1

A Tutorial on Image Analysis

Merrilee A. Hurn
Oddvar K. Husby
Håvard Rue

1.1 Introduction

1.1.1 Aims of image analysis

Data arise in the form of images in many different areas and using many different technologies. Within medical diagnostics, X-rays are probably the most well-known form of direct imaging, gathering structural information about the body by recording the transmission of X-rays. More recent advances have been the various emission-based technologies, PET and SPECT, which aim to map metabolic activity in the body, and MRI (Figure 1.1c) which again provides structural information. On two quite different scales, satellites (Figure 1.1a and b) and microscopes (Figure 1.1d) monitor and record useful scientific information; for example, aerial imaging in different wavebands and at different stages in the growing season can be used to detect crop subsidy fraud, while some types microscopy can be used to generate temporal sequences of three dimensional images, leading to a greater understanding of biological processes. There is an expectation that technological advances should soon provide solutions to problems such as automatic face or hand recognition, or unsupervised robotic vision.

A wide range of image processing tasks arise, not merely because of the range of different applications, but also because of the diversity of goals. In the medical context, some forms of imaging, such as PET, involve the use of radioactive material, and so the exposure should ideally be as small as possible. However this is a compromise with maintaining good image quality, and so the processing question may be one of making the visual appearance of the image as clear as possible. In other applications, it may not be the picture quality which is so much at issue as the extraction of quantitative information. We may be faced with questions ranging from "Can you make this picture less blurred" to "Can you describe the packing structure of any cells in this medium". Generally tasks of the former type, trying to improve the image quality, require local actions (deconvolution, or noise removal for example), and are known as *low level tasks*. Tasks which address more global properties of the scene, such as locating or identifying

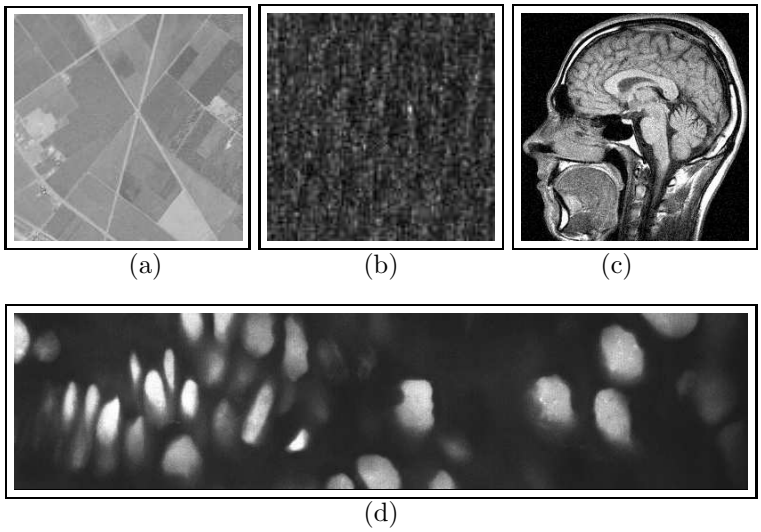


FIGURE 1.1. Some examples of image data; (a) A satellite image over an agricultural area, (b) a satellite image of ocean waves, (c) an MRI image of the brain, and (d) a confocal microscopy image of cells.

objects, are referred to as *high-level tasks*.

One unifying theme for the different types of imaging problems is that they may generally be regarded as being of the forms

$$\text{Signal} = \text{Image} \otimes \text{Noise}, \tag{1.1}$$

or

$$\text{Signal} = f(\text{Image}) \otimes \text{Noise}, \tag{1.2}$$

where $\otimes$ represents a suitable combination operator, and the function f indicates that the signal is not of the same format as the original image (for example, in emission tomography, the signals are emission counts along lines through the body). Notice that we are being rather vague about what the image actually is; this is a point to which we will return when we discuss modelling for different tasks.

1.1.2 Bayesian approach

What role can Bayesian statistics play in image analysis and processing? This is a particularly pertinent question given that there are many quick, often deterministic, algorithms developed in the computer science and electrical engineering fields. As we see it, there are three main contributions: The first is to model the noise structure adequately; the second is to regularise underdetermined systems through the use of a prior distribution; and the final, possibly most important role, is to be able to provide confidence statements about the output results. Increasingly, with the advance

in technology, the goals of the imaging are more complex than before and the questions being posed are more quantitative than qualitative. This is particularly an issue when studies also have a temporal aspect where it is important to be able to separate real change from noise.

In these notes, we will denote the underlying image by $\boldsymbol{x}$, although for the moment we postpone saying exactly what $\boldsymbol{x}$ is, and the corresponding signal by $\boldsymbol{y}$. Often, but not always, $\boldsymbol{y}$ will be a lattice of discrete values; in which case, the individual components of $\boldsymbol{y}$ are known as pixels (short for picture element). Bayes theorem allows us to write

$$\pi(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{\pi(\boldsymbol{y} \mid \boldsymbol{x})\pi(\boldsymbol{x})}{\pi(\boldsymbol{y})} \propto \pi(\boldsymbol{y} \mid \boldsymbol{x})\pi(\boldsymbol{x}) \tag{1.3}$$

where the likelihood, $\pi(\boldsymbol{y}|\boldsymbol{x})$, describes the data formation process for a particular underlying image, while the prior, $\pi(\boldsymbol{x})$, encodes any prior beliefs about the properties of such underlying images. The marginal for the data $\pi(\boldsymbol{y})$ is uninformative regarding $\boldsymbol{x}$. We will of course be interested in drawing inferences about $\boldsymbol{x}$ based on the posterior $\pi(\boldsymbol{x}|\boldsymbol{y})$. To do this we will need to specify various components of the model. First we must decide what a suitable representational form is for $\boldsymbol{x}$, and to a large extent this will depend on what the objectives of the imaging are. We must then decide upon appropriate prior and likelihood models. In the following two sections we will discuss some modelling possibilities, and also describe associated issues of the treatment of nuisance parameters and the machinery for inference.

1.1.3 Further reading

As this chapter is a tutorial and not a research paper, we have been relatively relaxed about referring to all the appropriate sources for some of the more basic material. Further, we do not claim to give a complete, or even almost complete, overview of this huge field. There are many topics which could have been discussed but which are not mentioned at all. For the reader who wishes to learn more about this field, there are several good sources. A good start might be the RSS discussion papers (Besag 1974, Besag 1986, Grenander & Miller 1994, Glasbey & Mardia 2001) (one in each decade), Geman & Geman (1984) and Geman (1990) are essential reading, as is Grenander (1993), although this might be a bit tough. The books by Winkler (1995) and Dryden & Mardia (1999) are also worth reading. Nice applications of high-level models can also be found in Blake & Isard (1998).

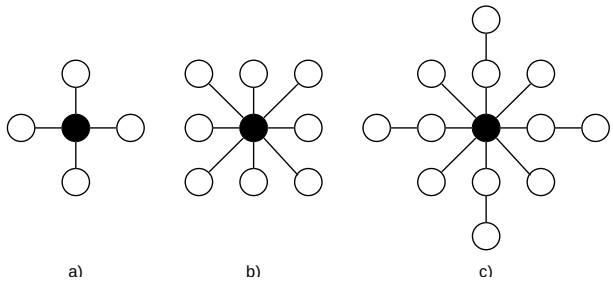


FIGURE 1.2. Common neighbourhood structures in imaging, four, eight or twelve nearest neighbours.

1.2 Markov random field models

We will begin with models where the representation for $\boldsymbol{x}$ is as that for $\boldsymbol{y}$, that is as a list of pixel values. We will need some definitions: Let $\mathcal{I}$ be the set of sites or pixels in the image which is assumed to be finite. Each $i \in \mathcal{I}$ is a coordinate in the lattice. In order to define a pixel-based model for images, we turn to the class of *Markov random fields*. We begin by defining a symmetric neighbourhood relation $\sim$ on $\mathcal{I}$, if i is a *neighbour* of j (written as $i \sim j$) then j is a neighbour of i. By convention i is not a neighbour of itself. A random field is then a collection of random variables $\{x_i : i \in \mathcal{I}\}$ where each x_i takes values in a finite set χ. Denote the neighbours of i by ∂i. For a subset of sites $A \subseteq \mathcal{I}$ we also use the notation $x_A = \{x_i : i \in A\}$ and $\boldsymbol{x}_{-A} = \{x_i : i \in \mathcal{I} \setminus A\}$. An $\boldsymbol{x}$ *configuration* is an element of $\chi^{|\mathcal{I}|}$. A random field is called a *Markov random field* (MRF) if the conditional distribution of any pixel given the rest (also called its *local characteristic*) only depends on the values of that pixel's neighbours,

$$\pi(x_i \mid \boldsymbol{x}_{-i}) = \pi(x_i \mid \boldsymbol{x}_{\partial i}). \tag{1.4}$$

Commonly used neighbourhood structures in imaging are four, eight or twelve nearest neighbours, see Figure 1.2.

MRFs are important in imaging for two main reasons:

1. Modelling the joint distribution of an image $\boldsymbol{x}$ on the lattice $\mathcal{I}$ is a daunting task because it is not immediately clear even how to begin. Approaching the issue through the *full conditionals* breaks the problem down into more manageable tasks, in the sense that we may be able to say more clearly how we think x_i behaves if we know the configuration of its neighbours.

2. There is an important connection between Markov chain Monte Carlo methods (MCMC) and MRFs, in that single-site updating schemes in MCMC only require evaluations of the local full conditionals (1.4). If we assume that the number of neighbours is very considerably

less than $n = |\mathcal{I}|$, as is the case for almost all applications, then a full sweep of the image using a Gibbs sampler, say, requires $\mathcal{O}(n)$ operations for a MRF, as opposed to $\mathcal{O}(n^2)$ operations for a random field lacking this local property.

Although we might therefore approach the modelling problem by specifying a neighbourhood and the full conditionals for each site, one very important issue arises: Given a set of local characteristics, under what conditions are we guaranteed that a legitimate joint density exists? Is this joint density unique and what is it? These are delicate questions. Suppose we have a set of full conditionals, and we wish to construct a corresponding joint density. Since the joint density sums to 1, it is enough to study $\pi(\boldsymbol{x})/\pi(\boldsymbol{x}^*)$, for some reference configuration $\boldsymbol{x}^*$. By considering cancellation of successive terms (and assuming $\pi(\cdot) > 0$), this can be written

$$\begin{aligned} \frac{\pi(\boldsymbol{x})}{\pi(\boldsymbol{x}^*)} &= \prod_{i=1}^{n} \frac{\pi(x_1^*, \ldots, x_i^*, x_{i+1}, x_{i+2}, \ldots, x_n)}{\pi(x_1^*, \ldots, x_i^*, x_{i+1}^*, x_{i+2}, \ldots, x_n)} \\ &= \prod_{i=1}^{n} \frac{\pi(x_{i+1} | x_1^*, \ldots, x_i^*, x_{i+2}, \ldots, x_n)}{\pi(x_{i+1}^* | x_1^*, \ldots, x_i^*, x_{i+2}, \ldots, x_n)}. \end{aligned} \tag{1.5}$$

Hence, we can obtain the joint density from a product of ratios of full conditionals, after renormalisation. Note that only the neighbours are needed in the above conditioning, but this we ignore for notational convenience. For a set of full conditionals to define a legitimate joint density, we must ensure that the joint density as defined in (1.5) is invariant to the ordering of the indices, and further, is invariant to the choice of the reference state $\boldsymbol{x}^*$. These are the consistency requirements on the full conditionals. Although it is possible, in theory, to verify these consistency requirements directly, we nearly always make use of the *Hammersley-Clifford theorem*. This theorem states that a set of full conditionals defines a legitimate joint density if and only if they are derived from a joint density of a particular form.

Theorem 1 (Hammersley-Clifford) *A distribution satisfying $\pi(\boldsymbol{x}) > 0$ for all configurations in $\chi^{|\mathcal{I}|}$ is a Markov random field if, and only if, it has a joint density of the form*

$$\pi(\boldsymbol{x}) = \frac{1}{Z} \exp\left(- \sum_{C \in \mathcal{C}} \Phi_C(\boldsymbol{x}_C) \right) \tag{1.6}$$

*for some functions $\{\Phi_C\}$, where $\mathcal{C}$ is the set of all*cliques *(a clique is defined to be any subset of sites where every pair of these sites are neighbours) and Z is the normalising constant*

$$Z = \sum_{\boldsymbol{x}} \exp\left(- \sum_{C \in \mathcal{C}} \Phi_C(\boldsymbol{x}_C) \right) < \infty. \tag{1.7}$$

The easier direction of the proof is to verify that a distribution of the form of (1.6) satisfies the Markov property (1.4). This follows from noting that

$$\pi(x_i \mid \boldsymbol{x}_{-i}) \propto \pi(\boldsymbol{x}) \propto \exp\left(-\sum_{C\in\mathcal{C}} \Phi_C(\boldsymbol{x}_C)\right). \tag{1.8}$$

Normalising the above expression over all possible values of x_i, notice that all terms $\Phi_C(\boldsymbol{x}_C)$ not involving x_i cancel out, and hence the result. It is far harder to prove the converse see, for example, Winkler (1995).

The theorem can be used in at least two ways. The first is to confirm that a collection of full conditionals (that is, the distribution of one component conditional on all the others) does define a legitimate joint density when we can find a distribution of the form (1.6) where the full conditionals match. Secondly, and more importantly, it says that instead of constructing full conditionals directly, we could construct them implicitly though the choice of so-called *potential functions* Φ_C.

1.3 Models for binary and categorical images

In this section we discuss the use of the*Ising model* and its extension, the *Potts model*, for *binary images* images with more than two unordered colours. We will present the material in a reasonable level of detail since many of the ideas generalise quite readily to grey-level images and even high level models as well. We will begin by describing the models themselves, and their contribution to posterior distributions for images, and then discuss how to simulate from such systems.

1.3.1 Models for binary images

Suppose the image of interest is binary, where we typically refer to each pixel x_i as foreground if $x_i = 1$ (black), or background if $x_i = 0$ (white). What are our prior beliefs about such scenes $\boldsymbol{x}$? For a start, we could think of how a pixel x_i might behave conditional on everything else in the image. Assuming a four nearest neighbours scheme, consider the situation in Figure 1.3a. Here the four nearest neighbours of x_i are black, so what probability do we wish to assign to x_i also being black? Of course, this will depend on the context and what type of images we are studying, but it seems reasonable that this probability should increase with an increasing number of neighbouring black pixels. One possibility would be to use

$$\pi(x_i \text{ is black} \mid k \text{ black neighbours}) \propto \exp(\beta k) \tag{1.9}$$

where β is a positive parameter. The normalising constant here is simply $\exp(\beta$ number of white neighbours$) + \exp(\beta$ number of black neighbours$)$.

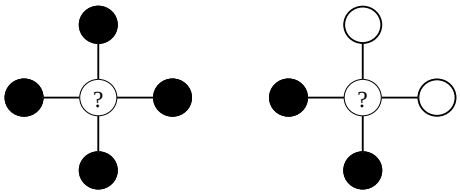


FIGURE 1.3. The pixel x_i marked as "?"and two different configurations for its four nearest neighbours

This choice implies that there is no reason to treat the background and the foreground differently, since the probabilities for $x_i = 1$ and $x_i = 0$ in Figure 1.3b are both 1/2. Also if we swap white pixels for black, and vice versa, the swapped configuration has the same probability as the original one.

We now have to appeal to the Hammersley-Clifford theorem. Can we find potential functions defined on the cliques such that the local characteristics are those we have suggested? Recall that the cliques are sets of sites such that each pair in the set are neighbours. In this case, using four nearest neighbourhoods, the cliques are the sets of nearest horizontal or nearest vertical pairs. If we try

$$\Phi_C(\boldsymbol{x}_C) = \begin{cases} -\beta, & \text{both sites in the clique } C \text{ have the same colour} \\ 0, & \text{else} \end{cases} \tag{1.10}$$

then we obtain a joint density

$$\begin{aligned} \pi(\boldsymbol{x}) &= \exp\left(-\sum_{C\in\mathcal{C}} \Phi_C(\boldsymbol{x}_C)\right) / \; Z(\beta) \\ &= \exp\left(\beta \sum_{i\sim j} I_{[x_i=x_j]}\right) / \; Z(\beta) \end{aligned} \tag{1.11}$$

where $i \sim j$ denotes neighbouring pairs. The normalising constant

$$Z(\beta) = \sum_{\boldsymbol{x}} \exp\left(\beta \sum_{i\sim j} I_{[x_i=x_j]}\right) \tag{1.12}$$

is a function of β. It is easy to verify that this has the same full conditionals as (1.9), and so our joint density is (1.11).

Obviously we did not make our full conditional suggestion at random! The joint density in (1.11) is the famous Ising model, named after E. Ising who presented the model in 1925. It has been used as a model for ferromagnetism where each site represents either an up spin or down spin. See Grimmett (1987) for a review.

