Statistics of Imaging

Mats Rudemo

May 11, 2018

STATISTICS OF IMAGING

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

E-mail: rudemo@chalmers.se

The current version has been updated by the addition of Chapter 10 (earlier there was just a sketch of this chapter).

Preface

The object of these notes is to provide an introduction to several subjects connected with statistical inference from images and spatial data. Image analysis and spatial statistics are extensive research fields 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 first chapter "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. The first part is concluded by a chapter on statistical models for images. One class of models discussed consists of Bayesian models with a Markov random field prior and with observation noise that is pixel-wise independent and identically distributed.

The chapter on "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 and remote sensing. The examples cover analysis of data from microarray (DNA chip) images, two-dimensional electrophoresis and aerial photographs of forests.

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.*

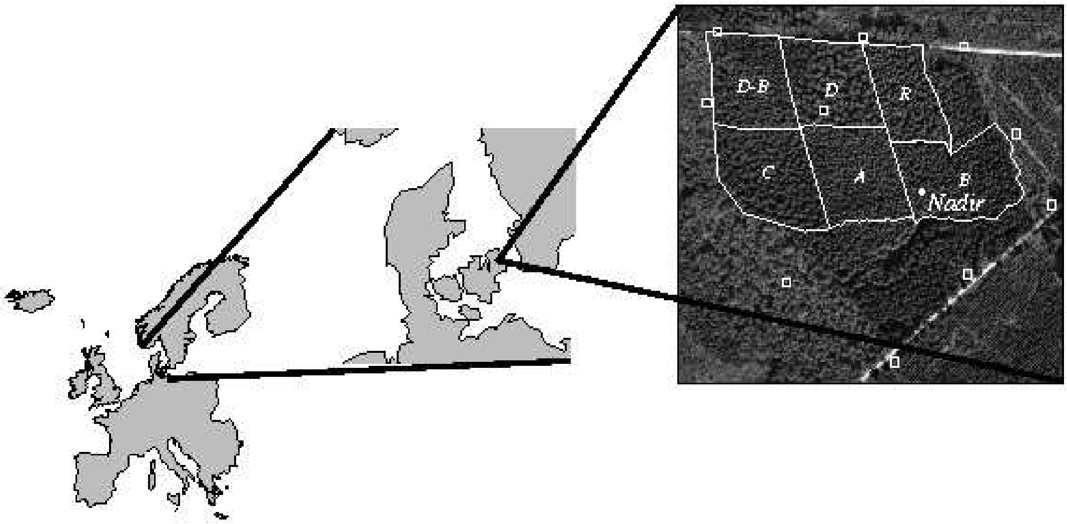


Figure 1.1: Aerial photograph of the thinning experiment KU in northern Sealand with Norway spruce trees. The position of the airplane at image acqusition was 560 m above “Nadir”.

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 airplane at the altitude 560 m above the point ”Nadir” in Figure 1.1. An enlargement of the subplot D is shown Figure 1.2.

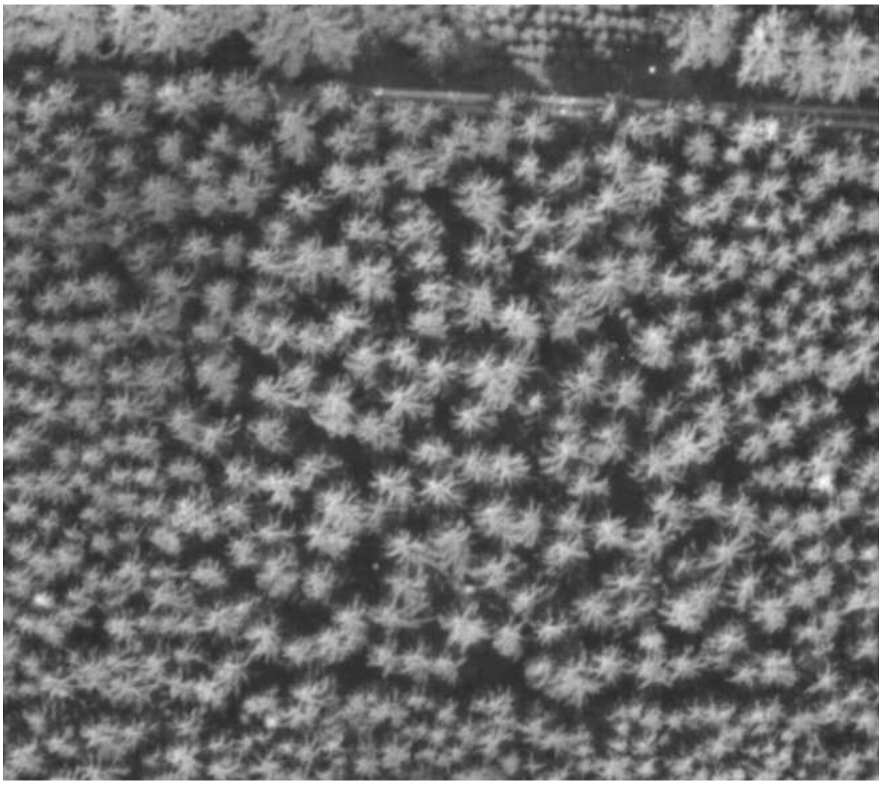


Figure 1.2: Detail of the aerial photograph in Figure 1.1 covering the subplot D with very heavy thinning.

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 airplane 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 10 in Part 3.

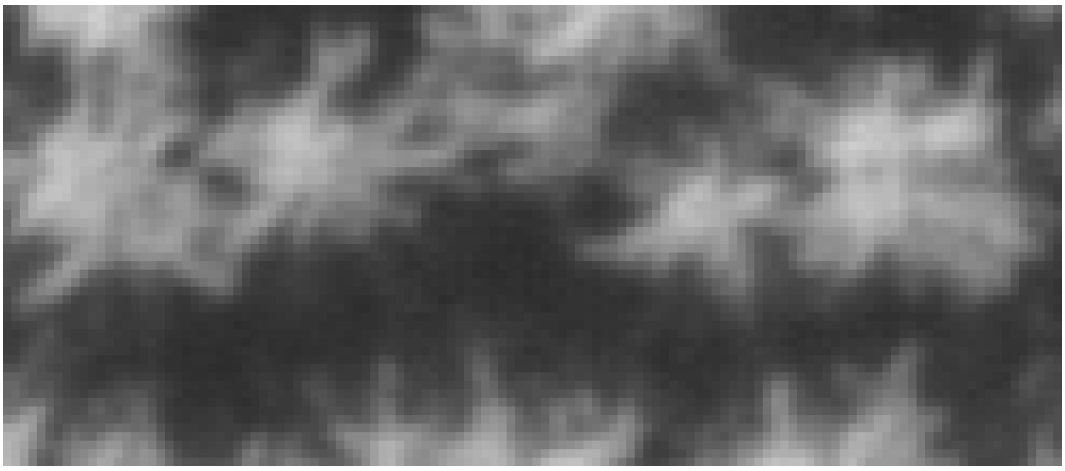


Figure 1.3: Detail of the aerial photograph in Figure 1.2 showing part of the southeastern corner of subplot D.

□

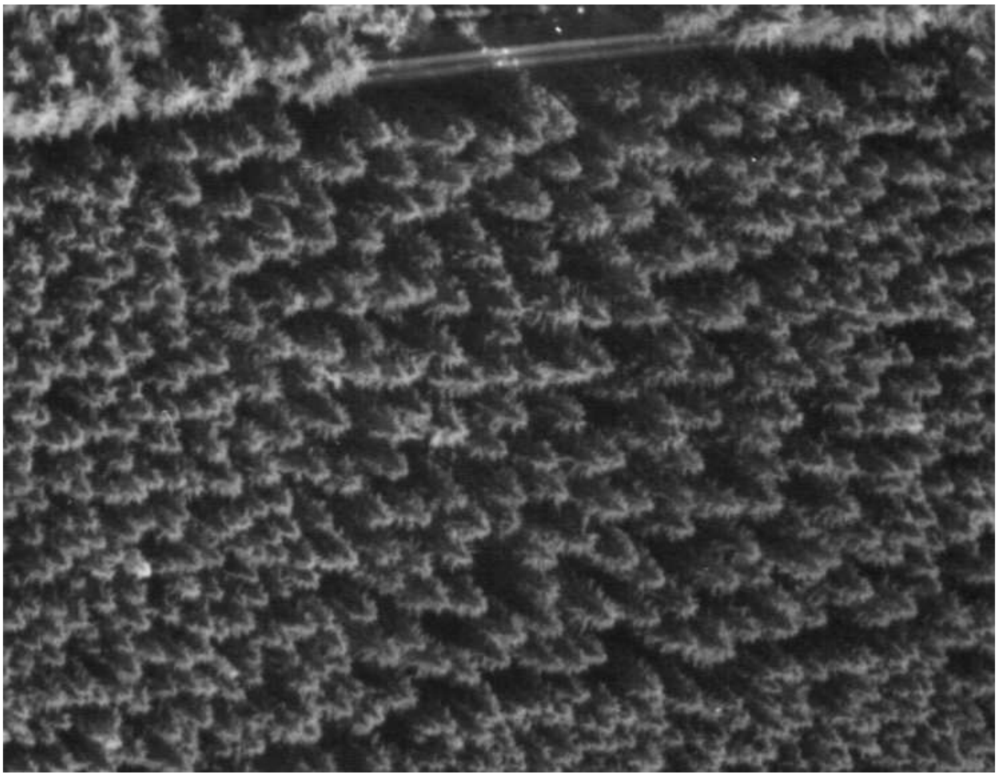


Figure 1.4: Detail of aerial photograph of subplot D of backlighted Norway spruce trees acquired from an oblique angle with the airplane located to the northwest of the experimental area shown in Figure 1.1.

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.

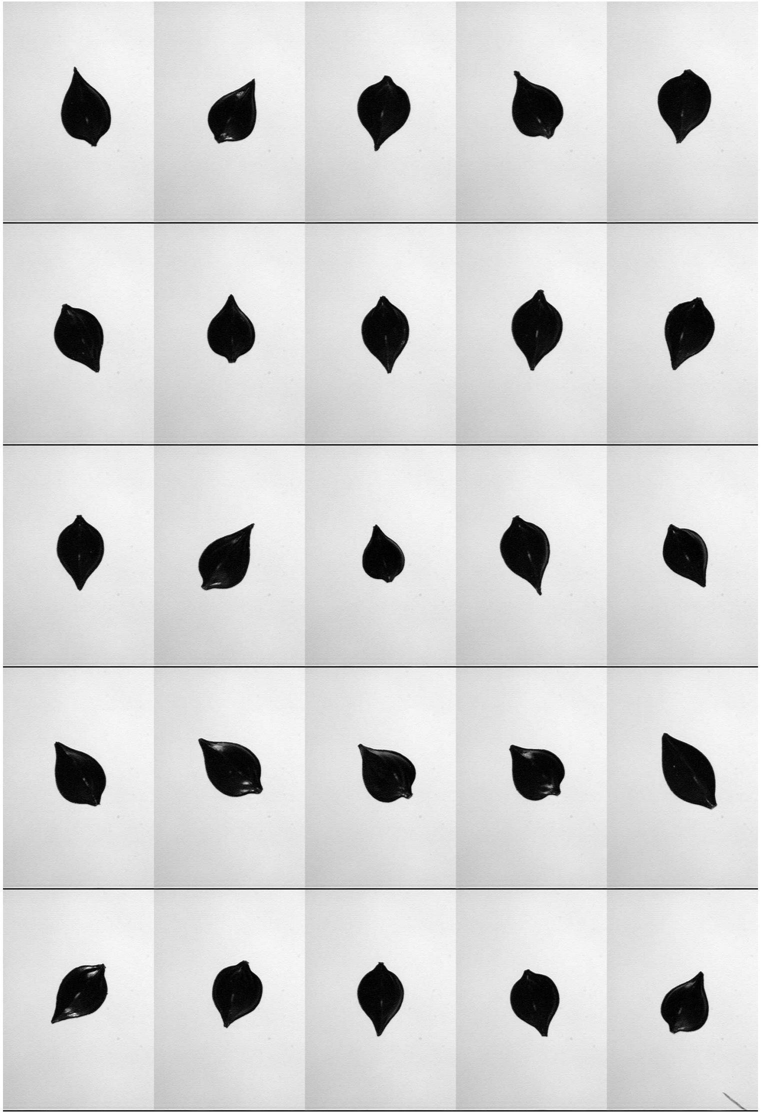


Figure 1.5: Images of seeds of *Rumex crispus.*

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 7 on image warping and image matching such problems will be treated.

□

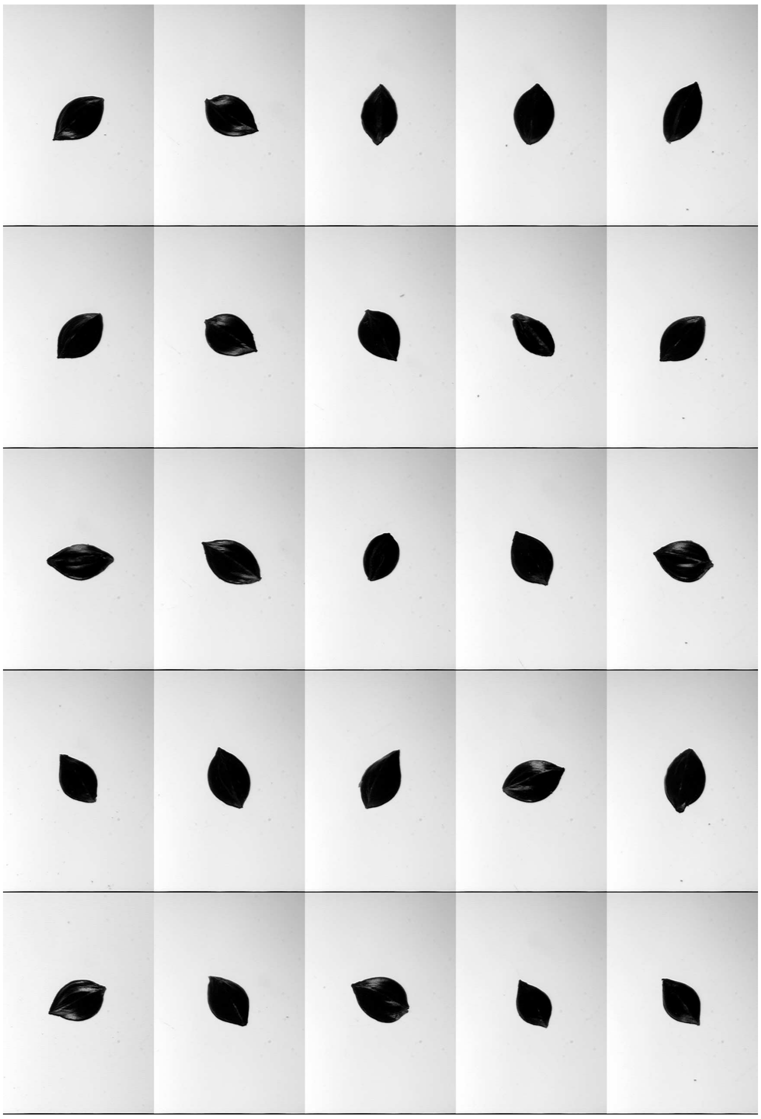


Figure 1.6: Images of seeds of *Rumex thyrsiflorus*.

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.

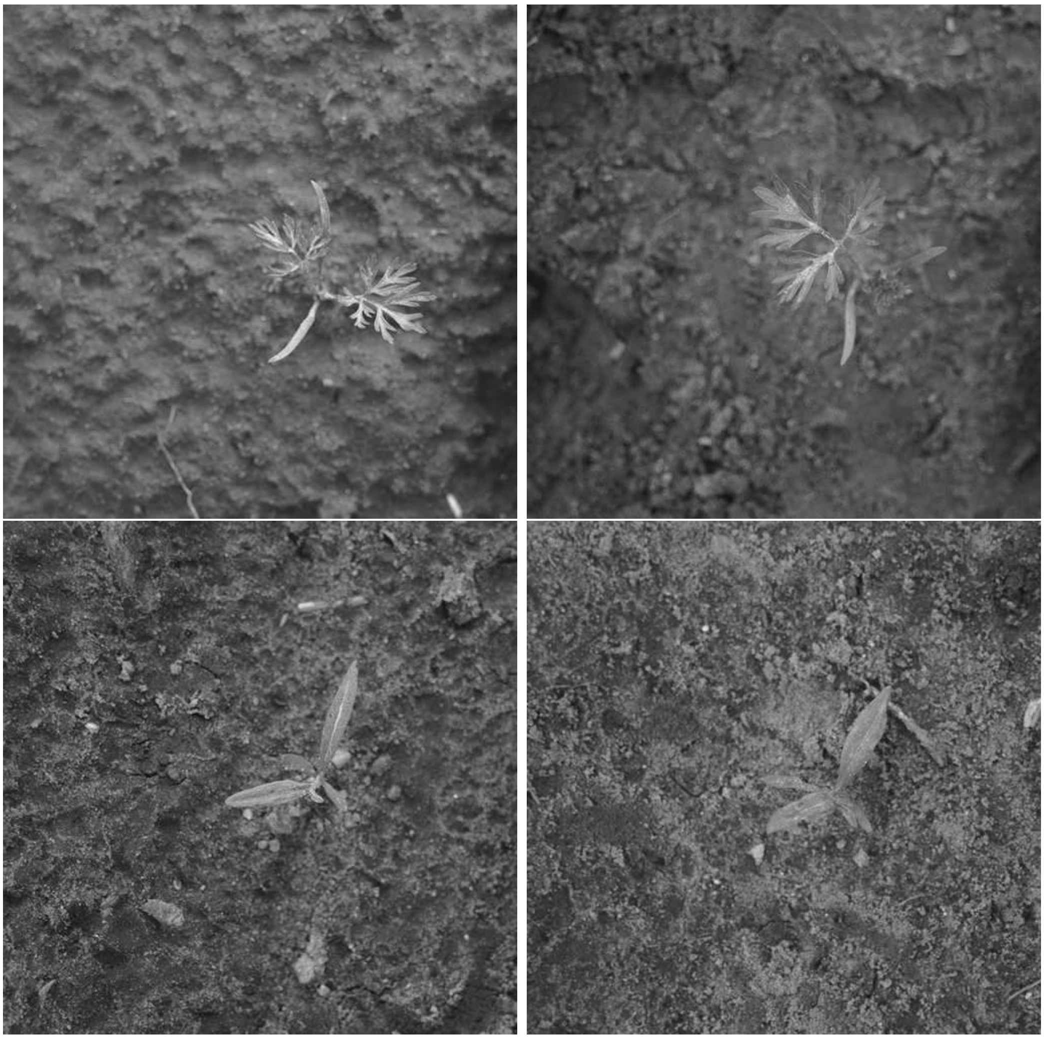


Figure 1.7: Above two images of plants of carrot, *Daucus carota*, L., and below two images of plants of ladythumb smartweed *Polygonum persicaria*, L.

