dimensional column vector x. One can show that a symmetric matrix is positive-definite or positive-semidefinite if all its eigenvalues are positive or nonnegative, respectively. Further, a positive definite matrix is invertible.

Exercises

Exercise 14.1. Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Determine $\det A$ by use of (14.3).

Exercise 14.2. Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ with $ad - bc \neq 0$. Determine the inverse of A by solving a linear equation system with four unknowns.

14.2 Optimization of a real funtion

Let us first consider Newton's method for optimization of a twice continuously differentiable real-valued function $f(x)$ of a real variable x. Suppose that f has a maximum or minumum at $x^\star$. Then $f'(x^\star) = 0$. Newton's iterative method for locating $x^\star$ is to put

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}. \tag{14.5}$$

Assuming that $f''(x^\star) \neq 0$ and that we start close enough to $x^\star$ one can show that $x^k \to x^\star$ as $k \to \infty$.

Let us now consider Newton's method for optimization of a twice continuously differentiable real-valued function $f(x)$ of an n-dimensional column vector x. As above we suppose that f has a maximum or minumum at $x^\star$. Let $\nabla f(x)$ denote the (column) gradient vector

$$\nabla f(x) = [\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n}]^T \tag{14.6}$$

and let $Hf(x)$ denote the Hessian matrix

$$Hf(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \tag{14.7}$$

Newton's iterative method for locating $x^\star$ is to put

$$x^{k+1} = x^k - (Hf(x^k))^{-1} \nabla f(x^k) \tag{14.8}$$

Assuming that $Hf(x^\star)$ is positive-definite and thus invertible, and that we start close enough to $x^\star$ one can show that $x^k \to x^\star$ as $k \to \infty$.

Newton's method is quite efficient but has drawbacks. Computation of derivatives can require a lot of programming. One may use finite differences to compute approximate derivatives but that then it requires extra programming to find suitable step lengths. Often it is more efficient to use so called quasi-Newton methods where the Hessian is automatically estimated from successively computed gradient vectors, see for instance (Press *et al.*, 2007). In MATLAB the FMINUNC function uses a quasi-Newton metod for minimization.

The Newton and quasi-Newton methods typically work quite well if you start close to the optimum. A much slower but quite robust optimizer, which does not require computation of any derivates, is the simplex method of (Nelder & Mead, 1965) which is available in MATLAB as the function NELDER_MEAD. A good strategy in applications can often be to begin with the simplex metod to get an overview and suitable starting values and then to use a quasi-Newton method.

14.3 Discrete probability distributions

Discrete distributions for a random variable X are characterized by the probability function $\Pr(X = x)$, $x \in V$, where V is the finite or countable set of values that X can take. For a real-valued discrete random variable the expectation μ, standard deviation σ and variance σ^2 are defined by $\mu = \mathbf{E}(X) = \sum_x x \Pr(X = x)$ and $\sigma^2 = \text{var}(X) = \sum_x (x - \mu)^2 \Pr(X = x)$.

A random variable X is said to be Poisson distributed with parameter λ if

$$\Pr(X = n) = \frac{\lambda^n}{n!} \exp(-\lambda), \;\; n = 0, 1, \ldots, \tag{14.9}$$

and for such a variable both the expectation and the variance are equal to λ.

A random variable X is said to be binomial (n,p) if

$$\Pr(X = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}, \;\; k = 0, \ldots, n, \tag{14.10}$$

and for such a variable the expectation is np and the variance is $np(1-p)$.

14.4 Continuous probability distributions

Continuous distributions for a real-valued random variable X are characterized by the probability density

$$f(x) = \frac{d}{dx}\Pr(X \leq x), \;\; x \in \mathbb{R}, \tag{14.11}$$

where $\mathbb{R} = (-\infty, \infty)$ is the set of real numbers. For a continuous random variable the expectation μ, standard deviation σ and variance σ^2 are defined by $\mu = \mathbf{E}(X) = \int_{\mathbb{R}} xf(x)dx$ and $\sigma^2 = \mathrm{var}(X) = \int_{\mathbb{R}}(x-\mu)^2 f(x)dx$.

A random variable X is said to have a uniform distribution on the interval (a, b) if the probability density is

$$f(x) = 1/(b-a), \;\; a < x < b, \tag{14.12}$$

and $f(x) = 0$ for $x < a$ and $x > b$, and for such a variable the expectation is $(a+b)/2$ and the variance is $(b-a)^2/12$.

A random variable X is said to have an exponential distribution with parameter β if the probability density is

$$f(x) = \beta \exp(-\beta x), \;\; x > 0, \tag{14.13}$$

and $f(x) = 0$ for $x < 0$, and for such a variable the expectation is $1/\beta$ and the variance is $1/\beta^2$.

A random variable X is said to be normal(μ,σ^2), or briefly $X \sim N(\mu,\sigma^2)$ if the probability density is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-(x-\mu)^2/\sigma^2), \;\; x \in \mathbb{R}, \tag{14.14}$$

and for such a variable the expectation is μ and the variance is σ^2.

14.5 Multivariate probability distributions

Let $X_1, \ldots, X_d$ be real-valued random variables. Then $X = [X_1 \ldots X_d]^T$ is a d-dimensional random (column) vector. The expectation of a random vector (or a random matrix) is defined componentwise. Thus the expectation vector $\mu = \mu_X = \mathbf{E}(X)$ of a random column vector X is the column vector with components $\mu_i = \mathbf{E}(X_i), i = 1, \ldots, d$. The covariance matrix $C = C_X = C(X)$ of X is the symmetric $d \times d$ matrix

$$C = \mathbf{E}(X-\mu)(X-\mu)^T = \begin{bmatrix} E(X_1-\mu_1)(X_1-\mu_1) & \cdots & E(X_1-\mu_1)(X_d-\mu_d) \\ \vdots & & \vdots \\ E(X_d-\mu_d)(X_1-\mu_1) & \cdots & E(X_d-\mu_d)(X_d-\mu_d) \end{bmatrix}. \tag{14.15}$$

The (i, j)-element of the covariance matrix of X is the covariance $\mathrm{cov}(X_i, X_j) = E(X_i - \mu_i)(X_j - \mu_j)$ of the ith and jth components of X, which for $i = j$ is the variance of X_i.