One remarkable feature about the Ising model, is the existence of *phase transition* behaviour. Assume for the moment an $n \times n$ lattice, where values outside the boundary are given, and let i^* denote the interior site closest to the centre. Will the values at the boundary affect the marginal distribution of x_{i^*} as $n \to \infty$? Intuitively one might expect that the effect of what happens at the boundary will be negligible as the lattice grows, but this is wrong. It can be shown that for

$$\beta > \beta_{\text{critical}} = \log(1 + \sqrt{2}) = 0.881373\ldots \tag{1.13}$$

the effect of the boundary does matter (below this value, it does not). In consequence, long-range interaction occurs over the critical value, while only short-range interaction occurs under the critical value. We will see what this implies clearly when we simulate from the Ising model. The existence of phase transition adds weight to the assertion that it is very hard to interpret the global properties of the model by only considering the full conditionals.

1.3.2 Models for categorical images

One often encounters the situation where there are more than two colours in the image, for example in a microscopy context these might correspond to the background, cells of type 1 and cells of type 2 . One of the simplest models for such categorical settings is the Potts model, which is the multicolour generalisation of the Ising model. Suppose now that $x_i \in \{0, 1, \ldots, n_c - 1\}$, where the number of colours n_c is a positive integer greater than two, then define

$$\pi(\boldsymbol{x} \mid \beta) \propto \exp\left(\beta \sum_{i \sim j} I_{[x_i = x_j]}\right). \tag{1.14}$$

Although this expression is similar to that for the Ising model, the configurations have changed from binary to multicolour. We see from (1.14) that each of the n_c colours has the same full conditional distribution for x_i as the Ising model if we merge all neighbour sites into the two classes "same colour as x_i" and its converse. This is not unreasonable as the colours are not ordered; colour 1 is not necessary closer to colour 3 than colour 0.

Further generalisations of the Ising/Potts model

Generalisations of the Ising and Potts models can also include other neighbourhood system than the four nearest neighbours; for example the nearest eight or twelve neighbours could be used. However, if we decide to use a larger neighbourhood system, we might also want to experiment with the potentials to promote a certain behaviour. Another natural generalisation

FIGURE 1.4. A section of a page of newsprint, displayed in reverse video.

is to allow more general β's, for example

$$\pi(\boldsymbol{x} \mid \{\beta_{\ldots}\}) \propto \exp\left(\sum_{i\sim j} \beta_{ijx_i} I_{[x_i=x_j]}\right). \tag{1.15}$$

where the strength of the interaction might depend on how far in distance site i is from site j and the colour they take. However, there is always a question as to whether it is more fruitful to consider cliques of higher orders instead of continuing this theme of pairwise interactions only. See, for example, Tjelmeland & Besag (1998), where some interesting experiments along these lines are conducted.

1.3.3 Noisy images and the posterior distribution

Generally we are not able to record images exactly, observing data $\boldsymbol{y}$ rather than $\boldsymbol{x}$. Using Bayes theorem however, we know how to construct the posterior distribution of $\boldsymbol{x}|\boldsymbol{y}$, and via this we can learn about the underlying images. We will now consider two noisy versions of binary images, illustrating these two using Figure 1.4 as the true underlying image. This image is a small section of a page of newsprint; character recognition is an important imaging task. We will make the assumption that y_i and y_j $(j \neq i)$ are conditional independent given $\boldsymbol{x}$, so that

$$\pi(\boldsymbol{y} \mid \boldsymbol{x}) = \pi(y_1 \mid x_1) \cdots \pi(y_n \mid x_n). \tag{1.16}$$

Gaussian additive noise

Suppose that the true image is degraded by additive *Gaussian noise*,

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon} \tag{1.17}$$

where $\boldsymbol{\epsilon}$ is Gaussian with zero mean, zero covariance and variance σ^2, and $\boldsymbol{\epsilon}$ and $\boldsymbol{x}$ are independent. Then

$$\pi(y_i \mid x_i) \propto \frac{1}{\sigma} \exp\left(-\frac{1}{2}(y_i - x_i)^2/\sigma^2\right). \tag{1.18}$$

Note that the observed image $\boldsymbol{y}$ is recorded on a continuous scale (which may later by discretised) even though $\boldsymbol{x}$ is binary. Gaussian additive noise is quite commonly assumed as it may mimic additive noise from several sources. The posterior distribution for the true scene $\boldsymbol{x}$, is

$$\pi(\boldsymbol{x} \mid \boldsymbol{y}) \propto \exp\left(\beta \sum_{i\sim j} I_{[x_i=x_j]} - \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i)^2/\sigma^2\right) \tag{1.19}$$

$$\propto \exp\left(\beta \sum_{i\sim j} I_{[x_i=x_j]} + \sum_i h_i(x_i, y_i)\right) \tag{1.20}$$

where $h_i(x_i, y_i) = -\frac{1}{2\sigma^2}(y_i - x_i)^2$. Note that additive constants not depending on x_i can be removed from h_i as they will cancel in the normalising constant. The form of the posterior as given in (1.20) is generic, covering all types of conditionally independent noise by defining the h_i's as

$$h_i(x_i, y_i) = \log(\pi(y_i \mid x_i)). \tag{1.21}$$

Flip noise

Flip, or binary, noise occurs when the binary images are recorded with a Bernoulli probability model that the wrong colour is recorded. We assume that the error probability p is constant and that each pixel value is recorded independently

$$\pi(y_i \mid x_i) = \begin{cases} 1-p & \text{if } y_i = x_i \\ p & \text{if } y_i \neq x_i \end{cases} \tag{1.22}$$

Note that $p = 1/2$ corresponds to the case where there is no information about $\boldsymbol{x}$ in $\boldsymbol{y}$. The posterior distribution can still be represented in the same way as (1.20), but with

$$h_i(x_i, y_i) = I_{[y_i=x_i]} \log(\frac{1-p}{p}), \tag{1.23}$$

where $I_{[]}$ is the indicator function.

Figure 1.5 shows some examples of degraded images using additive Gaussian noise (first column) and flip noise (second column). In the Gaussian case, we have rescaled the images for display purposes, using a linear grey scale from black to white so that 0 and 1 correspond to the minimum and maximum observed values of $\{y_i\}$. Our task might be to estimate or restore the images from such data; we will use images Figure 1.5e and Figure 1.5f in later examples.

1.3.4 Simulation from the Ising model

Models such as the Ising are sufficiently complex, despite their apparently simple structure, to require *Markov chain Monte Carlo methods*, see Chap-

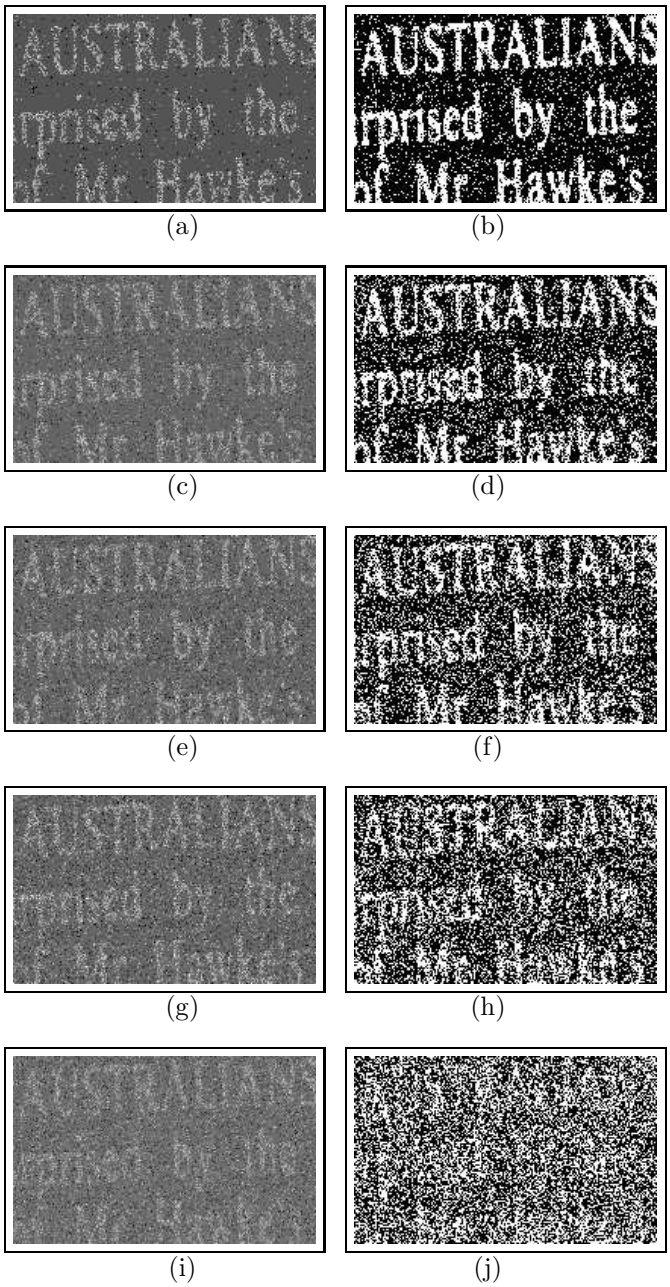


FIGURE 1.5. Newsprint degraded by additive Gaussian noise: (a) $\sigma^2 = 0.1$; (c) $\sigma^2 = 0.4$; (e) $\sigma^2 = 0.5$; (g) $\sigma^2 = 0.6$; (i) $\sigma^2 = 0.8$; or flip-noise:(b) $p = 0.1$; (d) $p = 0.2$; (f) $p = 0.25$; (h) $p = 0.3$; (j) $p = 0.4$.

ter 1. However, as already mentioned, the local structure of Markov random fields lends itself well to the component-wise structure of single site updating versions of the family of algorithms. Recall that the *Gibbs sampler* sequentially updates each site from its full conditional. This is particularly easy as the full conditional for the Ising model is

$$\pi(x_i \mid \boldsymbol{x}_{-i}) \propto \exp(\beta \sum_{j \sim i} I_{[x_i = x_j]}) \tag{1.24}$$

To simplify the notation, let $n_i^b = \sum_{j \sim i} I_{[x_j = 1]}$ be the number of black neighbours and $n_i^w = \sum_{j \sim i} I_{[x_j = 0]}$ be the number of white neighbours of i, then

$$\pi(x_i = 1 \mid \boldsymbol{x}_{-i}) = \frac{\exp(\beta n_i^b)}{\exp(\beta n_i^b) + \exp(\beta n_i^w)}. \tag{1.25}$$

The implementation of the Gibbs-sampler for the Ising model is shown in Algorithm 1. The function "NextSite" returns which site to update on the

Algorithm 1 Gibbs sampler for the Ising model, n_{iter} iterations

```
Set x = 0 or 1, or fill x with randomly chosen 0's and 1's.
for t = 1 to n_iter do
  for j = 1 to n do
    i =NextSite(j)
    U ~ Uniform(0, 1)
    p = exp(β n_i^b)/(exp(β n_i^b) + exp(β n_i^w)).
    if U < p then
      x_i = 1
    else
      x_i = 0.
    end if
  end for
end for
```

j'th iteration. The simplest choice is to use "NextSite(j) = return j" that is, updating the sites in numerical order. However, this need not be the case although it is convenient from a coding point of view. Two other *updating schemes* are also commonly used:

Random With this scheme, we chose which pixel to update at random from the whole image. A benefit here is that no potential "directional effects" occur. Although there is a small chance that some pixels will be updated only a few times, this is not a serious problem provided we run our sampler for sufficiently long (the expected time to visit all the pixels is $\mathcal{O}(n \log(n))$).

Permutation With this scheme, we again chose which pixels to update at random but now with the constraint that all other pixels are updated

before updating the same one again, in effect a random permutation. One implementation of this approach is to have initially a list of all sites so that each element contain a pixel index. At each call when the list is not empty, pick one element at random, return the contents and delete the element from the list. If the list is empty, then start a complete new list. There are also other approaches.

Obviously, the Gibbs proposals could be replaced equally well by a more general *Hastings proposals*, with the obvious modifications. One obvious choice in this binary situation is to propose to change pixel i to the other colour from its current value. If all the neighbours of i are black, the Gibbs sampler will favour x_i being black, but this will at the same time prevent the sampler from moving around, say to the symmetric configuration where all colours are reversed (and which has the same probability).

Algorithm 2 A Metropolis sampler for the Ising model

Set $\boldsymbol{x} = \mathbf{0}$ or $\mathbf{1}$, or fill $\boldsymbol{x}$ with randomly chosen 0's and 1's.
for $t = 1$ to n_{iter} **do**
  for $j = 1$ to n **do**
    $i =$NextSite(j)
    $x_i' = 1 - x_i$
    $d = \exp(\beta \sum_{j \sim i} I_{[x_i = x_j]})$
    $d' = \exp(\beta \sum_{j \sim i} I_{[x_i' = x_j]})$
    $p = \min\{1, d'/d\}$
    $U \sim$ Uniform$(0, 1)$
    if $U < p$ **then**
      $x_i = x_i'$
    end if
  end for
end for

In general, it is numerically unstable to compute the acceptance rate as in Algorithm 2. The problem arises when taking the ratio of *unnormalised* conditional densities. (Sooner or later this will give you severe problems or seemingly strange things happens with your MCMC-program, if you are not careful at this point!) A numerically more stable approach deals with the log densities for as long as possible, that is:

$d = \beta \sum_{j \sim i} I_{[x_i = x_j]}$
$d' = \beta \sum_{j \sim i} I_{[x_i' = x_j]}$
$p = \exp(\min\{0, d' - d\})$

Mixing issues

Both Algorithm 1 and 2 use single site updating, as is common in MCMC algorithms, and both perform poorly for simulating from the Ising model

for β higher than or close to the critical value. Assume the current configuration is mostly black due to a high value of β. We know that the configuration formed by flipping the colours has the same probability. Using a single site sampler, we have to change the colour of all the sites individually. This involves going though a region of very low probability states, which is of course quite unlikely (although it will happen eventually), so the convergence of single site algorithms can be painfully slow for large values of β. Alternative algorithms do exist for the Ising model, most notably the *Swendsen-Wang algorithm* (Swendsen & Wang 1987) which has good convergence properties even at β_{critical}. We now describe this algorithm specifically for simulating from the Ising model, but note that more general forms exist (Besag & Green 1993); see also Section 1.3.2.

A new, high dimensional variable $\boldsymbol{u}$ is introduced, with one component of $\boldsymbol{u}$ for each interaction $i \sim j$. These u_{ij} could be thought of as bond variables. The joint distribution of $\boldsymbol{x}, \boldsymbol{u}$ is constructed by defining the conditional distribution of $\boldsymbol{u}$ given $\boldsymbol{x}$, $\pi(\boldsymbol{u}|\boldsymbol{x})$. A Markov chain is then constructed alternating between transitions on $\boldsymbol{u}$ (by drawing from $\pi(\boldsymbol{u}|\boldsymbol{x})$) and transitions on $\boldsymbol{x}$. To ensure that this two step procedure retains $\pi(\boldsymbol{x})$ as its stationary distribution, the transition function $P(\boldsymbol{x} \to \boldsymbol{x}'|\boldsymbol{u})$ is chosen to satisfy detailed balance with respect to the conditional $\pi(\boldsymbol{x}|\boldsymbol{u})$; the simplest choice for this is $P(\boldsymbol{x} \to \boldsymbol{x}'|\boldsymbol{u}) = \pi(\boldsymbol{x}'|\boldsymbol{u})$.

So, given a realisation $\boldsymbol{x}$, define the u_{ij} to be conditionally independent with

$$u_{ij} \mid \boldsymbol{x} \sim \text{Uniform}(0, \exp(\beta I_{[x_i = x_j]})). \tag{1.26}$$

That is, given $\boldsymbol{x}$, the auxiliary variable u_{ij} is uniformly distributed either on $[0, \exp(\beta)]$, if $x_i = x_j$, or on $[0, 1]$ otherwise, both of which are clearly easy to simulate. Notice that the larger β is, the more likely it is that the u_{ij} generated for a neighbouring pair $x_i = x_j$ is greater than 1.

