University of Bath

PHD

Some topics in statistical image analysis

Stander, Julian

Award date:
1992

Awarding institution:
University of Bath

Some Topics in Statistical Image Analysis

Submitted by

Julian Stander

for the degree of PhD

of the

University of Bath

1992

J Stander

J. Stander

Abstract

In image analysis we aim to find a description of an image from observed data. In Chapters 1 to 4 we consider the reconstruction of an image observed directly, but with noise, on a grid of pixels as the description. In Chapter 1 we introduce statistical models both for the set of possible images *via* a prior distribution, and for the noise *via* a likelihood. These models are combined by Bayes's theorem to produce a posterior distribution. The reconstruction maximizes this posterior distribution, or, equivalently, minimizes a penalty function representing a trade off between the infidelity of a given image to the data and its roughness, controlled by an unknown smoothing parameter.

In Chapter 2 we investigate a modification to the usual penalty function that introduces an edge process to model the boundaries of the image.

Direct calculation of the global minimum of the penalty function is computationally prohibitive, except in one special case when the image comprises two colours and a fluid flow algorithm is employed. In Chapter 3 we exploit a special feature of this algorithm to estimate the smoothing parameter.

Simulated annealing is often used to attempt to find the global minimum. This algorithm is controlled by a temperature schedule, the effect of which we examine in Chapter 4.

In PET (positron emission tomography) the image represents the metabolic activity of a cross-section of an organ in the body and can be thought of as a density f. The image is observed indirectly by a detector ring. The description of the image is now provided by a linear functional of f, the estimation of which we consider in Chapter 5, both when the ring is continuous and when it comprises a finite number of detectors. In Chapter 6 we illustrate this theory with some numerical examples.

Acknowledgements

This research was carried out under the supervision of Professor Bernard W. Silverman. I am extremely grateful to him both for his guidance and encouragement during the course of this work, and for his kindness.

I would also like to thank Mario Cortina-Borja for many invaluable discussions and for his efficient and meticulous checking of this thesis. I am also grateful to James Duthie for reading over the manuscript, though of course any mistakes, omissions or obscurities remain entirely the responsibility of the author.

I would like to express gratitude to all the people who have encouraged and supported me during the course of this work; it is impossible to name them all here. In particular I would like to thank Luisa Franconi for many very encouraging comments and useful suggestions, especially with regard to Chapter 4 and Chapter 5. I am grateful to Guy Nason, Nigel Phelan, Glenn Stone and Mark Willis who were generous with help in particular concerning computer problems. Very special thanks are also due to John Eales, Tony Garratt, Stephen Gourley (especially for invaluable help with the analysis in Chapter 5), Howard Grubb, Merrilee Hurn, Mike Jubb, Marion Cortina-Borja, Sergio Pezzulli, Nuala Sheehan and Katerina Tzioli, as well as other members of the school of Mathematical Sciences for their advice, support and suggestions. Grazie anche a tutti i miei altri amici nel dipartimento e fuori.

Additional thanks are due to my Parents and family for their continual and unfailing support.

I would like to acknowledge my gratitude to the School of Mathematical Sciences for its excellent computational facilities without which much of this work would not have been possible. Finally I thank the Science and Engineering Research Council and the University of Bath for their financial support.

Contents

Chapter 1

Statistical Image Reconstruction: Introduction to the Direct Problem

The first part of this thesis comprises this chapter and Chapters 2, 3 and 4, and considers some topics in statistical image reconstruction when the data have been observed directly, but imperfectly, on a grid of picture elements known as pixels. In these chapters we define *image reconstruction* as the removal of noise from the observed data to reveal the image which would have been viewed under ideal conditions. The techniques associated with image reconstruction have many applications in subjects such as astronomy, remote sensing, industrial inspection, biological taxonomy, stereology and satellite imaging to name but a few.

We consider a two dimensional region partitioned into pixels. In this chapter we assume for simplicity that the region and the pixels are square (or possibly rectangular), but almost all the techniques can be easily generalized to irregular and uneven pixel arrays. Such arrays are discussed, for example, in Silverman, Jennison, Stander and Brown[39], where an edge process is introduced to approximate the boundaries that are present in the underlying real image regardless of the pixellation (see also Chapter 2) and Silverman, Jones, Wilson and Nychka[40], where a pixellation is introduced in the context of positron emission tomography (PET) that exploits circular symmetries and leads to substantial computational savings in both storage and time (see also Section 5.2.3).

Let us assume that the image comprises n pixels, indexed by i, and that the true value at pixel i is x_i^*. The image is, however, observed imperfectly and at pixel i a degraded record (or signal) y_i, related to x_i^*, is observed. In the main, we shall assume that the records $y_1, \dots, y_n$ are conditionally independent given the image, and that

$$y_i \sim \mathcal{N}(x_i^*, \kappa),$$

where the variance $\kappa > 0$ is assumed known. We shall refer to this type of degradation mechanism as 'normal noise'. In Section 1.9 we briefly discuss the estimation of the variance of the noise in the case when κ is unknown. In Chapter 3, where we consider images whose pixels can take only two colours (0 and 1, corresponding to white and black), we introduce a different type of degradation mechanism, in which each pixel switches colour with known probability. There we refer to this type of degradation mechanism as the 'binary channel'.

If we define x^* to be the $n \times 1$ vector with entries x_i^*, and y to be the $n \times 1$ vector with entries y_i, then in the case of normal noise we can write

$$y = x^* + \varepsilon, \tag{1.1}$$

where ε is an $n \times 1$ vector with multivariate normal distribution $\mathcal{N}(0, \kappa I)$ and I is the $n \times n$ identity matrix. Many authors work with a much more complicated set-up. For example, Geman and Reynolds[14] consider

$$y = Hx^* + \varepsilon, \tag{1.2}$$

where now y and ε are $m \times 1$ vectors, and H is a known $m \times n$ matrix, with $m \leq n$, representing the point spread function by means of which blur is modelled. In general $m < n$ due to the nature of optical blurring. We do not study blurring in this part of the thesis, but restrict our attention to the model given in equation (1.1). Our aim is to deduce the true image given the record y.

1.1 The role of statistics and the Bayesian paradigm

A very good introduction to the role of statistics in image reconstruction is given by Jubb[24]. Although image reconstruction has a long history, it is only since the 1980's that statisticians have begun to regard it as a legitimate part of their subject. But what have statisticians to add to what has been done by physicists, electronic engineers and computer scientists? A possible answer can be obtained from the recognition that statistics can offer a good treatment of the degradation mechanism and good models for the true image. The general Bayesian paradigm can be applied directly to the image problem. Let the set of all possible images be $\mathcal{X}$. Assume for the moment that we can write down a prior distribution $\Pr(x)$ for all $x \in \mathcal{X}$ that in some way reflects our beliefs about images. A very commonly held belief is that nearby pixels of the image should take similar values. We shall formalize this notion in Section 1.3 where we discuss the concept of neighbourhood. From our knowledge of the noise we can write down the likelihood of the record y given an image x, as $l(y \mid x)$. The prior distribution and the likelihood can be combined by means of Bayes's theorem to give us the posterior distribution of an image x given the (fixed) record y:

$$\Pr(x \mid y) \propto l(y \mid x) \Pr(x), \tag{1.3}$$

where the constant of proportionality does not depend upon x. In general, the idea is now to try to summarize this posterior distribution by giving a single estimate of x and we discuss possible summaries of the posterior distribution in Section 1.4. However, before we can make any further progress we must discuss the form of the likelihood and the prior distributions. From the assumptions that we have stated above it is quite easy to write down the likelihood, and we briefly discuss it in Section 1.2. The prior distribution is much more difficult and is discussed in detail in Section 1.3.

1.2 The likelihood, and simulating the noise distribution

We have already stated that we make the assumptions that given any image x, the records $y_1, \ldots, y_n$ are conditionally independent, and that each y_i has the same known conditional density function $f(y_i \mid x_i)$, dependent only on x_i. (In Section 3.3 we shall discuss the possibility of relaxing the second assumption.) Thus, the conditional density of the observed image y given an image x is simply

$$l(y \mid x) = \prod_{i=1}^{n} f(y_i \mid x_i). \tag{1.4}$$

If we assume that the distribution of the record y_i at pixel i given x_i is normal with mean x_i and known variance $\kappa > 0$, we have

$$f(y_i \mid x_i) = \frac{1}{\sqrt{2\pi\kappa}} \exp\left\{ \frac{-(y_i - x_i)^2}{2\kappa} \right\},$$

and from equation (1.4)

$$\log l(y \mid x) = -\frac{n}{2} \log(2\pi\kappa) - \frac{1}{2\kappa} \sum_{i=1}^{n} (y_i - x_i)^2. \tag{1.5}$$

We remark here that the first term of the right hand side of equation (1.5) does not depend upon x and that this log-likelihood form will be useful when we consider the penalty function approach in Section 1.5.