The d-dimensional vector X has a d-dimensional probability density $f = f_X$ if

$$\Pr(X \in A) = \int_A f(x)dx \tag{14.16}$$

for subsets A of d-dimensional space $\mathbb{R}^d$ for which the integral in (14.16) is well-defined.

Let μ be a d-dimensional column vector and let C be a positive-definite $d \times d$ matrix. The d-dimensional random vector X is said to be normal(μ,C) or briefly $X \sim N(\mu,C)$ if X has the d-dimensional density function

$$f_X(x) = \frac{1}{(2\pi)^{d/2}(\det C)^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)), \tag{14.17}$$

where $\det C$ denotes the determinant of the matrix C. One can show that then X has expectation vector μ and covariance matrix C.

An important special case is the two-dimensional normal distribution. Regard $X = [X_1\ X_2]^T$. Let μ_i and σ_i^2 denote the expectation and variance of X_i, $i = 1, 2$, and let $\rho = \mathrm{cov}(X_1, X_2)/(\sigma_1\sigma_2)$ denote the correlation between the two components of X. Thus the covariance matrix of X is

$$C = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}. \tag{14.18}$$

One can then show that the two-dimensional density funcion of X is

$$f(x) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\{-\frac{1}{2(1-\rho^2)} Q(x_1, x_2)\} \tag{14.19}$$

where

$$Q(x_1, x_2) = \frac{(x_1-\mu_1)^2}{\sigma_1^2} - 2\rho(\frac{x_1-\mu_1}{\sigma_1})(\frac{x_2-\mu_2}{\sigma_2}) + \frac{(x_2-\mu_2)^2}{\sigma_2^2} \tag{14.20}$$

14.6 Principal components, t-SNE

Suppose that we have a d-dimensional random vector X with covariance matrix C. Principal components can be used to transform the random vector. Define the first principal component

$$Y_1 = c_1^T X, \tag{14.21}$$

where c_1 is a d-dimensional column vector, determined by the condition that $\mathrm{var}(Y_1) = c_1^T C c_1$ is maximal subject to the restriction $c_1^T c_1 = 1$. Generally we define the ith principal component, $1 < i \leq d$ as

$$Y_i = c_i^T X, \tag{14.22}$$

where c_i is a d-dimensional column vector, determined by the condition that $\mathrm{var}(Y_i) = c_i^T C c_i$ is maximal subject to the restrictions $c_i^T c_i = 1$ and $c_j^T C c_i = 0$ for $1 \leq j < i$. The first two or three principle components are sometimes useful to visualize the distribution of X.

Principle components are often attributed to (Hotelling, 1933) although they are closely related to singular value decomposition which has a much older history. A recent quite effective machine-learning-inspired technique due to (van der Maaten & Hinton, 2008) for visualizing multidimensional distributions in two or sometimes three dimensions is *t-SNE*. The method is used in Figure 2.7, and a concise description of the method is given in (Longfils, 2018).

14.7 Random, Gaussian and Markov processes on the real line

A random process or stochastic process X on the real line consists a set of random variables $X = (X_t)$ indexed by time $t \in T$, where T is a subset of the real line $\mathbb{R}$. We suppose here that T is either a set of consecutive integers or an interval and then we talk about a discrete time or continuous time random process, respectively. The set V of values that X_t can take we call the state space. A real-valued process has the real line or a subset of it as state space. A real-valued random process may be characterized by its mean value function,

$$m_t = \mathbf{E}X_t \tag{14.23}$$

and its covariance function

$$C(s,t) = \mathbf{E}(X_s - m_s)(X_t - m_t). \tag{14.24}$$

A random process is said to be normal or Gaussian if $(X_{t_1}, \dots, X_{t_n})$ has an n-dimensional normal distribution for any choice of time points $t_1, \dots, t_n$. One can show that a Gaussian process is fully specified by its mean value and covariance functions.

A random process (X_t) is said to be stationary if its distribution is invariant under a translation τ, more precisely if for each choice of $n \geq 1$ and $(t_1, \dots, t_n)$ the distribution of the n-dimensional random vector $(X_{t_1+\tau}, \dots, X_{t_n+\tau})$ does not depend on τ. Consider the mean value and covariance functions of a stationary process. The mean value is a constant $m = \mathbf{E}X_t$ and the covariance function can be written as $C(s,t) = \sigma^2 \rho(t-s)$ where the variance $\sigma^2 = C(t,t)$ and $\rho(t)$ is the correlation function.

We say that $(X_t, t \in T)$ is a Markov process if the conditional distribution of X at a future time given the history up to time t only depends on the value of X at the current time t, more precisely if

$$\Pr(X_\tau \in A | X_s, s \leq t) = \Pr(X_\tau \in A | X_t), \;\; t < \tau. \tag{14.25}$$

A discrete time Markov process with finite state space V, for notational simplicity here denoted $V = \{1, \dots v\}$, is determined by its transition probability matrix P which is the $(v \times v)$ matrix with elements

$$p_{ij} = \Pr(X_{t+1} = j | X_t = i), \;\; 1 \leq i, j \leq v. \tag{14.26}$$

A zero-mean autoregressive process (X_t) of order p is recursively generated from

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + \epsilon_t, \tag{14.27}$$

where ϵ_t are independent and identically distributed random variables with zero mean and finite variance σ^2. Often ϵ_t is assumed to be normally distributed. Then X_t is also normally distributed, provided that starting values have a (multivariate) normal distribution. An autoregressive process of order $p = 1$ is a Markov process. An autogressive process of order one is stationary if $|a_1| < 1$ and the starting value in (14.27) is suitably chosen.