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.

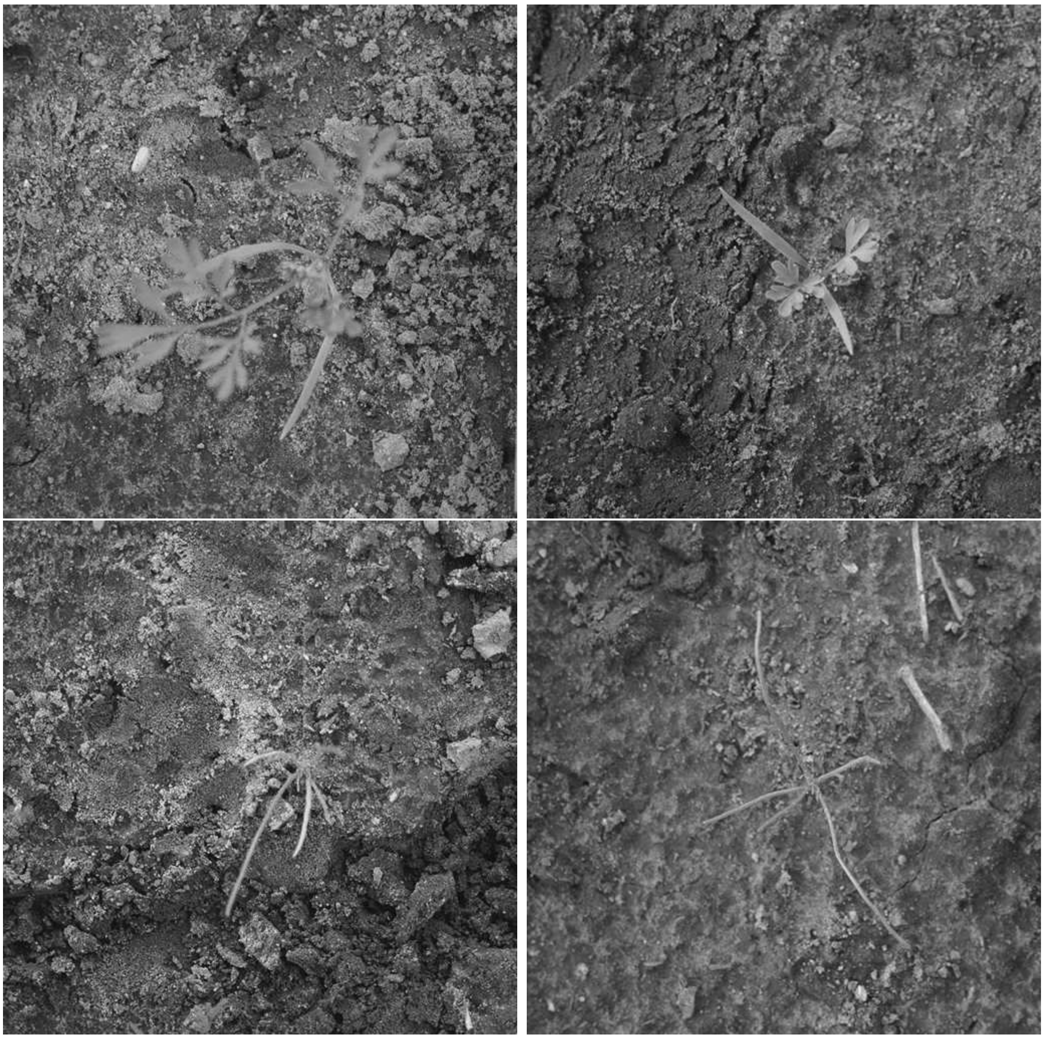


Figure 1.8: Above two images of plants of fumitory, *Fumaria officinalis*, L., and below two images of plants of corn spurry, *Spergula arvensis*, L.

□

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

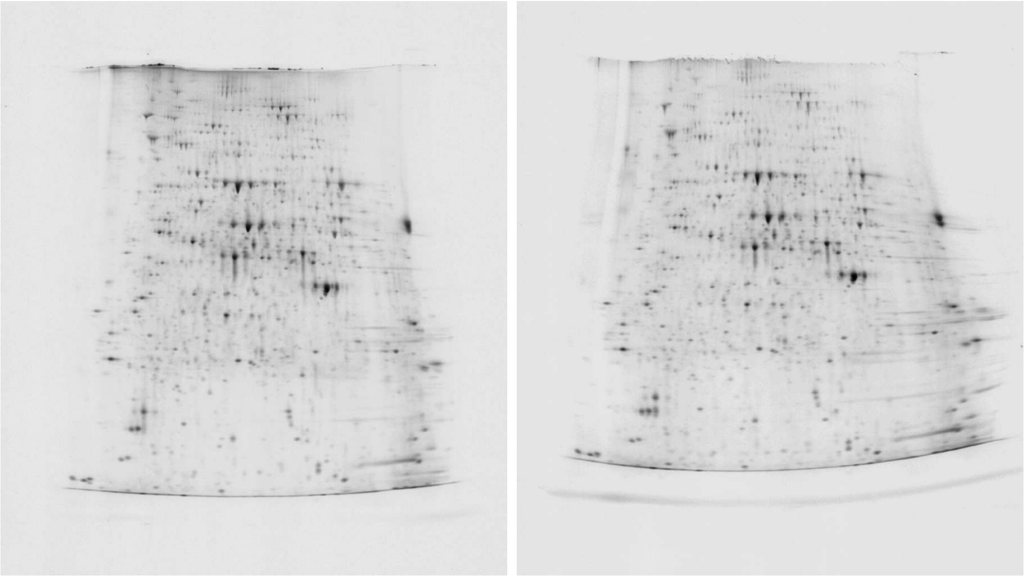


Figure 1.9: Images from 2D gel electrophoresis of baker's yeast grown in a standard solution.

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.

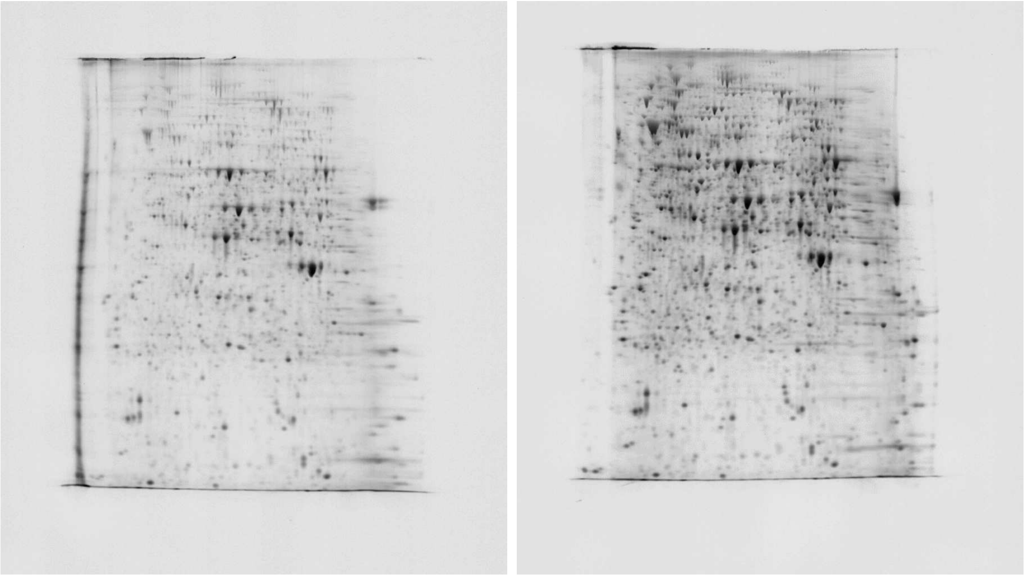
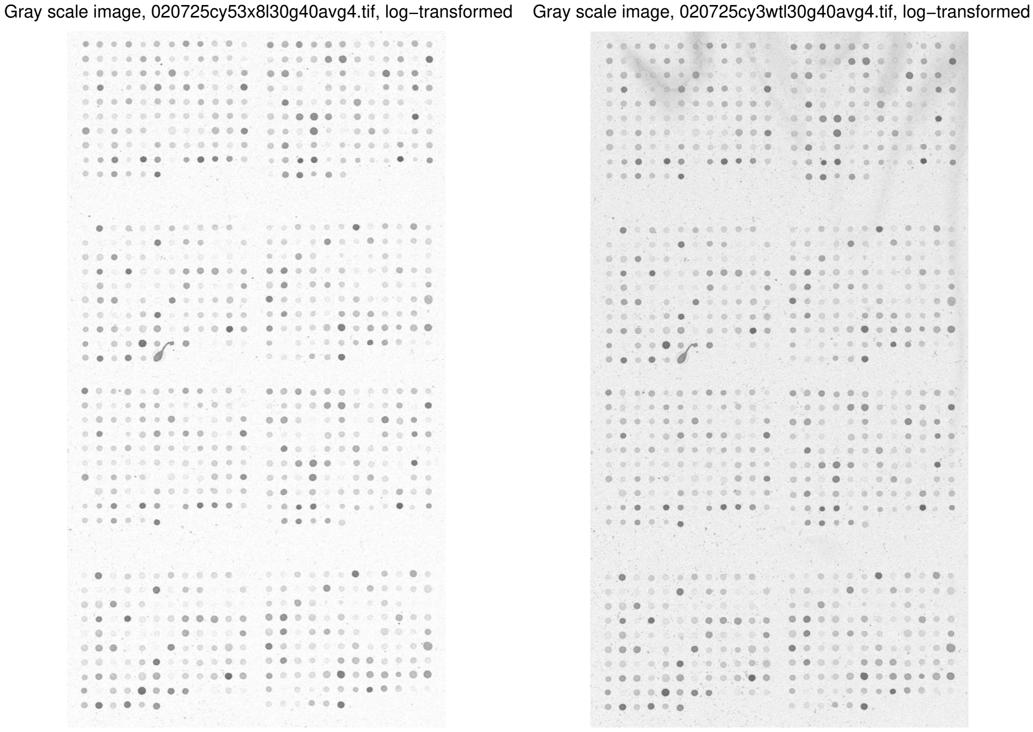


Figure 1.10: Images from 2D gel electrophoresis of baker's yeast grown under stress in a solution with salt added.

□

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.



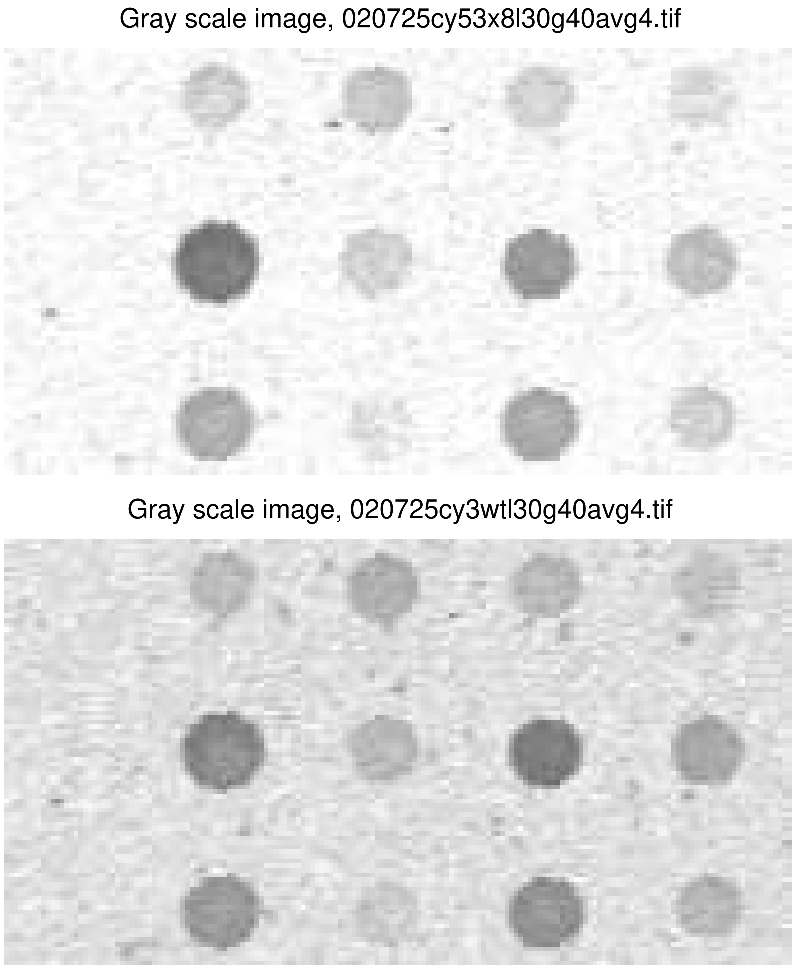
Gray scale image, 020725cy53x8l30g40avg4.tif, log−transformed

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.

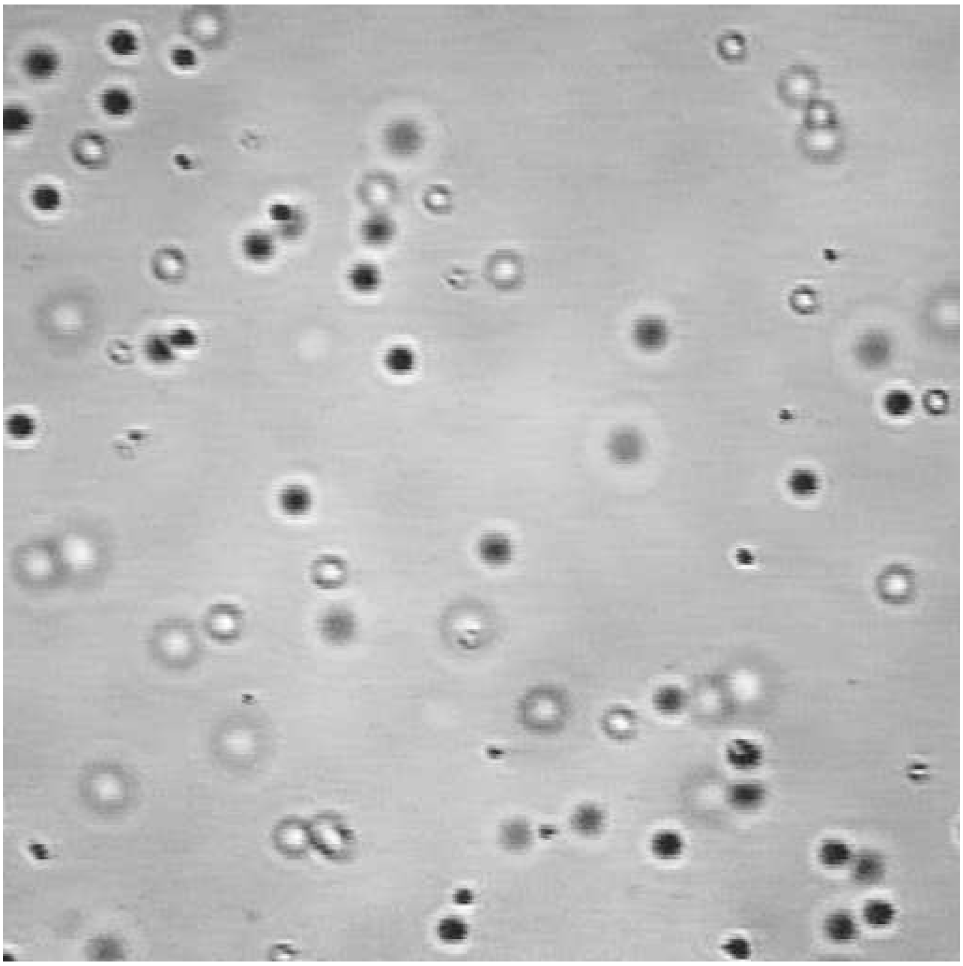


Figure 1.13: Image obtained by video microscopy showing diffusing particles. Particles in phocus are shown as small distinct black objects.

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 nd 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.