1.2.1 The noise distribution

We have seen that the records $y_1, \ldots, y_n$ are conditionally independent given the image and that $y_i \sim \mathcal{N}(x_i^*, \kappa)$, where the true value at pixel i is x_i^* and the variance $\kappa > 0$ is assumed known. For most of the experiments that are described in this part of the thesis, we add pseudo white noise of known variance κ to a known true scene. To generate this pseudo white noise numbers uniformly distributed between 0 and 1 are produced, and these are then transformed into standard normal variates. We find numbers that are uniformly distributed useful in their own right: for example, we use them with the Gibbs sampler (see Section 1.4.1), in performing simulated annealing (see Section 1.6.1) and in simulating the degradation mechanism known as the binary channel (see Chapter 3).

An algorithm due to Wichmann and Hill[45] was used to produce numbers uniformly distributed between 0 and 1, excluding the end points. Three simple congruential generators (see Ripley[32]) of the form

$$X_i = a X_{i-1} \bmod M$$

are used, where the a s and M s are specified by the authors. Each generator has a prime number for its modulus M and a primitive root for its multiplier a, that is, $a \neq 0$ and $a^{(M-1)/p} \not\equiv 1 \bmod M$ for each prime factor p of $M - 1$. This ensures that each generator has a full period or cycle-length. At each stage, the numbers produced by these three generators are combined in such a way as to generate realizations from the $\mathcal{U}(0, 1)$ distribution. The whole system is set-up in such a way as to ensure that the cycle length of this generator is the product of the individual cycle lengths, and is thus very long.

Each pair, U_i and U_{i+1} say, of independent $\mathcal{U}(0, 1)$ realizations produced are transformed into independent realizations, V_i and V_{i+1} say, from a $\mathcal{U}(-1, 1)$ variate. Next the following algorithm is employed to produce two independent normal variates from the two independent $\mathcal{U}(-1, 1)$ variates.

Algorithm 1 (Ripley[32], Algorithm 3.6 (polar))

1. *Repeat: Generate* V_i, $V_{i+1} \sim \mathcal{U}(-1, 1)$, *until* $W = V_i^2 + V_{i+1}^2 < 1$

2. *Let* $C = \sqrt{-2\, W^{-1} \log W}$

3. *Return* $X = C\, V_1$, $Y = C\, V_2$

Step 1 is a rejection method leaving (V_1, V_2) uniformly distributed in the unit disc. The X and Y so produced can be shown to be standard normal variates.

This algorithm performs well in as much as unwanted structure seemed not to be present. Other algorithms that we tried seemed to demonstrate such structure. For example, when we generated independent $\mathcal{U}(0, 1)$ variates by means of the congruential generator

$$X_i = (69069\, X_{i-1} + 1) \bmod 2^{32}, \qquad U_i = 2^{-32} X_i$$

(see Ripley[32], page 46), and transformed these into pairs of standard normal variates by means of the following algorithm

Algorithm 2 (Ripley[32], Algorithm 3.1 (Box-Muller))

1. *Generate* $U_1 \sim \mathcal{U}(0, 1)$, *set* $\Theta = 2\pi\, U_1$

2. *Generate* $U_2 \sim \mathcal{U}(0, 1)$, *set* $E = -\log U_2$, $R = \sqrt{2E}$

3. $X = R \cos \Theta$, $Y = R \sin \Theta$ *are independent standard normal variates*

we found that, if we displayed 65536 such standard normal variates on a 256 × 256 pixel grid by defining a pixel to be white if the appropriate variate was less than 0.0 and black otherwise, streaks appeared across the resulting image. Such problems did not occur with the uniform variates of Wichmann and Hill[45] and Algorithm 1. Moreover, Algorithm 1 does not require the calculation of the two trigonometric functions used by Algorithm 2 and is thus substantially faster, at the expense of a little extra complexity.

1.3 The prior distribution

Besag[4] points out that whereas the specification of $l(y \mid x)$ is in general relatively easy, being governed by physical considerations concerned with the sensing device for example, the specification of the prior is more of an art, and hence more difficult. The general aim is not to model the global features of the true image, but to try to capture some of the local characteristics. In this section we attempt to model the very commonly held belief that nearby pixels take similar values. First we must formalize the notion of 'nearby', and we do this by means of the concept of neighbourhood in Section 1.3.1. This concept of neighbourhood allows us to introduce locally dependent Markov random fields (or Gibbs distributions) in Section 1.3.2 as a way of modelling probabilistically the above mentioned commonly held belief about the set of images.

1.3.1 Neighbourhood systems

In this section we attempt to formalize the notion of 'nearby' by introducing the concept of neighbourhood system. We have seen that the true scene is observed on a grid of square pixels as illustrated in Figure 1.1. We can think of each pixel as a node of a finite graph, and we represent these nodes in Figure 1.1 by circles. Two pixels are said to be *neighbours* if and only if they are joined by an arc of the graph. In Figure 1.1 we illustrate both the first-order neighbourhood system (in the top left corner) and the second-order neighbourhood system (in the bottom right corner); the solid lines joining the pixels are the arcs of the graph, and thus represent the neighbourhood relationships. We can see from Figure 1.1 how the concept of neighbourhood relates to the notion of 'nearby'. A *clique c* is defined to be a set of pixels all of whose elements are neighbours. In Figure 1.2 we illustrate all the possible cliques corresponding to the neighbourhood systems illustrated in Figure 1.1. We denote the set of all pixel cliques in the image by C. Our definition has imposed symmetry in naming neighbours: that is, if pixel j is a neighbour of pixel i, pixel i must be a neighbour of pixel j. As a further piece of notation, we introduce ∂i to represent the set of neighbours of pixel i.

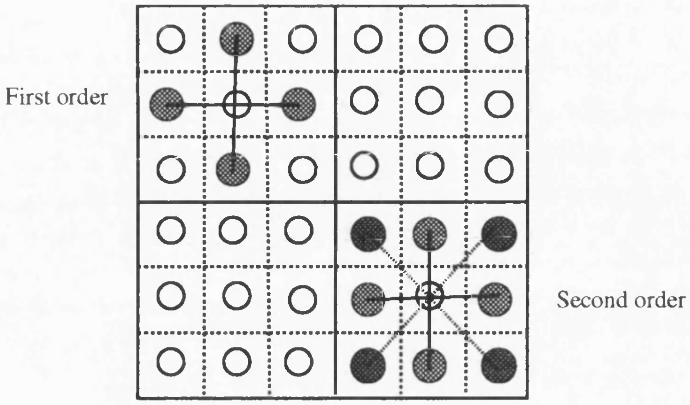


Figure 1.1: *The pixel grid showing the neighbourhood systems*

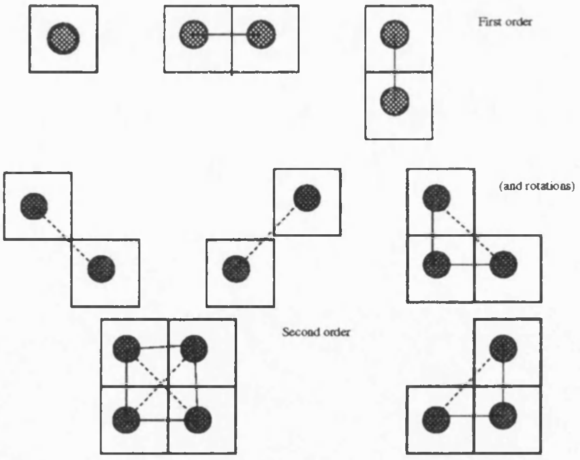


Figure 1.2: *The possible pixel cliques*

1.3.2 Markov random fields, the Hammersley-Clifford theorem and Gibbs distributions

To define the prior distribution $\Pr(x)$ we make the usual assumption that the true image x^* is a realization of a locally dependent Markov random field. Now that we have introduced the notion of neighbourhood we can define such a Markov random field. A *locally dependent Markov random field*, as defined by Besag[3], is a joint probability distribution on the set $\mathcal{X}$ of all possible images subject to the condition

$$\Pr(x_i \mid x_{S\setminus i}) = \Pr(x_i \mid x_{\partial i}),$$

where S is the set of all pixels. The condition, although specified locally, implies a global pattern and is known as the Markov condition. Geman and Geman[12] and Jubb[24] include the so-called positivity condition, namely

$$\Pr(x) > 0 \text{ for all } x \in \mathcal{X},$$

in their definition of a Markov random field; we shall do likewise throughout this thesis.