Then, via the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{u}$

$$\pi(\boldsymbol{x} \mid \boldsymbol{u}) \propto \prod_{ij} I_{[\exp(\beta I_{[x_i = x_j]}) \geq u_{ij}]}, \tag{1.27}$$

i.e. a random colouring of the pixels, subject to the series of constraints. Notice that whenever $u_{ij} \leq 1$, the constraint $\exp(\beta I_{[x_i = x_j]}) \geq u_{ij}$ is satisfied whatever the values of x_i and x_j. Conversely, if $u_{ij} > 1$, then for the constraint to be satisfied, x_i and x_j must be equal. Groups of $\boldsymbol{x}$ sites connected by some path of interactions for which each $u_{ij} > \exp(\beta I_{[x_i = x_j]})$ are known as clusters, and this definition segments $\boldsymbol{x}$ into disjoint clusters. Clusters are conditionally independent given $\boldsymbol{u}$, and can be updated separately, each to a single random colour. Notice that the larger β is, the larger the clusters are likely to be, and thus large changes can be made to $\boldsymbol{x}$ in precisely the cases where the usual algorithms struggle. It is worth remarking that, unfortunately, generalisations of the Swendsen-Wang algorithm have as yet generally lacked its spectacular success.

Some examples

We will now present some realisations from the Ising model using the Swendsen-Wang algorithm. The image is 200×125 (the same size used in examples later on). We have no boundary conditions, meaning that sites along the border have three neighbours, while the four corner sites have two neighbours. The examples are for a range of β values ranging from $\beta = 0.3$ in Figure 1.6a, to $\beta = 1.3$ in Figure 1.6h. Note how dramatically the image changes around β_{critical}.

It is straightforward to extend these sampling ideas to the Potts model. Figure 1.7 shows some realisations from the model on a 100×100 lattice with $n_c = 4$ for various β. We see the close resemblance to realisations from the Ising model, although the samples are rather more "patchy" as there are more colours present.

1.3.5 Simulation from the posterior

The posterior summarises knowledge of the true image based on our prior knowledge and the observed data. Hence, if we can provide samples from the posterior, they can be used to make inference about the true scene. We can now easily modify Algorithm 2 to account for observed data. The only change is to add the contribution from the likelihood. The convergence for this simple MCMC algorithm is, in most cases, quite good; the effect of long interactions from the prior when β is large, is reduced by the presence of the observations and so phase transition does not occur for the posterior (in most cases).

Algorithm 3 A Metropolis sampler for the noisy Ising model

Initialise $\boldsymbol{x}$
Read data $\boldsymbol{y}$ and noise-parameters
for $t = 1$ to n_{iter} **do**
  for $j = 1$ to n **do**
    $i =$NextSite(j)
    $x_i' = 1 - x_i$
    $d = \beta \sum_{j \sim i} I_{[x_i = x_j]} + h_i(x_i, y_i)$
    $d' = \beta \sum_{j \sim i} I_{[x_i' = x_j]} + h_i(x_i', y_i)$
    $U \sim$ Uniform$(0, 1)$
    $p = \exp(\min(d' - d, 0))$
    if $U < p$ **then**
      $x_i = x_i'$
    end if
  end for
end for

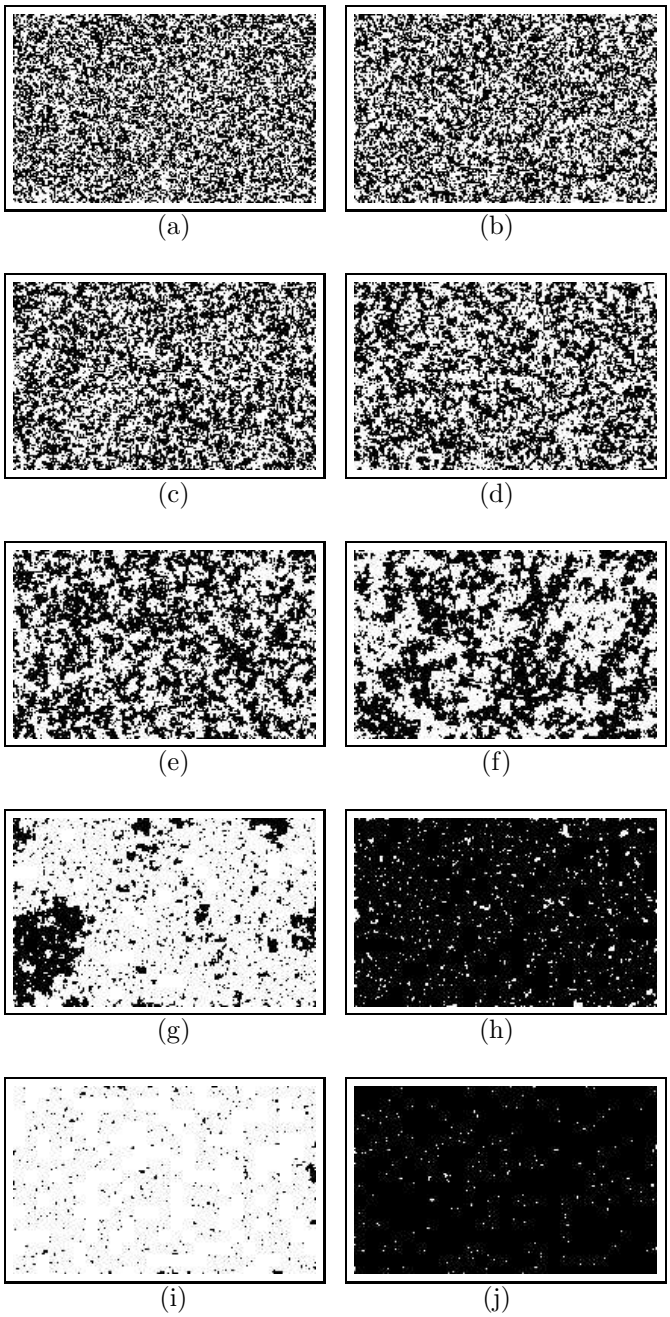


FIGURE 1.6. Simulation from the Ising model: (a) $\beta = 0.3$; (b) $\beta = 0.4$; (c) $\beta = 0.5$; (d) $\beta = 0.6$; (e) $\beta = 0.7$; (f) $\beta = 0.8$; (g) $\beta = 0.9$; (h) $\beta = 1.0$; (i) $\beta = 1.1$; (j) $\beta = 1.3$.

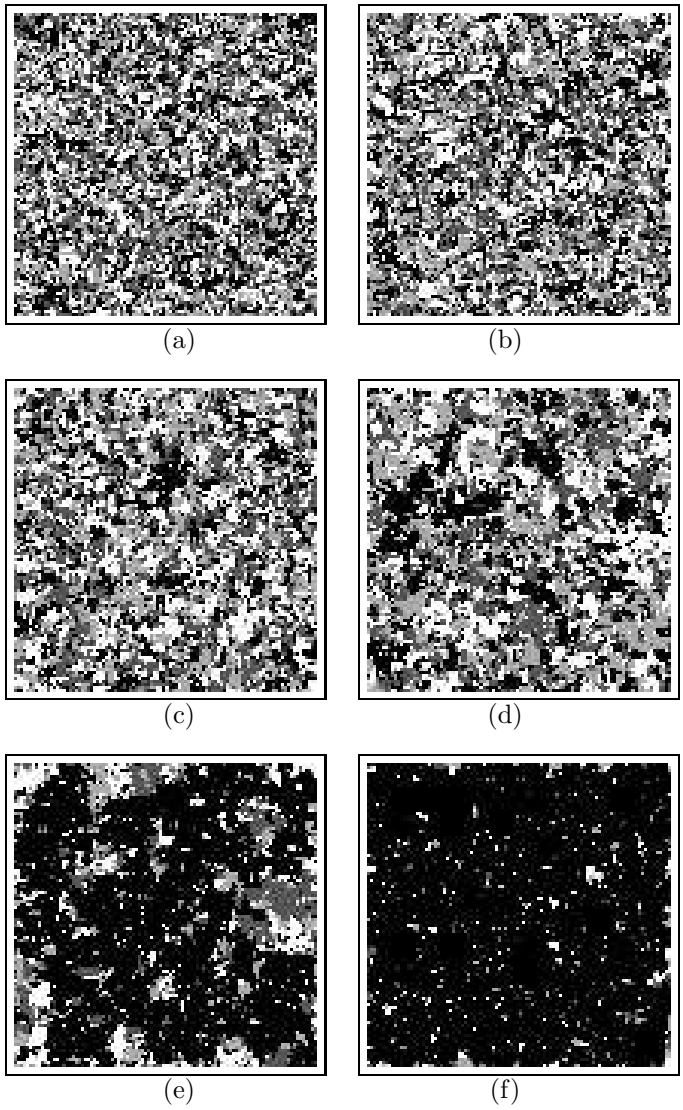


FIGURE 1.7. Realisations from the Potts-model with four colours on a 100×100 lattice: (a) $\beta = 0.7$; (b) $\beta = 0.8$; (c) $\beta = 0.9$; (d) $\beta = 1.0$; (e) $\beta = 1.1$; (f) $\beta = 1.2$.

1.4 Image estimators and the treatment of parameters

In this section, we consider issues of inference for image functionals, for images themselves and for the associated model parameters.

1.4.1 Image functionals

In some cases, it is not the image as a picture which is of primary concern, but rather some quantitative information carried by the image, for example the typical size of a cell, or the proportion of land used for growing a particular crop. It is usually possible to express such attributes as a function of $\boldsymbol{x}$, say $g(\boldsymbol{x})$. Finding posterior summaries of $g(\boldsymbol{x})$ is then recognisable as a fairly common problem in Bayesian statistics, with the most commonly used estimator being the posterior mean $\mathrm{E}_{\boldsymbol{x}|\boldsymbol{y}} g(\boldsymbol{x})$. Generally, of course, this expectation will be analytically intractable, and a run of MCMC must be used to approximate it by the usual ergodic average. This does at least have the advantage that other posterior summaries can be extracted in the course of the computation, for example credible intervals. It is perhaps this ability to get a handle on the uncertainty in imaging problems which may justify the use of computationally expensive Bayesian methods as opposed to the many other cheaper algorithms developed in the computer vision world.

1.4.2 Image estimation

For many types of images, particularly binary images, the use of posterior means as a summary does not provide sensible solutions; for that reason, we now consider the range of possibilities available. In the Bayesian framework, estimation is based upon the specification of appropriate *loss functions* (negative utility), for which we then derive the corresponding optimal estimators. Of course the posterior mean is an example of this, corresponding to the squared loss function. One interpretation of the loss function, in this context, is to consider it as a measure of distance between a true image $\boldsymbol{x}$ and an estimate $\boldsymbol{z}$. We are trying to capture the notion of how wrong we are if we mistakenly use $\boldsymbol{z}$ rather than the correct $\boldsymbol{x}$. Suppose we can find such a measure of distance, $L(\boldsymbol{x}, \boldsymbol{z})$, which defines numerically how close the two images are. It is more important that this L provides a reasonable measure of distance, rather than strictly satisfying the axioms of a metric (namely, $L(\boldsymbol{x}, \boldsymbol{z}) = L(\boldsymbol{z}, \boldsymbol{x}) \geq 0$ with equality iff $\boldsymbol{z} = \boldsymbol{x}$, and that $L(\boldsymbol{x}, \boldsymbol{z}) \leq L(\boldsymbol{x}, \boldsymbol{u}) + L(\boldsymbol{u}, \boldsymbol{z})$). Suppose we were to evaluate how close an estimate $\boldsymbol{z}$ is to the image $\boldsymbol{x}$ by using $L(\boldsymbol{x}, \boldsymbol{z})$. Different estimates could then be compared, and as an estimate $\boldsymbol{z}'$ is better than $\boldsymbol{z}''$ if $L(\boldsymbol{x}, \boldsymbol{z}') < L(\boldsymbol{x}, \boldsymbol{z}'')$. Although this is feasible when we are using a

known test image $\boldsymbol{x}$, the basic concept still applies when $\boldsymbol{x}$ is unknown and we have available its posterior distribution. We can define the posterior expected distance between any estimate $\boldsymbol{z}$ and the unknown $\boldsymbol{x}$, as

$$\mathrm{E}_{\boldsymbol{x}|\boldsymbol{y}} L(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{x}} \pi(\boldsymbol{x} \mid \boldsymbol{y}) L(\boldsymbol{x}, \boldsymbol{z}). \tag{1.28}$$

We define the optimal Bayes estimate as the configuration minimising (1.28),

$$\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{z}} \mathrm{E}_{\boldsymbol{x}|\boldsymbol{y}} L(\boldsymbol{x}, \boldsymbol{z}). \tag{1.29}$$

Although the general setup is straightforward, the practical implementation of these ideas is not trivial for two main reasons:

1. How should we construct a distance measure which conforms to our visual perception of how close two binary images are. Suppose there are four discrepant pixels, then the location of these pixels really matters! For example, if they are clumped, they may be misinterpreted as a feature. Ideally, a distance measure should take into account the spatial distribution of the errors as well as their number.

2. Assume we have a distance measure, how can we obtain the optimal Bayes estimate in practise, i.e. solve (1.29)?

To get some feeling as to how certain loss functions behave, consider the error vector $\boldsymbol{e}$, where $e_i = I_{[x_i \neq z_i]}$. We can expand any loss function based on a difference between images using a binary expansion

$$\begin{aligned} L(\boldsymbol{e}) &= a_0 - a_1 \sum_i (1 - e_i) - a_2 \sum_{i<j} (1 - e_i)(1 - e_j) \\ &- a_3 \sum_{i<j<k} (1 - e_i)(1 - e_j)(1 - e_k) \\ &- \ldots - a_n (1 - e_1)(1 - e_2) \cdots (1 - e_{n-1})(1 - e_n) \end{aligned} \tag{1.30}$$

where for simplicity we have assumed that the constants $(a_0, \ldots, a_n)$ depend only on the number of sites considered in each term, and the a_0 term is usually selected such that $L(\mathbf{0}) = 0$. We see that each summand is either 1 if there are no errors in the terms concerned, or 0 if there is at least one error. Two common image estimators are *Marginal Posterior Modes* (MPM) and *Maximum A Posteriori* (MAP) which correspond to a loss function which counts the number of pixel misclassifications and to a loss function which is zero if there is no error and is one otherwise, respectively:

$$L_{\mathrm{MPM}}(\boldsymbol{e}) = \sum_{i=1}^{n} e_i, \tag{1.31}$$

and

$$L_{\mathrm{MAP}}(\boldsymbol{e}) = 1 - \prod_{i=1}^{n}(1 - e_i). \tag{1.32}$$

Note that these choices corresponds to two extremes, namely that the non zero $\boldsymbol{a}$ values for MPM are $a_0 = n$, $a_1 = 1$, and for MAP $a_0 = 1$ and $a_n = 1$.

The MPM estimator

The loss function in (1.31) simply counts the number of errors, so the corresponding optimal Bayes estimate will be optimal in the sense of minimising the number of misclassifications. There is no extra penalty if the errors are clustered together as opposed to being scattered around, and in this sense the estimator is quite local. To obtain the estimate, we first compute the expected loss

$$\begin{aligned} \mathrm{E}_{\boldsymbol{x}|\boldsymbol{y}} \sum_i e_i &= \sum_i \mathrm{E}_{x_i|\boldsymbol{y}} e_i = \sum_i \mathrm{Prob}(x_i \neq z_i \mid \boldsymbol{y}), \\ &= \text{constant} - \sum_i \mathrm{Prob}(x_i = z_i \mid \boldsymbol{y}), \end{aligned} \tag{1.33}$$

hence by definition,

$$\begin{aligned} \boldsymbol{x}_{\mathrm{MPM}} &= \arg\min_{\boldsymbol{z}} \left\{ -\sum_i \mathrm{Prob}(x_i = z_i \mid \boldsymbol{y}) \right) \\ &= \sum_i \arg\max_{z_i} \mathrm{Prob}(x_i = z_i \mid \boldsymbol{y}). \end{aligned} \tag{1.34}$$

So the ith component of $\boldsymbol{x}_{\mathrm{MPM}}$ is the modal value of the posterior marginal. In our case, it is simply

$$x_{\mathrm{MPM},i} = \begin{cases} 1, & \text{if } \mathrm{Prob}(x_i = 1 \mid \boldsymbol{y}) > 1/2 \\ 0, & \text{if } \mathrm{Prob}(x_i = 1 \mid \boldsymbol{y}) \leq 1/2. \end{cases} \tag{1.35}$$

To compute an estimate of $\boldsymbol{x}_{\mathrm{MPM}}$, we can use N samples from the posterior, and if the number of times x_i is equal to 1 is greater or equal to $N/2$, then $\widehat{x}_{\mathrm{MPM},i} = 1$, else it is 0.

The MAP estimator

The zero-one loss function (1.32) giving rise to the MAP estimate is quite extreme; any image not matching $\boldsymbol{x}$ is as wrong as any other, independent of how many errors there are. As is well known for the zero-one loss function, the estimator is $\boldsymbol{x}_{\mathrm{MAP}}$, which yields

$$\boldsymbol{x}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{z}} \pi(\boldsymbol{z} \mid \boldsymbol{y}), \tag{1.36}$$

ie. the posterior mode. The mode may be an obvious candidate for an estimator if the posterior is unimodal or has one dominating mode which also contains essentially all of the probability mass. However, this is not always the case. Much of the probability mass can be located quite far away from the mode, which makes this estimator questionable. One indication of this happening is when the mode looks quite different from "typical" realisations from the posterior.