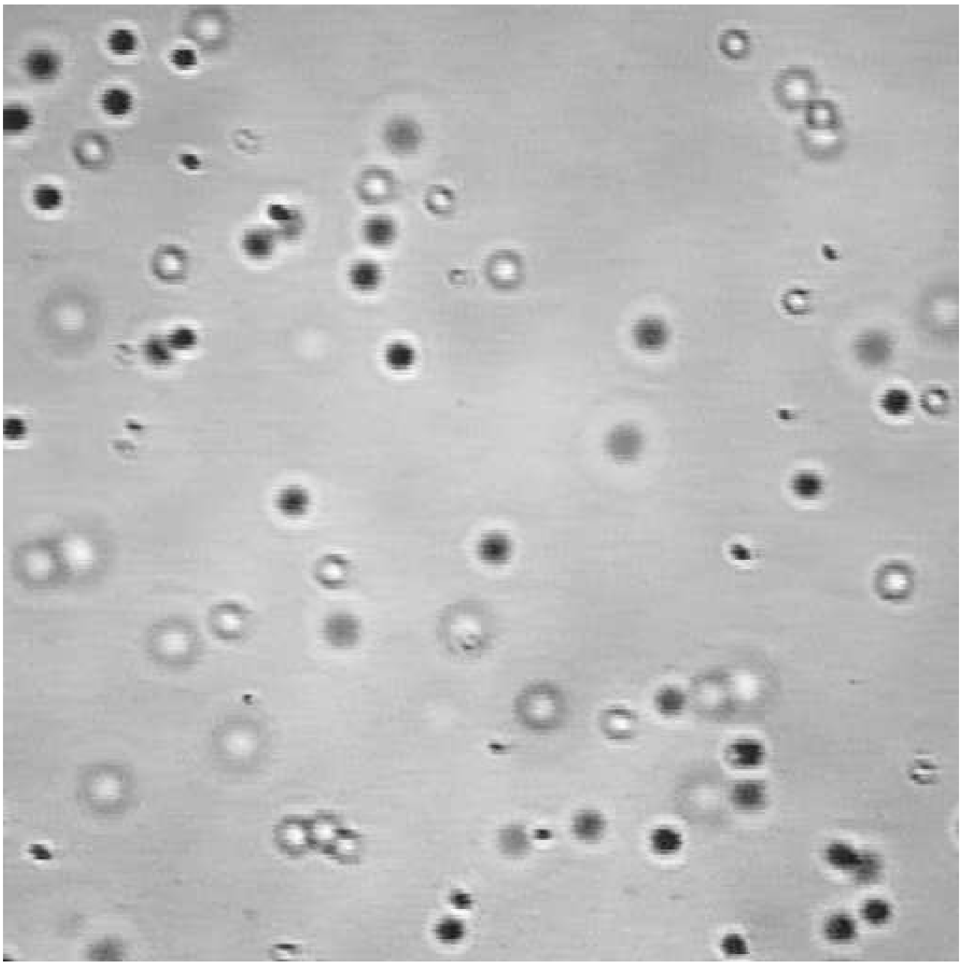


Figure 1.14: Image obtained by video microscopy showing diffusing particles. This image was obtained 0.2 seconds after the image in Figure 1.13.

□

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 28x28 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 in these notes in Chapter 2.

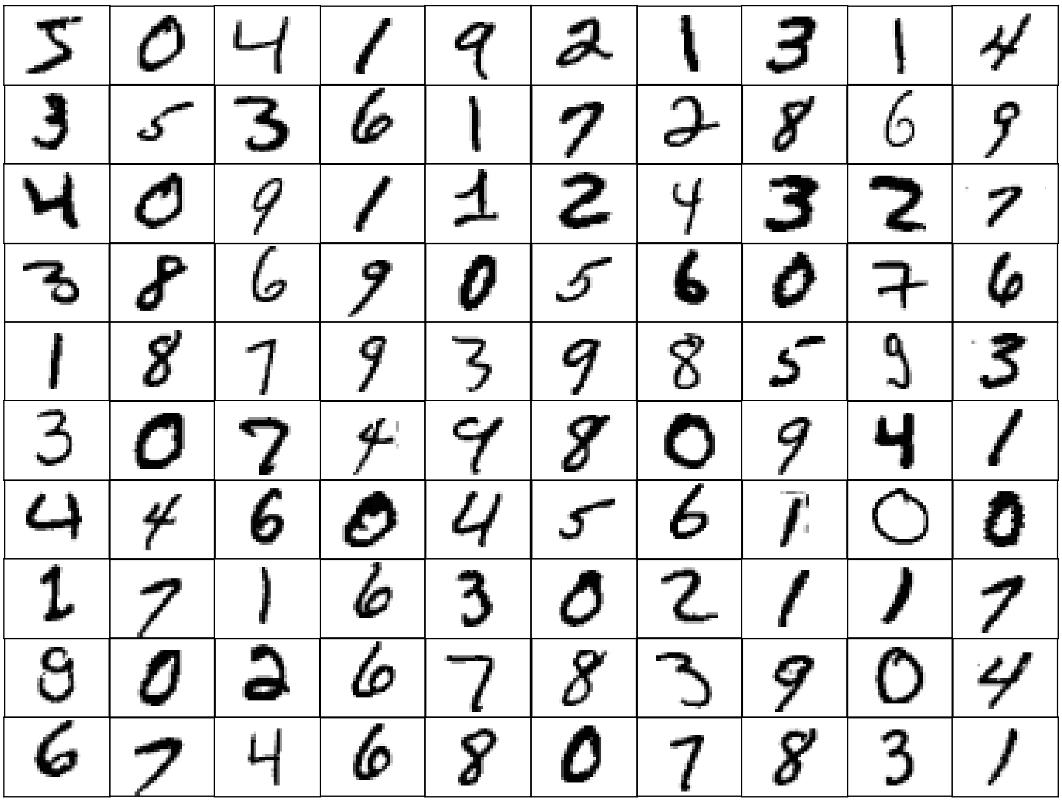


Figure 1.15: Examples of 100 handwritten digits from the MNIST database.

□

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.

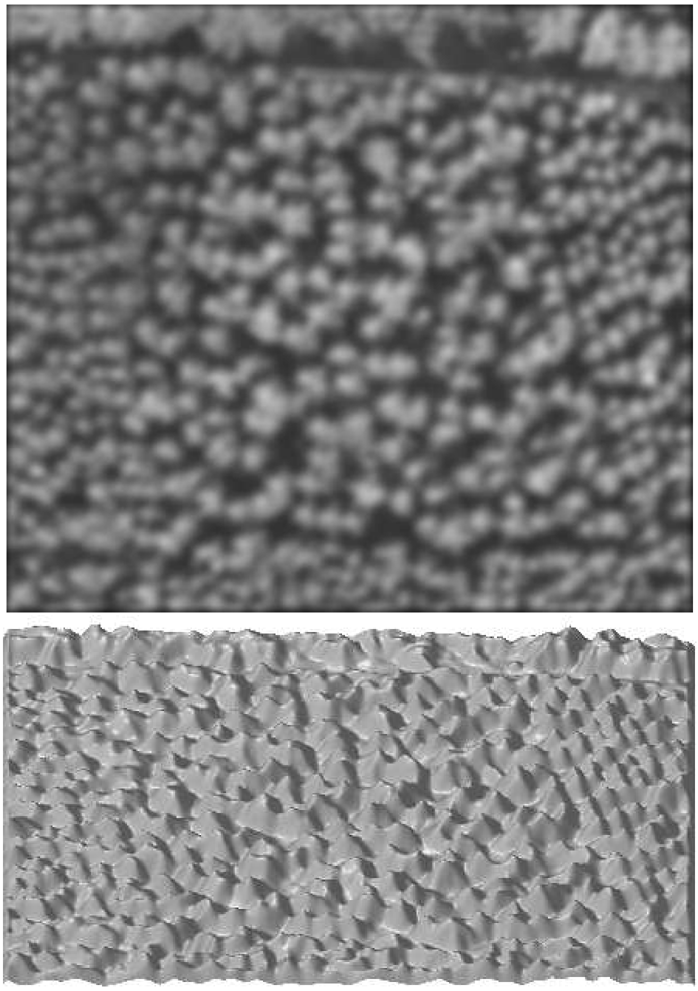


Figure 1.16: Upper part: Smoothed version of the image in Figure 1.2 by use of circular 2D Gaussian filter with $\sigma = 4.5$ pixel-widths. Lower part: The same image viewn in perspective as a 3D surface with light intensity as the vertical coordinate.

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.)

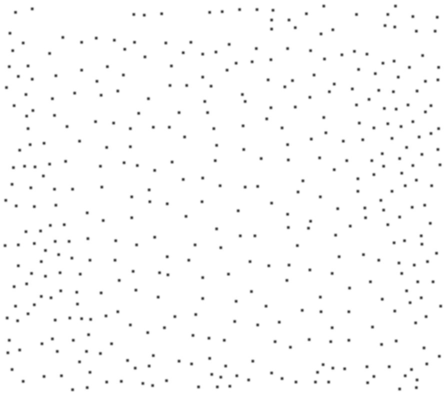


Figure 1.17: Location of maxima in Figure 1.16.

□

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.

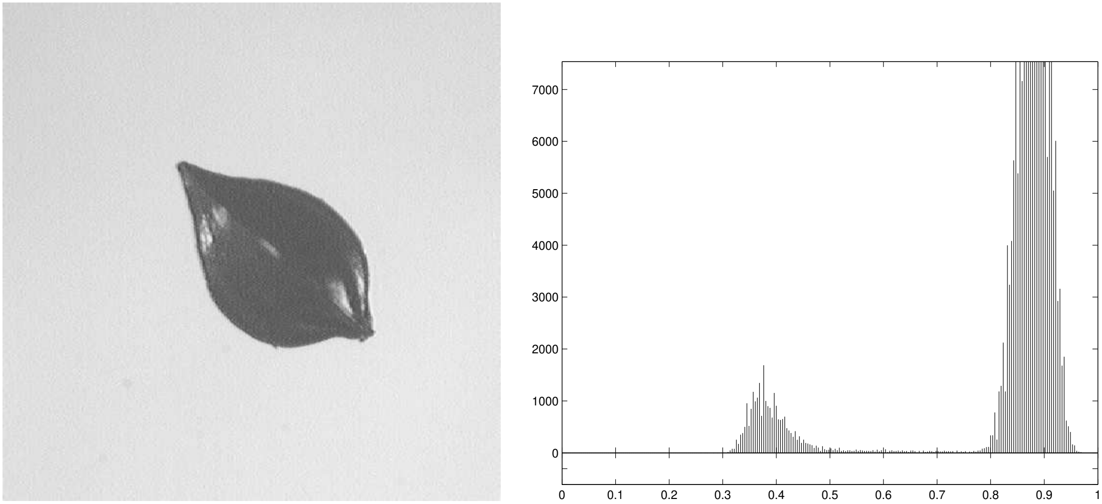


Figure 1.18: Above an image of a *Rumex crispus* seed and below the corresponding histogram.

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.

□

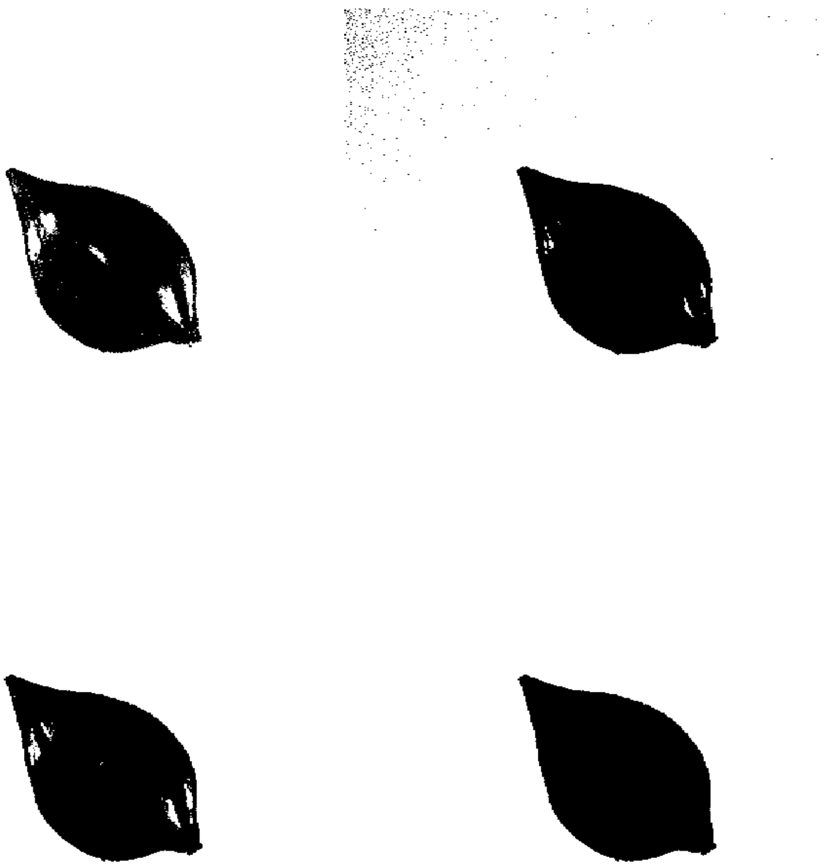


Figure 1.19: Binary images obtained by thresholding of the image in Figure 1.18 with the levels $t = 0.5$ (upper left), $t = 0.8$ (upper right), and $t = 0.65$ (lower left). The lower right image is obtained from the lower left image by filling out holes.

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

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

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

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

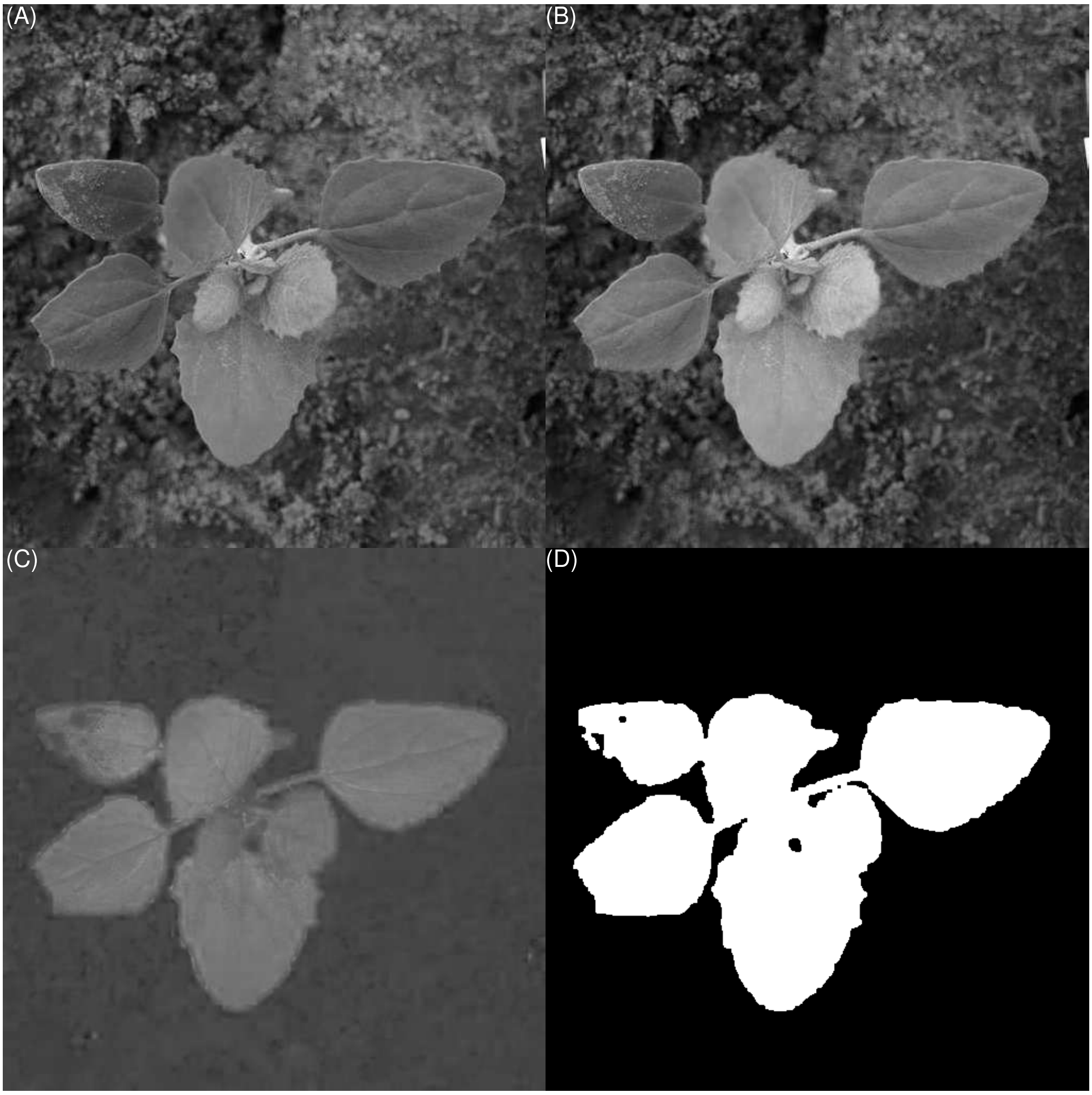


Figure 1.20: Images of a weed plant, lamb's quarter *Chenopodium album*, L., (A) grey scale image, (B) green channel image, (C) normalized green channel image, and (D) binary black and white image after thresholding.