Even when each pixel has only a few neighbours, unobvious consistency conditions, given by the Hammersley-Clifford theorem (see, for example, Besag[2]), delimit the functions admissible as conditional probability distributions. The Hammersley-Clifford theorem also provides a connection between the purely graph-theoretic neighbourhood relationships on the lattice with the algebraic form of the distribution function. This theorem may be stated as follows:

Theorem 1 (Hammersley-Clifford) *Any probability distribution on the set of all possible images $\mathcal{X}$ which is a locally dependent Markov random field (satisfying the positivity condition) is of the form*

$$\Pr(x) = \frac{1}{Z}\exp\left\{-\sum_{c\in C} V_c(x)\right\} \tag{1.6}$$

where Z is a constant of proportionality and $V_c(x)$ is a function only of the x_i with i in clique c.

The distribution given by equation (1.6) is often referred to as a *Gibbs distribution* relative to the appropriate graph; see, for example, Geman and Geman[12]. The term $\sum_{c\in C} V_c(x)$ is often referred to as the *energy function*, and the family $\{V_c(x) \mid c \in C\}$ is called a *potential*. From equation (1.6) it is easy to write down the conditional distribution of x_i given the values at the other pixels:

$$\Pr(x_i \mid x_{S\setminus i}) \propto \exp\left\{ -\sum_{c\in C_i} V_c(x) \right\} \tag{1.7}$$

where C_i is the set of all cliques that include pixel i. This conditional distribution can thus be seen to involve only the values at pixels that are neighbours of pixel i, and hence the so-called Markov condition in the definition of a Markov random field is satisfied.

The prior distribution that we use throughout this part of the thesis is of the form specified by equation (1.6) although not all the cliques shown in Figure 1.2 are involved:

$$\Pr(x) \propto \exp\left\{ -\beta \left(\sum_{[i,j]} \phi(|x_i - x_j|) + D \sum_{<i,j>} \phi(|x_i - x_j|) \right) \right\}, \tag{1.8}$$

where $\beta \geq 0$ can be thought of as a smoothing constant, $\sum_{[i,j]}$ indicates summation over horizontal and vertical neighbours, $\sum_{<i,j>}$ indicates summation over diagonal neighbours and $\phi(u)$ is a function that we shall discuss in Section 1.3.3. If $D = 0$ then the model is said to be *first-order*, as only horizontal and vertical neighbours are involved in the specification of the prior, whereas if $D \neq 0$ diagonal neighbours are also involved and the model is said to be *second-order*. Jubb[24] presents a discussion about the choice of D with several illustrative examples. Our experience is that, from the visual quality of reconstructions produced, little advantage is to be gained by using a second-order model as opposed to a first-order model despite the greater cost of computation and overall complexity. Accordingly, we concentrate almost entirely on the first-order model, although almost all our techniques apply equally well to the second-order model. One example using a second-order model can, however, be found in Section 1.9.

All that we now need to do to specify the prior distribution is to define the function ϕ. We discuss this function in Section 1.3.3.

1.3.3 Classes of images and the ϕ-function

We consider two classes of images. The first comprises images whose pixels can take any of a finite number c of *unordered colours*. These colours can be thought of as labels referring to attributes of the pixels. In this case we set $\phi(u) = I(u \neq 0)$, where I is the indicator function, and the prior distribution takes the form

$$\Pr(x) \propto \exp\left\{-\beta\left(v^{(1)}(x) + D\,v^{(2)}(x)\right)\right\}, \tag{1.9}$$

where $v^{(1)}(x)$ is the number of discrepant first-order pairs in the image and $v^{(2)}(x)$ is the number of discrepant second-order pairs. This prior is sensitive only to the existence of differences between the values taken by pixels, rather than to the size of the differences. If the colours have a natural ordering, usually ranging from black to white, they are referred to as *grey-levels*. Often we assume that each pixel can take one of g grey-levels, where $g = 64$ or $g = 256$. For grey-level images a different ϕ, and hence a different prior distribution, is employed that does take account of the size of the difference in intensities between neighbouring pixels. We follow the approach of Jubb[24] and Geman and McClure[13]. These authors in effect use a family of ϕs, indexed by a parameter $\alpha > 0$. The function ϕ_α is defined as

$$\phi_\alpha(u) \quad = \quad 1 - \frac{1}{1+\alpha u^2} \quad = \quad \frac{1}{1+(\alpha u^2)^{-1}}. \tag{1.10}$$

When $u = |x_i - x_j|$, this ϕ_α can be thought of as a penalty for the discrepancy of the grey-levels taken by pixel i and pixel j. The general idea behind this family is that for large discrepancies, and hence large values of u, the value of $\phi_\alpha(u)$ is about the same. Thus, the problem of 'over penalizing' very large discrepancies which might occur as a result of a natural boundary between regions is avoided. In Chapter 2 for both classes of images we consider a modification to the prior distribution that employs an explicit edge process to model the boundary between regions. In Figure 1.3 we present a graph of $\phi_\alpha(u)$, $u \geq 0$, for five different values of α. It can be seen from the graph that $\phi_\alpha(0) = 0$ and $\phi_\alpha(u) \to 1$ as $u \to \infty$, for all values of α. Moreover, it is clear that as α increases the value of $\phi_\alpha(u)$ for fixed u increases. Jubb[24] explains that α determines the amount of variation that is permitted within regions; large values of α restrict variation. He presents some analysis that leads to a suggestion about the choice of the value of α. However, in our experiment we choose α (and β) by trial and error to give good reconstructions. In fact, in the reconstruction experiment that we present in Section 1.7 we set $\alpha = 0.075$ and $\beta = 2.5$. With such a value of α, and small u ($u \leq 3$ for example), $\phi_\alpha(u)$ looks like a quadratic in u. We note that with both $\phi(u) = \phi_\alpha(u)$ and $\phi(u) = I(u \neq 0)$ images x that maximize $\Pr(x)$ are of constant intensity.

Green[15], working in the context of single-photon emission computerized tomography

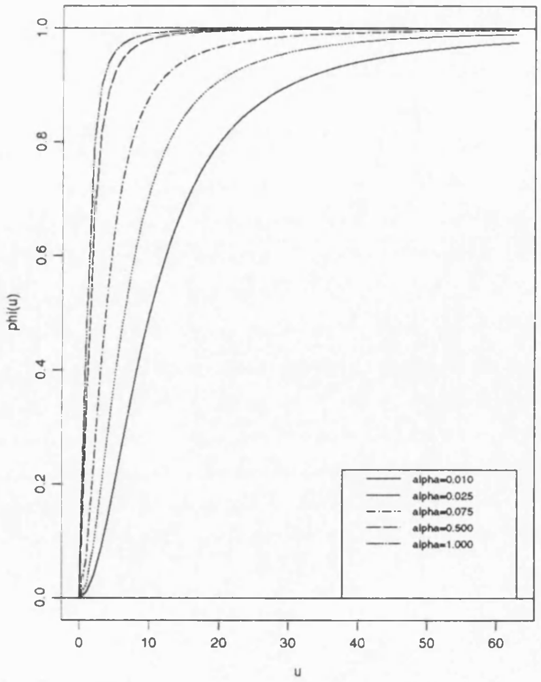


Figure 1.3: *The function ϕ_α for different values of α*

(SPECT), sets

$$\phi(u) = \psi_\alpha(u) = c_1 \log \cosh(c_2\, \alpha^{1/2} u) \tag{1.11}$$

where c_1 and c_2 are chosen to match the prior of Geman and McClure[13] in the sense that $\max \phi_1{}' = \max \psi_1{}'$ and $\phi_1{}''(0) = \psi_1{}''(0)$. With this choice of c_1 and c_2, $\phi_1(u)$ and $\psi_1(u)$ are very close for all u with $|u| < 1$. Again we have that $\psi_\alpha(0) = 0$. Green[15] explains that the ψ_α s yield quite a flexible family of prior distributions. In particular, if $\beta \rightarrow \infty$ and $\alpha \rightarrow 0$ in such a way that $\alpha\beta \rightarrow k$, then $\beta\, \psi_\alpha(u) \rightarrow k\, u^2$ for all u. The resulting prior is referred to as a Gaussian pixel prior by Besag[4]. He points out that such priors are unsatisfactory in the presence of real discontinuities, which they will smear out in a reconstruction experiment. If $\beta \rightarrow 0$ and $\alpha \rightarrow \infty$ such that $\alpha^{1/2}\beta \rightarrow k$, then $\beta\, \psi_\alpha(u) \rightarrow k\,|u|$. The resulting prior is referred to as a median pixel prior by Besag[4]. He points out that the resulting conditional distribution of x_i, given all other values (*i.e.*, given its neighbours), has its mode at the median rather than the mean of the neighbouring x_j s and hence performs better than the Gaussian pixel prior for reconstructing surfaces with discontinuities. In fact, Green[15] sets $\alpha = 1 / \sqrt{50} = 0.141$ and

$\beta = 0.2$.