Obviously, one way to try to find the MAP estimate would be to sample from the posterior, always keeping note of the state seen so far with the highest posterior probability. However, this is likely to be a highly inefficient strategy, and instead we next consider two alternatives, one stochastic and the other deterministic.

Algorithms for finding the MAP estimate

One algorithm which is known to converge to the MAP estimate, at least in theory, is *simulated annealing*, cf. Section 1.5.4. The basic idea is as follows: Suppose $\pi(\boldsymbol{x})$ is the distribution of interest and let $\boldsymbol{x}^*$ be the unknown mode. It would clearly be a slow and inefficient approach to search for $\boldsymbol{x}^*$ simply by sampling from $\pi(\boldsymbol{x})$, but the search could be made more efficient if we instead sample from $\pi_T(\boldsymbol{x}) \propto \pi(\boldsymbol{x})^{1/T}$ for small values of T, known as the temperature, $0 < T \ll 1$. Note that $\pi_T(\boldsymbol{x})$ has the same mode for all $0 < T < \infty$, and as $T \to 0$ will have most of its probability mass on this mode. The fact that we do not know the normalising constant as a function of temperature will not be important as we will use MCMC. So if we were to chose $T = 0^+$ and construct a MCMC algorithm to sample from $\pi_T(\boldsymbol{x})$, a sample would most likely be the mode and the problem is apparently solved! The catch is of course that the smaller T gets, the harder it is for an MCMC algorithm to mix and produce samples from $\pi_T(\boldsymbol{x})$, rather than getting trapped at a local mode of the posterior. So, the following trick is used: At iteration t, the target distribution is $\pi(\boldsymbol{x})^{1/T(t)}$, where $T(t)$ is the temperature which varies with time (hence, we have a non-homogeneous Markov chain). The temperature schedule is decreasing in such a way that $T(t) \leq T(t')$ if $t \geq t'$, and $T(t) \to 0$ as $t \to \infty$. If we decrease the temperature slowly enough, then hopefully the MCMC algorithm will reach the global mode. Theoretical analysis of the algorithm clarifies what is required of the speed of the temperature schedule. We have to lower the temperature not faster than

$$T(t) = C/\log(t+1) \tag{1.37}$$

where C is a constant depending on $\pi(\boldsymbol{x})$. Hence, the time it takes to reach $T = \epsilon$, is at least

$$t = \exp(C/\epsilon) - 1 \tag{1.38}$$

which quickly tends to infinity as ϵ tends to zero. In other words, the required schedule is not implementable in practise. Stander & Silverman

(1994) give some recommendations for schedules which perform well in finite time.

From a computational point of view, simulated annealing is easy to implement if one already has an MCMC algorithm to sample from $\pi(\boldsymbol{x})$. Note that if $\log \pi(\boldsymbol{x}) = -U(\boldsymbol{x}) + \text{constant}$, then

$$\pi(\boldsymbol{x})^{1/T(t)} \propto \exp\left(-\frac{1}{T(t)} U(\boldsymbol{x})\right), \tag{1.39}$$

so the effect of the temperature is simply a scaling of $U(\boldsymbol{x})$. In Algorithm 4 we have implemented simulated annealing using Algorithm 3. The user has to provide the "temperature" function, which returns the temperature as a function of the time. We chose n_{iter} to be finite and lower the temperature faster than (1.37), for example as

$$T(t) = T_0 \times \eta^{t-1} \tag{1.40}$$

where $T_0 = 4$ and $\eta = 0.999$. T_0, η and n_{iter} should be chosen to reach a predefined low temperature in n_{iter} iterations. Note that we may also keep track of which configuration that has highest probability of those visited so far, and return that configuration as the final output.

Algorithm 4 The Simulated Annealing algorithm for the noisy Ising model

```
Initialise x, set T = T_0.
Read data y and noise parameters
for t = 1 to n_iter do
  for j = 1 to n do
    i =NextSite(j)
    x'_i = 1 − x_i
    d = β Σ_{j∼i} I_{[x_i=x_j]} + h_i(x_i, y_i)
    d' = β Σ_{j∼i} I_{[x'_i=x_j]} + h_i(x'_i, y_i)
    U ∼ Uniform(0, 1)
    p = exp(min(d' − d, 0)/T)
    if U < p then
      x_i = x'_i
    end if
  end for
  T = Temperature(t)
end for
return x
```

Besag (1986) introduced the method of *Iterated conditional modes* (ICM) as a computationally cheaper alternative to simulated annealing. (There are also arguments for considering ICM as an estimator on its own.) It is equivalent to using simulated annealing at temperature zero taking the Gibbs sampler as the MCMC component. At each iteration, the most likely value

for each pixel is chosen in turn, conditional on the current values of all the others. By considering the expression $\pi(\boldsymbol{x}|\boldsymbol{y}) = \pi(x_i|\boldsymbol{x}_{-i}, \boldsymbol{y})\pi(\boldsymbol{x}_{-i}|\boldsymbol{y})$, it is clear that each iteration increases the posterior probability until the algorithm reaches a mode, most likely a local mode. The algorithm converges fast in general, but can be very sensitive to the choice of starting point.

Finally, for the Ising model, an algorithm for locating the mode exactly exists (Greig, Porteous & Scheult 1989). The algorithm is rather technical, and not extendable beyond the Ising model, and so we do not present it here.

Examples

We will now show some estimated MPM and MAP estimates based on Figure 1.5e and f, Gaussian noise with $\sigma^2 = 0.5$ and flip noise with $p = 0.25$. We assume the noise parameter to be known, so our only *nuisance parameter* is β. It is not that common to estimate β together with $\boldsymbol{x}$, so usually the inference for $\boldsymbol{x}$ is based on

$$\pi(\boldsymbol{x} \mid \boldsymbol{y}, \beta) \tag{1.41}$$

for a fixed value of β. This is then repeated for a range of β values, and the value producing the "best" estimate is then selected. This process clearly underestimates the uncertainty regarding β, and usually also the uncertainty in other derived statistics from the posterior distribution. We will later demonstrate how this could be avoided by taking the uncertainty in β into account.

Figure 1.8 shows the case with Gaussian noise with MAP estimates in the left column and the MPM estimates in the right column. Similarly with Figure 1.9, but for the flip noise case. The effect of increasing β is clear in both sets of figures. Increasing β makes the estimate smoother. In this instance, there is not much difference between the MAP and MPM estimates.

1.4.3 Inference for nuisance parameters

We will concentrate on a binary classification model with noisy Gaussian observations of typical response levels associated with the two states (for example, our newsprint images degraded by noise). Hence, we now treat the level of background (μ_0) and foreground (μ_1) as unknown instead of being known as 0 and 1. Additionally, the noise variance σ^2 is unknown as well. The posterior then is

$$\pi(\boldsymbol{x} \mid \boldsymbol{y}) \propto \frac{(\sigma^2)^{-n/2}}{Z(\beta)} \exp\left(\beta \sum_{i\sim j} I_{[x_i=x_j]} - \frac{1}{2\sigma^2}\sum_i (y_i - \mu_{x_i})^2\right). \tag{1.42}$$

FIGURE 1.8. MAP (left column) and MPM (right column) estimates for various values of β when the true scene is degraded by Gaussian noise with variance 0.5. First row: $\beta = 0.3$, second row: $\beta = 0.5$, third row: $\beta = 0.7$, fourth row: $\beta = 0.9$, fifth row: $\beta = 1.3$.

FIGURE 1.9. MAP (left column) and MPM (right column) estimates for various values of β when the true scene is degraded by flip noise with $p = 0.25$. First row: $\beta = 0.3$, second row: $\beta = 0.5$, third row: $\beta = 0.7$, fourth row: $\beta = 0.9$, fifth row: $\beta = 1.3$.

The posterior of interest is leaving us four nuisance parameters to deal with, $\beta, \sigma^2, \mu_0, \mu_1$. We will consider likelihood approaches, with and without training data, as well as a fully Bayesian description.

Likelihood approaches with training data

Suppose we are in the fortunate position where we have a data set $\boldsymbol{y}$ generated from a known configuration $\boldsymbol{x}$. In this situation, we could use *likelihood approaches*:

$$(\hat{\sigma}^2, \hat{\mu}_0, \hat{\mu}_1) = \arg\max \pi(\boldsymbol{y}|\boldsymbol{x}; \sigma^2, \mu_0, \mu_1) \tag{1.43}$$

$$\hat{\beta} = \arg\max \pi(\boldsymbol{x}; \beta). \tag{1.44}$$

For the former, it is straightforward to show that

$$\hat{\mu}_j = \frac{1}{|i : x_i = j|} \sum_{i:\boldsymbol{x}_i=j} y_i, \qquad j = 0, 1 \tag{1.45}$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_i (y_i - \hat{\mu}_{x_i})^2. \tag{1.46}$$

However for $\hat{\beta}$, we have significant difficulties because the normalising constant $Z(\beta)$ for the prior distribution is not tractable. In addition, we are in effect attempting to estimate one parameter based on a single "data" point, $\boldsymbol{x}$, because although we may have many pixels, and thus considerable information about typical response levels and variance levels, we only have this single realisation from the process controlled by β. Working with high-dimensional problems with complex interaction parameters, this sort of problem arises quite frequently. We describe one possible alternative to true maximum likelihood estimation which has been suggested in this context, *maximum pseudo-likelihood*: Consider factorising the joint distribution $\pi(\boldsymbol{x}|\beta)$

$$\pi(\boldsymbol{x} \mid \beta) = \pi(x_1 \mid x_2, \ldots, x_n, \beta)\pi(x_2 \mid x_3, \ldots, x_n, \beta) \ldots \pi(x_n \mid \beta) \tag{1.47}$$

Despite the Markov structure, the terms on the right-hand side above become increasingly difficult to compute. The idea behind pseudo-likelihood is to replace each of the terms $\pi(x_i|x_{i+1}, \ldots, x_n, \beta)$ by the complete conditional $\pi(x_i|\boldsymbol{x}_{-i}, \beta)$ which by the Markov property is $\pi(x_i|\boldsymbol{x}_{\partial i}, \beta)$. So

$$PSL(\beta) = \prod_{i=1}^{n} \pi(x_i|\boldsymbol{x}_{\partial i}, \beta). \tag{1.48}$$

It should be noted that this is not a likelihood except in the case of no dependence, that is when $\pi(\boldsymbol{x}_i|\boldsymbol{x}_{\partial i}) = \pi(\boldsymbol{x}_i)$. The benefit of this approach

is that these conditional probabilities are particularly easy to express for the Ising model:

$$\pi(x_i \mid \boldsymbol{x}_{\partial i}, \beta) = \left(1 + \exp(-\beta \sum_{j \in \partial i} (I_{[x_i = x_j]} - I_{[x_i \neq x_j]}))\right)^{-1}. \tag{1.49}$$

This function is easily calculated for a given $\boldsymbol{x}$, and so we can write down the pseudo-likelihood function. Maximisation of the function over β will have to be numerical. Obviously the higher the true value of β, the worse the estimates, since we ignore the true dependence structure.

Likelihood approaches without training data

It is more typically the case that we do not have training data either for the likelihood or the prior parameters; it is then hard to implement the approaches described in the previous section exactly. One commonly used approach is to alternate iterations of whichever updating scheme is being used for $\boldsymbol{x}$ using the current parameter estimates, with steps which update the current parameter estimates treating the current $\boldsymbol{x}$ as if it were a known ground-truth. See, for example, Besag (1986).

Fully Bayesian approach

The obvious alternative to an approach which fixes the parameters at some estimated value, is to treat them as *hyperparameters* in a *fully Bayesian approach*. That is we treat $\sigma^2, \{\mu_i\}, \beta$ as variables, specify prior distributions for them and work with the full posterior

$$\begin{aligned} \pi(\boldsymbol{x}, \sigma^2, \{\mu_i\}, \beta \mid \boldsymbol{y}) &\propto \pi(\boldsymbol{y} \mid \boldsymbol{x}, \sigma^2, \{\mu_i\}, \beta)\pi(\boldsymbol{x}, \sigma^2, \{\mu_i\}, \beta) \\ &\propto \pi(\boldsymbol{y} \mid \boldsymbol{x}, \sigma^2, \{\mu_i\})\pi(\boldsymbol{x} \mid \beta)\pi(\sigma^2)\pi(\mu_0)\pi(\mu_1)\pi(\beta) \end{aligned} \tag{1.50}$$

where we assume independent priors for each parameter. If we have no additional prior information about any of the parameters, then common choices of priors are the following (for a data set where the recorded values are in the set $\{0, \ldots, 2^8 - 1\}$, the priors for the means have been restricted to $[0, 255]$), where $1/\sigma^2$ has been reparameterised as τ

$$\begin{aligned} \tau &\sim \exp(1) \\ \mu_0 &\sim \text{Uniform}(0, 255) \\ \mu_1 &\sim \text{Uniform}(0, 255) \\ \beta &\sim \text{Uniform}(0, \beta_{max}) \text{ where say } \beta_{max} = 2. \end{aligned} \tag{1.51}$$

For the purposes of sampling, we will most likely need to know the conditional distributions of each of the parameters:

$$\begin{aligned}
\pi(\tau \mid \cdots) &\propto (\tau)^{n/2} \exp\left(-\tau(1 + 1/2 \sum_i (y_i - \mu_{x_i})^2)\right) \\
\pi(\mu_i \mid \cdots) &\propto \exp\left(-\tau/2 \sum_{j:x_j=i} (y_j - \mu_i)^2\right) I_{[0<\mu_i<255]}, \quad i = 0, 1 \\
\pi(\beta \mid \cdots) &\propto \frac{1}{Z(\beta)} \exp\left(\beta \sum_{i\sim j} I_{[x_i=x_j]}\right) I_{[0<\beta<\beta_{max}]}. \qquad (1.52)
\end{aligned}$$

The first of these conditionals is recognisable as a gamma distribution with parameters $(n/2 + 1)$ and $(1 + 1/2 \sum_i (y_i - \mu_{x_i})^2)^{-1}$, and as such would lend itself well to the Gibbs sampler. Notice that the conditional mean is quite closely related to the maximum likelihood estimate of τ. The conditionals for the two mean level parameters are recognisable as truncated Gaussians; again notice that the conditional mean is closely related to the maximum likelihood estimate. In this case, a Metropolis algorithm could be used, perhaps taking the untruncated Gaussian as the proposal density. Only the conditional for β is not recognisable as being related to a standard distribution. This suggests using some form of Metropolis-Hastings algorithm for sampling. However, once again we run into difficulties because of the intractability of the normalising constant $Z(\beta)$; in order to implement a Metropolis-Hastings algorithm here, we would need to be able to evaluate values of Z at different values of β. We now describe one approach to this problem.

Estimating $Z(\beta)/Z(\beta')$

Recall that $Z(\beta)$ is defined by

$$Z(\beta) = \sum_{\boldsymbol{x}} \exp(\beta S(\boldsymbol{x})) \qquad (1.53)$$

where $S(\boldsymbol{x})$ is the sufficient statistic $\sum_{i\sim j} I_{[x_i=x_j]}$. One possibility is to see whether the derivative of $Z(\beta)$ with respect to β is easier to estimate than $Z(\beta)$ itself.

$$\begin{aligned}
\frac{dZ(\beta)}{d\beta} &= \sum_{\boldsymbol{x}} S(\boldsymbol{x}) \exp(\beta S(\boldsymbol{x})) \\
&= Z(\beta) \sum_{\boldsymbol{x}} (S(\boldsymbol{x})) \exp(\beta S(\boldsymbol{x})) / Z(\beta) \\
&= Z(\beta)\, \mathrm{E}_{\boldsymbol{x}|\beta} S(\boldsymbol{x}). \qquad (1.54)
\end{aligned}$$

By solving this differential equation, we obtain that

$$\log\left(Z(\beta')/Z(\beta)\right) = \int_{\beta}^{\beta'} \mathrm{E}_{\boldsymbol{x}|\tilde{\beta}} S(\boldsymbol{x})\, d\tilde{\beta}, \tag{1.55}$$

see also Sections 1.5.4, 1.7.2 and 4.7.4. As we see, this trick has reduced the difficulty of the problem to one we can tackle using the following procedure:

1. Estimate
$$\mathrm{E}_{\boldsymbol{x}|\beta} S(\boldsymbol{x}) \tag{1.56}$$
for a range of various β values using posterior mean estimates based on the output from a sequence of MCMC algorithms. (These values will depend on the image size and so will need to be recalculated for each new problem, although a proper rescaling to account for (not too different) dimensions will do fine.)

2. Construct a smoothing spline $f(\beta)$ to smooth the estimated values of $\mathrm{E}_{\boldsymbol{x}|\beta} S(\boldsymbol{x})$.