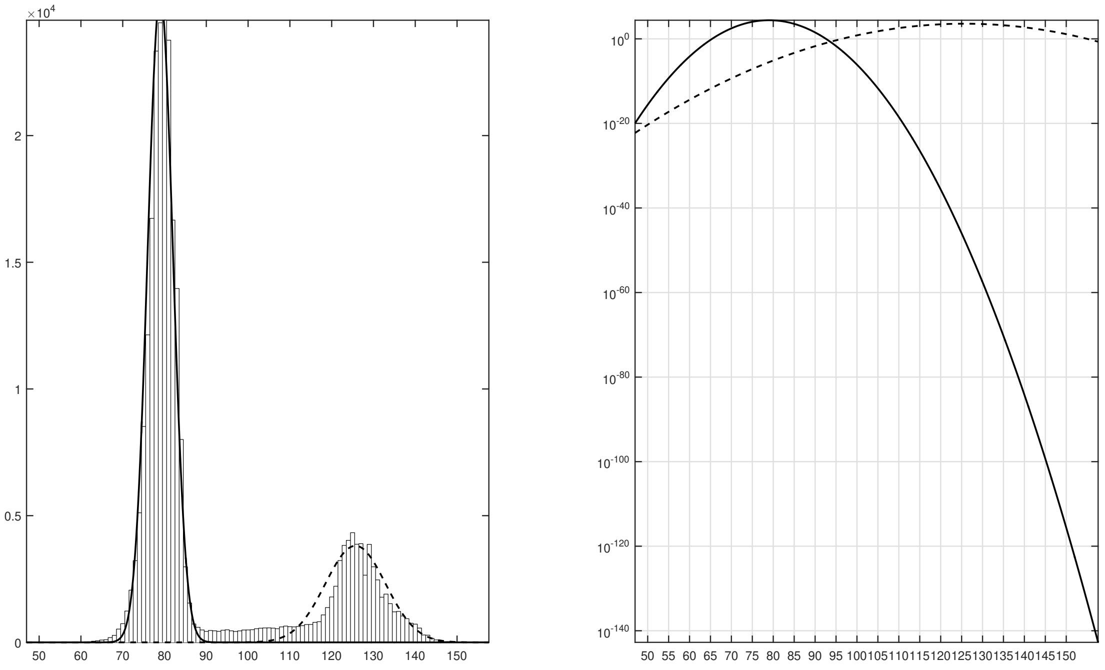


Figure 1.21: Left: histogram for the normalized green channel shown in the lower left part of Figure 1.20 and the two components shown as fully drawn and dashed curves. Right: the two components shown with a log scale on the vertical axis; here the threshold where the two curves cross can be seen.

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.

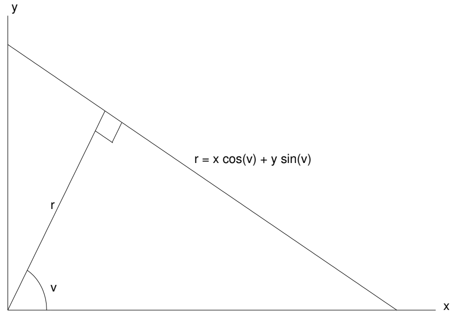


Figure 1.22: Representation of line in terms of angle and distance to origo.

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.

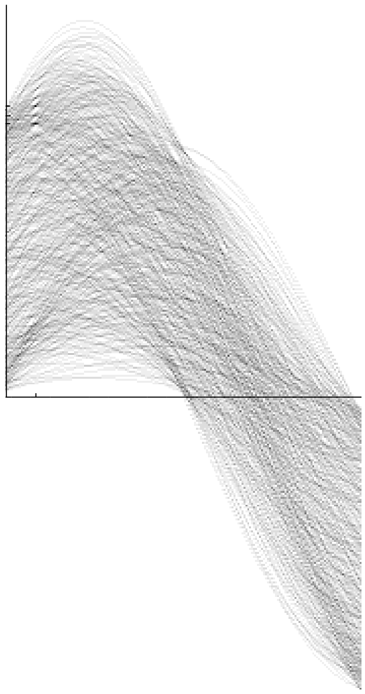


Figure 1.23: Hough transform for Figure 1.17 with angle v on the horizontal axis extending from 0 to 180 degrees and distance r on the vertical axis extending from $-r_{max}$ to r_{max}, where r_{max} is the length of the diagonal in Figure 1.2.

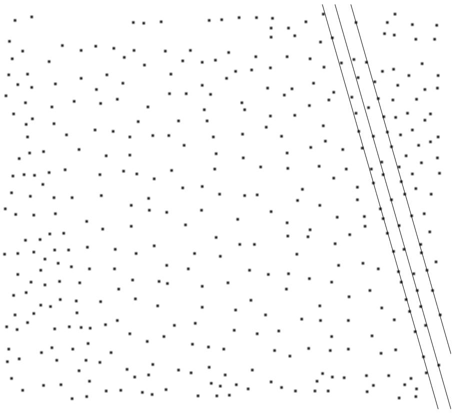


Figure 1.24: Location of maxima in Figure 1.16 together with three lines found by the Hough transform.

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 yse 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. □

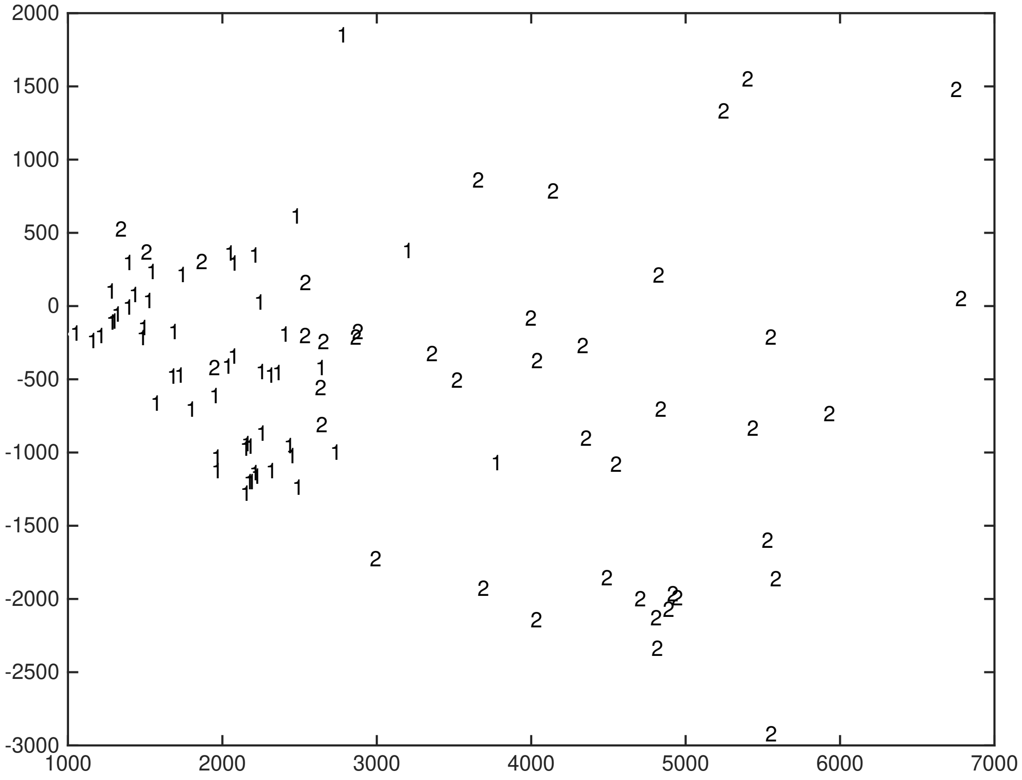


Figure 1.25: Plot of μ_{11} versus moment μ_{20} for handwritten digits digits 1 and 2 among the first 400 digits in the MNIST data base.

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

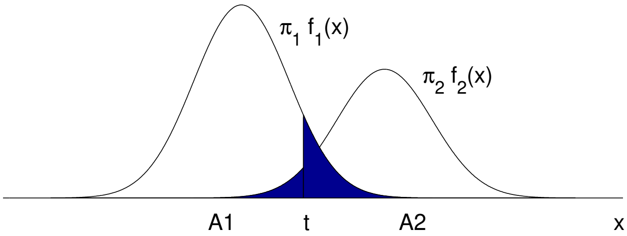


Figure 2.1: Probability of misclassification is given by the coloured area. The set A_1 where class ω_1 is chosen extends here up to the threshold t, while A_2 is chosen above t.

to be equal or unequal. Let the observations be denoted X_{im}, $m = 1, \dots, n_i$, $i = 1, 2$. 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 $\omega_i, i = 1, \dots, k$, 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, $i = 1, \dots, k$. 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 $A_1, \dots, A_k$ with $\cup_{i=1}^k A_i = \mathbb{R}^d$. If $X \in A_i$ we assign our object to class $\omega_i, i = 1, \dots, k$.

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

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

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 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

$$\pi_i f_i(x) > \pi_j f_j(x) \tag{2.7}$$

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 N(μ,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

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

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 N(μ_i,C_i) in class ω_i, $i = 1, \ldots, k$. Sometimes the covariance matrices are assumed to be equal, that is

$$C_i = C, \quad i = 1, \ldots, k. \tag{2.9}$$

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

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

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

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

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

$$\hat{\mu}_i = \frac{1}{n_i}\sum_{m=1}^{n_i} X_{im}, \quad i = 1, 2. \tag{2.12}$$

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

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

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

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

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 $j, j = 1, \dots, k$. 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*

We use the first 8000 digits in the MNIST database, see Example 1.7, and consider discrimination between the 10 types of digits 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 full drawn curves and for the quadratic discrimination 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.

□

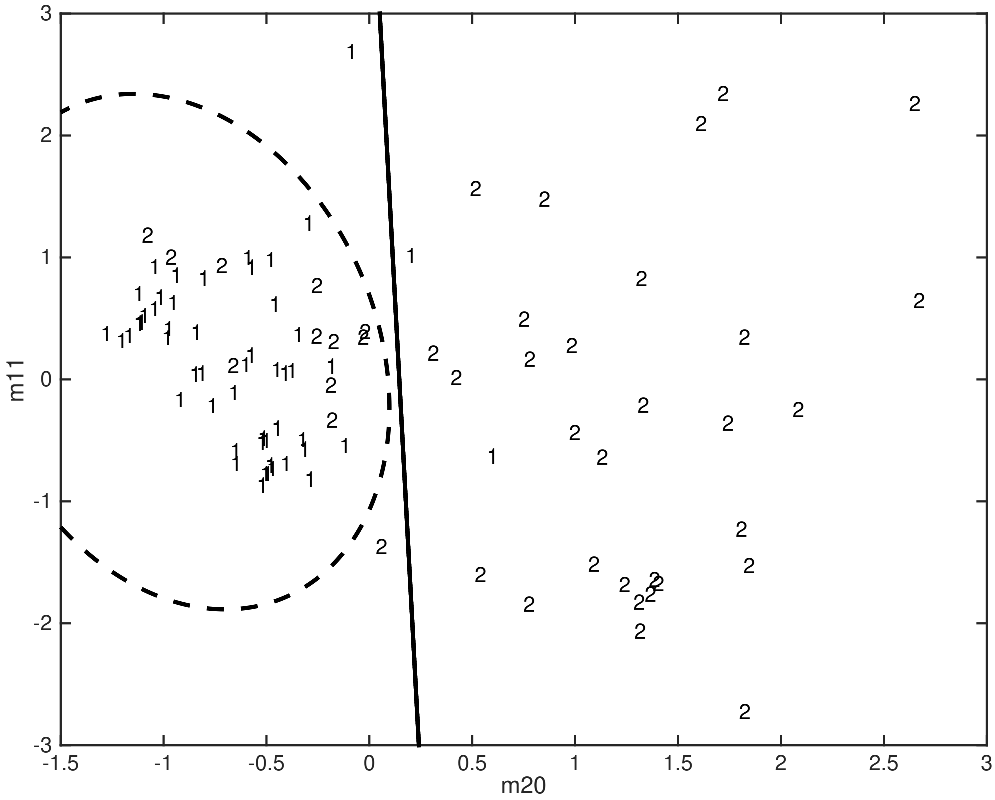


Figure 2.2: Plot of standardized moments μ_{11} versus μ_{20} for handwritten digits 1 and 2 among the first 400 digits in the MNIST data base together with the class boundaries corresponding to linear and quadratic discrimination.

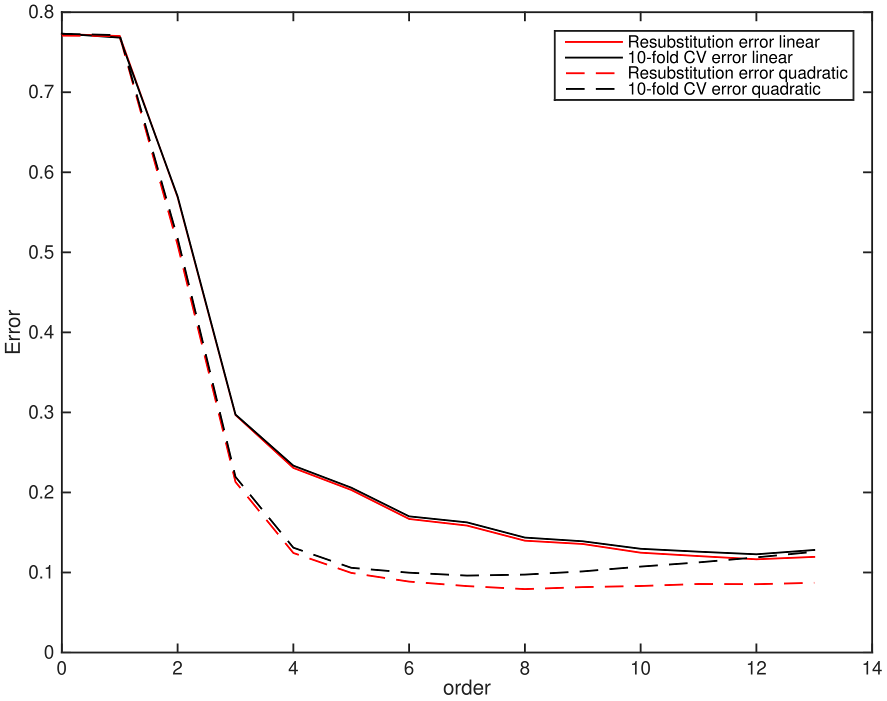


Figure 2.3: Plot of error probabilities for linear discrimination, full drawn curves, and quadratic discrimination, dashed curves. Resubstitution error curves are in grey and cross-validation error curves are in black. Order K on the horizontal axis means that all moments μ_{pq} with $p + q \leq K$ are 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 crossvalidated error is obtained for $m = 5$ and equals 12 %. □

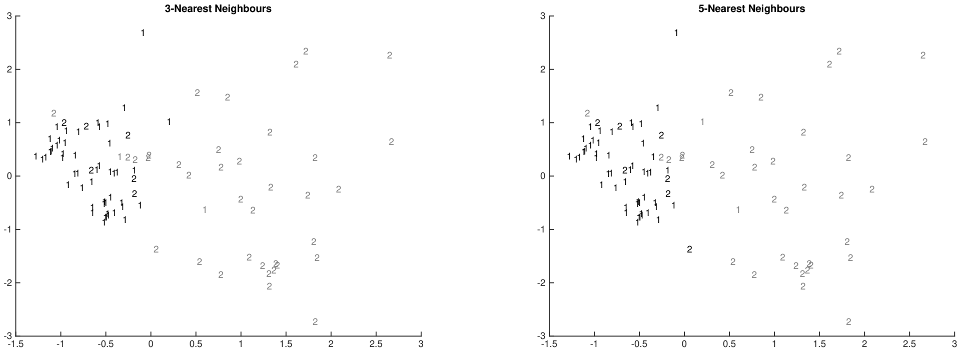


Figure 2.4: Plot of standardized moments μ_{11} versus μ_{20} for handwritten digits digits 1 and 2 among the first 400 digits in the MNIST data base together classifications from m-nearest neighbour classification for $m = 3$ and $m = 5$. Digit colours indicate classification: black digits are classified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

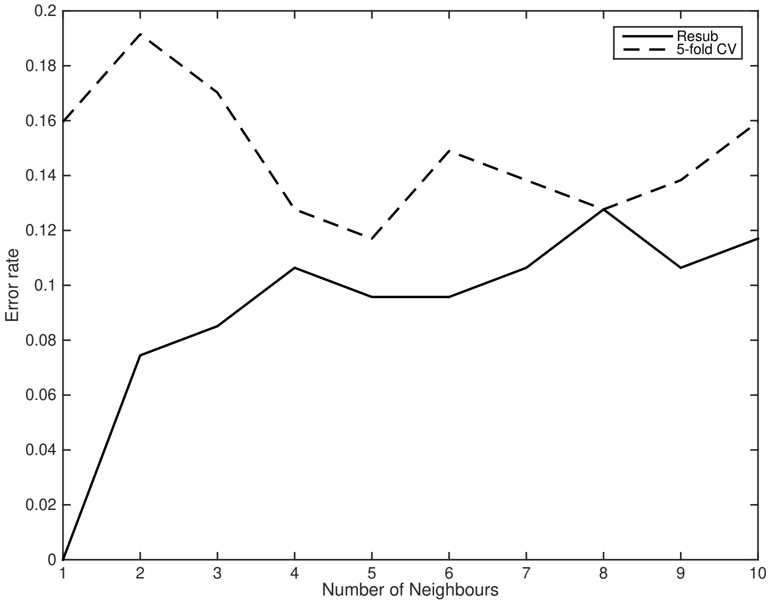


Figure 2.5: Plot of resubstitution and 5-fold cross validation error estimates for m-nearest neighbour classications for $m = 1, \ldots, 10$.

2.6 Multinomial logistic regression

Logistic regression with two classes is briefly described in Section 13.9. Here we will generalize to k classes $\omega_1, \ldots, \omega_k$. 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

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

and

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

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 of illustrating the results of a discrimination analysis is to compute a *confusion matrix*

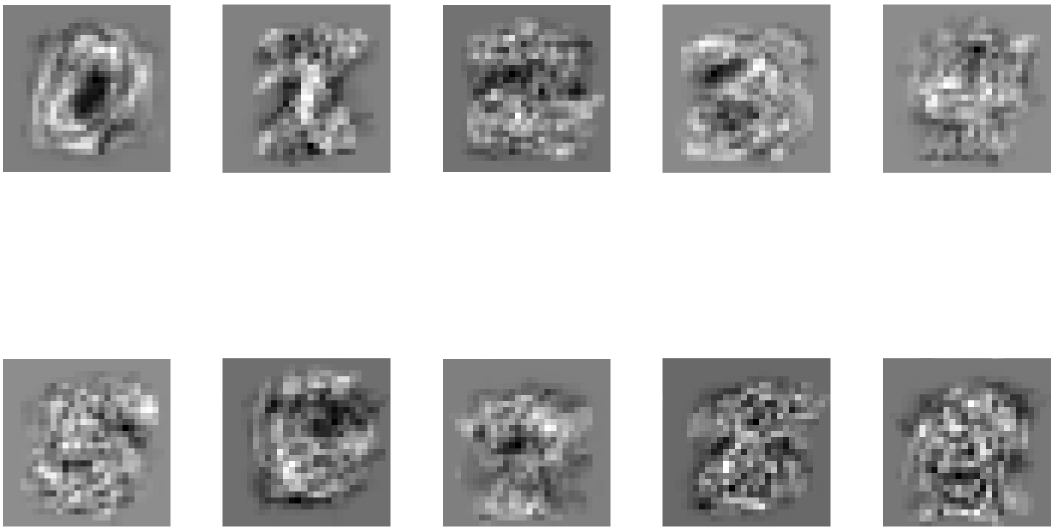


Figure 2.6: Parameter vectors β_i for digits $0, \ldots, 4$ in the upper row and digits $5, \ldots, 9$ in the lower row estimated from 10000 digits in the MNIST database.