Geman and Reynolds[14], working with blurred images, take

$$\phi(u) = \frac{1}{1+(\alpha u)^{-1}} \tag{1.12}$$

up to an additive constant. This ϕ shares many of the properties of Geman and McClure[13]'s ϕ, as given in equation (1.10), *e.g.* $\phi(0) = 0$ and $\phi(u) \to 1$ as $u \to \infty$; both priors belong to the general family

$$\phi(u) = \frac{1}{1+(\alpha u^{\gamma})^{-1}},$$

indexed by γ. Function (1.12) is concave, whereas the log cosh function (1.11) is convex for $u \in (0, \infty)$. Geman and Reynolds[14] note that the function (1.12) has strictly positive derivative (from above) at the origin, and explain in their Section 1.2 that this property together with concavity discourages the interpolation of a reconstruction towards the data. This is shown to be especially useful for the recovery of discontinuities in the presence of blurring. Convex functions considerably simplify the computational problem (see Besag[4] Section 4.1, or Green[15] Section V), but often lack this non-interpolating property.

Finally, we should point our that although the prior distributions that we consider are widely used in the literature, they are relatively simple and only attempt to model the general features of an image. More sophisticated priors that attempt to model more specific knowledge of the class of images under consideration are sometimes employed.

1.4 Summaries of the posterior distribution

Now that we have defined the likelihood in equation (1.5) and the prior distribution in equation (1.8) we can use Bayes's theorem, as expressed in equation (1.3), to combine our model for the noise with our prior knowledge of the true scene. Hence the posterior probability can be written as

$$\Pr(x \mid y) \propto \exp\left\{-\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 - \beta\left(\sum_{[i,j]}\phi(|x_i - x_j|) + D\sum_{<i,j>}\phi(|x_i - x_j|)\right)\right\}. \tag{1.13}$$

Inspection of equation (1.6) reveals that this posterior distribution is a Gibbs distribution relative to the same neighbourhood graph as the prior distribution. Thus, as we have seen in equation (1.7), the conditional distribution at each pixel, given the values at all the other pixels, can be computed easily from essentially local information. This fact is extremely useful when it comes to using the Gibbs sampler (Section 1.4.1), and the simulated annealing (Section 1.6.1) or ICM (Section 1.6.2) algorithms.

We try to summarize this posterior distribution by giving a single estimate of x. We discuss two possible summaries, the maximum *a posteriori* (MAP) estimate and the maximum posterior marginal (MPM) probability estimate, in Section 1.4.2. First, in Section 1.4.1 we discuss a way of sampling images $x \in \mathcal{X}$ from the posterior distribution $\Pr(x \mid y)$ known as the Gibbs sampler.

1.4.1 The Gibbs sampler

In this section we describe how samples can be generated from a given posterior distribution. The method that we shall describe was introduced in the context of image reconstruction by Geman and Geman[12]. It is, however, a special case of the method proposed and discussed by Hastings[19], which itself is a generalization of the famous Metropolis method. In fact, the Gibbs sampler can be used to generate images from any suitable distribution; in Section 3.4.3 we employ it to generate binary images from the prior distribution (1.9) with various values of β. We saw above that the posterior distributions that we use in image reconstruction are Gibbs distributions as defined in Section 1.3.2, and this motivates the use of the term 'Gibbs sampler' by Geman and Geman[12].

The idea is to produce a Markov chain (see, for example, Stander, Farrington, Hill and Altham[42]), with state space $\mathcal{X}$ and limit distribution $\Pr(x \mid y)$. After an initial (perhaps long) period in which the process settles down, a simulation of this chain produces a sequence of (dependent) images sampled from $\Pr(x \mid y)$. The actual implementation is simple and proceeds according to the following algorithm:

Algorithm 3 (Gibbs sampler)

1. *Produce an initial image (for example, assign a colour to each pixel at random)*

2. *Visit each pixel in the image by means of a raster scan and replace the current value by one sampled from the conditional distribution of the value at that pixel, given the current states of all the other pixels*

3. *Repeat many times*

Geman and Geman[12] show that asymptotically the choice of the initial image is not important. Of course, we must be able to compute the associated conditional distribution in Step 2. Because the distributions that we consider in image reconstruction are Gibbs distributions, this is an easy task. To sample from the conditional distribution in Step 2 we need to be able to generate numbers uniformly distributed between 0 and 1. We described an algorithm for doing this in Section 1.2.1.

1.4.2 The MAP and MPM summaries of the posterior distribution

In this part of the thesis we shall concentrate almost exclusively on the summary of the posterior distribution known as the maximum *a posteriori* (MAP) estimate. This is the image $x \in \mathcal{X}$ that maximizes $\Pr(x \mid y)$. In the context of decision theory, the MAP estimator corresponds to a zero-one loss function, according to whether the reconstruction is perfect or imperfect. Thus, all incorrect choices for the reconstruction are equally penalized. This has been cited as a criticism of the MAP estimate (see Besag[4]).

Another summary is the maximum posterior marginal (MPM) probability estimate. This is the image that maximizes the marginal posterior probability $\Pr(x_i \mid y)$ at each pixel i. In the context of decision theory, the MPM estimate corresponds to a loss function that counts the number of misclassified pixels. The MPM estimate can be found by sampling repeatedly from $\Pr(x \mid y)$ by means of the Gibbs sampler, and at each pixel storing a histogram of the values of x_i taken. The MPM reconstruction at pixel i is then the value corresponding to the mode of this histogram. Other appropriate summaries of the histogram can also be considered. In addition, in reconstructing a grey-level image, we can attach an approximate Bayesian confidence interval to each pixel, and with colour images we can assign a probability estimate, rather than single colours. Of course, as Besag[4] points out, such confidence interval or probability estimates must not be interpreted too rigorously because of the known defects of the prior.

1.5 The Bayesian formulation and the penalty function approach

We have seen in Section 1.4 that we try to summarize the posterior distribution by giving a single estimate of x, and that in this part of the thesis we concentrate upon the MAP estimate, *i.e.* the image $x \in \mathcal{X}$ that maximizes the right hand side of the posterior distribution given in (1.13). Equivalently, we can attempt to find the image $x \in \mathcal{X}$ that maximizes

$$-\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 - \beta\left(\sum_{[i,j]}\phi(|x_i - x_j|) + D\sum_{<i,j>}\phi(|x_i - x_j|)\right). \tag{1.14}$$

This is a penalized log-likelihood (see equation (1.5)): the logarithm of the likelihood of the image x given the record y is penalized by a term that measures the roughness of x, namely

$$\beta\left(\sum_{[i,j]}\phi(|x_i - x_j|) + D\sum_{<i,j>}\phi(|x_i - x_j|)\right).$$

It is more common, however, to consider the minimization of the following *penalty function*

$$\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left(\sum_{[i,j]}\phi(|x_i - x_j|) + D\sum_{<i,j>}\phi(|x_i - x_j|)\right) \tag{1.15}$$

over the set of all images $\mathcal{X}$. This penalty function is often referred to as the (posterior) *energy*, and thus we shall also refer to it as $E(x)$. It represents a trade off between infidelity of the reconstruction x to the data y, and the roughness of the image x. The balance of this trade off is in effect controlled by the unknown parameter $\beta \geq 0$. If, on the one hand, $\beta = 0$, the second or roughness term of (1.15) makes no contribution to the penalty function and the image that minimizes $E(x)$ is the one in which every pixel i takes on a value that is closest to its record y_i. This reconstruction, which uses no spatial information, will be referred to as the *maximum likelihood estimate.* If, on the other hand, β is infinitely large, the contribution of the first term of (1.15) becomes unimportant and the image that minimizes (1.15) is such that every pixel has the same value. We shall refer to the parameter β as the *smoothing parameter.* Often in reconstruction experiments in this part of the thesis we shall choose β by eye so as to give reconstructions that appear good. This is how we proceed in Section 1.7. Much research has, however, been done concerning the estimation of the smoothing parameter. We present a brief review of some of this in Section 1.8. In Chapter 3 we discuss methods of estimating β when the true image comprises only two colours.