3. Use numerical or analytical integration of $f(\beta)$ to compute an estimate of (1.55),
$$\log\left(\widehat{Z(\beta')/Z(\beta)}\right) = \int_{\beta}^{\beta'} f(\tilde{\beta})\, d\tilde{\beta} \tag{1.57}$$
for each pair (β, β') required.

Example

Let us now apply the fully Bayesian approach to the same examples as in Section 1.4.2. Our first task is to estimate (1.56) to compute $f(\beta)$ for evaluating the normalising constant: We ran Algorithm 2 using $3,000$ iterations after burn-in, for values of β from 0 to 1.5 in steps of 0.01. We then estimated the smooth curve $f(\beta)$ using a local polynomial smoother to reduce the noise and to compute interpolated values on a fine grid, as shown in Figure 1.10. The computation of (1.57) is then trivial.

Recall that Algorithm 2, which was use to estimate Figure 1.10, has severe mixing problems for high βs due to the invariance when flipping colours. How does this effect the estimated curve in Figure 1.10? In fact very little, as the contribution to the normalising constant from each of the two modes is the same. We might expect small errors at and around the critical β, from the contribution from all the "intermediate" configurations. These may not be properly explored by Algorithm 2. However, a careful re-estimation of Figure 1.10 using the Swendsen-Wang algorithm, which does not suffer from such mixing problems, gave an indistinguishable estimate even at and around $\beta_{\mathrm{critical}}$.

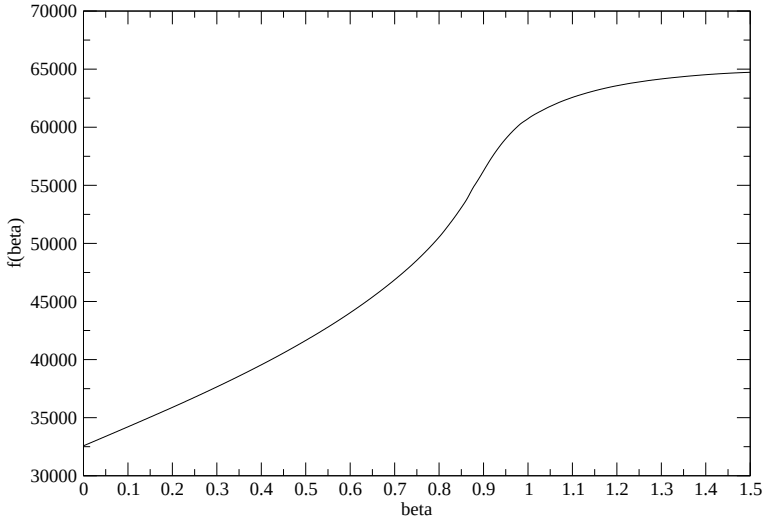


FIGURE 1.10. Estimated $f(\beta)$ for the Ising model using Algorithm 2.

We now need to extend Algorithm 3 to include a move changing β. We adopt a simple Metropolis strategy, and propose a new value β' by

$$\beta' = \beta + \text{Uniform}(-h, h) \tag{1.58}$$

where h is a small constant, in the order of $h = 0.025$ or so. The corresponding acceptance probability becomes

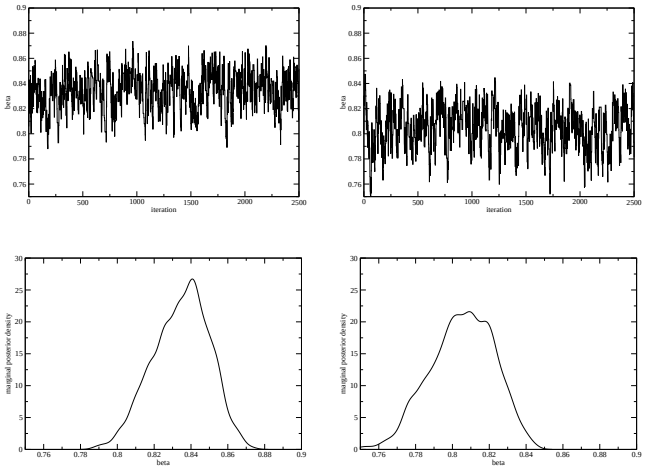
$$\begin{aligned} \alpha(\beta, \beta') &= \min\left\{1,\ \frac{\exp(\beta' S(\boldsymbol{x}) + \sum_i h_i(x_i, y_i))}{\exp(\beta S(\boldsymbol{x}) + \sum_i h_i(x_i, y_i))} \times \frac{Z(\beta)}{Z(\beta')} \times \frac{\pi(\beta')}{\pi(\beta)}\right\} \\ &= \min\left\{1,\ \exp\left[S(\boldsymbol{x})(\beta' - \beta) - \log\left(\frac{Z(\beta')}{Z(\beta)}\right)\right] \frac{\pi(\beta')}{\pi(\beta)}\right\} \end{aligned} \tag{1.59}$$

where $\pi(\beta)$ is the prior for β. Note that as $\beta' - \beta \to d\beta$, the exponential term in (1.59) reduces to

$$\exp\left(d\beta(S(\boldsymbol{x}) - \mathrm{E}_{\boldsymbol{x}|\beta} S(\boldsymbol{x}))\right), \tag{1.60}$$

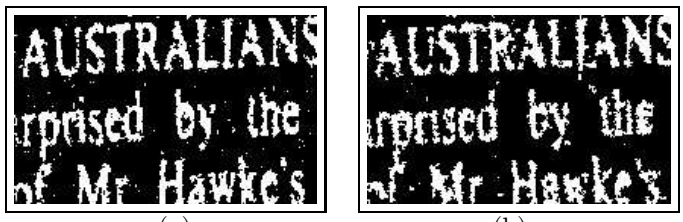
and so we see that β will tend to vary around the maximum likelihood estimate, $S(\boldsymbol{x}) = \mathrm{E}_{\boldsymbol{x}|\beta} S(\boldsymbol{x})$, for β given $\boldsymbol{x}$, but with the added variability coming from $\boldsymbol{x}$ itself being unknown.

The extended sample is then run for 2 500 iterations after burn-in for the same two noisy data sets as in Figures 1.8 and 1.9. As is clear from (1.60), the value of β is determined only by $\boldsymbol{x}$ and not by the data themselves, so by looking at Figure 1.9 and Figure 1.8, we see that the "effective" noise level is slightly higher for the flip noise case, and the estimates of the true scene are slightly more noisy. Hence, it is expected that β in the flip



(c) (d)

FIGURE 1.11. Figure (a) shows 2 500 values of β from the MCMC algorithm 3 and (c) the density estimate, for Gaussian noise with $\sigma^2 = 0.5$. Figures (b) and (d) are similar, but for flip noise with $p = 0.25$.



(a) (b)

FIGURE 1.12. Posterior marginal mode estimates of the true scene with the fully Bayesian approach: (a) Gaussian noise with $\sigma^2 = 0.5$; (b) flip noise with $p = 0.25$.

noise case is slightly lower than in the Gaussian case. Figure 1.11 shows the trace plot of β and its density estimate using a kernel method, for the Gaussian case on the left, the flip noise case on the right. In both cases β varies around $0.8 - 0.85$, which is just below the critical value 0.88. The uncertainty in β might seem high for such a large data set, but in effect we have only one relevant piece of data (the whole image). Figure 1.12 shows the corresponding MPM estimates, which seem reasonable.

1.5 Grey-level images

In this section, we remain with pixel-based image models, but widen the range of application. We turn our attention to images which are *piecewise smooth*, consisting of smooth regions separated by edges (where we will be

rather more precise about what we mean by smooth later on). Such images might arise in the context of any type of imaging device which records on an intensity scale and where, unlike for categorical image data, the order of these values is of interest per se.

1.5.1 Prior models

Whereas models for binary and categorical data tend to penalise any discrepancy between neighbouring pixel values, here it will be more appropriate to penalise large discrepancies more heavily than small ones. We will consider MRF prior distributions of the form $\pi(\boldsymbol{x}) \propto \exp(-\Phi(\boldsymbol{x}))$, where

$$\Phi(\boldsymbol{x}) = \beta \sum_{C \in \mathcal{C}} w_C \phi \left(D_C(\boldsymbol{x}) \right), \tag{1.61}$$

ϕ is a real function, the w_Cs are positive weights, β is a positive scaling parameter, and the functions $D_C(\boldsymbol{x})$ are discrete approximations to some order of $\boldsymbol{x}$ derivative at clique C. This choice of order of derivative is one way in which to control the order of smoothness desired; penalising differences in first order derivative will favour constant regions, penalising second order derivatives will favour planar regions, and so on. The D_C's are taken to be discrete difference operators corresponding to approximations of first- and second-order derivatives, so for example an approximation to a first order derivative in the grey-level is simply the difference between values at neighbouring pixels. An approximation to the second order derivative is the difference in differences, and so on. The weights w_Cs are used to accommodate the differences in distance between diagonal and vertical or horizontal sites, assuming a neighbourhood structure larger than simply the four nearest neighbours is used.

Using the above formulation, we can write the unnormalised negative log prior as

$$\Phi(\boldsymbol{x}) = \beta \sum_{m=1}^{M} w_m \sum_i \phi \left(D_i^{(m)} \boldsymbol{x} \right), \tag{1.62}$$

where $D_i^{(m)} \boldsymbol{x}$ is the m-th discrete derivative of $\boldsymbol{x}$ at position i, e.g. $D_i^{(1)} \boldsymbol{x} = (x_{i+1} - x_i)/\delta$, where δ is a scaling parameter. The choice of the potential function ϕ has implications for the properties of the system. It is natural to assume ϕ to be symmetric, so that positive and negative gradients of equal magnitude are penalised equally, but what other properties do we desire? One particular and important possibility for ϕ is the quadratic potential $\phi(u) = u^2$ which leads to a Gaussian prior, and, combined with a Gaussian likelihood, an unimodal posterior distribution. This of course simplifies sampling; Gaussian models can be efficiently sampled using sparse matrix methods (Rue 2001), or fast Fourier transforms in the special case of homogeneous likelihood and toroidal boundary conditions (Wood & Chan 1994).

However, this choice of ϕ is not suited for estimation of piecewise constant or planar fields because the rapid growth as $u \to \infty$ severely penalises the intensity jumps which may occur across edges. In addition, the slow variation around the origin might cause excessive smoothing and interpolation. So, ideally we would also like to be able to detect changes between smooth regions. The choice of potential functions for edge-preserving restoration is widely discussed in the literature. We will here consider such implicit edge-preserving models; alternatives are to model the edges explicitly using discrete line processes (Geman & Geman 1984) or to use deformable templates which act on region descriptions directly (Grenander & Miller 1994).

We will largely follow Geman & Yang (1995) and consider potential functions in the class of continuous (but not necessary derivative) functions

$$\begin{aligned} \mathcal{E} \quad = \quad & \Big\{ \phi(\cdot) \in \mathrm{C}^{(0)}(\mathbb{R}) \mid \phi(0) = 0, \phi(u) = \phi(-u), \\ & \lim_{u \to \infty} \phi(u) < \infty, \frac{d\phi}{du} \geq 0, u \in \mathbb{R}^+ \Big\}. \end{aligned} \tag{1.63}$$

The arguments in favour of this class of models are largely based on heuristics, but it is clear that the finite limit and slow growth for large u ensure that intensity jumps over edges are not too severely penalised. The following example, taken from Blake & Zisserman (1987), shows that using potential functions in $\mathcal{E}$ has links to both *line processes* and robust inference.

Example 1 *Let u be a Markov random field with neighbourhood relation $\sim$, and define the line process*

$$l_{ij} = \begin{cases} 1 & \exists \text{ edge between } u_i \text{ and } u_j \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, define the negative log prior

$$\Phi(u, l) = \sum_{i \sim j} \left((u_i - u_j)^2 - 1 \right) (1 - l_{ij})$$

smoothing within the disjoint regions defined by the line process l. Then Blake & Zisserman (1987) observed that

$$\begin{aligned} \inf_l \Phi(u, l) &= \sum_{i \sim j} \left((u_i - u_j)^2 - 1 \right) I_{[(u_i - u_j)^2 < 1]} \\ &= \sum_{i \sim j} \left((u_i - u_j)^2 - 1 \right)^- = \Phi^*(u), \end{aligned}$$

where $x^- = \min(x, 0)$. Thus $\inf_u \inf_l \Phi(u, l) = \inf_u \Phi^*(u)$ and, in terms of modal behaviour, there is no need to model the edges explicitly, and thus instead use the truncated quadratic.

In addition to the behaviour as u grows large, the behaviour around the origin is important. Charbonnier, Blanc-Feraud, Aubert & Barlaud (1997) advocate strictly concave functions, such as

$$\phi(u) = \frac{-1}{1+|u|}, \qquad \phi(u) = \frac{|u|}{1+|u|}, \tag{1.64}$$

basing their argument on consideration of *coordinate-wise minima* (that is, a change in the value of any single pixel, results in an increase in the negative log prior). Let $\boldsymbol{x}^*$ be a coordinate-wise minimum and consider a small perturbation $x_i^* + tu$ toward the data y_i. This will lead to a decrease of order tu in the likelihood component, but an increase in the prior energy since $\phi'(0+) > 0$. By appropriately choosing the scaling parameter β the combined effect will very likely be an increase of the posterior distribution. This is in contrast to the case where $\phi'(0) = 0$, where there will be interpolation. (As an aside, note that it is generally difficult to say as much about the choice of ϕ when we consider estimators such as the posterior mean, since we then need to consider distributional properties rather than just the mode). Some potentials which have been used in the literature are given below.

Potential function	Reference
$\min(1, u^2)$	Blake & Zisserman (1987)
$u^2/(1+u^2)$	Geman & McClure (1987)
$\log\cosh u$	Green (1990)
$\log(1+u^2)$	Hebert & Leahy (1989)
$2\sqrt{1+u^2}-2$	Charbonnier (1994)

TABLE 1.1. Some *edge preserving potentials*

1.5.2 Likelihood models

Generally speaking, *continuous likelihood models* for grey-level images and binary images are not so different. However, at this stage we will introduce one further aspect to the image degradation model, which is *blurring*. Blurring occurs when for some reason, which might be motion or perhaps a defect in the imaging device, spatial resolution is lost so that the value recorded at site i is actually a convolution of $\boldsymbol{x}$ values in a region around i. Denote this convolution by $\boldsymbol{z}$, then

$$z_i = (\boldsymbol{h} * \boldsymbol{x})_i = \sum_j h_j x_{i-j} \tag{1.65}$$

where the kernel $\boldsymbol{h}$ is called a *point spread function* (psf). The components of $\boldsymbol{h}$ often sum to 1, and in most cases, h_j has its largest value at $j = 0$. If

FIGURE 1.13. Examples of two types of blurring; (left) motion blur, (right) out-of-focus blur.

we have an additive noise structure, then the data $\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{\epsilon}$. It is possible to show that the posterior remain a Markov random field with an enlarged neighbourhood structure.

As an example of the visual effect of blurring, Figure 1.13 simulates the effects either of constant relative motion of object and recording device or of a picture taken out of focus. The corresponding ideal point spread functions are a line of uniform weights for the motion blur, and for the out-of-focus case, radially symmetric weights defined over a pixel approximation to a circular disc.