An example of a continuous time Markov process is the Poisson process with intensity λ which is characterized by the fact that the increment $X_t - X_s$ is Poisson distributed with expecation

$$\mathbf{E}(X_t - X_s) = \lambda(t - s), \;\; s < t, \tag{14.28}$$

and the increments over disjoint time intervals are independent.

Suppose that points are randomly placed on the real line such that

(i) the number of points in disjoint intervals are independent,

(ii) the probability that two points are placed in an interval of length h tends to zero faster than the probability that one point is placed in the same interval when $h \to 0$,

(iii) the distribution of the number of points in an interval depends only on the length of the interval and not on where it is placed.

One can then show that if X_t denotes the number of points in the interval $(0, t)$, then $(X_t, t > 0)$ is Poisson process with intensity λ equal to the expected number of points in an interval of unit length. For an arbitrary time t let further W denote the waiting time for the first point after t. One can then show that W has an exponential distribution with parameter λ.

Another example of a continuous time Markov process is the Brownian motion or Wiener process on the interval $[0, \infty)$ characterized by having independent increments over disjoint time intervals and that X_t is normal$(0, \sigma^2 t)$ for $t \geq 0$.

A third example of a continuous time Markov process is the Ornstein-Uhlenbeck process, which is Gaussian process with mean zero and correlation function

$$\rho(t) = \exp(-\lambda t) \tag{14.29}$$

for some positive constant λ.

14.8 Estimation of parameters. Likelihood and least squares

Suppose that we observe a random variable or vector X with a distribution that depends on a parameter θ that may be a vector. Let $\hat{\theta} = \hat{\theta}(X)$ be an estimate of θ. We say that $\hat{\theta}$ is an unbiased estimate of θ if

$$\mathbf{E}(\hat{\theta}) = \theta. \tag{14.30}$$

Typically we observe a sample of a random variable which means that we have a sequence of independent and identically distributed random variables. We say that $\hat{\theta}$ is a consistent estimate of θ if for an arbitrary $\epsilon > 0$

$$\Pr(|\hat{\theta} - \theta| > \epsilon) \to 0 \tag{14.31}$$

as the number n of observations goes to infinity. One can for instance show that $\hat{\theta}$ is a consistent estimate of θ if $\mathbf{E}(|\hat{\theta}-\theta|^2) \to 0$ as $n \to \infty$.

Let X be a discrete or continuous random vector that we observe and that has a probability distribution depending on θ. If X is discrete we put $f(x,\theta) = \Pr(X = x)$ and if X is continuous $f(x,\theta)$ denotes the probability density of X. The likelihood value corresponding to an observed value x of X is written

$$L(\theta) = L(\theta|x) = f(x,\theta). \tag{14.32}$$

In particular, if we have a sample $X = (X_1,\ldots,X_n)$ of a random variable assumed to be either discrete with probability function $\Pr(X_i = x_i) = f(x_i,\theta)$ or continuous with probability density $f(x_i,\theta)$ the corresponding likelihood function is

$$L(\theta) = L(\theta|x) = \prod_{i=1}^{n} f(x_i,\theta), \tag{14.33}$$

where $x = (x_1,\ldots,x_n)$.

A maximum likelihood estimate $\hat{\theta}$ of θ is a value that maximizes the likelihood function. In practice it is often more convenient to maximize the log-likelihood function

$$\ell(\theta) = \log(L(\theta)), \tag{14.34}$$

where log (as always in this book) denotes the natural logarithm.

As an example, suppose that $X = (X_1,\ldots,X_n)$ is a sample of a variable that is Poisson distributed with parameter λ, that is $X_1,\ldots,X_n$ are independent and identically Poisson distributed. The log-likelihood function is

$$\ell(\lambda) = \log(\prod_{i=1}^{n} \frac{\lambda^{X_i}}{X_i!} \exp(-\lambda)) = c - n\lambda + \log(\lambda)\sum_{i=1}^{n} X_i, \tag{14.35}$$

where c does not depend on λ and thus can be disregarded during the maximization. One finds that the maximum likelihood estimate of λ is

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{14.36}$$

which one can show is a both unbiased and consistent estimate of λ. (In the computations in this example we have used the notation X_i rather than x_i which is often convenient.)

A useful complement to the maximum likelihood method to estimate parameters is the least squares method which, when applicable, is often easier to use. Suppose that $X_1 \ldots, X_n$ are independent random variables with the same variance and with an expection that depends on a parameter θ. The least squares estimate $\hat{\theta}$ is obtained by minimizing

$$Q(\theta) = \sum_{i=1}^{n} (X_i - \mathbf{E}(X_i))^2. \tag{14.37}$$

Let us again consider a sample $(X_1, \ldots, X_n)$ of a random variable that is Poisson distributed with parameter λ. The sum of squares (14.37) now becomes

$$Q(\lambda) = \sum_{i=1}^{n} (X_i - \lambda)^2, \tag{14.38}$$

which is minimized for $\lambda = \hat{\lambda}$ in (14.36). Thus the least squares and the maximum likelihood estimates coincide in this example.