From a philosophical point of view there is a difference between the β that appears in the prior (and posterior) distribution, and the β that appears in the penalty function. In the former β is a smoothing constant, whereas in the latter it is a smoothness parameter. We shall, however, not be rigorous in making this distinction in this thesis.

Our interest now turns to finding the global minimum of $E(x)$. Of course, we have not presented any theory to suggest that this global minimum will necessarily be a good reconstruction, in terms of the percentage of misclassified pixels, for example. (For further comments on the percentage of misclassified pixels as a measure of the quality of a reconstruction see Section 1.7.) Indeed some researchers feel that the MAP estimate can often give too much emphasis to the global properties of the prior distribution (see Besag[4]). We feel, however, that this minimization problem is worthy of consideration in its own right. In essence, this is the subject of Chapter 4. The global minimization of the penalty function (1.15) can, in theory, be achieved by a direct search over all c^n possible images, where c is the number of colours or grey-levels in the image, and n is the total number of pixels. In practice, however, for even moderate values of c and n such a search is not computationally feasible, and other techniques to minimize (1.15) have to be employed. We discuss these in Section 1.6.

1.6 Minimization techniques

We have seen that our aim is to find the image $x \in \mathcal{X}$ that corresponds to the global minimum of the penalty function $E(x)$ given by expression (1.15). In theory, the global minimum of $E(x)$ can be obtained by simulated annealing, as proposed by Geman and Geman[12]. We outline simulated annealing briefly in Section 1.6.1. It is not well known, however, how simulated annealing performs in practice. Accordingly, in Chapter 4 we present a thorough study of this stochastic optimization technique. Simulated annealing is very computationally expensive, and an alternative simple deterministic algorithm, known as iterated conditional modes (ICM) is often used. We outline this technique in Section 1.6.2. Both simulated annealing and ICM are iterative algorithms. In effect, at each iteration a new image is generated by visiting each pixel in turn and updating it in an appropriate fashion. The final image in this sequence is the reconstruction.

1.6.1 Simulated annealing

The basic idea behind simulated annealing is that, instead of sampling from the (posterior) distribution

$$\Pr(x \mid y) \propto \exp\{-E(x)\}$$

directly by means of the Gibbs sampler (Section 1.4.1) for example, we sample from a probability measure defined on $\mathcal{X}$ by

$$\pi_\tau(x) \propto \exp\left\{-\frac{E(x)}{\tau}\right\},$$

where $\tau > 0$ is a control parameter known as the *temperature.* In the limit, as $\tau \searrow 0$, $\pi_\tau(x)$ assigns unit probability to the MAP image. Thus, it is credible that if τ is decreased to zero sufficiently slowly during the sampling process, the MAP image should result. We discuss this in greater detail in Section 4.2.2. In essence the algorithm escapes from local minima by allowing changes that increase $E(x)$, as well as decrease it. As we shall see in Section 1.6.2 ICM only allows images that decrease $E(x)$ and so that technique does not permit an escape from a local minimum. The way in which τ is changed is known as the *temperature schedule.* Geman and Geman[12] state and prove a theorem confirming this convergence for a temperature schedule in effect of the form $\tau(t) \geq C / \log(1 + t)$, where C is some (possibly very large) constant that depends upon the function $E(x)$ to be minimized and $t \rightarrow \infty$ is the number of iterations of the image that have been started. Geman and Geman[12] show that this convergence does not depend upon the initial image used by the algorithm. This theorem is, however, an asymptotic result. In practice the algorithm can only be run for a finite time, thus giving an approximation to the MAP image. (Greig *et al.*[17] show that when the image comprises only two colours the global minimum of $E(x)$, sometimes referred to as the exact MAP estimate, can be found by means of a fluid flow algorithm. This is discussed further is Chapter 3.) In Chapter 4 we discuss the finite behaviour of the simulated annealing algorithm, suggest some modifications to the standard annealing algorithm, and examing different temperature schedules. We have already seen that simulated annealing is very computationally expensive. This is especially true when the number of possible values that can be taken at each pixel is high, such as is the case for grey-level images. Geman and Reynolds[14], working with these images, propose a slight modification to the standard annealing algorithm in order to reduce the computation required. They refer to this modification as the 'truncated algorithm'. When updating the value of the estimate of the image at pixel i, instead of sampling from the actual conditional distribution of x_i that puts positive weight on all g grey-levels, the support of the distribution is reduced to the values obtained by taking the union of small intervals (of five grey-levels in our case) about the current value at site i, the current values at the neighbours of i, and the data value y_i. We examine the truncated algorithm further in Section 4.5.

We finish this section by remarking that the x produced by the simulated annealing algorithm may not even correspond to a local minimum of the penalty function. In Section 1.6.2 we describe a simple modification that overcomes this defect. An example of the use of the truncated algorithm of simulated annealing with this simple modification is given in Section 1.7.

1.6.2 Iterated conditional modes

A very clear description of the ICM method can be found in Besag[3] or Besag[4]. ICM is a simple deterministic algorithm closely related to the Gibbs sampler. However, instead of sampling randomly at each stage, ICM selects the mode of the relevant conditional distribution. In our case this is equivalent to selecting at pixel i the value of x_i that minimizes

$$\frac{1}{2\kappa}(y_i - x_i)^2 + \beta \left\{ \sum_{j \in \partial i^{(1)}} \phi(|x_i - x_j|) + D \sum_{j \in \partial i^{(2)}} \phi(|x_i - x_j|) \right\}, \tag{1.16}$$

where $\partial i^{(1)}$ are the first-order neighbours of pixel i and $\partial i^{(2)}$ are the second-order neighbours, with the appropriately chosen ϕ-function. It can be easily shown that this procedure cannot decrease $\Pr(x \mid y)$—or increase $E(x)$—at any stage, and hence the method will converge to a local minimum of the penalty function $E(x)$. ICM is computationally inexpensive, usually requiring less than 10 iterations for convergence. Moreover, only pixels whose neighbours have changed since they were last visited need be considered in any given iteration, and this reduces computation even further. The reason for this can be seen from expression (1.16). If the neighbours of pixel i have not been changed since that pixel was last considered, then the minimization problem is unchanged and thus the value that achieves the minimum will also be unchanged. Such computational savings are especially important when dealing with grey-level images. However, ICM is sensitive to the choice of the initial image. Besag[3] suggests the use of the maximum likelihood estimate as the initial estimate, and we shall follow his suggestion in all our experiments. Jubb[24] examines the effect of different initial estimates, such as images all of whose pixels take the same value. He shows that, when the value of the smoothing parameter β is high, the ICM algorithm cannot move away from such initial estimates. We present an example of the ICM algorithm in action in Section 1.7.

We have seen that the ICM algorithm always converges to a reconstruction corresponding to a local minimum of the penalty function $E(x)$, whereas the simulated annealing algorithm need not do so. If we apply the ICM algorithm with the reconstruction that is produced by the simulated annealing algorithm as initial image, we will obtain a new reconstruction that not only corresponds to a local minimium, but that has a lower value of the penalty function, as ICM can only decrease $E(x)$. We shall see an example of such a procedure in Section 1.7. Our experience is that if a reasonable reconstruction is produced by the simulated annealing algorithm, the inclusion of ICM will have little effect from a visual point of view other than to remove speckle error. ICM can be thought of as zero-temperature simulated annealing to convergence. Thus, we can interpret the inclusion of ICM at the end of simulated annealing as the addition of some zero temperature steps to the temperature schedule.

1.7 A reconstruction experiment

We now attempt to illustrate simulated annealing and ICM by means of a reconstruction experiment. Although our main concern is with the value of the penalty function $E(x)$, as given by (1.15), obtained by the algorithms, we also present the reconstructions produced. In order to provide some indication as to the quality of the reconstruction, we give the percentage of misclassified pixels. Unfortunately, this can be misleading for several reasons. Ripley[33] illustrates one of them by an example which shows two reconstructions of a binary scene both with error rate 3.5%. One reconstruction, however, is visually much more acceptable than the other. The example that we shall consider is based on a grey-level image. In this case, there is another reason why the number of misclassified pixels fails to provide us with an accurate assessment of the similarity of the reconstruction to the true image: a pixel in the reconstructed image that has only a slightly different grey-level from that in the original image would be considered to be misclassified. In an attempt to overcome this problem, we present histograms of the grey-levels taken by a reconstruction.