1.5.3 Example

We end with a simulated example performing restoration of *confocal microscopy* images of human melanoma cancer cells. One such image is shown in Figure 1.14a. The true image $\boldsymbol{x}$ is degraded by blurring with a Gaussian kernel $\boldsymbol{h}$ with standard deviation of 3 pixels, and adding independent zero mean Gaussian noise with $\sigma = 15$. The resulting image $\boldsymbol{y}$ is shown in Figure 1.14b. The true image is divided into piecewise smooth regions, so it makes sense to use the edge preserving prior (1.62) for recovering the edges in the image. Obviously it should be possible to use single site MCMC algorithms here. However, because of the non-convexity of the prior and the long-range spatial interactions introduced by the point spread function $\boldsymbol{h}$, such standard samplers will converge very slowly for this model. Experience shows that updating all or parts of the variables jointly in blocks will lead to improved mixing, but for the present model, *block sampling* is only possible after reformulating the model using an idea in Geman & Yang (1995), which we present here. Introduce M auxiliary arrays $\boldsymbol{b} = (\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(M)})$,

and define a distribution π^* with distribution

$$\pi^*(\boldsymbol{x}, \boldsymbol{b}) \propto \exp\left(-\beta \sum_{m=1}^{M} w_m \sum_{s \in S} \left(\frac{1}{2}\left(D_s^{(m)}\boldsymbol{x} - b_s^{(m)}\right)^2 + \psi(b_s^{(m)})\right)\right), \tag{1.66}$$

where the function $\psi(\cdot)$ is related to ϕ by the identity

$$\phi(u) = -\log \int \exp(-1/2(u-v)^2 - \psi(v))\, dv. \tag{1.67}$$

Then it is easy to show that

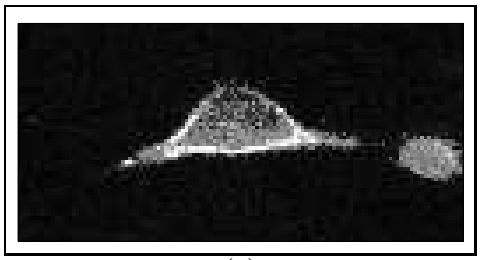
$$\pi(\boldsymbol{x}) = \int \pi^*(\boldsymbol{x}, \boldsymbol{b})\, d\boldsymbol{b}, \tag{1.68}$$

which means that we can use π^* to estimate the posterior mean of $\boldsymbol{x}$ under π. The motivation for this is that under the so called *dual model* π^*, $\boldsymbol{x}$ is Gaussian conditional on the data $\boldsymbol{y}$ and the auxiliary array $\boldsymbol{b}$. Let $\boldsymbol{D}^{(m)}$ be matrices representing the difference operators $\{D_s^{(m)}\}$, and let $\boldsymbol{D}^T = (\boldsymbol{D}^{(1)T}, \ldots, \boldsymbol{D}^{(M)T})$. Furthermore, define $\boldsymbol{W} = \mathrm{diag}(\omega_1, \ldots, \omega_M) \otimes \boldsymbol{I}_n$, and let $\boldsymbol{H}$ be a matrix representing the point spread function $\boldsymbol{h}$. Then the full conditional distribution for the true image $\boldsymbol{x}$ has distribution

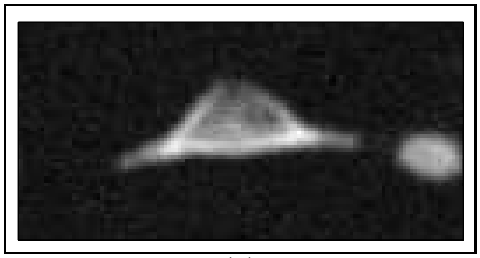
$$\pi^*(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{b}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T\left(\beta \boldsymbol{D}^T\boldsymbol{W}\boldsymbol{D} + \frac{1}{\sigma^2}\boldsymbol{H}^T\boldsymbol{H}\right)\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{W}\boldsymbol{D}\boldsymbol{x}\right), \tag{1.69}$$

which is a *Gaussian Markov random field* with inverse covariance matrix $\boldsymbol{Q} = \beta \boldsymbol{D}^T\boldsymbol{W}\boldsymbol{D} + \sigma^{-2}\boldsymbol{H}^T\boldsymbol{H}$ and mean vector $\boldsymbol{\mu}^T = \boldsymbol{b}^T\boldsymbol{W}\boldsymbol{D}\,\boldsymbol{Q}^{-1}$. Assuming toroidal boundary conditions $\boldsymbol{x}$ can be sampled very efficiently using *fast Fourier transforms* as detailed in Geman & Yang (1995). In the general situation one can use *Cholesky decompositions* and *sparse matrix methods* as in Rue (2001). The components of the auxiliary array $\boldsymbol{b}$ are conditionally independent given $\boldsymbol{x}$, and can be sampled using e.g. the Metropolis-Hastings algorithm or rejection sampling.

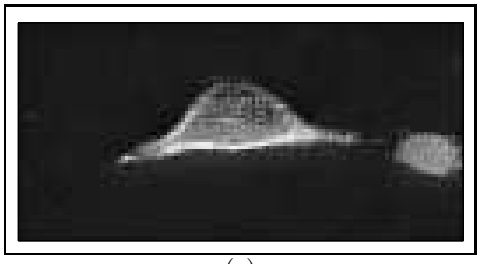
Figure 1.14c shows a posterior mean estimate of the cell image based on 1000 iterations of the block sampler with parameters $\beta = 200$, $\delta = 50$, and $\psi(u) = |u|/(1 + |u|)$. Visually the restoration is quite close to the original image, although some smoothing has taken place. The results were similar for a wide range of parameters, with smaller values of the ratio β/δ^2 leading to smoother images. Since the truth is known, a qualitative comparison between the restoration and the true image can be made. Figure 1.14d plots the *double integral distance* (Friel & Molchanov 1998) between the Markov chain samples and the true image Figure 1.14a. A constant image was used as an initial point, but the sampler seems to reach a stable state very quickly. Experiments indicate that the convergence is much faster than for the single site sampler, so the dual model formulation combined with a good block sampling algorithm seems well suited for recovering discontinuities in smooth images.



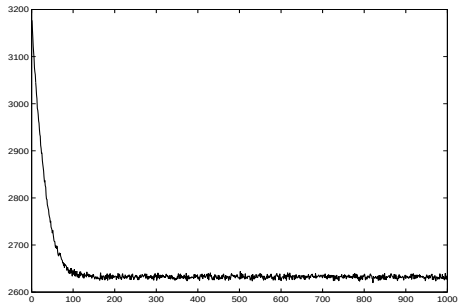
(a)



(b)



(c)



(d)

FIGURE 1.14. (a) Confocal microscopy image of a human melanoma cancer cell; (b) data simulated by blurring and adding Gaussian noise; (c) restoration of the image using the Geman & Yang model and the GMRF algorithm of Rue (2001); (d) trace plot of the double integral distance between the true image and the Markov chain samples.

1.6 High-level imaging

We now turn our attention to a completely different class of image representations and models, motivated by some of the tasks for which Markov random field representations may be inadequate. For example, suppose we are trying to identify and measure an unknown number of cells viewed under a microscope. There may be variations in the size and shape of cells, and there may even be more than one type of cell. Typical tasks might be to label the identified cell and perhaps to identify their shape characteristics, or simply to count them. A pixel-based description of the underlying $\boldsymbol{x}$ may not be the most effective one in this context for a number of reasons. Firstly, using a pixel grid, we will be "building" cells out of square building blocks. Second, if we are then trying to identify cells and we see a conglomeration of "cell" pixels, how do we know whether this is one cells or several, i.e. what constitutes a cell in terms of pixels? Third, how do we incorporate any prior information which we have about size or shape information? For problems of this sort, we may have to turn to high-level modelling.

There are various approaches to high-level modelling, mainly based on looking at the range of variation of some prototypical version of the object(s) under study (prototypical in terms of shape and/or other information such as grey-level). We will concentrate on a subset of approaches, those based on *deformations* of *polygonal templates* for objects.

1.6.1 Polygonal models

Considering our cell example above, one possible way to describe the microscope data (or at least that which is relevant to our purposes) would be to use a representation

$$\boldsymbol{x} = \{\text{Cell}_1, \text{Cell}_2, \ldots, \text{Cell}_k\} \tag{1.70}$$

where k itself may be unknown, and each Cell_i is a collection of information sufficient to locate and label that particular cell. One way to achieve this is to use a *marked point process* (see also Chapter 4) as a prior with the model for a random configuration of objects built around a stochastic template model for a single object embedded in a *mixture model* of different object types which is used as the *mark distribution* of a marked point process model. We begin, in this section, by describing the polygonal model for the outline of an object. We will describe the embedding into a point process to handle an unknown number of objects later.

The prototypical object is described by an n–sided template defined by a set of vectors $\boldsymbol{g}_0, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_{n-1}$ which give the edges of the polygon, see Figure 1.15. For example, if one type of object is characteristically circular, then these edges describe a polygonal approximation to a circle. The closure

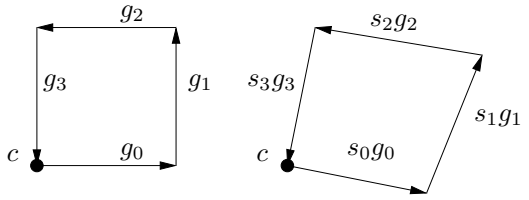


FIGURE 1.15. The left figure shows an undeformed square template. The right figures shows a deformed template holding the location c fixed.

of the polygonal template is equivalent to the condition that $\sum_{i=0}^{n-1} \boldsymbol{g}_i = \mathbf{0}$. Described in this way, the template does not have any location information, and so we will consider its first vertex to be located at the origin, the second to be located at $\boldsymbol{g}_0$, the third at $\boldsymbol{g}_0 + \boldsymbol{g}_1$ and so on. It is possible to accommodate *scaling and rotational effects* , the template may be globally scaled by a scalar R and rotated through an angle α; however we shall ignore this in this exposition. To model natural shape variability occurring between objects of the same type, each edge $\boldsymbol{g}_i$ is subject to a *stochastic deformation* which incorporates an edge–specific Gaussian deformation in length and direction. This edge-specific effect describes the change in length and direction between the undeformed $\boldsymbol{g}_i$ and the new edge. Writing the deformed edge as $\boldsymbol{s}_i\boldsymbol{g}_i$ where $\boldsymbol{s}_i$ is the 2×2 matrix representing these changes, we thus have

$$\boldsymbol{s}_i\boldsymbol{g}_i - \boldsymbol{g}_i = r_i \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \boldsymbol{g}_i. \tag{1.71}$$

Writing $t_i^{(0)} = r_i \cos(\theta_i)$ and $t_i^{(1)} = r_i \sin(\theta_i)$, determines that

$$\boldsymbol{s}_i = \begin{bmatrix} 1 + t_i^{(0)} & t_i^{(1)} \\ -t_i^{(1)} & 1 + t_i^{(0)} \end{bmatrix}. \tag{1.72}$$

Specifying the distribution of r_i and θ_i to be the angular and radial components of a bivariate Gaussian with zero correlation, then $t_i^{(0)}$ and $t_i^{(1)}$ are independent Gaussians with mean zero. Ignoring for a moment the constraint that the deformed template must be closed, i.e. $\sum_{i=0}^{n-1} \boldsymbol{s}_i\boldsymbol{g}_i = \mathbf{0}$, Grenander, Chow & Keenan (1991) suggest a *first order cyclic Markov structure* on the $\{t_i^{(0)}\}$ and the $\{t_i^{(1)}\}$ independently with each having an n–dimensional Gaussian distribution with mean $\mathbf{0}$ and circulant inverse covariance matrix

$$\boldsymbol{\Sigma}^{-1} = \begin{bmatrix} \beta & \delta & & & & \delta \\ \delta & \beta & \delta & & & \\ & \delta & \beta & \delta & & \\ & & \ddots & \ddots & \ddots & \\ & & & \delta & \beta & \delta \\ \delta & & & & \delta & \beta \end{bmatrix}, \tag{1.73}$$

where all other entries are zero. We will in Section 1.7 use the second order model in (1.89), but we use the first order model in this section to avoid unnecessary details.

Define the length $2n$ vector $\boldsymbol{t} = (t_0^{(0)}, t_1^{(0)}, \ldots, t_{n-1}^{(0)}, t_0^{(1)}, \ldots, t_{n-1}^{(1)})^T$ then, still considering only the unconstrained non-closed polygon case

$$\boldsymbol{t} \sim N_{2n}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma} & 0 \\ 0 & \boldsymbol{\Sigma} \end{bmatrix}\right). \tag{1.74}$$

Imposing the closure constraint on the deformed template will destroy the simple structure of (1.74). However, for the purposes of simulation, which will of course require MCMC methods, the unconstrained density suffices because in the acceptance ratio all we will need to evaluate is the ratio of the constrained density at two values of $\boldsymbol{x}$. Since the closure constraint is linear in $\boldsymbol{x}$, the ratio of the constrained densities is also the ratio of the unconstrained densities at the same $\boldsymbol{x}$ values.

1.6.2 Simulation

Simulation from the prior is straight forward as the joint density for the deformations are joint Gaussian and so also with the constrained density as the constraints are linear. Figure 1.16 shows some realisations from the second order model in (1.89) which we will use later in Section 1.7, for various parameter-setting. As we see, the samples mimic quite well circular-like objects.

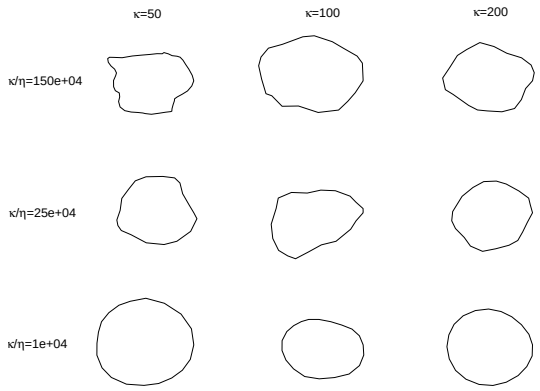


FIGURE 1.16. Samples for the edge transformation template model with precision matrix (1.89), and with different values of the parameters κ and η.

Simulation from the posterior usually requires an MCMC approach. Consider the square template shown in Figure 1.17c. We will use this template in locating the object observed in the data shown in Figure 1.17b. These data have been generated from the underlying pixellated image $\boldsymbol{x}$ in Figure

1.17a by setting the black background pixels to have value $\mu_0 = 0$ and the white foreground pixels to have value $\mu_1 = 1$. Pixel-wise independent zero mean Gaussian noise with variance $\sigma^2 = 0.5^2$ has then been added. Using a uniform prior density for location c, and the model for the deformations $\boldsymbol{s}$ described previously conditioned on the closure of the polygon, together with the likelihood $L(\boldsymbol{y}\,|\boldsymbol{s}, c)$, the posterior density for c and $\boldsymbol{s}$ becomes

$$\pi(\boldsymbol{s}, c \mid \boldsymbol{y}) \propto \pi(\boldsymbol{s}) \prod_{i \in \mathcal{I}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mu(i;\, \boldsymbol{s}, c))^2\right), \tag{1.75}$$

where $\mu(i;\, \boldsymbol{s}, c)$ is equal to μ_1 if pixel i is inside the deformed template, and equal to μ_0 if it is outside.

There are various ways in which we can propose a change to the current realisation of $\boldsymbol{x} = \{c, \boldsymbol{s}\}$. Proposing a change in c alters the location of the current polygon without altering its shape; in the early stages if the object is not well located, moves of this type are quite useful. By proposing symmetric changes in c, the acceptance ratio will depend only on the likelihood ratio of the new and old states. How can the shape of the polygon be changed? Proposed changes can be made either by altering $\boldsymbol{s}$ itself, or by altering the position of one or more of the vertices directly. Notice that the vertex locations can be written as a one-to-one linear transformation of c and $\boldsymbol{s}$ (subject to keeping the labelling of the vertices constant). This means we can propose to move a randomly chosen vertex, perhaps uniformly in some small disc around its existing value (a symmetric proposal). Because the transformation from c and $\boldsymbol{s}$ to the vertices is linear, the Jacobian is a constant, which will therefore cancel in the acceptance ratio. This allows a fast evaluation of the density $\pi(\boldsymbol{s})$ by using its first order Markov property. Figure 1.17d shows some samples of the posterior density using the template in Figure 1.17c.

There are two additional tasks that require our attention working with polygon models, that is to check if the deformed polygon is simple, and whether a site is inside or outside the deformed polygon. These tasks are classical problems in *computational geometry* , see e.g. O'Rourke (1998) for solutions and algorithms.