14.9 Linear and logistic regression

Let us first consider linear regression with one explaining real variable x. Suppose that we observe

$$Y_i = \alpha + \beta x_i + \epsilon_i, \;\; i = 1, \ldots n, \tag{14.39}$$

with independent zero-mean random errors ϵ_i, $i = 1, \ldots, n$, with identical variances. The least squares estimates $\hat{\alpha}$ and $\hat{\beta}$ are obtained by minimizing

$$Q(\alpha, \beta) = \sum_{i=1}^{n} (Y_i - \alpha - \beta x_i)^2, \tag{14.40}$$

which gives

$$\hat{\alpha} = \overline{Y} - \hat{\beta}\,\overline{x}, \qquad \hat{\beta} = \frac{\sum_{i=1}^{n} (Y_i - \overline{Y})(x_i - \overline{x})}{\sum_{i=1}^{n} (x_i - \overline{x})^2}, \tag{14.41}$$

where $\overline{x} = (1/n) \sum_i x_i$ and $\overline{Y} = (1/n) \sum_i Y_i$.

Let us now consider multiple linear regression with m explaining variables. We assume that we have observations

$$Y_i = \beta_1 x_{i1} + \ldots + \beta_m x_{im} + \epsilon_i, \;\; i = 1, \ldots n, \tag{14.42}$$

with independent zero-mean random errors ϵ_i, $i = 1, \ldots, n$, with identical variances. We can write our observations on vector-matrix form as

$$Y = X\beta + \epsilon, \tag{14.43}$$

where

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}, \quad X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}. \tag{14.44}$$

It turns out that the least squares estimate of the parameter vector β is

$$\hat{\beta} = (X^T X)^{-1} X^T Y, \tag{14.45}$$

provided that the matrix $X^T X$ is invertible.

Let us now consider logistic regression where we observe independent variables $Y_1, \ldots, Y_n$ taking values 0 or 1. We suppose that the probability $p_i = \Pr(Y_i = 1) = 1 - \Pr(Y_i = 0)$ depends on m explaining variables such that

$$\log(\frac{p_i}{1-p_i}) = \beta_1 x_{i1} + \ldots + \beta_m x_{im}, \; i = 1, \ldots n. \tag{14.46}$$

To estimate the parameters $\beta_1, \ldots, \beta_m$ we can maximize the likelihood function

$$L(\beta_1, \ldots, \beta_m) = \prod_{i=1}^{n} (p_i^{Y_i}(1-p_i)^{1-Y_i}). \tag{14.47}$$

There is no analytical expression for the maximum likelihood estimates so to maximize (14.47) one may use computational optimization methods such as those describe in Section 14.2 and then it is typically more convenient to maximize the log-likelihood function.

14.10 Confidence intervals and tests, observations from a normal distribution, the t- and chi-square distributions

Let X denote observations from a distribution depending on a real-valued parameter θ. We say that the interval $(L(X), U(X))$ is a confidence interval for θ with confidence degree p if

$$\Pr(L(X) < \theta < U(X)) = p. \tag{14.48}$$

Let $X = (X_1, \ldots, X_n)$ be a sample from a normal(μ, σ^2) distribution. Then

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \quad \text{and} \quad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2 \tag{14.49}$$

are unbiased and consistent estimates of μ and σ^2, respectively. To compute confidence intervals for μ and σ^2 we introduce the chi-square and t-distributions.

A random variable is said to be chi-square distributed with r degrees of freedom if it has the same distribution as

$$\chi^2 = \sum_{i=1}^{r} Z_i^2, \tag{14.50}$$

where $Z_1, \ldots, Z_r$ are independent and normal$(0, 1)$. Let us note that a variable that is chi-square distributed with r degrees of freedom has expectation r. A random variable is said to be t-distributed with r degrees of freedom if it has the same distribution as

$$t = \frac{Z}{\sqrt{\chi^2/r}} \tag{14.51}$$

where Z and χ^2 are independent and distributed normal$(0, 1)$ and chi-squared with r degrees of freedom, respectively.

Let us define quantiles for random variables with a continuous distribution function $F(x) = \Pr(X \leq x)$. A pth quantile x_p corresponding to such a distribution satisfies $F(x_p) = p$. Let χ^2_p denote the pth quantile of a chi-square distribution with $n-1$ degrees of freedom. For s^2 defined by (14.49) one can then show that

$$\Pr(\chi^2_{(1-p)/2} < (n-1)s^2/\sigma^2 < \chi^2_{(1+p)/2}) = p \tag{14.52}$$

which gives a confidence interval for σ^2 with confidence degree p,

$$\Pr(\frac{(n-1)s^2}{\chi^2_{(1+p)/2}} < \sigma^2 < \frac{(n-1)s^2}{\chi^2_{(1-p)/2}}) = p. \tag{14.53}$$

Similarly we let t_p denote the pth quantile of a t-distribution with $n-1$ degrees of freedom. Then

$$\Pr(\overline{X} - t_{(1-p)/2}\ s/\sqrt{n} < \mu < \overline{X} + t_{(1-p)/2}\ s/\sqrt{n}) = p, \tag{14.54}$$

which gives a confidence interval for μ with confidence degree p.

Let us also briefly describe one type of test of an hypothesis $H_0 : \theta = \theta_0$. Suppose that we have a test variable $T = T(X)$ tending to take large values when the hypothesis H_0 is not true and that we for our observations obtain an observed value T_{obs} of T. The strategy can then be to reject the hypothesis H_0 if the probability under H_0 to obtain a T-value at least as large as the observed value is small enough. More precisely we reject H_0 if the p-value

$$p = \Pr_0(T \geq T_{obs}) \tag{14.55}$$

is small enough. Here $\Pr_0$ denotes a probability evaluated under the probability distribution corresponding to H_0.