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. 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 13.6 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.

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

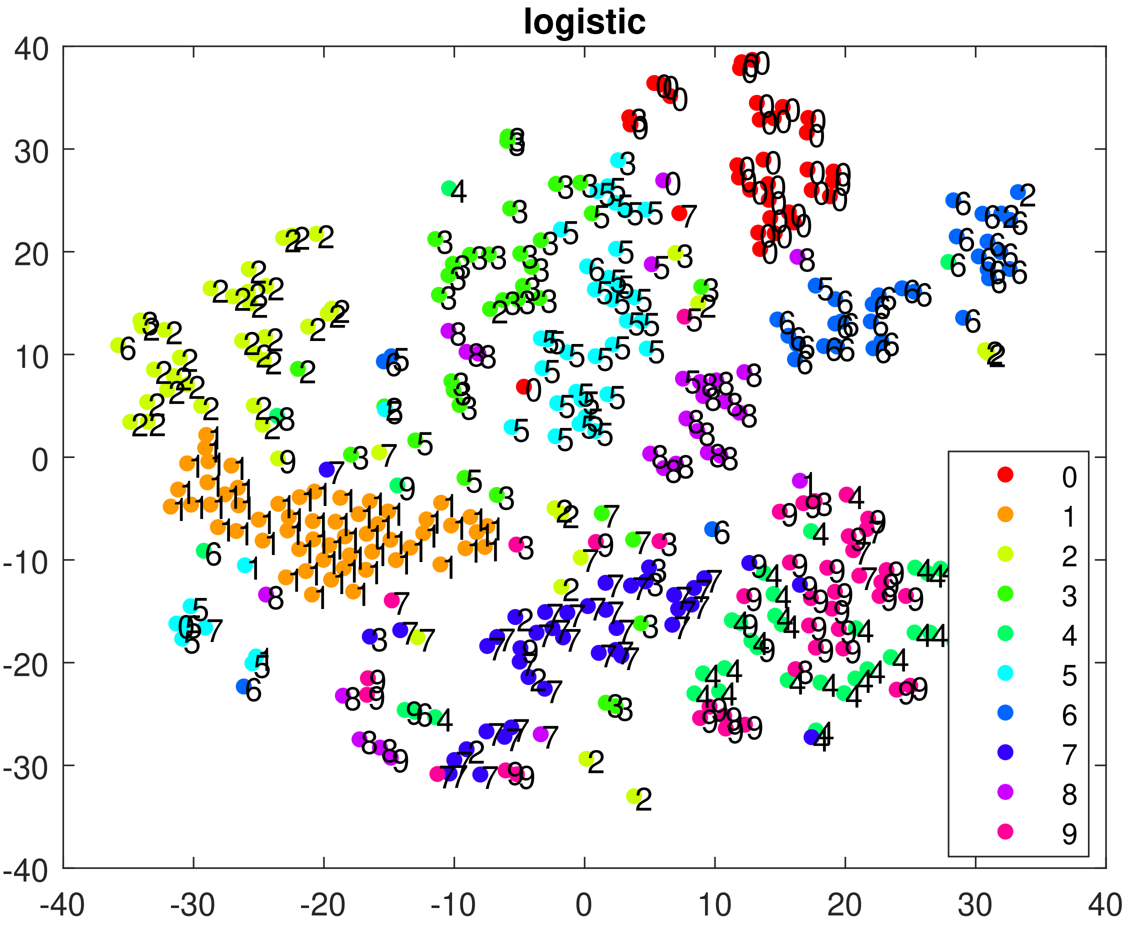


Figure 2.7: Visualization by use of t-SNE for the first 400 digits in the test set used in Example 2.17. The numbers close to points are the labels predicted by the logistic regression method, and the colours of points correspond to the true labels as given in the box in the lower right part of the image.

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 MNIST data in Example 2.17.

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, $j = 1, \ldots, n$, 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, $i = 1, \ldots, k$, on the basis of our observations $X_1, \ldots, X_n$. 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 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, $j = 1, \ldots, n$. Then we alternate between two types of steps. In the *observation allocation step* we suppose that we have cluster centers $C_i, i = 1, \ldots, k$, 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. □

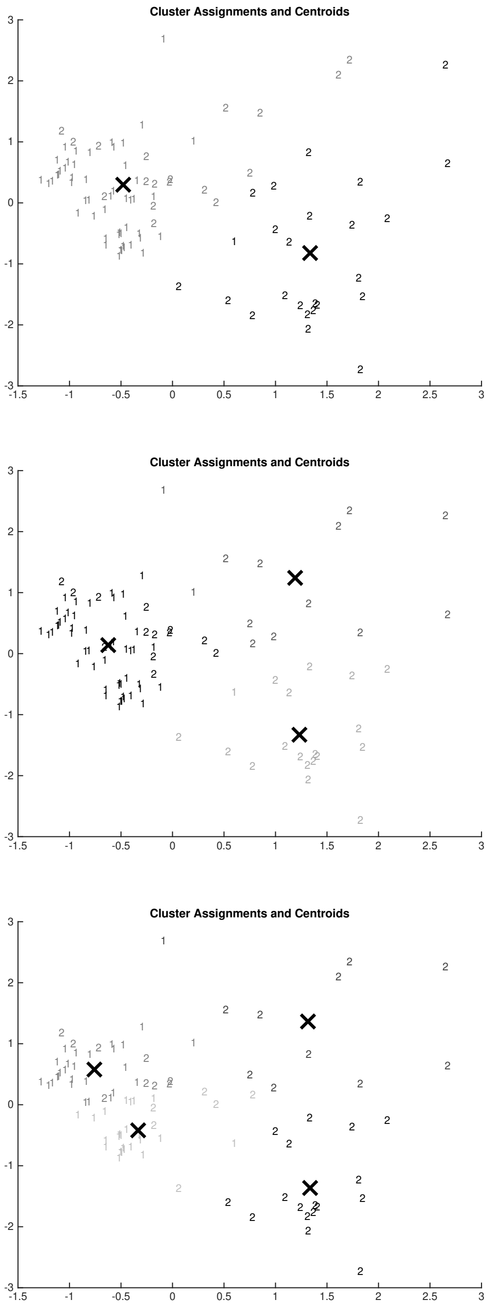


Figure 2.8: Results from k-means clustering with $k = 2$, 3 and 4 of the same data as used in Example 2.14. Crosses mark estimated cluster centers.

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 $x_i, i = 1, \dots, n_1$, 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 T of $|T|$ 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

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

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 $f_k(x, \theta), k = 1, \ldots, K$, for the possible class values. For a pair (x, y) where k_c is the correct class we can define $y_k, k = 1, \ldots, K$, 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

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

We can minimize $L(\theta, T)$ and obtain an estimate $\hat{\theta} = \hat{\theta}(T)$. The result is then that we often get a good fit to the observations in 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 $\theta = (\theta_1, \ldots, \theta_{|\theta|})$. Then we estimate θ by minimizing the regularized loss function

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

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 T' of pairs (x, y) or use cross-validation.

Convolutional neural nets

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

$$(w * g)_{ij} = \sum_k \sum_\ell w_{k\ell}\, g_{i-k,j-\ell}, \tag{3.12}$$

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

$$a_{ij}^{(r+1)} = \sigma \left(\sum_{k=-p}^{p} \sum_{\ell=-p}^{p} w_{k\ell}^{(r)}\, a_{i-k,j-\ell}^{(r)} \right), \tag{3.13}$$

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.

Let us conclude this short introduction to neural nets with mentioning two recent references, both with the title 'Deep Learning' which is a current term for advanced neural nets: (LeCun *et al.*, 2015) giving an overview and (Goodfellow *et al.*, 2016) with giving a thorough and up-to-date coverage of the field.

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

$$yf(x) > 0 \quad \text{for all} \quad (x, y) \in T. \tag{3.14}$$

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

$$\max_{\beta_0,\beta}\left\{M : \text{subject to } \frac{1}{||\beta||}y(\beta_0 + x^T\beta) \geq M \text{ for all } (x, y) \in T\right\}, \tag{3.15}$$

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

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

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

$$\min_{\beta_0,\beta}\left\{\sum_{(x,y)\in T}[1 - y(\beta_0 + x^T\beta)]_+ + \lambda||\beta||^2\right\}, \tag{3.17}$$

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. Similar 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-vectoer to 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

$$f(x) = \beta_0 + x^T\beta = \beta_0 + \sum_{i=1}^{|T|} \alpha^i x^T x^i, \tag{3.18}$$

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

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

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

$$k(u, v) = \mathrm{e}^{-||u-v[[^2}. \tag{3.20}$$

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

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.

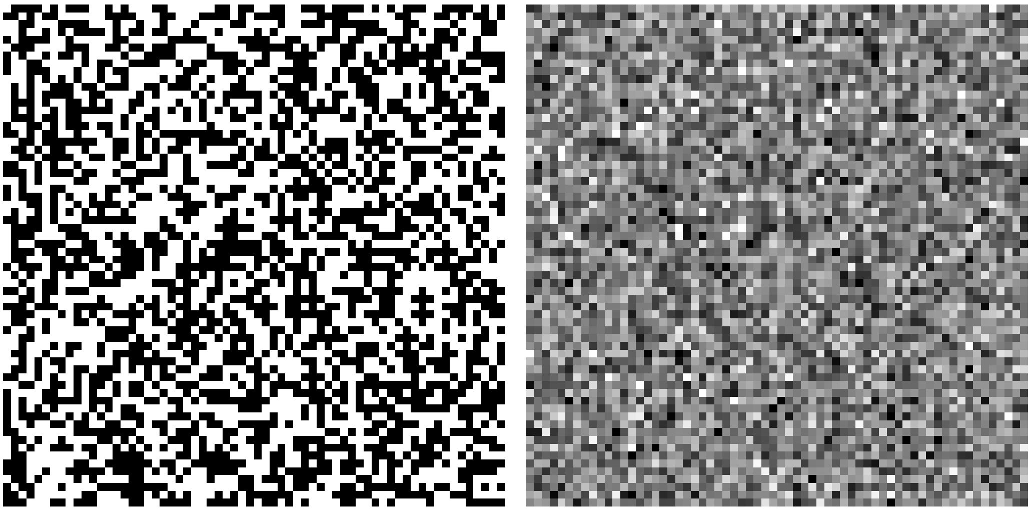


Figure 4.1: Images of size 64×64 obtained by simulation from models with independent pixel values: to the left a black-and-white image with equal probabilities for the two colours, and to the right a grey-level image with values from a normal distribution with expectation $\mu = 0.5$, a standard deviation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

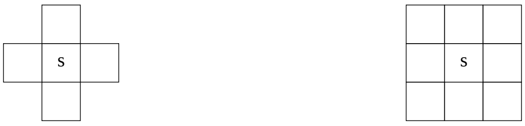


Figure 4.2: Two often used neighbourhood systems: to the left the site s has four neighbours and to the right it has eight neighbours.

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 3.1. 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 3.2. 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. □

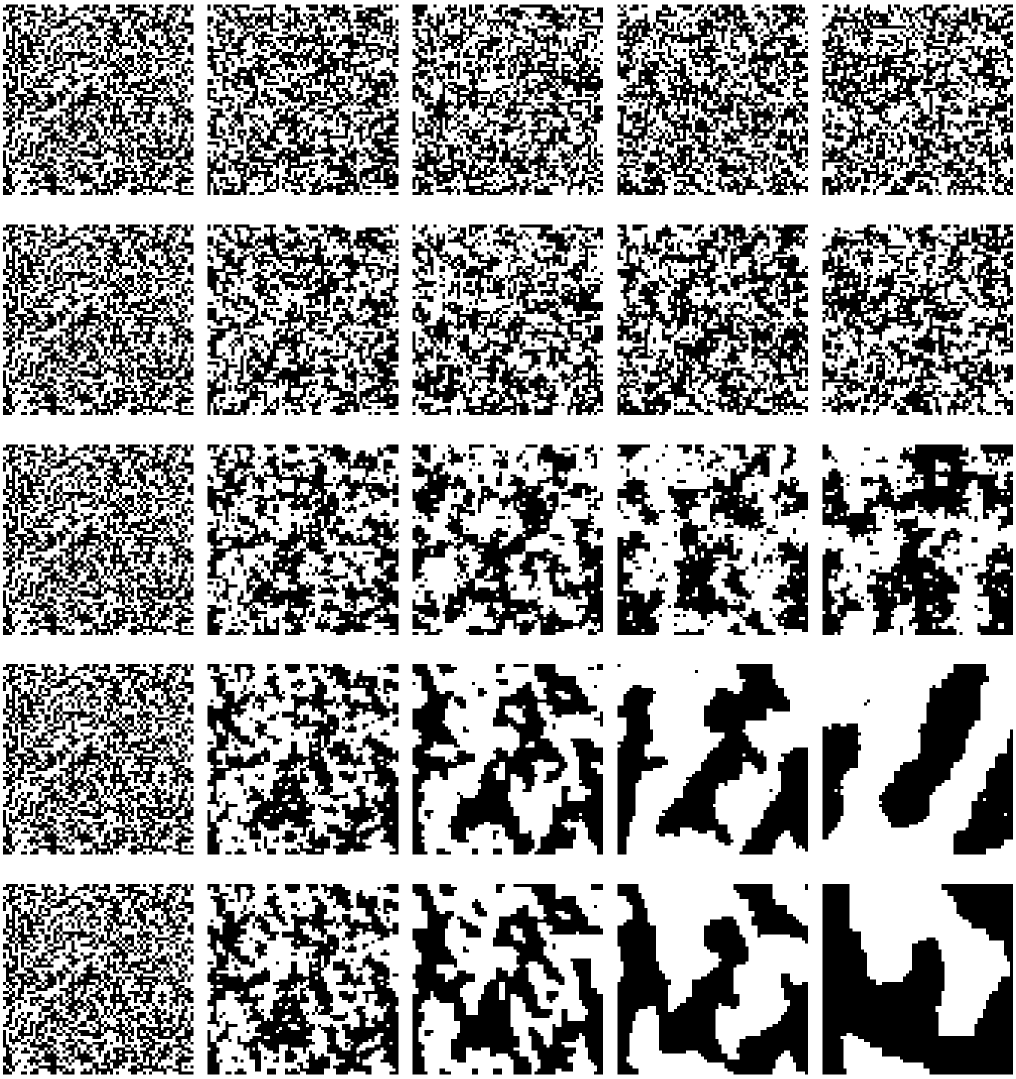


Figure 4.3: Binary images obtained by simulation for the Ising model with $\beta = 0.11$, 0.22, 0.4407, 0.88 and 1.76 in rows 1 to 5, respectively. In the columns we have to the left a purely random start configuration and then the result after 1 sweep, after 4 sweeps, after 16 sweeps and after 64 sweeps,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

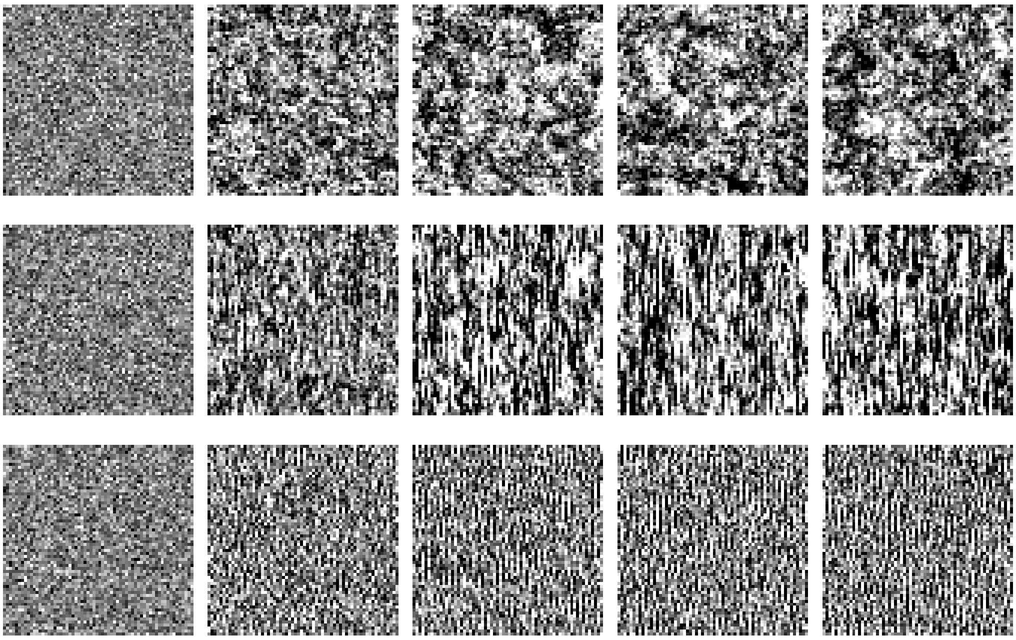


Figure 4.4: Grey-scale images obtained by simulation for autonormal models. In the columns we have to the left a purely random start configuration and then the result after 1 sweep, after 16 sweeps, after 128 sweeps and after 256 sweeps, respectively. The parameters in (4.11) are in the upper row $\beta_W = \beta_E = \beta_N = \beta_S = 0.24$, 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 3.3. 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. A recent 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 3.1. 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 3.2. 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 3.3. 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 3.4. 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

Let $X = (X_s, s \in S)$ be a spatial random process, 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 these notes 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 Gaussian random process is completely specified by its mean value and covariance functions. It should, however, 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 covariance function $C(s,t)$ is called *stationary* if $C(s, s+t)$ only depends on t, and it is called *isotropic* if it can be written on the form

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

where $|s-t|$ is the Euclidean distance between s and t. Examples of ρ-functions that give valid (positive-definite) covariance functions are

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

$$\rho(r) = \exp(-ar^2) \tag{5.7}$$

with a positive constant a, and

$$\rho(r) = (1 + r^2/b^2)^{-\beta} \tag{5.8}$$

with positive constants b and β.