1.6.3 Marked point process priors

Within this framework, each object x_i comprises a point which gives location (unspecified by the deformation model) and a set of marks which then specify its other attributes, in this case its outline (Baddeley & Van Lieshout 1993). The points lie in a window Λ related to the data observation coordinates, and the marks in the space $\mathcal{M}$ associated with the object shapes. A configuration of objects is described as a finite unordered set $\boldsymbol{x} = \{x_1, \ldots, x_k\}$ where $\boldsymbol{x}$ follows an object process, i.e. a marked point process on $\Lambda \times \mathcal{M}$ with a *Poisson object process* as the basic reference

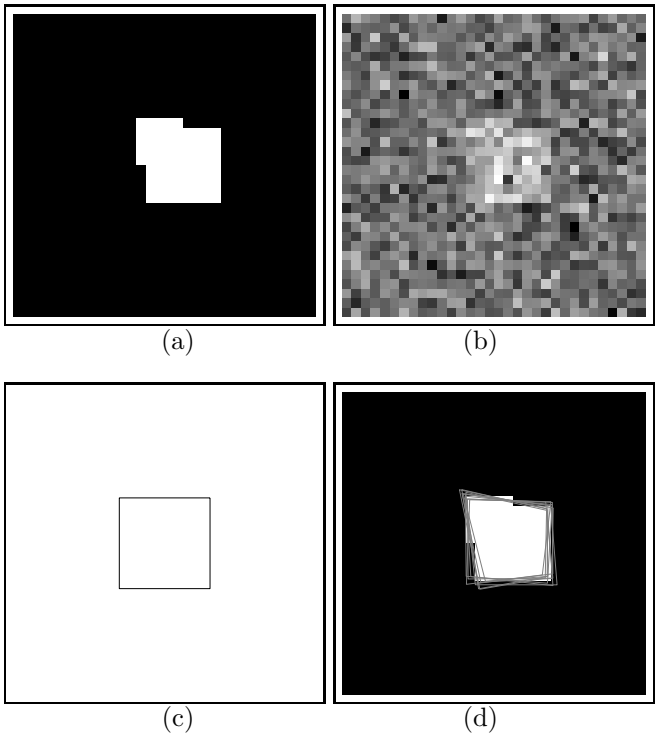


FIGURE 1.17. (a) The true pixellated image; (b) the data; (c) the square template; (d) samples of the deformed template overlaid on the true scene.

process (see Chapter 4). Under the Poisson process, conditional on the number of objects k, the objects are uniformly and identically distributed. The joint density of $\{x_1, \ldots, x_k\}$ is defined by a density $f(\boldsymbol{x})$ relative to the Poisson object process. For example, to model *pairwise interactions* between objects which are defined to be neighbours by some relation $\sim$, the function

$$f(\boldsymbol{x}) \propto \gamma^k \prod_{i \sim j} h(x_i, x_j), \quad \gamma > 0, \tag{1.76}$$

could be used; Van Lieshout (1995) discusses various interaction models. To model a situation where objects are not allowed to overlap, such as confocal microscopy where the images are optical sections, all objects are defined to be neighbours and the interaction function $h(x_i, x_j)$ is taken to be zero if objects x_i and x_j overlap and one otherwise. This model is known as the *hard core object process* . We take the point, denoted $\boldsymbol{c}$, to be the location of the first vertex for each object. The marks are the deformations $\boldsymbol{t}$ of the standard template. It is possible to allow objects of different types by forming a mixture model using different basic templates with different deformation parameters, where the weights in the mixture model represent the relative frequencies of occurrence; refer to Rue & Hurn (1999) for details. This mixture distribution is then used as the mark distribution of the marked point process model to model an unknown number of objects.

Extending the MCMC algorithm to handle an unknown number of objects requires modifications to accommodate the dimensionality changes. Essentially, as well as proposing fixed dimension changes to the current cells, the method has to allow for *births and deaths* of cells. A framework has been provided for this by Geyer & Møller (1994), Green (1995) and Geyer (1999); see Section 4.7.7. One way to decrease the number of cells by one is to delete a randomly selected cell. The complimentary move, increasing the number of cells by one, is to propose a new cell by simulating a location uniformly in the window, and a set of marks at random from the mixture model of template types. In these move types, it is necessary to "*dimension match*" $\boldsymbol{x}$ and $\boldsymbol{x}'$, including an additional Jacobian term in the acceptance ratio. Both types of move are required in sampling our posterior density; moves which keep the dimension of $\boldsymbol{x}$ fixed, for example altering the location of shape of one of the cells, and moves which alter the dimension by altering the number of cells.

1.6.4 Parameter estimation

There are clearly some parameters in the deformable template set-up which are hard to interpret intuitively, in particular β and δ of (1.73). This is a barrier to their more widespread use in applied work. It would be nice to be able to treat them in a fully Bayesian manner, so that uncertainty could be propagated through. Unfortunately, the normalising constants of

this type of model are extremely complex, and this is likely to preclude this approach. Instead in this section we will consider maximum likelihood estimation of β and δ based on recorded vertex information (as could, for example, be gathered using the mouse on display of training data).

We begin by considering a single object, and transforming from the deformation model for the polygon edges to the model for the corresponding vertex locations. Recall that the first vertex defines the location of the entire polygon; if the first vertex is at location $\boldsymbol{c} = \boldsymbol{v}_0$, then the second vertex $\boldsymbol{v}_1$ is located at $\boldsymbol{v}_0 + \boldsymbol{s}_0\boldsymbol{g}_0$ and so on,

$$\boldsymbol{v}_j = \boldsymbol{v}_0 + \sum_{i=0}^{j} \boldsymbol{s}_i \boldsymbol{g}_i, \qquad j = 1, \ldots, n.$$

There are $n+1$ vertices in the non-closed polygon. Considering the x and y components separately, the vertices can be written

$$\begin{bmatrix} v_1^x \\ v_2^x \\ \vdots \\ v_n^x \\ v_1^y \\ v_2^y \\ \vdots \\ v_n^y \end{bmatrix} = \begin{bmatrix} g_0^x & & & & g_0^y & & & \\ g_0^x & g_1^x & & & g_0^y & g_1^y & & \\ \vdots & \vdots & \ddots & & \vdots & \vdots & \ddots & \\ g_0^x & g_1^x & \cdots & g_{n-1}^x & g_0^y & g_1^y & \cdots & g_{n-1}^y \\ g_0^y & & & & -g_0^x & & & \\ g_0^y & g_1^y & & & -g_0^x & -g_1^x & & \\ \vdots & \vdots & \ddots & & \vdots & \vdots & \ddots & \\ g_0^y & g_1^y & \cdots & g_{n-1}^y & -g_0^x & -g_1^x & \cdots & -g_{n-1}^x \end{bmatrix} \begin{bmatrix} t_0^{(0)} \\ t_1^{(0)} \\ \vdots \\ t_{n-1}^{(0)} \\ t_0^{(1)} \\ t_1^{(1)} \\ \vdots \\ t_{n-1}^{(1)} \end{bmatrix} + \bar{\boldsymbol{v}}_0 \tag{1.77}$$

where $\bar{\boldsymbol{v}}_0$ is the vector of vertex x and y positions of the undeformed template with first vertex located at $\boldsymbol{v}_0$. We will write (1.77) in the form $\boldsymbol{v} = \boldsymbol{G}\boldsymbol{t} + \bar{\boldsymbol{v}}_0$. The distribution of $\boldsymbol{v}$ unconstrained by closure given the observed $\boldsymbol{v}_0$ is therefore

$$\boldsymbol{v}^T \mid \boldsymbol{v}_0 \sim N_{2n}\left(\bar{\boldsymbol{v}}_0 \ , \ \boldsymbol{G} \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma} \end{bmatrix} \boldsymbol{G}^T \right). \tag{1.78}$$

To find the constrained distribution of $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1})^T | (\boldsymbol{v}_n = \boldsymbol{v}_0)$, it is simpler to reorder the components of $\boldsymbol{v}$ from x then y components to the vertex pairs, rewriting (1.78) as

$$(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_n)^T \mid \boldsymbol{v}_0 \sim N_{2n}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{v}_0 \end{bmatrix} \ , \ \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right), \tag{1.79}$$

where the partitioning of the mean and variance correspond to (1.78) partitioned into the sets $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1})^T$ and $\boldsymbol{v}_n$. Note that $\mathrm{E}(\boldsymbol{v}_n | \boldsymbol{v}_0) = \boldsymbol{v}_0$ by closure of the undeformed template. Denote the partitioned inverse of the

variance matrix

$$\begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{\Psi}_{11} & \boldsymbol{\Psi}_{12} \\ \boldsymbol{\Psi}_{12}^T & \boldsymbol{\Psi}_{22} \end{bmatrix}. \tag{1.80}$$

Denote $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-1}$ by $\boldsymbol{v}_{-n}$, then

$$\pi(\boldsymbol{v}_{-n}, \boldsymbol{v}_n \mid \boldsymbol{v}_0) = \frac{(2\pi)^{-n}}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2} \begin{bmatrix} \boldsymbol{v}_{-n} - \boldsymbol{\mu}_1 \\ \boldsymbol{v}_n - \boldsymbol{v}_0 \end{bmatrix}^T \boldsymbol{\Sigma}^{-1} \begin{bmatrix} \boldsymbol{v}_{-n} - \boldsymbol{\mu}_1 \\ \boldsymbol{v}_n - \boldsymbol{v}_0 \end{bmatrix} \right) \tag{1.81}$$

$$\pi(\boldsymbol{v}_n \mid \boldsymbol{v}_0) = \frac{(2\pi)^{-1}}{|\boldsymbol{\Sigma}_{22}|^{1/2}} \exp\left(-\frac{1}{2} \left[\boldsymbol{v}_n - \boldsymbol{v}_0\right]^T \boldsymbol{\Sigma}_{22}^{-1} \left[\boldsymbol{v}_n - \boldsymbol{v}_0\right] \right). \tag{1.82}$$

Then the conditional density of interest is

$$\begin{aligned} \frac{\pi(\boldsymbol{v}_{-n}, \boldsymbol{v}_n \mid \boldsymbol{v}_0)|_{\boldsymbol{v}_n = \boldsymbol{v}_0}}{\pi(\boldsymbol{v}_n \mid \boldsymbol{v}_0)|_{\boldsymbol{v}_n = \boldsymbol{v}_0}} &= (2\pi)^{-(n-1)} \left(\frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}_{22}|} \right)^{-1/2} \\ &\times \exp\left(-\frac{1}{2} (\boldsymbol{v}_{-n} - \boldsymbol{\mu}_1)^T \boldsymbol{\Psi}_{11} (\boldsymbol{v}_{-n} - \boldsymbol{\mu}_1) \right). \end{aligned} \tag{1.83}$$

By some manipulation of the variance matrices, this can be seen to be the density of a $N_{2n-2}(\boldsymbol{\mu}_1, \boldsymbol{\Psi}_{11}^{-1})$. Assuming that $\boldsymbol{v_0}$ is uniformly distributed in the observation window, and under an assumption of independence of the polygon shapes, the likelihood for m cells will be the product of these individual likelihoods. This will have to be maximised numerically.

1.7 An example in ultrasound imaging

In this final section, we present an analysis of a real data set. Our goal here is to demonstrate how complex tasks can be tackled using techniques based on the type of ideas presented in the previous sections.

1.7.1 Ultrasound imaging

Ultrasound is widely used in medical settings, mainly because of its ease of use and its real-time imaging capability. However, the diagnostic quality of ultrasound images is low due to noise and image artifacts (*speckle*) introduced in the imaging process. The principle of ultrasound imaging is simple: A pulse of ultra-high frequency sound is sent into the body, and the backscattered return signal is measured after a time delay corresponding to depth. When the pulse hits a boundary between tissues having different acoustic impedances, it is partially reflected and partially transmitted. In addition there is reflection within homogeneous material due to small spatial variations in acoustical impedance, called scatterers. Thus variations in

acoustic impedance is the basis for identifying regions of interest in the imaged tissue. We concentrate on the second mode of variation, called diffuse scattering, and use a Bayesian model developed in Hokland & Kelly (1996) and Husby, Lie, Langø, Hokland & Rue (2001). In this model the area Ω_i imaged in pixel i is assumed to consist of a large number of uniformly distributed scatterers, and the received signal is the sum of the reflections from all these scatterers. Assuming that all scatterers are independent, and invoking the central limit theorem, the resulting radio frequency signal x_i is assumed to be a Gaussian random variable with mean zero and a variance determined by the scattering properties of the tissue in Ω_i,

$$x_i \mid \sigma_i^2 \sim \mathrm{N}\left(0, \sigma_i^2\right), \forall i. \tag{1.84}$$

It is in other words he variance which characterises the different tissues, and we can thus segment the image into different tissue types by identifying regions where the Gaussian echoes have approximately equal variances. Note that given the variances, the radio-frequency signal contains no additional information, and can thus be regarded as a nuisance parameter.

The observed image $\boldsymbol{y}$ is modelled as resulting from a convolution of the radio frequency signal $\boldsymbol{x}$ with the imaging system point spread function $\boldsymbol{h}$, with the addition of independent and identically distributed Gaussian noise. We assume the point spread function to be spatially invariant, thus

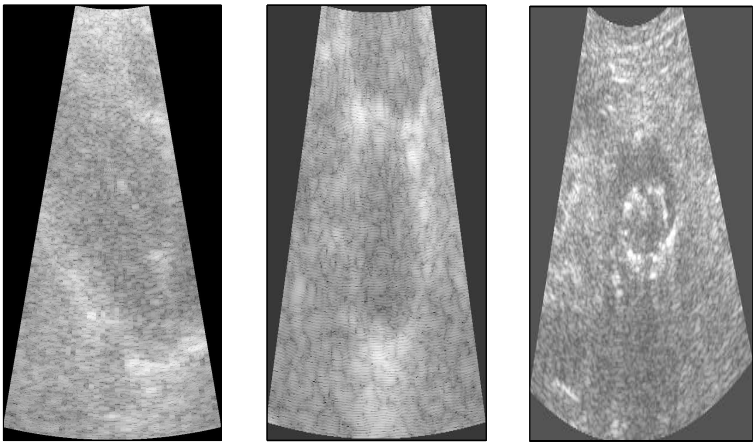
$$y_i \mid \boldsymbol{x} \sim \mathrm{N}\left(\sum_k h_k x_{i+k}, \tau^2\right), \quad \forall i, \tag{1.85}$$

where τ^2 is the noise variance. The pulse function is modelled as a separable Gaussian function with a sine oscillation in the radial direction, i.e.

$$h_{k,l} \propto \exp\left(-\frac{k^2}{2\sigma_r^2} - \frac{l^2}{2\sigma_l}\right) \cos\frac{2\pi k}{\omega}. \tag{1.86}$$

Empirical studies indicate that this is a good approximation which seems to be quite robust with respect to misspecification of the parameters.

Figure 1.18 shows examples of medical ultrasound images. The images are log-compressed before display to make it easier to see the anatomy. Figure 1.18a shows a part of the right ventricle of a human heart, while Figure 1.18b shows a cross-section of the carotid artery. Figure 1.18c shows an abdominal aorta aneurism, that is a blood-filled dilatation of the aorta. In the middle of the aorta it is possible to spot a vessel prothesis. Common to the images is the characteristic speckle pattern that makes it difficult for the untrained eye to spot the important anatomical features. We will focus on the cross-sectional images of the carotid artery, doing both image restoration and estimation of the cross-sectional area. An interval estimate of the artery area might also be useful as a mean of diagnosing atherosclerosis.



(a) (b) (c)

FIGURE 1.18. Real ultrasound images. Panel (a) and (b) show log-compressed radio frequency images of (a) the right ventricle of the heart, and (b) a cross section through the carotid artery. Panel (c) shows a B-scan ultrasound image of an aorta aneurism with a vessel prothesis in the middle.

1.7.2 Image restoration

We first consider restoration of the true radio-frequency image given the observed image $\boldsymbol{y}$. In this respect the most important modelling step is the specification of a prior model for the underlying variance field $\boldsymbol{\sigma}^2$, since this parameter contains information about the anatomy of the imaged tissue. In fact, the radio frequency field $\boldsymbol{x}$ contains no additional information, and can thus be seen as a nuisance parameter. However, as argued in Husby et al. (2001), a model formulation containing $\boldsymbol{x}$ has great computational advantages, since the distribution for the variance field $\boldsymbol{\sigma}^2$ given the data has no local Markov structure, whereas the distribution for $\boldsymbol{\sigma}^2$ given the data and the radio frequency field has a neighbourhood structure depending on the support of the point spread function $\boldsymbol{h}$.

To avoid problems with positivity we reparameterised the model and define a log-variance field $\boldsymbol{\nu} = (\log \sigma_i \,:\, i \in \mathcal{I})$. The choice of prior model for this field should be justified from physical considerations about the imaged tissue, and we use the following assumptions:

- the scattering intensity tends to be approximately constant within regions of homogeneous tissue,
- abrupt changes in scattering intensity may occur at interfaces between different tissue types.

Based on these assumptions it is reasonable to model the log-variance field

$\boldsymbol{\nu}$ as being piecewise smooth with homogeneous subregions corresponding to the different tissue types in the imaged region. As explained in Section 1.5.1, edge-preserving functionals are well suited for modelling such fields. Thus we define the prior distribution for $\boldsymbol{\nu}$ as

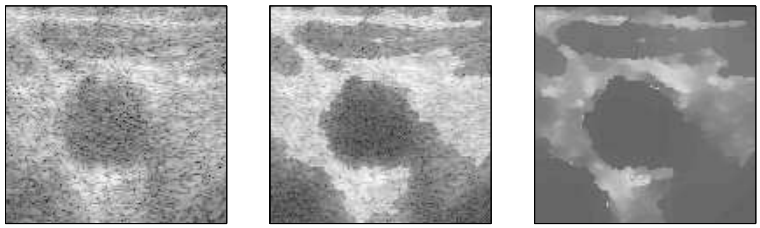
$$\pi(\boldsymbol{\nu}) \propto \exp\left(-\beta \sum_{m=1}^{M} w_m \sum_{i \in \mathcal{I}} \phi\left(D_i^{(m)} \boldsymbol{\nu}\right)\right), \tag{1.87}$$

where ϕ is a functional from the edge preserving class defined in (1.62), $w_1, \ldots, w_M$ are positive constants, β is a positive scaling factor, and $D_i^{(m)}$ are difference operators approximating first order derivatives, e.g. $D_i^{(1)}\boldsymbol{\nu} = (\nu_{i+1} - \nu_i)/\delta$. Unless otherwise stated, we will use the four first order cliques (and so eight nearest neighbours) with corresponding constants $w_1 = w_2 = 1$, $w_3 = w_4 = 1/\sqrt{2}$. The scaling parameters β and δ are assumed to be known constants, although it is possible to integrate them out using a fully Bayesian approach. δ should be selected to match the intensity jumps in the image, while a suitable choice for β can be found by trial and error. Large values of β tend to give smooth realisations, while small values give noisy realisations.