As an example let us suppose that we have a random sample $(X_1, \ldots, X_n)$ from a $N(\mu, \sigma^2)$ distribution and that we want to test the hypothesis $H_0 : \mu = \mu_0$ with the alternative hypothesis that μ is either larger or smaller than μ_0. Let X and s^2 be defined as in (14.49) and put

$$t_{obs} = \frac{\overline{X} - \mu_0}{s/\sqrt{n}}. \tag{14.56}$$

The corresponding p-value is then

$$p = P(|t| \geq |t_{obs}|) \tag{14.57}$$

evaluated with the assumption that t is t-distributed with $n-1$ degrees of freedom.

14.11 The F-distribution, analysis of variance

A random variable is F-distributed with (r_1, r_2) degrees of freedom if it has the same distribution as

$$F = \frac{\chi^2_1/r_1}{\chi^2_2/r_2}, \tag{14.58}$$

where χ_1^2 and χ_2^2 are independent chi-square distributed variables with r_1 and r_2 degrees of freedom, respectively. The F-distribution can be used to compare two variance estimates and in analysis of variance (ANOVA) models. Let us consider a simple ANOVA model.

Assume that X_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n_i$ are independent normal variables with identical variance σ^2 and expectations

$$\mathbf{E}(X_{ij}) = \mu_i, \;\; i = 1, \ldots, m, \, j = 1 \ldots, n_i. \tag{14.59}$$

To test the hypothesis $H_0 : \mu_1 = \ldots = \mu_m$ we can use the test variable

$$F = \frac{\sum_{i=1}^m n_i(\overline{X_{i\cdot}} - \overline{X_{\cdot\cdot}})^2 \; / \; (m-1)}{\sum_{i=1}^m \sum_{j=1}^{n_i} (X_{ij} - \overline{X_{i\cdot}})^2 \; / \; (\sum_i (n_i - 1))} \tag{14.60}$$

where $\overline{X_{i\cdot}} = (1/n_i) \sum_j X_{ij}$ and $\overline{X_{\cdot\cdot}} = (\sum_i \sum_j X_{ij})/(\sum_i n_i)$. It turns out that under H_0 the test variable F in (14.60) is F-distributed with $(m-1, \sum_i (n_i - 1))$degrees of freedom and we reject the hypothesis H_0 if F is large enough.

14.12 Approximate statistical methods, bootstrap

In the previous sections we have seen how confidence intervals with exact confidence degree and exact p-values for tests can be computed for simple models with normal random variables. Otherwise such exact statistical inference is typically not possible. However, for large samples good approximate methods are often available. Let us give some examples of how such approximate methods can look.

Suppose that we have a sample $X = (X_1, \ldots, X_n)$ of a random variables with log-likelihood $\ell(\theta)$, see (14.34), depending on a parameter vector $\theta = (\theta_1, \ldots, \theta_d)$. Under suitable regularity conditions, see for instance (Pawitan, 2001), one can then show that for large n the maximum likelihood estimate $\hat{\theta}$ has an approximate d-dimensional normal distribution, which we write

$$\hat{\theta} \xrightarrow{d} N(\theta, \mathcal{I}(\hat{\theta})^{-1}). \tag{14.61}$$

Here $\mathcal{I}(\hat{\theta})$ is the Fisher information matrix with matrix elements

$$\mathcal{I}_{ij}(\hat{\theta}) \; = \; -\frac{\partial^2}{\partial\theta_i \partial\theta_j} \ell(\theta)|_{\theta=\hat{\theta}} \tag{14.62}$$

and we suppose that $\mathcal{I}(\hat{\theta})$ is invertible. From this we can compute confidence intervals with approximate p-values for the components of θ and more generally for linear combinations of these components. Let us note that the Fisher information matrix is the Hessian (see Section 14.2) of the log-likelihood function and as discussed in Section 14.2 the Hessian can be obtained by use of quasi-Newton optimization methods.

Let us now consider two hypotheses H_0 and H_1, which are nested in such a way that H_0 is obtained from H_1 by imposing r linear restrictions on the parameters, for instance by putting r parameters equal to zero. Let $\ell(\hat{\theta}_0)$ and $\ell(\hat{\theta}_1)$ denote the log-likelihoods corresponding to the maximum likelihood estimates obtained under H_0 and H_1. Put

$$\chi^2 = 2(\ell(\hat{\theta}_1) - \ell(\hat{\theta}_0)). \tag{14.63}$$

We note that as $\ell(\hat{\theta}_1)$ is obtained as a maximum under fewer restrictions than $\ell(\hat{\theta}_0)$ it follows that $\ell(\hat{\theta}_1) \geq \ell(\hat{\theta}_0)$. One can show that under the hypothesis H_0 the variable χ^2 in (14.63) is approximately chi-square distributed with r degrees of freedom for large samples. We can reject the hypothesis H_0 if the observed χ^2-value is large enough, that is if the corresponding p-value

$$p = \Pr(\chi^2 \geq \chi^2_{obs}) \tag{14.64}$$

evaluated for a chi-square distribution with r degrees of freedom is small enough.

One method for obtaining approximate inference that has been much used since its introduction 1979 is the bootstrap which is based on resampling from observed distributions in such a way that confidence intervals and test variables can be computed, see for instance (Efron & Tibshirani, 1993).

14.13 Random numbers, simulation

An important method to study random systems is to use simulation and this requires generation of random numbers, or more precisely pseudo-random numbers, with computers. A basic random number generator is the linear congruential generator

$$X_{n+1} = (aX_n + b) \mod m \tag{14.65}$$

with suitable integers a, b and m and a starting value X_0 called seed. This generates a sequence with approximately independent random number equidistributed on the set of integers $\{0, 1, \ldots, m-1\}$. This type of generators with some variations are used as basic random generators in computer languges such as for MATLAB. Putting $U_n = X_n/m$ gives a sequence of random numbers with an approximate uniform distribution on the unit interval $[0, 1]$.