We present the true image in the top two pictures of Figure 1.4. We consider this image again in Section 2.5.3 and in Section 4.5. It comprises 32×32 pixels, and four distinct regions based on the three grey-levels, 15, 30 and 45. The histogram gives the numbers of pixels taking these grey-levels, and we assume that there are $g = 64$ possible grey-levels. In general, fluctuation within regions is allowed, but this example does not exhibit this phenomenon, as our main interest is with the minimization of the penalty function.

Next the true image is corrupted by the addition of independent normal noise with mean 0.0

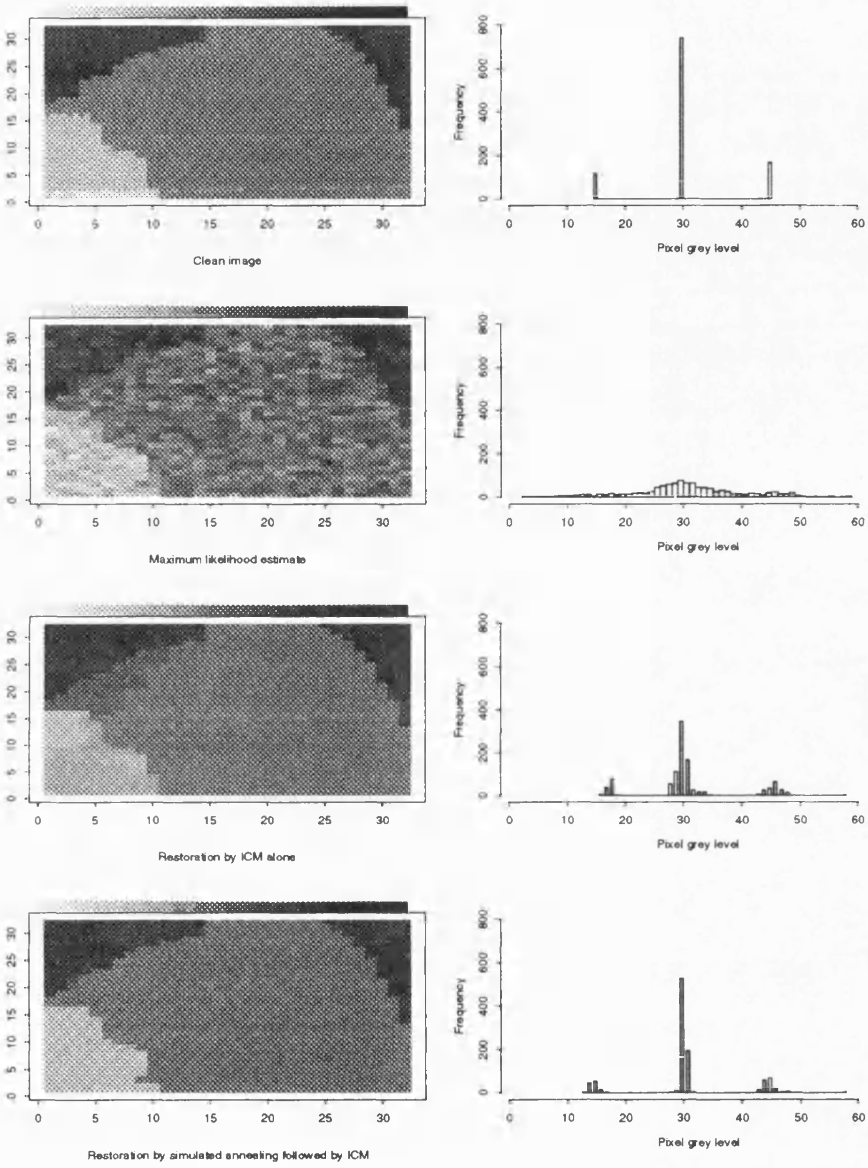


Figure 1.4: *A reconstruction experiment on a grey-level image*

and known variance $\kappa = 20.0$. Thus, if the grey-level of pixel i is x_i, the record at that pixel y_i is a random variable with distribution $\mathcal{N}(x_i, 20.0)$, independent of all other pixels. In the second two pictures of Figure 1.4 we present the maximum likelihood estimate of the true image: to each pixel i we assign the grey-level that is closest to the record y_i. The associated histogram shows how the number of grey-levels taken by the maximum likelihood estimate is greater that the number taken by the true image. In this maximum likelihood estimate there are 923 (90.14%) misclassified pixels.

The penalty function $E(x)$ that we now attempt to minimize is obtained from (1.15) by using the ϕ-function defined by equation (1.10). Accordingly, it takes the form

$$E(x) = \frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left\{\sum_{[i,j]}\phi_\alpha(|x_i - x_j|) + D\sum_{<i,j>}\phi_\alpha(|x_i - x_j|)\right\}, \tag{1.17}$$

where

$$\phi_\alpha(u) = \frac{1}{1 + (\alpha u)^{-1}}.$$

We take $\alpha = 0.075$ and $\beta = 2.5$, and consider a first-order model by setting $D = 0.0$. The true image has a value of $E(x)$ equal to 724.20, whereas the value of $E(x)$ for the maximum likelihood estimate is much higher, at 2671.39. The third set of two pictures shows the reconstruction obtained by applying the ICM algorithm starting from the maximum likelihood estimate. The ICM algorithm requires 9 iterations for convergence (no pixels are changed on the final iteration) and yields a reconstruction with 644 (62.89%) misclassified pixels and a value of $E(x)$ as given by (1.17) equal to 781.03. The appropriate histogram of Figure 1.4 shows three distinct regions corresponding to the grey levels 15, 30 and 45. Examination of the reconstruction itself shows basically the four regions of the true image but with some problems at the boundaries of the regions. Figure 1.5 shows how the penalty function $E(x)$ behaves when the ICM algorithm is applied. The initial value of $E(x)$ is given by the maximum likelihood estimate and is shown as the value at iteration 0. It can be seen that each iteration of ICM brings about a decrease in the penalty function. The horizontal line indicates the value of $E(x)$ given by the true image.

The bottom pair of pictures of Figure 1.4 shows one reconstruction obtained by the stochastic simulated annealing algorithm followed by ICM to convergence. We used 64 iterations of simulated annealing with a straight line temperature schedule as advocated by