Suppose now 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

$$C_0(s,t) = \begin{cases} \sigma_0^2 + C(s,t) & \text{if } s = t \\ C(s,t) & \text{if } s \neq t. \end{cases} \tag{5.9}$$

The constant σ_0^2 in (5.9) is sometimes called a *nugget* effect with regard to applications in mining. Another interpretation of the added quantity σ_0^2 in (5.9) is that it just corresponds to adding independent noise with variance σ_0^2 to all our original observations.

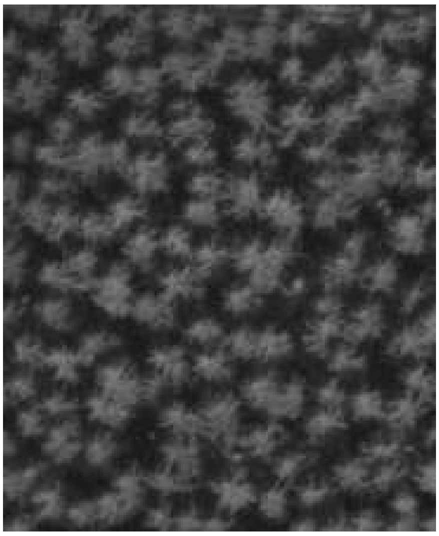


Figure 5.1: Aerial photograph of Norway spruce trees.

5.1 Prediction (kriging)

Suppose that

$$X_s = m_s + \epsilon_s, \tag{5.10}$$

where m_s is a slowly varying *trend* function, known or with a known parametric form, and that ϵ_s is a zero-mean random process with a covariance function, also assumed to be either known or of a known parametric form.

Suppose that we have observed X_{s_i}, $i = 1, \ldots, n$, and that we want to predict X_t. In mining this problem is often called kriging after the South African mining engineer D. G. Krige.

Assume first that the functions m and C are known. By regarding $X_s - m_s$ instead of X_s we can transform the problem into one where $m_s = 0$, which we now assume.

Consider a linear predictor

$$\hat{X}_t = \sum_{i=1}^{n} a_i X_{s_i} = a^T X_{(n)}, \tag{5.11}$$

where $a = [a_1 \ldots a_n]^T$ and $X_{(n)} = [X_{s_1} \ldots X_{s_n}]^T$ denotes the observations. We choose a to minimize the expected squared error

$$\mathbf{E}(\hat{X}_t - X_t)^2 = a^T G a - 2a^T g_t + \sigma^2(t), \tag{5.12}$$

where G is the $n \times n$-matrix with elements $G_{ij} = C(s_i, s_j)$, $g_t^T = [C(s_1, t) \ldots C(s_n, t)]$, and $\sigma^2(t) = C(t, t)$. It is straightforward to show that (5.12) is minimized for $a = G^{-1} g_t$, and the optimal predictor thus becomes

$$\hat{X}_t = X_{(n)}^T G^{-1} g_t. \tag{5.13}$$

The corresponding expected squared error becomes

$$\sigma^2_{\text{opt}}(t) = \sigma^2(t) - g_t^T G^{-1} g_t. \tag{5.14}$$

It should be noted that in practice we often only assume that m and C are of known parametric forms but with unknown parameters, and our observations $X_{(n)}$ have to be used to estimate these parameters.

5.2 Exercises

Exercise 4.1. Regard the image in Figure 5.1. The image $X_s, s \in S$ with $S = \{1, \ldots, 223\} \times \{1, \ldots, 183\}$ is available as `ku94-148Dpart.tif`
(*a*). Assume first that the random function $X_s, s \in S$, has a stationary covariance function that can be written on the form $C(s, s+t) = R(t_1, t_2)$ for $t = (t_1, t_2)$. Estimate the covariance function $R_1(t_1) = R(t_1, 0)$ and the covariance function $R_2(t_2) = R(0, t_2)$ in two orthogonal directions and plot the estimated functions R_1 and R_2 in the same diagram. Does it seem as the covariance function C is identical in the two directions studied?
(*b*). Assume now that the random function $X_s, s \in S$, has an isotropic covariance function. Try to estimate the corresponding ρ-function in (5.5).
(*c*). Assume that the random function $X_s, s \in S$, is stationary such that the distribution of X_s is the same for all $s \in S$. Try to estimate this distribution, often called the marginal distribution of X.

Exercise 4.2. Delete, say, the three bottom rows in the image in Figure 5.1. See how well you can reconstruct these three rows by use of prediction according to (5.13). Assume that the mean value function is a constant, which you estimate from the data. Use an isotropic covariance function with one of the three forms (5.6) – (5.8) with parameter(s) adapted to the result of Exercise 4.1(b). To limit computations in the prediction, use as $X_{s_1}, \ldots, X_{s_n}$ a limited set of observations from, say, the last two remaining rows. Note that if you want to use (5.13) for several g_t it is computationally advantageous to multiply together $X_{(n)}^T$ and G^{-1} before starting to vary g_t.

Exercise 4.3. Consider the three images in the rightmost column of Figure 4.4. Estimate the covariance function in two orthogonal directions (horisontal and vertical in the figure) as in Exercise 4.1 above. Can any of the three covariance functions be assumed to be isotropic?

Exercise 4.4. Show that if C is a valid covariance function, that is satisfies the inequality (5.3), then C_0 in (5.9) is also a valid covariance function.

Exercise 4.5. Verify that $\hat{X}_t$ in (5.13) minimizes (5.12) and that (5.14) gives the corresponding expected squared error.

5.3 Literature on spatial random processes

See (Ripley, 1981) and (Cressie, 1993).

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 6.19. *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 $X_1, \ldots, X_n$ as independent and identically distributed variables, each with a uniform distribution over A. (See Section 13.13 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 $X_1, \ldots, X_n$, that is $X = \{X_1, \ldots, X_n\}$.

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

□

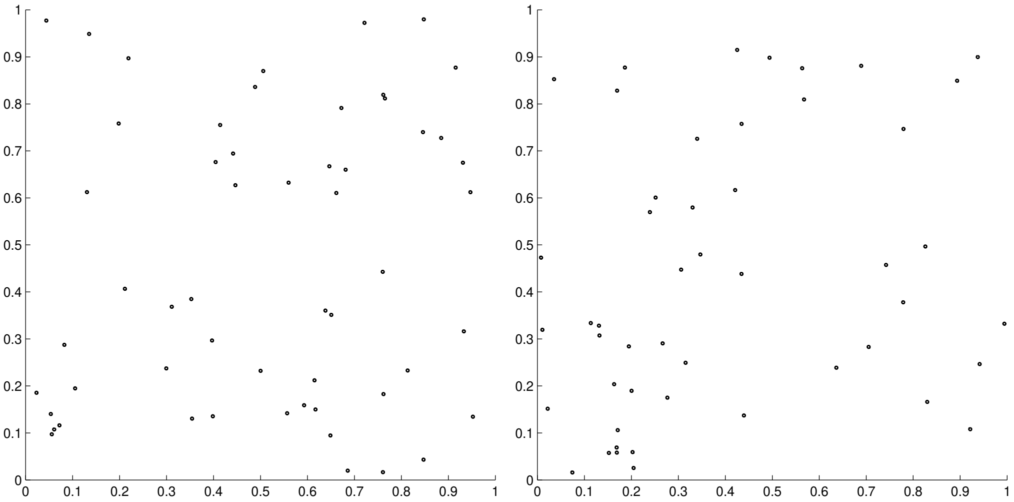


Figure 6.1: Two examples of Poisson point processes generated in the unit square with $\lambda = 50$. The generated number of points is to the left $N = 55$ and to the right $N = 49$.

Example 6.20. *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 6.21. *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.

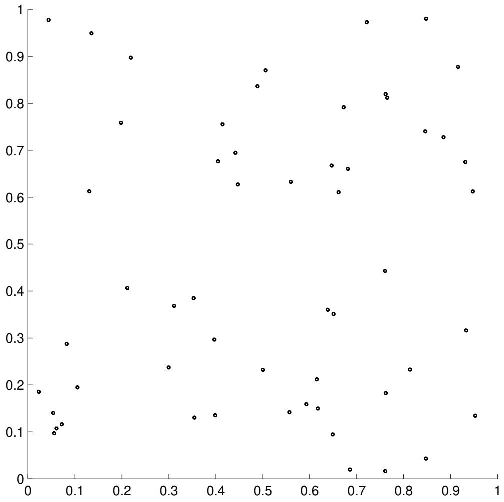


Figure 6.2: A simulation of a Neyman-Scott process with mother points as circles and daughter points as dots. OBS OBS a new figure must be generated.

□

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 $X_i, i = 1, \ldots, N$, 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,

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

or more precisely

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

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

$$K(r) = \pi r^2. \qquad (6.6)$$

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

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

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 (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 parameters of these mechanisms are studied by consideration of the conditional likelihood $L(Y|X)$ of Y given X.

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 $X_1, \ldots, X_M$ of a Poisson process on A conditioned on $X_m(A) = n, m = 1, \ldots, M$. 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, $m = 1, \ldots, M$. 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.

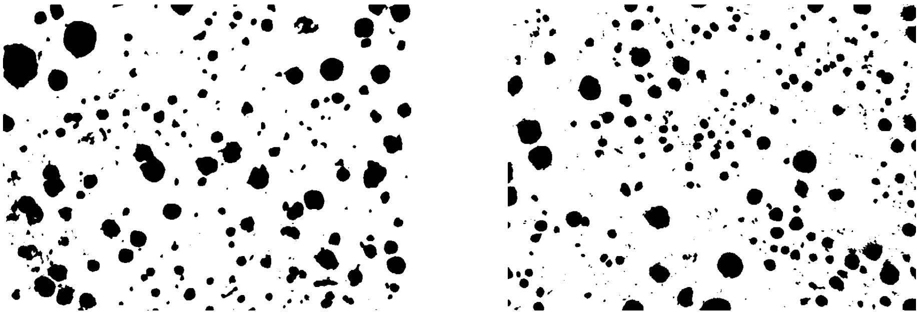


Figure 7.1: Binary images of two cuts in cast iron showing approximately disk-shaped defects. Data from Beretta (2000) and Månsson and Rudemo (2002).

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.

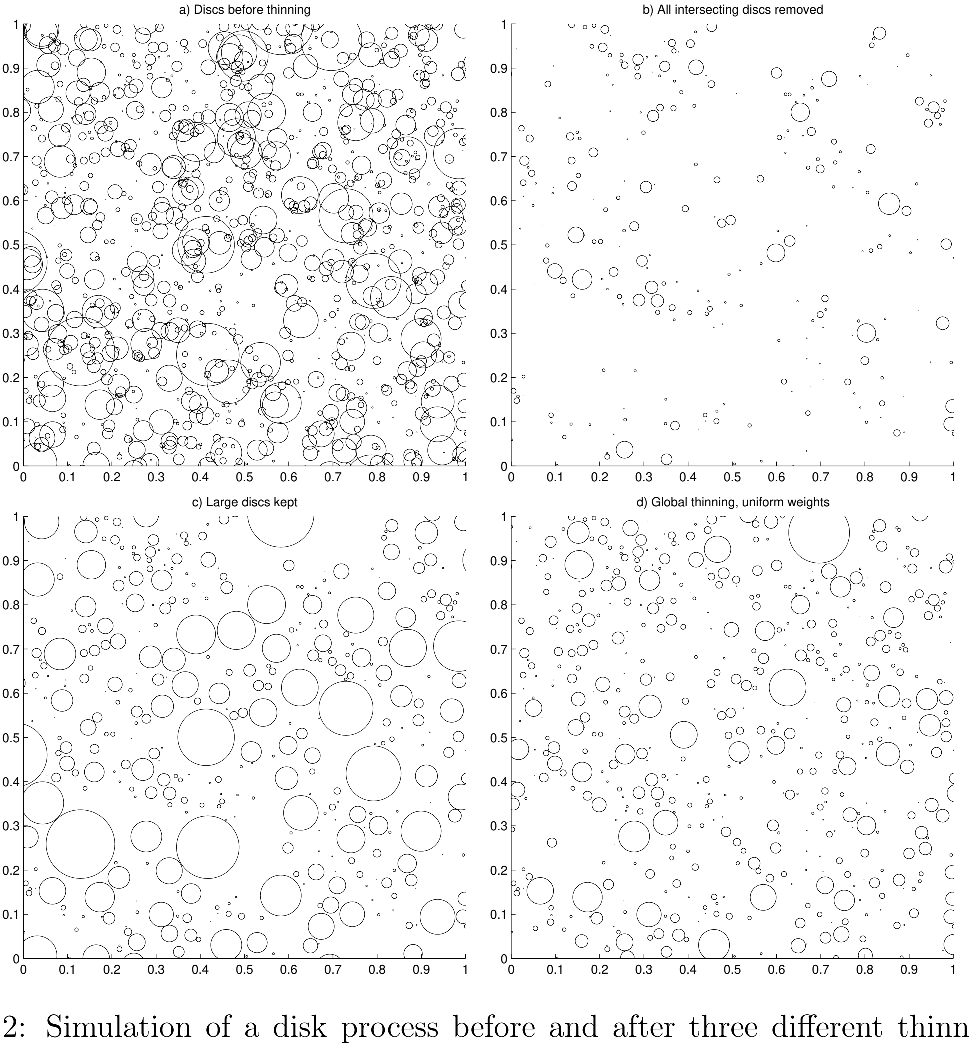


Figure 7.2: Simulation of a disk process before and after three different thinning procedures. In the first step a Poisson process with intensity 1000 in the unit square is generated with exponentially distributed disk radii with expectation 0.01.

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

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

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 $x_1'x_2'$-plane (but generally not with pairwise parallell sides).

Another nonlinear warping function is the perspective transformation

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

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 8.22. *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.

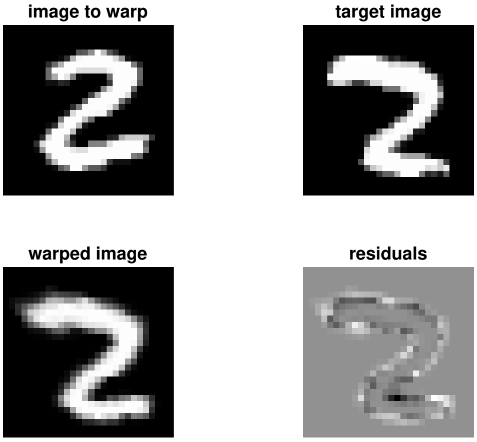


Figure 8.1: Warping of the digit "2" left in upper row to the digit "2" right in the same row. The lower row shows the warped image and the residuals relative to the target upper right.

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 $\mu_j, j = 0, \ldots, 9$, denote average iteratively warped image for digits j as described above, and where μ_5 is shown in Figure 8.4. To

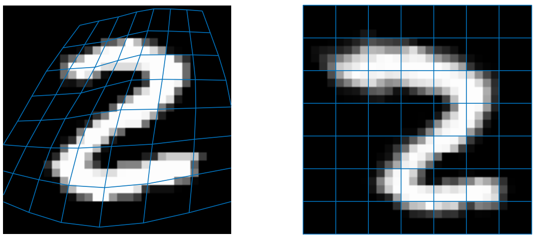


Figure 8.2: Original and warped handwritten digits also shown in Figure 8.1, upper left and lower left, here with the 7×7-grid for the bilinear transformations. The target is the upper right digit in Figure 8.1.

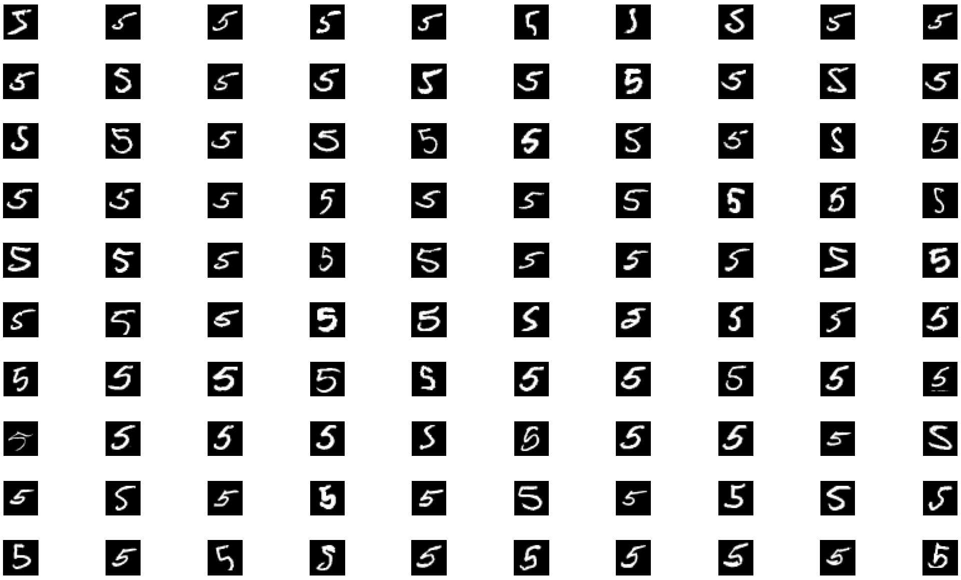


Figure 8.3: First 100 digits “5” in the MNIST database.