Suppose now that we have a random number U with a uniform distribution on the interval $(0, 1)$ and that we want a random number X with a given distribution function $F(x) = \Pr(X \leq x)$. This can be obtained by putting

$$X = F^{-1}(U), \tag{14.66}$$

where F^{-1} denotes the inverse of F. Putting

$$X = -\frac{1}{\beta}\log(1-U) \tag{14.67}$$

gives for instance a random variable that is exponentially distributed with parameter β.

Sometimes one wants a random variable $X = (X_1, X_2)$ with a uniform distribution on a bounded two-dimensional set A. One can then use *rejection sampling* by first finding a rectangle $R_0 = \{(x_1, x_2) : a_1 \leq x_1 \leq b_1, a_2 \leq x_2 \leq b_2\}$ containing A as a subset. Generate then two independent random numbers U_1 and U_2 with uniform distributions on the unit interval. Put $X = (a_1 + (b_1 - a_1)U_1, a_2 + (b_2 - a_2)U_2)$. If $X \in A$ accept X, otherwise reject X and repeat the procedure until we get a point in A.

14.14 Bayesian inference, Markov chain Monte Carlo

In Bayesian inference we have in addition to a model describing the distribution of observations X given the parameter θ also a random distribution for θ called the prior distribution. After obtaining observations of X the distribution of θ is modified to the posterior distribution. Let us show how this goes when both θ and X are discrete variables, the formulas when one or both of these variables have continuous distributions being similar. We let π_i denote the prior probability, $\pi_i = \Pr(\theta = \theta_i)$.

From the definition of conditional probabilities for events A and B we have $\Pr(A|B) = \Pr(A \cap B)/\Pr(B)$. This gives the posterior distribution for θ when we observe $X = x$ as

$$\Pr(\theta = \theta_i | X = x) = \frac{\Pr(X = x|\theta_i)\pi_i}{\Pr(X = x)} = \frac{\Pr(X = x|\theta_i)\pi_i}{\sum_j \Pr(X = x|\theta_j)\pi_j}. \tag{14.68}$$

In Bayesian analysis of noisy observations of complicated high-dimensional objects such as images it is not easy to evaluate or sample from the posterior distribution. One general method that has ben much used in recent years is Markov chain Monte Carlo, abbreviated MCMC. Here you construct a Markov chain which has the distribution of interest as its stationary distribution. Useful algorithms for constructing and analyzing such Markov chains are the Gibbs sampler and the Metropolis algorithm, see Section 4.7 in this book for a brief summary and (Gilks *et al.*, 1996) for more details.

14.15 Prediction, Kalman filtering

Let us look at prediction and filtering by use of Kalman filters. We let the d-dimensional column vector $X_t, t = 0, 1, \ldots,$ denote the state of a system at time t. Assume that $X_0 \sim N(\mu_0, P_0)$ and that

$$X_t = F_t X_{t-1} + W_t, \quad t = 1, 2, \ldots, \tag{14.69}$$

where F_t is a $d \times d$ matrix. Suppose that the dynamic d-dimensional noise vectors $W_t \sim N(0, Q_t)$ are independent mutually and of the initial state X_0. Assume further that we observe the r-dimensional vectors

$$Y_t = H_t X_t + V_t, \quad t = 1, 2, \ldots, \tag{14.70}$$

where H_t is a $r \times d$ matrix and the measurement noise vectors $V_t \sim N(0, R_t)$ are independent mutually and of (W_t) and the initial state X_0. Let $Y_{1:t} = (Y_1, \ldots, Y_t)$ denote the accumulated observations up to time t. We are interested in computing the optimal estimate of X_t given observations up to time t. It turns out that given $Y_{1:t}$ the conditional distribution of X_t is normal with expectation

$$\hat{X}_{t|t} = \mathbf{E}(X_t | Y_{1:t}) \tag{14.71}$$

and covariance matrix $P_{t|t}$. We will give a recursive algorithm for computing $\hat{X}_{t|t}$ and $P_{t|t}$ which also gives the conditional expectation and covariance matrix $\hat{X}_{t|t-1}$ and $P_{t|t-1}$ for prediction of X_t from observations $Y_{1:t-1}$ up to time $t-1$. The algorithm consists of the following six equations in going from $\hat{X}_{t-1|t-1}$ and $P_{t-1|t-1}$ to $\hat{X}_{t|t}$ and $P_{t|t}$,

$$\hat{X}_{t|t-1} = F_t \hat{X}_{t-1|t-1}, \tag{14.72}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t, \tag{14.73}$$

$$S_t = H_t P_{t|t-1} H_t^T + R_t, \tag{14.74}$$

$$K_t = P_{t|t-1} H_t^T S_t^{-1}, \tag{14.75}$$

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t(Y_t - H_t \hat{X}_{t|t-1}), \tag{14.76}$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1}, \tag{14.77}$$

where I denotes the unit $d \times d$-matrix.

Consider as an example motion of an object with centre at (x_t, y_t) and velocity $(\dot{x}_t, \dot{y}_t)$ with a sampling interval Δt and observation of the position but not the velocity. We can then put

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix}, \quad F_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{14.78}$$

Bibliography

Anderson, E. (1935) The irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, **59**, 2–5.

Andersson, M. (1998) Weed and crop classification by automated digital image processing. Master thesis, Chalmers University of Technology, Department of Applied Electronics and Department of Mathematical Statistics.

Baddeley, A. & Jensen, E. B. V. (2005) *Stereology for Statisticians*. Chapman & Hall/CRC, London.

Baddeley, A., Rubak, E. & Turner, R. (2015) *Spatial Point Patterns: Methodology and Applications with R*. CRC Press, London.

Besag, J. (1986) On the statistical analysis of dirty pictures (with Discussion). *J. Royal Statistical Soc. B*, **48**, 259–302.

Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*. Springer, London.

Braeckmans, K., Buyens, K., Bouquet, W., Verwaet, C., Joye, P., Vos, F. D., Plawinski, L., Douvre, L., Angles-Cano, E., Sanders, N., Demeester, J. & Smedt, S. D. (2010) Sizing nanomatter in biological fluids by fluorescence single particle tracking. *Nano Letters*, **10**, 4435–4442.

Breu, H., Gil, J., Kirkpatrick, D. & Werman, M. (1995) Linear time Euclidean distance transform algorithms. *IEEE Trans. Pattern Anal. Machine Intell.*, **17**, 529–533.

Chiu, S., Stoyan, D., Kendall, W. & Mecke, J. (2013) *Stochastic Geometry and its Applications, 3rd Edition*. John Wiley & Sons, New York.

Cressie, N. (1993) *Statistics for Spatial Data, Revised edition*. Wiley, New York.

Daley, D. J. & Vere-Jones, D. (2003) *An Introduction to the Theory of Point Processes. Volume I: Elementary Theory and Methods, Second Edition*. Springer-Verlag, New York.

Daley, D. J. & Vere-Jones, D. (2008) *An Introduction to the Theory of Point Processes. Volume II: General Theory and Structure, Second Edition*. Springer-Verlag, New York.

Diggle, P. (2013) *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns, Third Edition*. CRC Press, London.

Digman, M. A., Brown, C. M., Sengupta, P., Wiseman, P. W., Horwitz, A. R. & Gratton, E. (2005) Measuring fast dynamics in solutions and cells with a laser scanning microscope. *Biophysical Journal*, **89**, 1317–1327.

Dralle, K. & Rudemo, M. (1996) Stem number estimation by kernel smoothing of aerial photos. *Canadian J. Forest Research*, **26**, 1228–1236.

Dralle, K. & Rudemo, M. (1997) Automatic estimation of individual tree positions from aerial photos. *Canadian J. Forest Research*, **27**, 1728–1736.

Duda, R. & Hart, P. (1972) Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM*, **15**, 11–15.

Dudoit, S., Fridlyand, J. & Speed, T. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, **97**, 77–87.

Efron, B. & Hastie, T. (2016) *Computer Age Statistical Inference*. Cambridge University Press, New York.

Efron, B. & Tibshirani, R. (1993) *An introduction to the bootstrap*. Chapman & Hall, London.

Einstein, A. (1905) Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, **17**, 549–560.

Ekstrøm, C. T., Bak, S., Christensen, C. & Rudemo, M. (2004) Spot shape modelling and data transformations for microarrays. *Bioinformatics*, **20**, 2270–2278.

Ekstrøm, C. T., Bak, S. & Rudemo, M. (2005) Spot shape modelling with spacial correlation for microarrays. *Statistical Applications in Genetics and Molecular Biology*, **4**, Article 6.

Fisher, R. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**, 179–188.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition, Second edition*. Academic Press, Boston.

Garrels, J. (1989) The QUEST system for quantitative analysis of two-dimensional gels. *Journal of Biological Chemistry*, **264**, 5269–5282.

Gelfand, A., Diggle, P., M.Fuentes & Guttorp, P. (2010) *Handbook of Spatial Statistics*. Chapman & Hall/CRC, Boca Raton, USA.

Geman, S. & Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, **6**, 721–741.

Georgii, H.-O., Häggström, O. & Maes, C. (2001) The random geometry of equlibrium phases. *Phase Transitions and Critical Phenomena* (edited by C. Domb & J. Lebowitz), vol. 18, 1–142, Academic Press, London.

Gilks, W., Richardson, S. & Spiegelhalter, D. (1996) *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

Glasbey, C. & Horgan, G. (1995) *Image Analysis for the Biological Sciences*. John Wiley & Sons, New York.

Glasbey, C. & Mardia, K. (1998) A review of image warping methods. *J. Applied Statistics*, **25**, 155–171.

Glasbey, C. & Mardia, K. (2001) A penalized likelihood approach to image warping. *J. Royal Statistical Soc. B*, **63**, 465–514.

Goffeau, A., Barrel, B., Bussey, H., Davis, R. *et al.* (1996) Life with 6000 genes. *Science*, **274**, 563–567.

Goodfellow, I., Bengio, Y. & Courville, A. (2016) *Deep Learning*. MIT press, http://www.deeplearningbook.org.

Grenander, U. (1983) *Tutorial in Pattern Theory*. Division of Applied Mathematics, Brown University, Providence, Rhode Island.

Gustafsson, J., Blomberg, A. & Rudemo, M. (2002) Warping two-dimensional electrophoresis gel images to correct for geometric distortions of the spot pattern. *Electrophoresis*, **23**, 1731–1744.

Häggström, O. (2002) *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press, Cambridge.

Hotelling, H. (1933) Analysis of complex statistical variables into principal components. *Journal of Educational Psychology*, **24**, 417–441 and 498–520.

Hough, P. (1959) Machine analysis of bubble chamber pictures. *Proc. Int. Conf. High Energy Accelerators and Instrumentation.*

Illyan, J., Penttinen, A., Stoyan, H. & Stoyan, D. (2008) *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley & Sons, New York.

Jain, A., Murty, M. & Flynn, P. (1999) Data clustering: a review. *ACM Computing Surveys*, **31**, 264–323.

Johns, C., Nychka, D., Kittel, T. & Daly, C. (2003) Infilling sparse records of spatial fields. *Journal of the American Statistical Association*, **98**, 796–806.

Jonasson, J., Lorén, N., Olofsson, P., Nydén, M. & Rudemo, M. (2008) A pixel-based likelihood framework for analysis of fluorescence recovery after photobleaching data. *Journal of Microscopy*, **232**, 260–269.