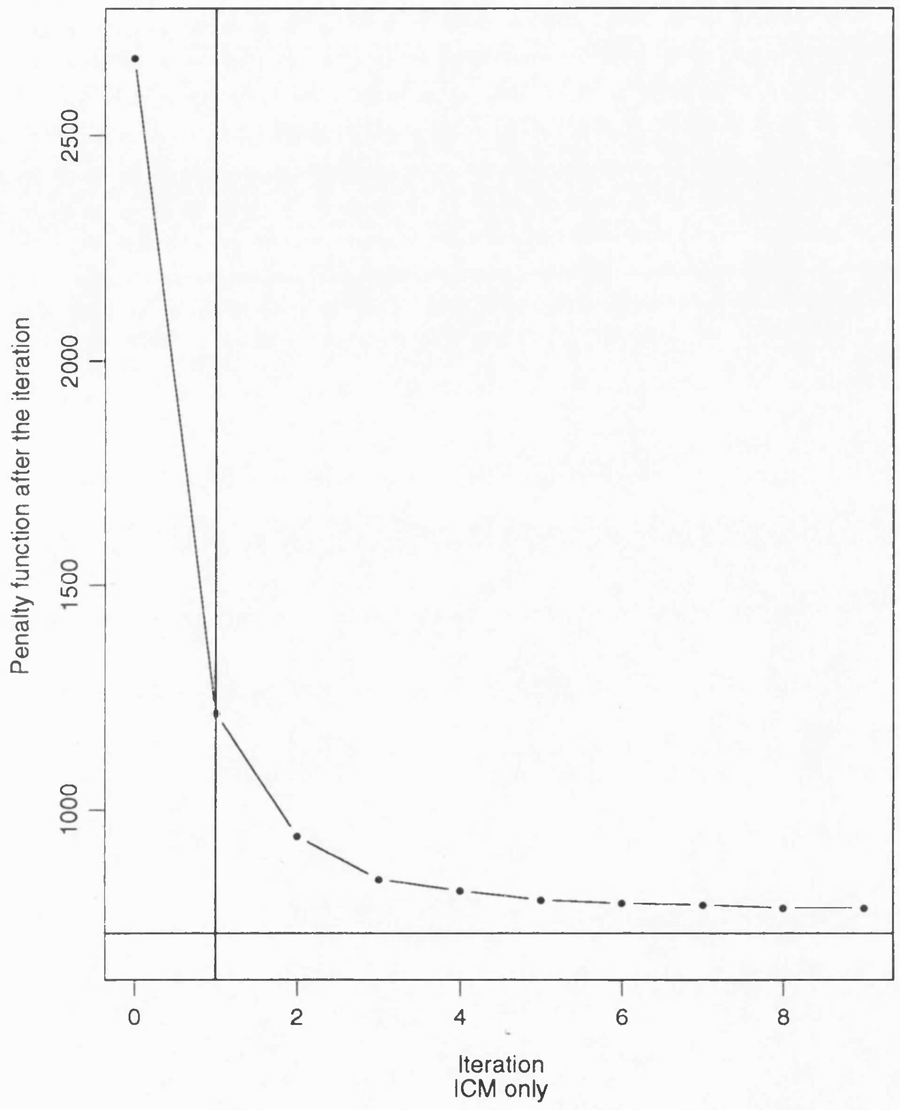


Figure 1.5: *The behaviour of the penalty function using ICM*

Geman and Reynolds[14] (for further details see Chapter 4). The temperature on the first iteration was 0.3, and the temperature on the final iteration was 0.05. Although for this reconstruction we used the truncated algorithm as described in Section 1.6.1, very similar results were obtained using the standard algorithm. The iteration of simulated annealing that yielded the lowest value of $E(x)$ as given by (1.17) at 721.41 was the final iteration. At this stage there were 296 (28.91%) pixels misclassified. Convergence to a local minimum from this image was achieved after 3 iterations of ICM (no pixels were changed on the final iteration, and 24 and 2 pixels were changed on the first and second iterations, respectively). The value of $E(x)$ was then 718.93 and 287 (28.03%) pixels were misclassified. Thus, the use of ICM at the end of the simulated annealing algorithm has reduced the penalty function, but only by a very small amount. This value of $E(x)$ is a little lower than the value of 724.20 achieved by the true image. Turning to the reconstruction itself, we see that the four regions are well reconstructed. The associated histogram is very similar to that of the true image. Figure 1.6, which should be contrasted with Figure 1.5, shows how the penalty function $E(x)$ behaves. The unbroken horizontal line again indicates the value of (1.17) achieved by the true image, while the broken horizontal line shows the value achieved by ICM alone. The left hand vertical line indicates the first iteration of simulated annealing, whereas the right hand vertical line indicates the first iteration of ICM. We note from Figure 1.6 the lack of monotonicity of the penalty function.

1.8 Estimation of the smoothing parameter

We have seen that the reconstruction process can be thought of as the minimization of a penalty function (1.15) that represents a trade off between infidelity of the reconstruction to the data and roughness controlled by a smoothing parameter β. This smoothness parameter β originates from the prior distribution $\Pr(x)$, which we shall write as $\Pr(x;\beta)$ in this section where this makes the argument clearer. In Section 1.7 and elsewhere in this part of the thesis we chose β by eye to give reconstructions that appear good. This method is especially appropriate when the reconstruction method is computationally inexpensive. An excellent example of such an approach when the image comprises just two colours is shown in Section 3.6. There, for a sequence of increasing βs, images that corresponds to the global minimum of (1.15) can be found relatively inexpensively. We refer to these images as exact MAP estimates. In the same chapter we go on to discuss automatic methods of estimating β for binary images. Jubb[24]

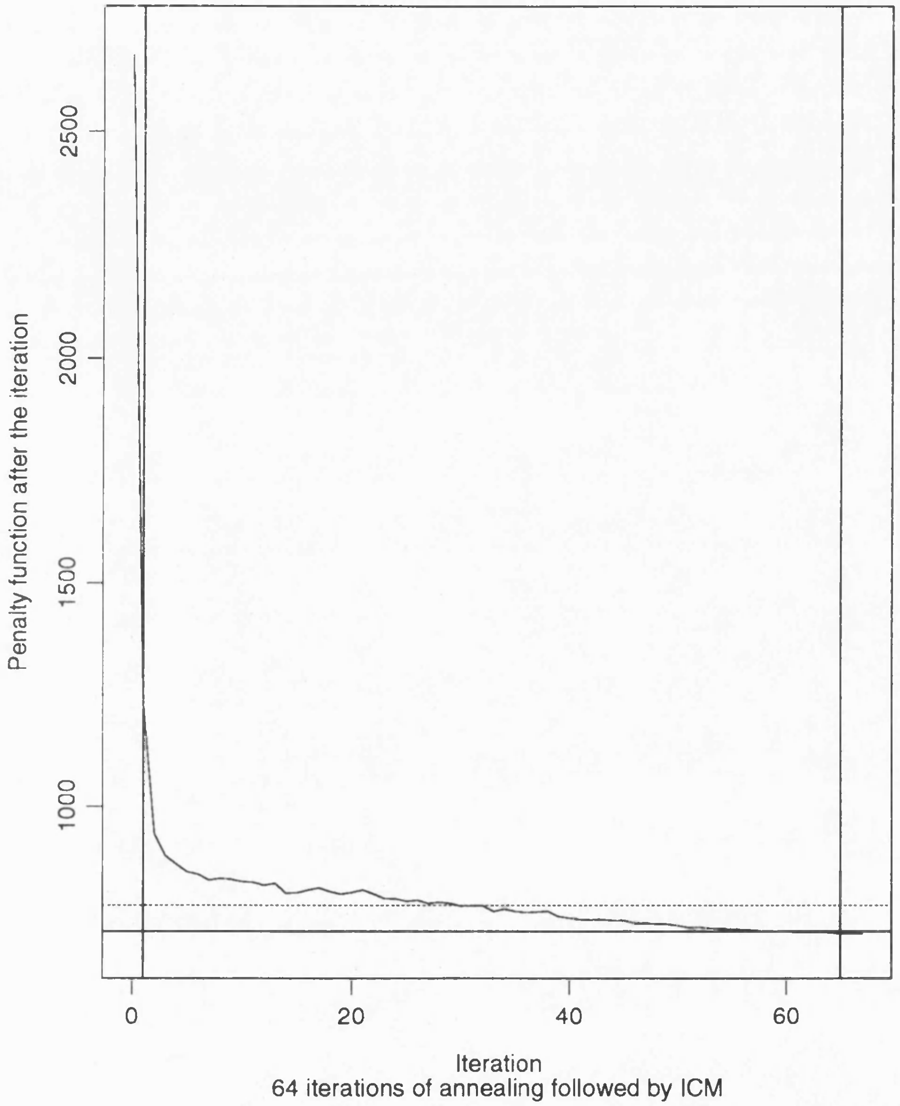


Figure 1.6: *The behaviour of the penalty function using simulated annealing followed by ICM*

also studied the problem of specifying β when considering the exact MAP estimate of binary images, but from a different angle. He observed that the exact MAP estimation is sensitive to small changes in the choice of β (see also Section 3.2), and thus in his Chapter 6 proposed a test for detecting oversmoothing. His method is based upon the difference, known as the residual, between the fitted value at pixel i and its record y_i. The basic idea is that if an area of colour has been obliterated as a result of oversmoothing then the residuals at those pixels that have been misclassified are expected to be larger than those occurring at pixels that have been correctly classified. Such 'informative' residuals are identified and their groupings investigated. Frigessi and Piccioni[11], again working with binary images, outlined a method for choosing β when the degradation method is the binary channel. We describe that method in Section 3.7.

Besag[3] outlines a method for general images to estimate β during ICM. We describe a slight modification of that method in Section 3.9. The method is based on maximizing the *pseudo-likelihood*

$$\prod_{i=1}^{n} \Pr(x_i \mid x_{\partial i}; \beta)$$

over β, where for example x is the current ICM reconstruction. (Sometimes the boundary pixels are not included in the product.) This pseudo-likelihood method is, according to Besag[3], a neater and more efficient variation of the 'coding method' (see Besag[2] or Cross and Jain[7]). The coding method is based on the product

$$\prod_{i \in M} \Pr(x_i \mid x_{\partial i}; \beta)$$

where M is a (maximal) set of pixels such that no two are neighbours. As Jubb[24] explains the idea is that, under the assumptions of the model, the colourings of each of the pixels in any such set, given the colouring of all the other pixels, are conditionally independent of one another, and thus maximum likelihood estimates can be obtained from the conditional likelihoods. The different coding estimates for β are then combined in a suitable way.