Combining equations (1.84), (1.85) and (1.87) we obtain the full conditional distribution for the radio-frequency image $\boldsymbol{x}$ and the log-variance field $\boldsymbol{\nu}$ as

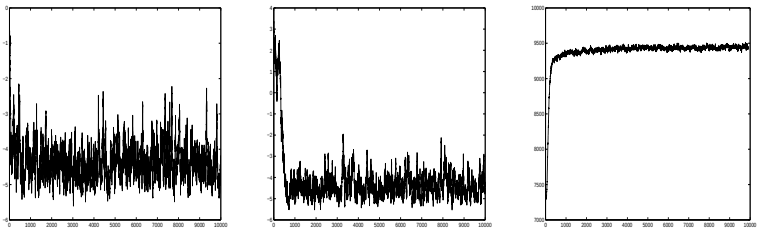
$$\begin{aligned} \pi(\boldsymbol{x}, \boldsymbol{\nu} \mid \ldots) \quad &\propto \quad \prod_{i \in \mathcal{I}} \exp\left(-\frac{1}{2\tau^2}\left(y_i - \sum_k h_k x_{i+k}\right)^2\right) \\ &\times \quad \exp\left(-\frac{x_i^2}{2}\exp(-2\nu_i) - \nu_i - \beta \sum_m \omega_m \phi\left(D_i^{(m)}\boldsymbol{\nu}\right)\right). \end{aligned} \tag{1.88}$$

A point estimate $\hat{\boldsymbol{x}}$ of the true radio frequency image $\boldsymbol{x}^*$ can be constructed using MCMC output, and a natural first choice of estimator is the posterior mean. The simplest way of constructing the Markov chain is to use a single-site Metropolis-Hastings algorithm for $\boldsymbol{\nu}$, and the Gibbs sampler for $\boldsymbol{x}$. Alternatively, one might update $\boldsymbol{\nu}$ and $\boldsymbol{x}$ as blocks by using the same dual formulation as before and utilising an algorithm for efficient sampling of Gaussian Markov random fields (Rue 2001). The single site sampler was run for $10,000$ iterations on the blood vessel image in Figure 1.19a, and posterior mean estimates of the radio frequency- and log-variance-fields are shown in Figure 1.19b and c, respectively. The images are plotted in polar coordinates. To get a feel for the convergence of the chain, we have plotted traces of the log-variance at two different positions (Figure 1.20a and b), as well as the functional $f(\boldsymbol{\nu}) = \beta \sum_m \omega_m \sum_i \phi(D_i^{(m)}\boldsymbol{\nu})$ (Figure 1.20c).



(a) (b) (c)

FIGURE 1.19. Image restoration using an edge-preserving model: (a) log-compressed radio frequency image of a cross section through the carotid arteryl in polar coordinates; (b) a-posterior mean estimate of the true radio frequency image; (c) the corresponding posterior mean estimate of the underlying log-variance field.



(a) (b) (c)

FIGURE 1.20. Convergence diagnostics for the experiment in Figure 1.19. Panels (a) and (b) show trace plots of the log-variance at different positions, while panel (c) shows a trace plot of the functional $f(\boldsymbol{\nu}) = \beta \sum_m \omega_m \sum_i \phi(D_i^{(m)} \boldsymbol{\nu})$.

1.7.3 Contour estimation

In this example we will use a template model for detecting the outline of an artery wall in an ultrasound image. As noted already, this could be used in diagnosing atherosclerosis, since diseased arteries are less likely to dilate in response to infusion of achetylcholine. For this procedure to be useful, we need to quantify the uncertainty of the given answer, for instance by means of an interval estimate. Thus, doing a segmentation of the image, or using a standard contour-detection method, would not be satisfactory, as they only provide point estimates. Moreover, procedures such as segmentation are not robust with respect to image artifacts such as the *missing edge* at the lower right of the artery wall in Figure 1.18b. Such artifacts can easily be dealt with in a template model, see Husby (2001) for details.

We model the artery outline $\boldsymbol{e}$ as the result of applying a transformation to a predefined circular template $\boldsymbol{e}^0$ with m edges. The transformation vector $\boldsymbol{s}$ is modelled as a second order circulant Gaussian Markov random field with precision matrix $\boldsymbol{Q}_s = \boldsymbol{I}_2 \otimes \boldsymbol{Q}$, where $\boldsymbol{Q}$ is a circulant Toeplitz matrix with entries

$$Q_{ij} = \begin{cases} \frac{\kappa}{m} + 6\eta m^3, & j = i \\ -4\eta m^3, & j = i-1, i+1 \bmod m \\ \eta m^3, & j = i-2, i+2 \bmod m, \end{cases} \qquad \kappa, \eta > 0 \tag{1.89}$$

With this parametrisation the behaviour of the model is approximately independent of the number m of edges. We assign independent Gamma priors $\Gamma(a_\kappa, b_\kappa)$ and $\Gamma(a_\eta, b_\eta)$ to the parameters κ and η. See Figure 1.16 for some realisations from this model and Hobolth & Jensen (2000) for some explicit results of the limiting process.

Having an explicit model for the artery wall, we no longer need the implicit edge model, but a model for the log-variance field is still needed. A very simple approach is to assume that there are two smooth Gaussian fields $\boldsymbol{\nu}_0$ and $\boldsymbol{\nu}_1$ associated with the back- and foreground, respectively. The fields are defined on the whole image domain, but are only observed within their respective regions; thus, letting $\mathcal{T}_{\boldsymbol{s}} \subset \mathcal{I}$ be the set of pixels enclosed by the template deformed by $\boldsymbol{s}$, the conditional distribution for the radio frequency field $\boldsymbol{x}$ is

$$x_i \mid \nu_{0,i}, \nu_{1,i}, \boldsymbol{s} \sim \begin{cases} \mathrm{N}\left(0, \exp(-2\nu_{0,i})\right), & i \in \mathcal{T}_{\boldsymbol{s}}^C \cap \mathcal{I} \\ \mathrm{N}\left(0, \exp(-2\nu_{1,i})\right), & i \in \mathcal{T}_{\boldsymbol{s}} \cap \mathcal{I} \end{cases} \qquad \forall i \in \mathcal{I}. \tag{1.90}$$

For simplicity we use an intrinsic Gaussian Markov random field model for the log-variance fields,

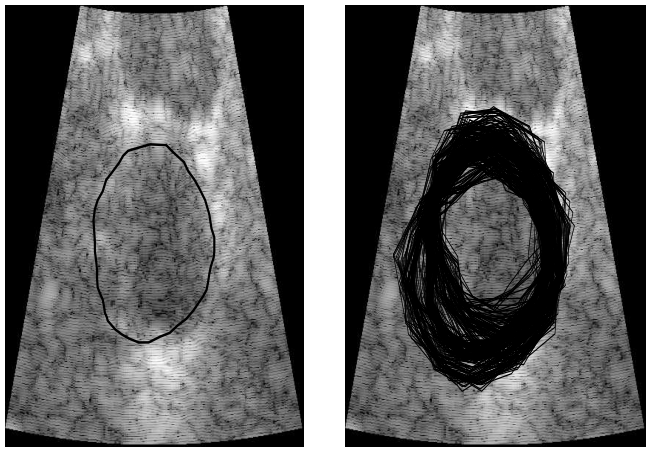
$$\pi(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1) \propto \exp\left(-\sum_{k=0}^{1} \tau_k \sum_{i \sim j} q_{ij} \left(v_{k,i} - v_{k,j}\right)^2 \right), \tag{1.91}$$

where

$$q_{ij} = \begin{cases} |\partial_i|, & i = j \\ -1, & i \sim j, \end{cases}$$

and 0 otherwise. The precisions τ_0 and τ_1 are given Gamma priors with common hyperparameters c and d.

Sampling is again done most simply using single site random walk Metropolis-Hastings algorithms, but experience shows that this leads to slow convergence for our model. Instead we have used a block sampling algorithm, see Husby (2001) for details. Figure 1.21a shows a point estimate of the artery wall based on $200,000$ iterations of the sampler. To get a measure of the uncertainty we have plotted samples from the posterior distribution in Figure 1.21b. The variation seems to correspond well with the human perception of uncertainty. A trace plot of the cross-sectional area of the blood vessel is shown in Figure 1.22; the plot indicates that the chain mixes well, and confirms that there is a great deal of uncertainty. A density estimate of the cross-sectional area is shown in Figure 1.23.



(a) (b)

FIGURE 1.21. Contour estimation: (a) Point estimate of the vessel contour; (b) Samples from the posterior distribution, taken with a separation of 500 iterations.

Acknowledgments: Thanks to Prof. Adrian Baddeley for providing the newspaper image, and to the referees and editors for their comments.

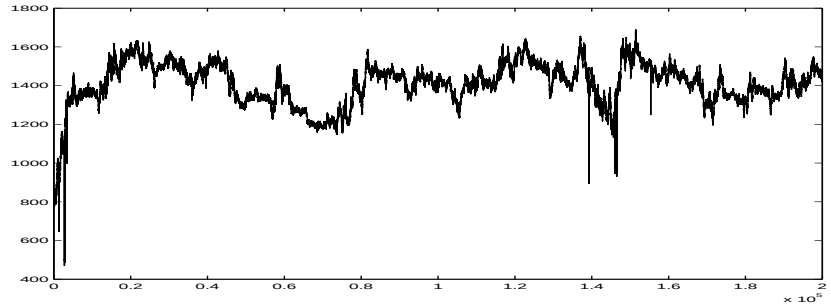


FIGURE 1.22. Trace plot of the cross-sectional area of the template in Figure 1.21.

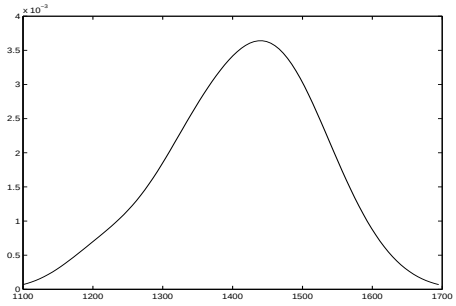


FIGURE 1.23. Density estimate of the cross-sectional area of the blood vessel in Figure 1.18 (a).

1.8 References

Baddeley, A. J. & Van Lieshout, M. N. M. (1993). Stochastic geometry models in high–level vision, *in* K. V. Mardia & G. K. Kanji (eds), *Statistics and Images*, Vol. 20, Carfax Publishing, Abingdon, pp. 235–256.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion), *Journal of the Royal Statistical Society, Series B* **36**: 192–225.

Besag, J. (1986). On the statistical analysis of dirty pictures (with discussion), *Journal of the Royal Statistical Society, Series B* **48**: 259–302.

Besag, J. & Green, P. J. (1993). Spatial statistics and Bayesian computation (with discussion), *Journal of the Royal Statistical Society, Series B* **55**: 25–37.

Blake, A. & Isard, M. (1998). *Active Contours*, Springer-Verlag, Berlin.

Blake, A. & Zisserman, A. (1987). *Visual Reconstruction*, MIT Press, Cambridge, MA.

Charbonnier, P. (1994). *Reconstruction d'image: Régularization avec prise en compte des discontinuités*, PhD thesis, Univ. Nice, Sophia Antipolis, France.

Charbonnier, P., Blanc-Feraud, L., Aubert, G. & Barlaud, M. (1997). Deterministic edge-preserving regularization in computed imaging, *IEEE Transaction on Image Processing* **6**: 298–311.

Dryden, I. & Mardia, K. (1999). *Statistical Shape Analysis*, John Wiley and Sons, Chichester.

Friel, N. & Molchanov, I. (1998). Distances between grey-scale images, *Mathematical morphology and its applications to image and signal processing*, Vol. 12 of *Comput. Imaging Vision*, Amsterdam, The Netherlands, pp. 283–290.

Geman, D. (1990). Random fields and inverse problems in imaging, *in* P. L. Hennequin (ed.), *Ecole d'ete de probabilites de Saint-Flour XVIII, 1988*, Springer, Berlin. Lecture notes in mathematics, no 1427.

Geman, D. & Yang, C. (1995). Nonlinear image recovery with half-quadratic regularization, *IEEE Transaction on Image Processing* **4**: 932–946.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**: 721–741.

Geman, S. & McClure, D. (1987). Statistical methods for tomographic image reconstruction, *Proc. 46th Sess. Int. Stat. Inst. Bulletin ISI*, Vol. 52.

Geyer, C. (1999). Likelihood inference for spatial point processes, *in* O. E. Barndorff-Nielsen, W. S. Kendall & M. N. M. van Lieshaut (eds), *Stochastic Geometry: Likelihood and Computation*, Chapman and Hall/CRC, London, Boca Raton, pp. 79–140.

Geyer, C. J. & Møller, J. (1994). Simulation procedures and likelihood inference for spatial point processes, *Scandinavian Journal of Statistics* **21**: 359–373.

Glasbey, C. A. & Mardia, K. V. (2001). A penalized likelihood approach to image warping (with discussion), *Journal of the Royal Statistical Society, Series B* **63**: 465–514.

Green, P. J. (1990). Bayesian reconstruction from emission tomography data using a modified EM algorithm, *IEEE Transaction on Medical Imaging* **9**: 84–93.

Green, P. J. (1995). Reversible jump MCMC computation and Bayesian model determination, *Biometrika* **82**: 711–732.

Greig, D. M., Porteous, B. T. & Scheult, A. H. (1989). Exact maximum a posteriori estimation for binary images, *Journal of the Royal Statistical Society, Series B* **51**: 271–279.

Grenander, U. (1993). *General Pattern Theory*, Oxford University Press, Oxford.

Grenander, U., Chow, Y. & Keenan, D. M. (1991). *Hands: a Pattern Theoretic Study of Biological Shapes*, Research Notes on Neural Computing, Springer, Berlin.

Grenander, U. & Miller, M. I. (1994). Representations of knowledge in complex systems (with discussion), *Journal of the Royal Statistical Society, Series B* **56**: 549–603.

Grimmett, G. (1987). Interacting particle systems and random media: An overview, *International Statistics Review* **55**: 49–62.

Hebert, T. & Leahy, R. (1989). A generalized EM algorithm for 3D Bayesian reconstruction form Poisson data using Gibbs priors, *IEEE Transaction on Medical Imaging* **8**: 194–202.

Hobolth, A. & Jensen, E. B. V. (2000). Modelling stochastic changes in curve shape, with application to cancer diagnostics, *Advances in Applied Probability (SGSA)* **32**: 344–362.

Hokland, J. & Kelly, P. (1996). Markov models of specular and diffuse scattering in restoration of medical ultrasound images, *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control* **43**: 660–669.

Husby, O. (2001). High-level models in ultrasound imaging, *Preprint*, Department of mathematical sciences, Norwegian University of Technology and Science, Trondheim.

Husby, O., Lie, T., Langø, T., Hokland, J. & Rue, H. (2001). Bayesian 2d deconvolution: A model for diffuse ultrasound scattering, *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control* **48**: 121–130.

O'Rourke, J. (1998). *Computational Geometry in C*, Cambridge University, Cambridge. Press.

Rue, H. (2001). Fast sampling of Gaussian Markov random fields with applications, *Journal of the Royal Statistical Society, Series B* **63**: 325–338.

Rue, H. & Hurn, M. A. (1999). Bayesian object identification, *Biometrika* **86**: 649–660.

Stander, J. & Silverman, B. W. (1994). Temperature schedules for simulated annealing, *Statistics and Computing* **4**: 21–32.

Swendsen, R. & Wang, J. (1987). Nonuniversal critical dynamics in Monte Carlo simulations, *Physical Review Letters* **58**: 86–88.

Tjelmeland, H. & Besag, J. (1998). Markov random fields with higher order interactions, *Scandinavian Journal of Statistics* **25**: 415–433.

Van Lieshout, M. N. M. (1995). Markov point processes and their applications in high-level imaging (with discussion), *Bulletin of the International Statistical Institute* **LVI, Book 2**: 559–576.

Winkler, G. (1995). *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, Springer, Berlin.

Wood, A. T. A. & Chan, G. (1994). Simulation of stationary Gaussian processes in $[0, 1]^d$, *Journal of Computational and Graphical Statistics* **3**: 409–432.

Index