Kristensen, C., Morant, M., Olsen, C. E., Ekstrøm, C. T., Galbraith, D. W., Møller, B. L. & Bak, S. (2005) Metabolic engineering of dhurrin in transgenic arabidopsis plants with marginal inadvertent effects on the metabolome and transcriptome. *Proceedings of the National Academy of Science of the United States of America*, **102**, 1779–1784.

Kvarnström, M. (2005) Position estimation and tracking in colloidal particle microscopy. Ph.D. thesis, Chalmers University of Technology, Department of Mathematical Sciences.

Kvarnström, M. & Glasbey, C. (2007) Estimation of centres and radial intensity profiles of spherical nano-particles in digital microscopy. *Biometrical Journal*, **49**, 300–311.

Larsen, M. (1997) Crown modelling to find tree top positions in aerial photographs. *Proceedings of the Third International Airborne Remote Sensing Conference and Exhibition*, II–428 – II–435, ERIM International, Copenhagen, Denmark.

Larsen, M. & Rudemo, M. (1998) Optimizing templates for finding for finding trees in aerial photographs. *Pattern Recognition Letters*, **19**, 1153–1162.

LeCun, Y., Bengio, Y. & Hinton, G. (2015) Deep learning. *Nature*, **521**, 436–444.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**, 2278–2324.

Longfils, M. (2018) Pattern Recognition Methods Applied on the MNIST Database. Technical report, Chalmers University of Technology, Gothenburg.

Longfils, M., Röding, M., Altskär, A., Schuster, E., Lorén, N., Särkkä, A. & Rudemo, M. (2018) Single particle raster image analysis of diffusion for particle mixtures. *Journal of Microscopy*, **269**, 269–281.

Longfils, M., Schuster, E., Lorén, N., Särkkä, A. & Rudemo, M. (2017) Single particle raster image analysis of diffusion. *Journal of Microscopy*, **266**, 3–14.

Lund, J. & Rudemo, M. (2000) Models for point processes observed with noise. *Biometrika*, **87**, 235–249.

Matérn, B. (1986) *Spatial variation. Second edition (First edition 1960). Lecture notes in Statistics 36*. Springer, Berlin.

Matheron, G. (1971) *The Theory of Regionalized Variables and Its Applications. Les cahiers du CMM de Fontainebleau, Fascicule 5*. Ecole des Mines,Paris.

Møller, J. & Waagepetersen, R. (2003) *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall/CRC, London.

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H. & Hahn, U. (2017) Global envelope tests for spatial processes. *J. Royal Statistical Soc. B*, **79**, 381–404.

Nelder, J. A. & Mead, R. (1965) A simplex method for function minimization. *The Computer Journal*, **7**, 308–313.

Nisslert, R., Kvarnström, M., Lorén, N., Nydén, M. & Rudemo, M. (2007) Identification of the three-dimensional gel microstructure from transmission electron micrographs. *Journal of Microscopy*, **225**, 10–21.

Nordin, M., Abrahamsson, C., Blomqvist, C. H., Häbel, H., Röding, M., Olsson, E., Nydén, M. & Rudemo, M. (2014) Estimation of mass thickness response of embedded aggregated silica nanospheres from high angle annular dark-field scanning electron micrographs. *Journal of Microscopy*, **253**, 166–170.

Onsager, L. (1944) Crystal statistics, i. a two-dimensional model with an order-disorder transformation. *Physical Review*, **65**, 117–149.

Otsu, N. (1979) A threshold selection method from gray-level histograms. *IEEE Trans. Syst., Man, Cybernetics*, **9**, 62–66.

Pawitan, Y. (2001) *In all likelihood: Statistical modelling and inference using likelihood*. Oxford University Press.

Petersen, P. (1992) Weed seed identification by shape and texture analysis of microscope images. Ph.D. thesis, Agricultural University, Copenhagen, Department of Mathematics and Physics.

Pollock, R. J. (1994) A model-based approach to automatically locating individual tree crowns in high-resolution images of forest canopies. *Proceedings of the First International Airborne Remote Sensing Conference and Exhibition*, Strasbourg, France.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (2007) *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York.

Ripley, B. (1981) *Spatial Statistics*. Wiley, New York.

Ripley, B. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK.

Robert, C. (2016) The Metropolis-Hastings algorithm. arXiv:1504.01896v3 [stat.CO].

Röding, M., Deschout, H., Braeckmans, K. & Rudemo, M. (2011) Measuring absolute number concentrations of nanoparticles using single-particle tracking. *Physical Review E*, **84**, 031920.

Röding, M., Deschout, H., Braeckmans, K. & Rudemo, M. (2013) Measuring absolute nanoparticle number concentrations from particle count time series. *Journal of Microscopy*, **251**, 19–26.

Rosenfeld, A. & Kak, A. (1982) *Digital Picture Processing, Second Edition*. Academic Press, San Diego, Ca.

Sonka, M., Hvalac, V. & Boyle, R. (2015) *Image Processing, Analysis, and Machine Vision, Fourth Edition*. Cengage Learning.

van der Maaten, L. & Hinton, G. (2008) Visualizing Data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605.

van Lieshout, M. N. M. (2000) *Markov Point Processes and their Application*. Imperial College Press, London.

von Smoluchowski, M. (1906) Zur kinetischen Theorie der Brownschen Molekularbewegung und der Suspensionen. *Annalen der Physik*, **326**, 756–780.

Winkler, G. (2003) *Image Analysis, Random Fields and Dynamic Monte Carlo Methods, Second Edition*. Springer-Verlag, Berlin.

Zhu, S. & Mumford, D. (1997) Prior learning and Gibbs reaction-diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, **19**, 1236–1250.