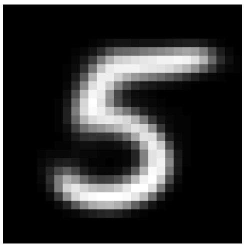


Figure 8.4: Average handwritten digit “5” obtained by sequential warping and averaging.

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.

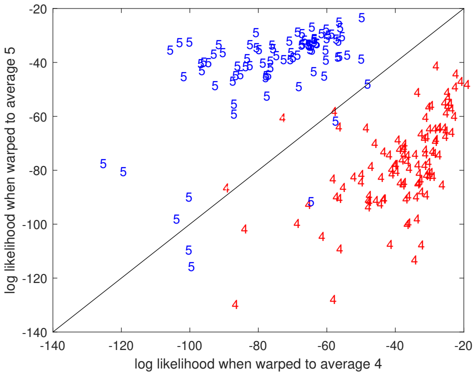


Figure 8.5: Classification of 110 handwritten digits “4” and 87 digits “5” by warping classification. Penalized log-likelihoods for the two types of digits are shown on the axes. Six digits are miss-clasified.

□

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 $r_s(x) =\parallel x - c_s \parallel$ 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

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

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

The polynomial-hyperbolic shape model. Here

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

with

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

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 $Y(x) \geq \ell_c$. Let $A_s' = \{x \in A_s : Y(x) < \ell_c\}$ and $A_s'' = \{x \in A_s : Y(x) \geq \ell_c\}$ 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

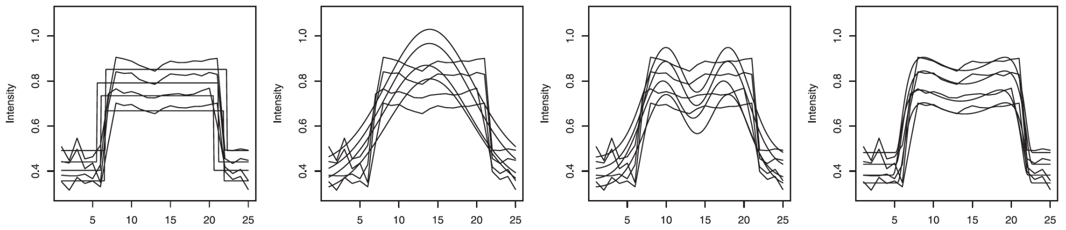


Figure 9.1: One-dimensional intensity profiles (through spot center) for observed intensities of one spot, four photometric gains and maximum likelihood fits for the four spot shape models (9.6), (9.7), (9.8) and (9.9).

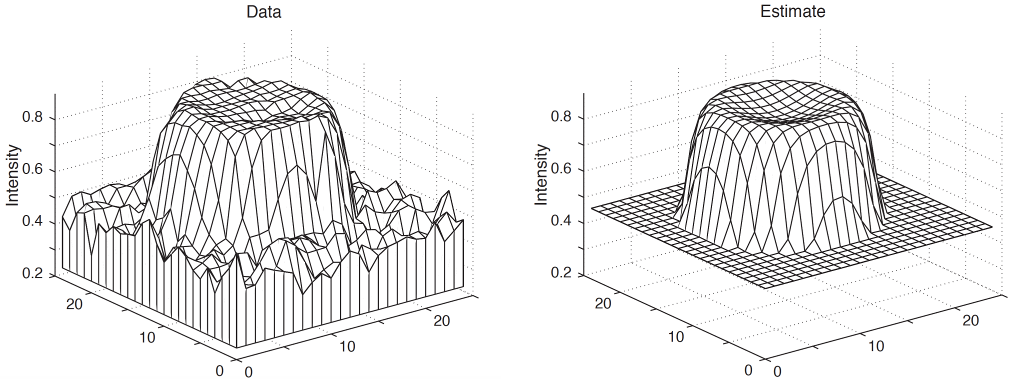


Figure 9.2: Three-dimensional plot (for one photometeric gain) of observed intensities (left surface) for the same spot as in Figure 9.1 and the corresponding estimated spot shape for the polynomial-hyperbolic shape model (right surface).

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.

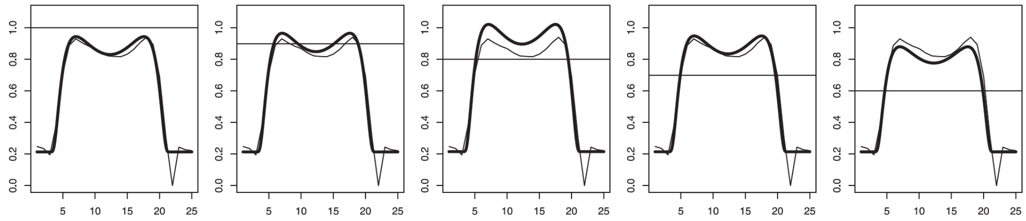


Figure 9.3: One-dimensional intensity profiles through the center of one spot together with reconstructions by use of the polynomial-hyperbolic spot shape model for different levels of artificial saturation indicated by horizontal lines. Both data (thin curves) and reconstructions (heavy curves) are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

The *exponential correlation function*

$$\rho(r,c) = \exp(-r/c), \tag{9.16}$$

the *Gaussian correlation function*

$$\rho(r,c) = \exp(-(r/c)^2), \tag{9.17}$$

the *linear correlation function*

$$\rho(r,c) = (1 - r/c)1(r < c), \tag{9.18}$$

the *rational quadratic correlation function*

$$\rho(r,c) = \frac{1}{1 + (r/c)^2} \tag{9.19}$$

and the *spherical correlation function*

$$\rho(r,c) = (1 - \frac{2}{3}(r/c) + \frac{1}{2}(r/c)^3)1(r < c). \tag{9.20}$$

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

$$\log L = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log(\det C) - \frac{1}{2}(Y - \mu)^T C^{-1}(Y - \mu), \tag{9.21}$$

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

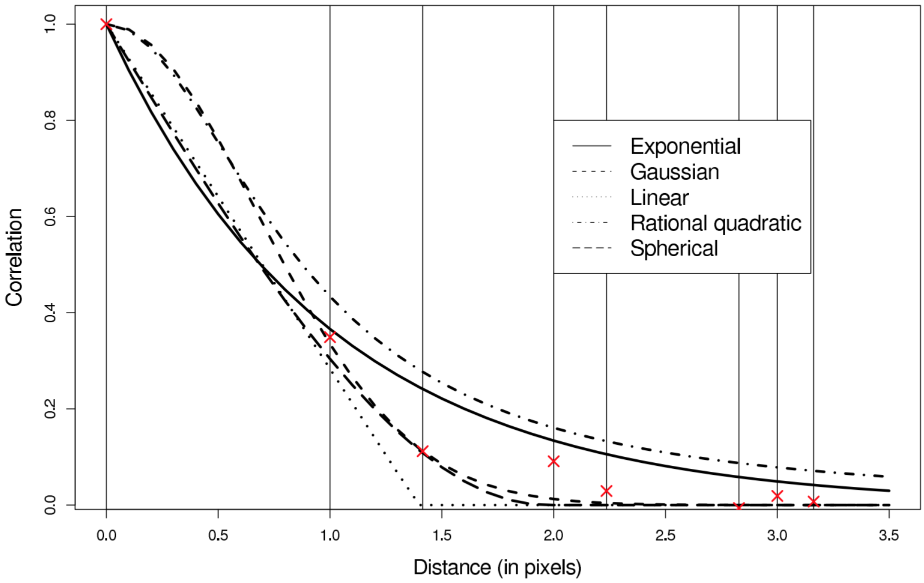


Figure 9.4: Median estimated correlation functions for the five studied correlation structures. The possible observable distances between pixel centres are shown by vertical lines and the crosses on these lines show the median empirical correlation coefficients.

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.

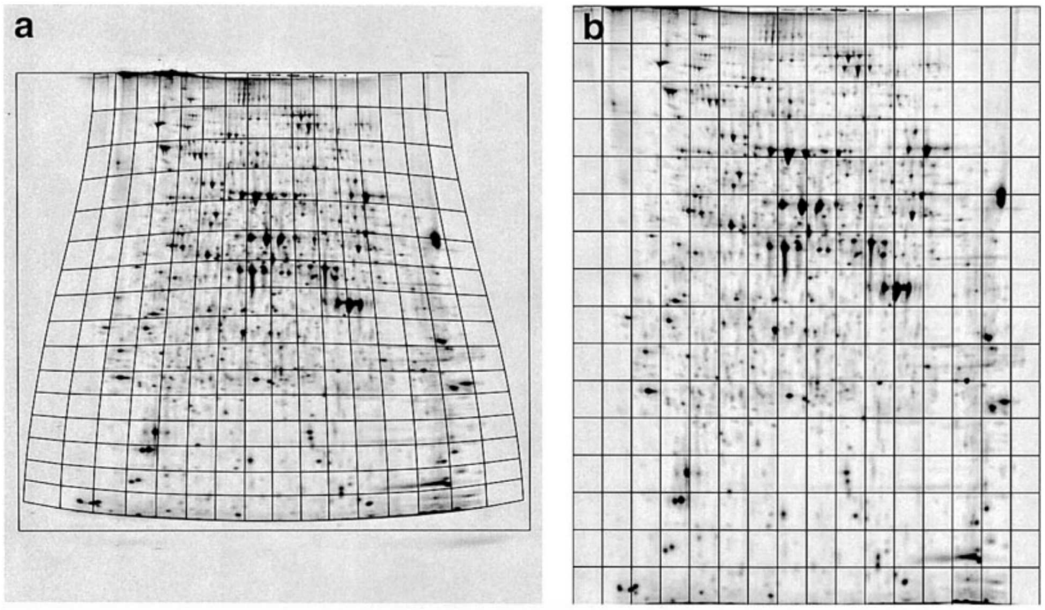


Figure 10.1: Illustration of warping step I with correction for current leakage sideways through the left and right boundaries during the second-dimensional gel electrophresis. Part **a** of the figure shows the original image and part **b** shows the warped current-leakage corrected image.

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.

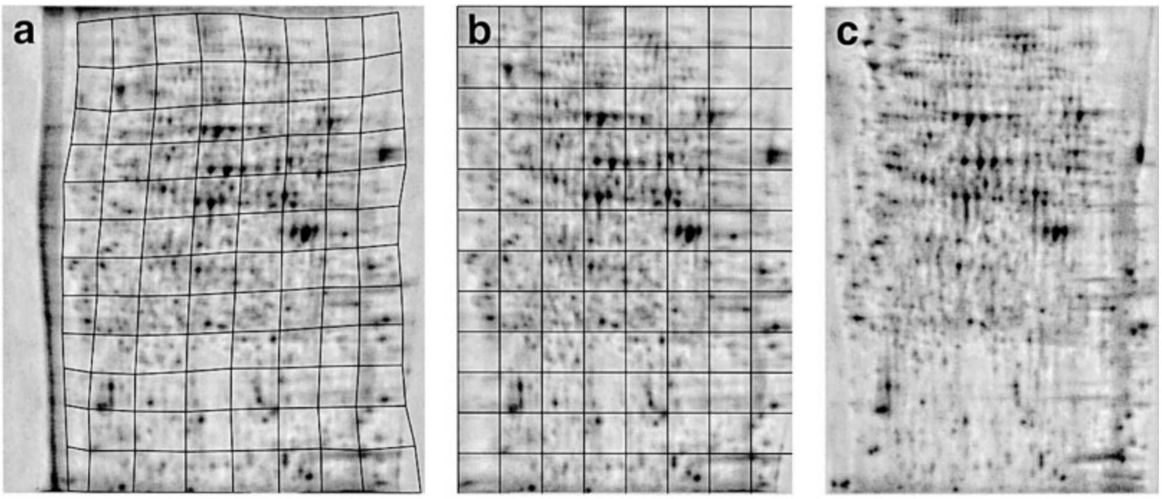


Figure 10.2: Illustration of warping step II. The image in **a** is warped onto the reference image in **c** by use of the grid shown in **a** warped to the grid in **b**.

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.

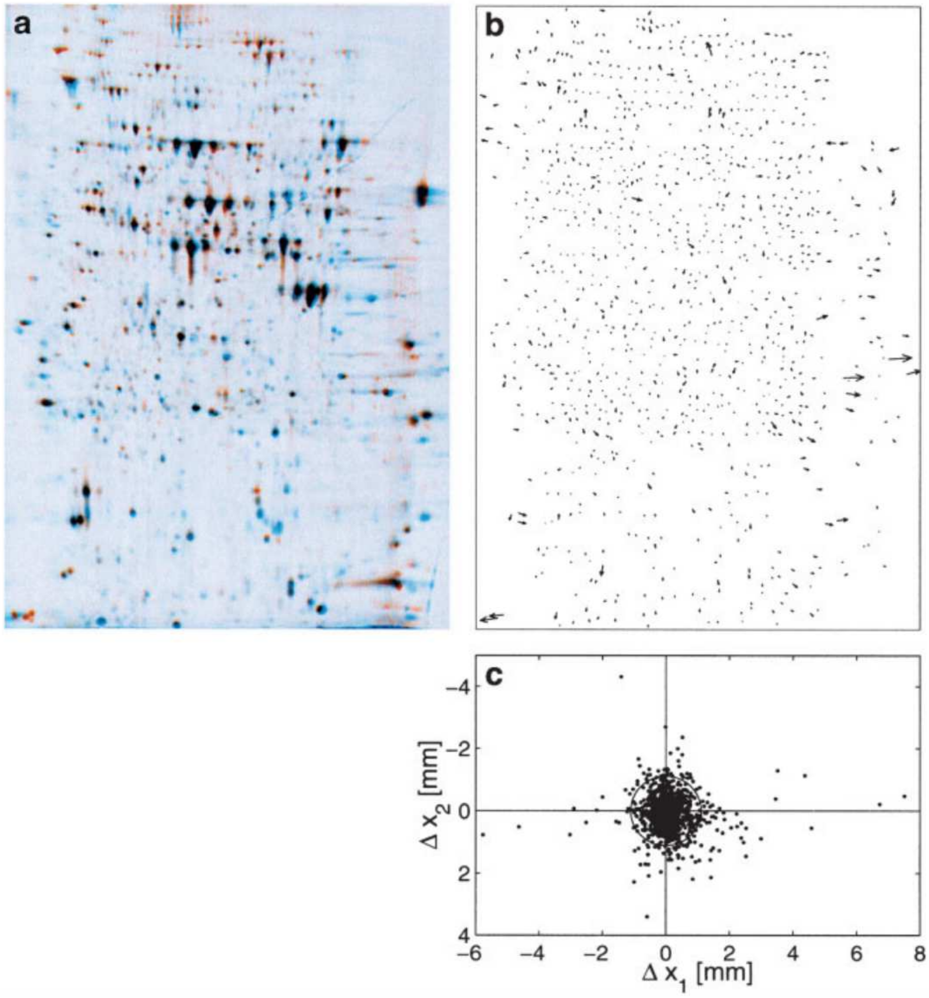


Figure 10.3: Further illustration of warping step II. In part **a** the reference image coloured red and the warped image coloured blue are superimposed. Displacement vectors for spots are shown in part **b**, and also in part **c**, here as relocated vectors starting at the origin and ending at dots.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

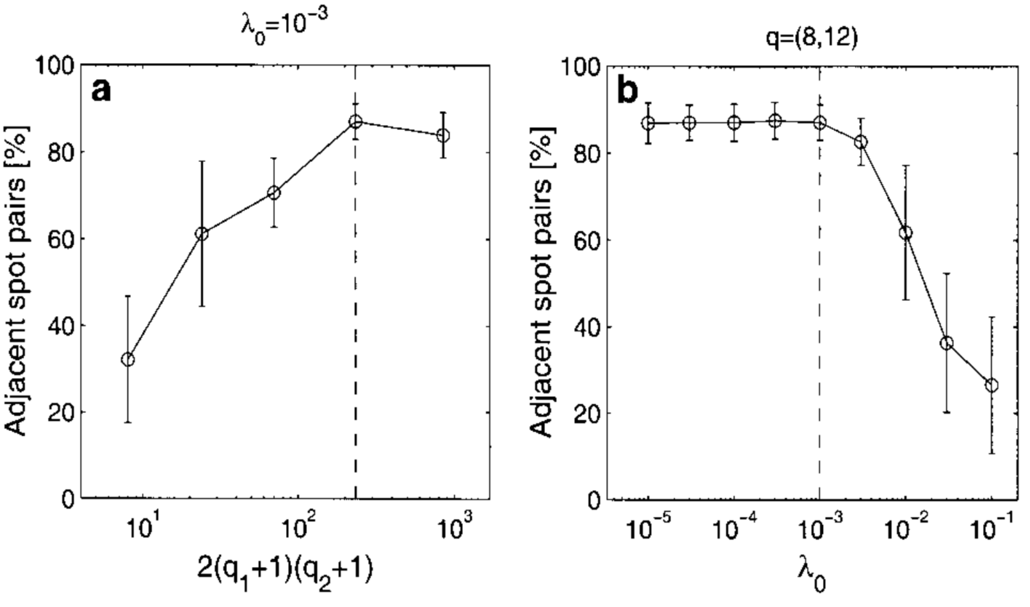


Figure 10.4: Illustration of spot pattern similarity in aligned images. The left part **a** shows the effect of changing grid size for the particular λ-value 10^{-3}. The graph shows the percentage of adjacent spot pairs as a function of the number of grid size parameters. The right part **b** shows 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

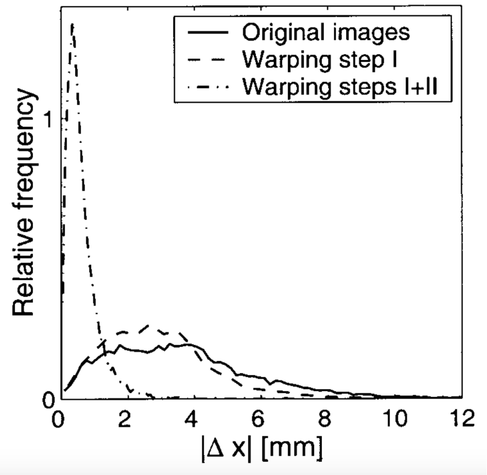


Figure 10.5: Length distribution of spot displacement vectors for the original data (solid line), after the current leakage warping step (dashed line) and after both warping steps (dash-dot line).

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.

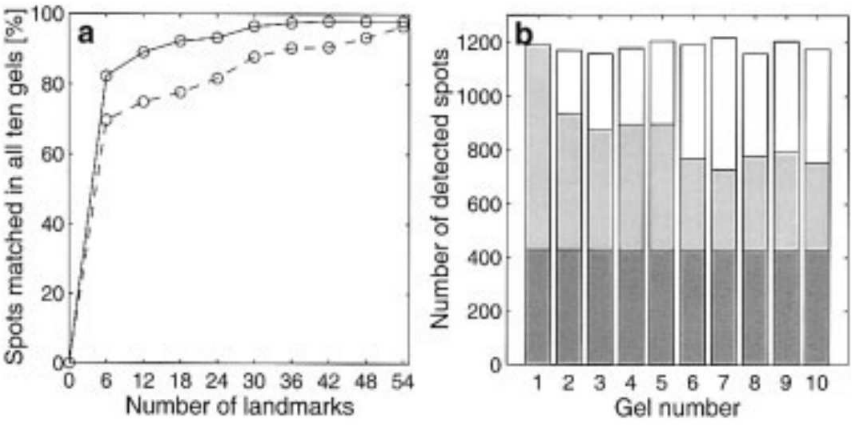


Figure 10.6: Efficiency profiles in the left part **a** showing the number of automatically matched spots in all ten gels (with gel images two-step warped) by the software PDQuest as a function of an initial manual matching of a number of spots (in the image called landmarks) both for the original set 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 the number of spots detected in all ten gels is 430.

10.1 Exercises

Exercise 10.1. As mentioned above a top-hat transformstion 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

Aerial photographs of forests

Read the following parts of Dralle & Rudemo (1997):
Abstract
Introduction (skim this part)
Data (skim this part)
Problem specification
A model for the grey-level maxima given tree positions and heights
Parameter estimation
Results
Discussion
Conclusions

Read the following parts of Larsen & Rudemo (1998):
Abstract
1. Introduction
2. The opticaL model (skim this part)
3. Local correlation maxima
4. Experiment
5. Discussion
6. Conclusions

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 A pixel-based likelihood framework for analysis of fluorescence recovery after photobleaching

Read the following parts of Jonasson et al. (2008):
Read Summary
Skim Introduction
In Theory:
  Read Model
  Skip Fluorescence intensity and fluorochrome concentration
  Skip The detection point spread function
Skip Materials and methods, but read the last part starting with “To maximize the log-likelihood” on top of page 265
In Results read only the last part Diffusion in PEG solutions
Read Discussion, conclusions and outlook

12.3 Estimation of particle concentration from fluorescent particle counting

Read the following parts of Röding et al. (2011):
Read Abstract
Skim I. Introduction
Read II. Theory till A. Trajectory length distribution
Skim A. Trajectory length distribution
Skim B. Number concentration
Skim III. Simulation study
Skim IV. Experimental results
Read V. Discussion and conclusion
Skip Appendix A, B and C but have a look at Fig. 8

Read the following parts of Röding et al. (2013):
Read Summary
Skim Introduction
Read Theory and methods, Concentration measurements till equation (3) on p 21, skim the rest of this section till Bootstrap confidence intervals
Skip Bootstrap confidence intervals
Skim Simulation study
Skim Experimental results
Read Discussion and conclusion
Skip Appendix A-D

Chapter 13

Appendix. Mathematical, computational and statistical background

Below you can find condensed descriptions of concepts and methods used in these notes. 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.

13.1 Some matrix algebra

A matrix with m rows and n columns, or briefly a matrix of type $m \times n$, is a rectangular array

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \tag{13.1}$$

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

$$AA^{-1} = A^{-1}A = I \tag{13.2}$$

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

$$\det A = \sum_{j=1}^{n} (-1)^{1+j} a_{1,j} A_{1,j}. \tag{13.3}$$

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

$$Ax = \lambda x. \tag{13.4}$$

A symmetric real $n \times n$ matrix A is said to be positive-definite or positive-semidefinite if $x^T Ax > 0$ or $x^T Ax \geq 0$, 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 11.1. Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. Determine $\det A$ by use of (13.3).

Exercise 11.2. 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.

13.2 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

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}. \tag{13.5}$$

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

$$\nabla f(x) = [\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n}]^T \tag{13.6}$$

and let $Hf(x)$ denote the Hessian matrix

$$Hf(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \tag{13.7}$$

Newton's iterative method for locating $x^\star$ is to put

$$x^{k+1} = x^k - (Hf(x^k))^{-1} \nabla f(x^k) \tag{13.8}$$

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.

13.3 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 $\sigma^2 = \mathrm{var}(X) = \sum_x (x-\mu)^2 \Pr(X = x)$.

A random variable X is said to be Poisson distributed with parameter λ if

$$\Pr(X = n) = \frac{\lambda^n}{n!} \exp(-\lambda), \;\; n = 0, 1, \ldots, \tag{13.9}$$

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

$$\Pr(X = k) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k}, \;\; k = 0, \ldots, n, \tag{13.10}$$

and for such a variable the expectation is np and the variance is $np(1-p)$.

13.4 Continuous probability distributions

Continuous distributions for a real-valued random variable X are characterized by the probability density

$$f(x) = \frac{d}{dx}\Pr(X \leq x), \;\; x \in \mathbb{R}, \tag{13.11}$$

where $\mathbb{R} = (-\infty, \infty)$ is the set of real numbers. For a continuous random variable the expextation 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

$$f(x) = 1/(b-a), \;\; a < x < b, \tag{13.12}$$

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

$$f(x) = \beta \exp(-\beta x), \;\; x > 0, \tag{13.13}$$

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

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-(x-\mu)^2/\sigma^2), \;\; x \in \mathbb{R}, \tag{13.14}$$

and for such a variable the expectation is μ and the variance is σ^2.

13.5 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

$$C = \mathbf{E}(X-\mu)(X-\mu)^T = \begin{bmatrix} E(X_1-\mu_1)(X_1-\mu_1) & \cdots & E(X_1-\mu_1)(X_d-\mu_d) \\ \vdots & & \vdots \\ E(X_d-\mu_d)(X_1-\mu_1) & \cdots & E(X_d-\mu_d)(X_d-\mu_d) \end{bmatrix}. \tag{13.15}$$

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

$$\Pr(X \in A) = \int_A f(x)dx \tag{13.16}$$

for subsets A of d-dimensional space $\mathbb{R}^d$ for which the integral in (13.16) 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

$$f_X(x) = \frac{1}{(2\pi)^{d/2}(\det C)^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)), \tag{13.17}$$

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

$$C = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}. \tag{13.18}$$

One can then show that the two-dimensional density funcion of X is

$$f(x) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\{-\frac{1}{2(1-\rho^2)}Q(x_1, x_2)\} \tag{13.19}$$

where

$$Q(x_1, x_2) = \frac{(x_1-\mu_1)^2}{\sigma_1^2} - 2\rho(\frac{x_1-\mu_1}{\sigma_1})(\frac{x_2-\mu_2}{\sigma_2}) + \frac{(x_2-\mu_2)^2}{\sigma_2^2} \tag{13.20}$$

13.6 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

$$Y_1 = c_1^T X, \tag{13.21}$$

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 minimal subject to the restriction $c_1^T c_1 = 1$. Generally we define the ith principal component, $1 < i \leq d$ as

$$Y_i = c_i^T X, \tag{13.22}$$

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 minimal 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 (Longfils, 2018).

13.7 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,

$$m_t = \mathbf{E}X_t \tag{13.23}$$

and its covariance function

$$C(s,t) = \mathbf{E}(X_s - m_s)(X_t - m_t). \tag{13.24}$$

A random process is said to be normal or Gaussian if $(X_{t_1}, \ldots, X_{t_n})$ has an n-dimensional normal distribution for any choice of time points $t_1, \ldots, t_n$. 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 $(t_1, \ldots, t_n)$ the distribution of the n-dimensional random vector $(X_{t_1+\tau}, \ldots, X_{t_n+\tau})$ 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

$$\Pr(X_\tau \in A | X_s, s \leq t) = \Pr(X_\tau \in A | X_t), \;\; t < \tau. \tag{13.25}$$

A discrete time Markov process with finite state space V, for notational simplicity here denoted $V = \{1, \ldots v\}$, is determined by its transition probability matrix P which is the $(v \times v)$ matrix with elements

$$p_{ij} = \Pr(X_{t+1} = j | X_t = i), \;\; 1 \leq i,j \leq v. \tag{13.26}$$

A zero-mean autoregressive process (X_t) of order p is recursively generated from

$$X_t = \sum_{i=1}^{p} a_i X_{t-i} + \epsilon_t \tag{13.27}$$

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. 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 (13.27) 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

$$\mathbf{E}(X_t - X_s) = \lambda(t - s), \;\; s < t, \tag{13.28}$$

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

$$\rho(t) = \exp(-\lambda t) \tag{13.29}$$

for some positive constant λ.

13.8 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

$$\mathbf{E}(\hat{\theta}) = \theta. \tag{13.30}$$

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 $\epsilon > 0$

$$\Pr(|\hat{\theta} - \theta| > \epsilon) \to 0 \tag{13.31}$$

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

$$L(\theta) = L(\theta|x) = f(x,\theta). \tag{13.32}$$

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

$$L(\theta) = L(\theta|x) = \prod_{i=1}^{n} f(x_i,\theta), \tag{13.33}$$

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

$$\ell(\theta) = \log(L(\theta)), \tag{13.34}$$

where log (as always in these notes) 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

$$\ell(\lambda) = \log(\prod_{i=1}^{n} \frac{\lambda^{X_i}}{X_i!}\exp(-\lambda)) = c - n\lambda + \log(\lambda)\sum_{i=1}^{n} X_i, \tag{13.35}$$

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

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{13.36}$$

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

$$Q(\theta) = \sum_{i=1}^{n}(X_i - \mathbf{E}(X_i))^2. \tag{13.37}$$

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 (13.37) now becomes

$$Q(\lambda) = \sum_{i=1}^{n} (X_i - \lambda)^2, \tag{13.38}$$

which is minimized for $\lambda = \hat{\lambda}$ in (13.36). Thus the least squares and the maximum likelihood estimates coincide in this example.

13.9 Linear and logistic regression

Let us first consider linear regression with one explaining real variable x. Suppose that we observe

$$Y_i = \alpha + \beta x_i + \epsilon_i, \ \ i = 1, \ldots n, \tag{13.39}$$

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

$$Q(\alpha, \beta) = \sum_{i=1}^{n} (Y_i - \alpha - \beta x_i)^2, \tag{13.40}$$

which gives

$$\hat{\alpha} = \overline{Y} - \hat{\beta}\,\overline{x}, \qquad \hat{\beta} = \frac{\sum_{i=1}^{n} (Y_i - \overline{Y})(x_i - \overline{x})}{\sum_{i=1}^{n} (x_i - \overline{x})^2}, \tag{13.41}$$

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

$$Y_i = \beta_1 x_{i1} + \ldots + \beta_m x_{im} + \epsilon_i, \ \ i = 1, \ldots n, \tag{13.42}$$

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

$$Y = X\beta + \epsilon, \tag{13.43}$$

where

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}, \quad X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}. \tag{13.44}$$

It turns out that the least squares estimate of the parameter vector β is

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \tag{13.45}$$

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

$$\log(\frac{p_i}{1-p_i}) = \beta_1 x_{i1} + \ldots + \beta_m x_{im}, \ \ i = 1, \ldots n. \tag{13.46}$$

To estimate the parameters $\beta_1, \ldots, \beta_m$ we can maximize the likelihood function

$$L(\beta_1, \ldots, \beta_m) = \prod_{i=1}^{n} (p_i^{Y_i} (1 - p_i)^{1-Y_i}). \tag{13.47}$$

There is no analytical expression for the maximum likelihood estimates so to maximize (13.47) one may use computational optimization methods such as those describe in Section 13.2 and then it is typically more convenient to maximize the log-likelihood function.

13.10 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

$$\Pr(L(X) < \theta < U(X)) = p. \tag{13.48}$$

Let $X = (X_1, \ldots, X_n)$ be a sample from a normal(μ, σ^2) distribution. Then

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \text{and} \quad s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X})^2 \tag{13.49}$$

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

$$\chi^2 = \sum_{i=1}^{r} Z_i^2, \tag{13.50}$$

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

$$t = \frac{Z}{\sqrt{\chi^2/r}} \tag{13.51}$$

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 (13.49) one can then show that

$$\Pr(\chi^2_{(1-p)/2} < (n-1)s^2/\sigma^2 < \chi^2_{(1+p)/2}) = p \tag{13.52}$$

which gives a confidence interval for σ^2 with confidence degree p,

$$\Pr(\frac{(n-1)s^2}{\chi^2_{(1+p)/2}} < \sigma^2 < \frac{(n-1)s^2}{\chi^2_{(1-p)/2}}) = p. \tag{13.53}$$

Similarly we let t_p denote the pth quantile of a t-distribution with $n-1$ degrees of freedom. Then

$$\Pr(\overline{X} - t_{(1-p)/2}\ s/\sqrt{n} < \mu < \overline{X} + t_{(1-p)/2}\ s/\sqrt{n}) = p, \tag{13.54}$$

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

$$p = \mathrm{Pr}_0(T \geq T_{obs}) \tag{13.55}$$

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 (13.49) and put

$$t_{obs} = \frac{\overline{X} - \mu_0}{s/\sqrt{n}}. \tag{13.56}$$

The corresponding p-value is then

$$p = P(|t| \geq |t_{obs}|) \tag{13.57}$$

evaluated with the assumption that t is t-distributed with $n-1$ degrees of freedom.

13.11 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

$$F = \frac{\chi^2_1/r_1}{\chi^2_2/r_2}, \tag{13.58}$$

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

$$\mathbf{E}(X_{ij}) = \mu_i, \;\; i = 1, \ldots, m, \, j = 1 \ldots, n_i. \tag{13.59}$$

To test the hypothesis $H_0 : \mu_1 = \ldots = \mu_m$ we can use the test variable

$$F = \frac{\sum_{i=1}^m n_i(\overline{X_{i\cdot}} - \overline{X_{\cdot\cdot}})^2 \;/\; (m-1)}{\sum_{i=1}^m \sum_{j=1}^{n_i} (X_{ij} - \overline{X_{i\cdot}})^2 \;/\; (\sum_i (n_i - 1))} \tag{13.60}$$

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 (13.60) 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.

13.12 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 (13.34), 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

$$\hat{\theta} \xrightarrow{d} N(\theta, \mathcal{I}(\hat{\theta})^{-1}). \tag{13.61}$$

Here $\mathcal{I}(\hat{\theta})$ is the Fisher information matrix with matrix elements

$$\mathcal{I}_{ij}(\hat{\theta}) \;=\; -\frac{\partial^2}{\partial\theta_i \partial\theta_j} \ell(\theta)|_{\theta=\hat{\theta}} \tag{13.62}$$

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 13.2) of the log-likelihood function and as discussed in Section 13.2 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

$$\chi^2 = 2(\ell(\hat{\theta}_1) - \ell(\hat{\theta}_0)). \tag{13.63}$$

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 (13.63) 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

$$p = \Pr(\chi^2 \geq \chi^2_{obs}) \tag{13.64}$$

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).

13.13 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

$$X_{n+1} = (aX_n + b) \mod m \tag{13.65}$$

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

$$X = F^{-1}(U), \tag{13.66}$$

where F^{-1} denotes the inverse of F. Putting

$$X = -\frac{1}{\beta} \log(1 - U) \tag{13.67}$$

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.

13.14 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 follows.

$$\Pr(\theta = \theta_i | X = x) = \frac{\Pr(X = x|\theta_i)\pi_i}{\Pr(X = x)} = \frac{\Pr(X = x|\theta_i)\pi_i}{\sum_j \Pr(X = x|\theta_j)\pi_j} \tag{13.68}$$

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 **??** in this book for a brief summary and (Gilks *et al.*, 1996) for more details.

13.15 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

$$X_t = F_t X_{t-1} + W_t, \quad t = 1, 2, \ldots, \tag{13.69}$$

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

$$Y_t = H_t X_t + V_t, \quad t = 1, 2, \ldots, \tag{13.70}$$

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

$$\hat{X}_{t|t} = \mathbf{E}(X_t | Y_{1:t}) \tag{13.71}$$

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 $P_{t|t}$,

$$\hat{X}_{t|t-1} = F_t \hat{X}_{t-1|t-1}, \tag{13.72}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t, \tag{13.73}$$

$$S_t = H_t P_{t|t-1} H_t^T + R_t, \tag{13.74}$$

$$K_t = P_{t|t-1} H_t^T S_t^{-1}, \tag{13.75}$$

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t (Y_t - H_t \hat{X}_{t|t-1}), \tag{13.76}$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1}, \tag{13.77}$$

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

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix}, \quad F_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{13.78}$$

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., 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.

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.

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.

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.

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.

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.

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.

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.

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.

Onsager, L. (1944) Crystal statistics, i. a two-dimensional model with an order-disorder transformation. *Physical Review*, **65**, 117–149.

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.

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].

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.

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.