In Section 1.8.1 we present a brief review of a recent paper by Thompson, Brown, Kay and Titterington[43]. This paper, although not concerning itself with simulated annealing or ICM, describes a thorough study of some other methods for choosing the smoothing parameter in image reconstruction.

1.8.1 A review of Thompson *et al.*

Thompson *et al.*[43], working mainly with grey-level images, consider the model given in equation (1.2) that includes symmetric blurring by means of an $n \times n$ known point-spread matrix H (which may be the identity). The distribution of the noise vector ε is again considered to be $\mathcal{N}(0, \kappa I)$ where I is the $n \times n$ identity matrix. The variance $\kappa > 0$ is possibly unknown in this case.

Thompson *et al.*[43] proceed as we do by considering the minimization of the penalty function

$$\|y - Hx\|^2 + \lambda \Phi(x),$$

where $\|z\|^2 = z^T z$ and $\Phi(x) = x^T Cx$, C being a prescribed non-negative definite matrix. They think of the function $\Phi(x)$ as a roughness penalty and $\lambda \geq 0$ as a smoothing parameter. The estimate of x, denoted by $\hat{x}(\lambda)$, is thus the solution of

$$\min_x \left\{ \|y - Hx\|^2 + \lambda x^T Cx \right\}, \tag{1.18}$$

and therefore takes the form

$$\hat{x}(\lambda) = (H^T H + \lambda C)^{-1} H^T y. \tag{1.19}$$

Thompson *et al.*[43] discuss four basic ways of choosing the smoothing parameter λ to be used in equation (1.18) which we now outline:

TPMSE The first approach is the *minimization of total predicted mean square error*. The idea is to choose λ in such a way that, on average, the mean of the observed data would have been most closely predicted. In algebraic terms, this involves choosing λ to minimize

$$\text{TPMSE}(\lambda) = \mathbf{E} \left\{ \|H(x - \hat{x}(\lambda))\|^2 \right\},$$

where the expectation is taken with respect to the probability distribution of the error ε. Unfortunately, there are drawbacks with this approach, not least of which is the fact that the optimal λ in this sense, $\hat{\lambda}_{TP}$, is a function of the unknown x, and of κ. The authors use $\hat{\lambda}_{TP}$ and the estimate of x derived from it mainly for comparisons with the other, more practical methods.

GCV The next approach is a version of λ_{TP} that is completely data based and is referred to as the *generalized cross-validatory choice of λ.* The generalized cross-validatory choice $\hat{\lambda}_{GCV}$ is the minimizer of

$$GCV(\lambda) = \frac{\mathrm{RSS}(\lambda)}{[\mathrm{trace}\{I - K(\lambda)\}]^2},$$

where

$$\mathrm{RSS}(\lambda) = \|y - H\hat{x}(\lambda)\|^2 \tag{1.20}$$

and

$$K(\lambda) = (I + \lambda Q)^{-1} \tag{1.21}$$

where

$$Q = (H^T)^{-1} C H^{-1}.$$

CHI The third approach is referred to as the *χ^2 choice of λ.* In this, $\hat{\lambda}_{CHI}$ is the solution of

$$\mathrm{RSS}(\lambda) = n\,\kappa.$$

The solution is not explicit and requires an estimate for κ, or the true value itself.

EDF The final approach is the *equivalent degrees of freedom choice of λ.* Here $\hat{\lambda}_{EDF}$ satisfies

$$\frac{\mathrm{RSS}(\lambda)}{\mathrm{trace}\{I - K(\lambda)\}} = \kappa$$

where RSS(λ) is defined in equation (1.20) and $K(\lambda)$ is defined in equation (1.21).

One of the measures often used to assess a reconstruction $\hat{x}(\lambda)$, based on λ, is the total mean-squared error, or any quantity proportional to it. The authors consider the quantity

$$\sum_{i=1}^{n} (\hat{x}_i(\lambda) - x_i)^2, \tag{1.22}$$

where x_i represents the true grey-level at pixel i. In fact, they use the λ that minimizes the expression (1.22) as a mark against which to judge the λs produced by the other methods that they examine. The values of (1.22) are presented and discussed for the various λ s.

After discussing some theoretical results about the above choices for λ, the authors consider variations and computational short-cuts. For example, they show how the use of a preliminary eigen-analysis reduces the computation required for determining the choices of smoothing parameter and for reconstructing the image. Even so, some approximations have to be made for computational feasibility, but it is stated that little is lost by such approximations. These authors do not consider iterative algorithms such as simulated annealing and ICM. In addition, four estimators of κ are introduced. These may be needed in the CHI and EDF methods for choosing the smoothing parameter λ. We do not discuss these estimators here. Thompson *et al.*[43] present an extensive simulation study based on four grey-level images, three different types of blurring of size 3×3, 7×7 and 15×15 (see their Section VI for the precise definitions), and the addition of independent Gaussian noise having two different variances $\kappa = 1$ (low noise) and $\kappa = 100$ (high noise). The results presented are based on 1000 independent realizations of the noise process. The roughness penalty, $\Phi(x) = x^T Cx$, was taken as the quadratic

$$\Phi(x) = \sum_{i=2}^{m-1} \sum_{j=2}^{m-1} \left\{ (x_{i,j} - x_{i-1,j})^2 + (x_{i,j} - x_{i,j-1})^2 \right\}$$

throughout. Space does not allow us to discuss these results here, but the main conclusions are clearly stated and can be summarized as follows:

- The TPMSE method, not surprisingly, on average produced the best recovered image. However, it is totally impractical for any real situation as it requires *a priori* knowledge of the true image.
- The totally data based GCV method usually performs well. It can, however, fail catastrophically in some circumstances, producing a grossly underestimated smoothing parameter that may even be negative.
- The χ^2 method tends to overestimate the value of the smoothing parameter, producing stable but distorted images.
- Provided a good estimator of κ is available, the EDF method provides a good data-based choice of the smoothing parameter. Problems do, however, occur with this method in the presence of large blurring.

1.9 Estimation of the variance

Often the variance κ of the added noise is known from the physics of the set-up, or can be estimated accurately from training data. If this is not the case, the estimation of κ can be included as part of the reconstruction procedure. This is generally an easier task than the estimation of the smoothing parameter β.

As stated in Section 1.8.1, Thompson *et al.*[43] propose some ways of estimating κ, but they do not use iterative algorithms such as simulated annealing or ICM. We follow the approach outlined by Besag[3] in his Section 5.1.2. He suggests that κ is estimated by maximizing the likelihood $l(y \mid \hat{x}) = l(y \mid \hat{x}; \kappa)$ over κ at the start of each iteration of the algorithm, where $\hat{x}$ is the current estimate of the image. We now illustrate this procedure, which we have found to work well, by an example. The true image comprises 64×64 pixels and has $c = 3$ unordered colours. We present the true image in the top left picture of Figure 1.7. This is the same image that we use in Section 2.5.2. We add independent Gaussian noise with variance $\kappa = 1.0$ to the true image. The maximum likelihood estimate is shown in the top right picture of Figure 1.7. It is easy to see that the image has been heavily distorted. (In Section 2.5.2 the distortion is less as there $\kappa = 0.5$; see Figure 2.5.) We now attempt to reconstruct the image using ICM alone. We use the appropriate prior distribution given in equation (1.9): on this occasion we use a second-order model, with downweight $D = 1/\sqrt{2}$, and we set $\beta = 1.5$. In the bottom left hand picture of Figure 1.7 we see the reconstruction obtained under the assumption that the variance κ of the noise is known. This reconstruction was obtained after 6 iterations of ICM (no pixels were changed on the final iteration), and 490 (11.96%) pixels were misclassified. In the bottom right hand picture of Figure 1.7 we see the reconstruction obtained under the assumption that the variance κ of the noise is unknown. The variance is estimated at the start of each iteration by maximizing $l(y \mid \hat{x}; \kappa)$ over κ, where $\hat{x}$ is the current estimate of the image, the initial estimate of the image being the maximum likelihood estimate shown in the top right picture of Figure 1.7. The ICM algorithm converged after 16 iterations (no pixels were changed on the final iteration), and 479 (11.69%) pixels were misclassified. Visually, there is little to choose between the two reconstructions, both of which are quite poor!

In Figure 1.8 we present a graph showing the estimate of the variance κ used for each iteration of ICM. The estimate of κ based on the maximum likelihood estimate of the image was 0.48. This rises quickly until about iteration 6 when it settles down (very few pixels are changed by further iterations) at around 1.0, the value of the true variance which we indicate

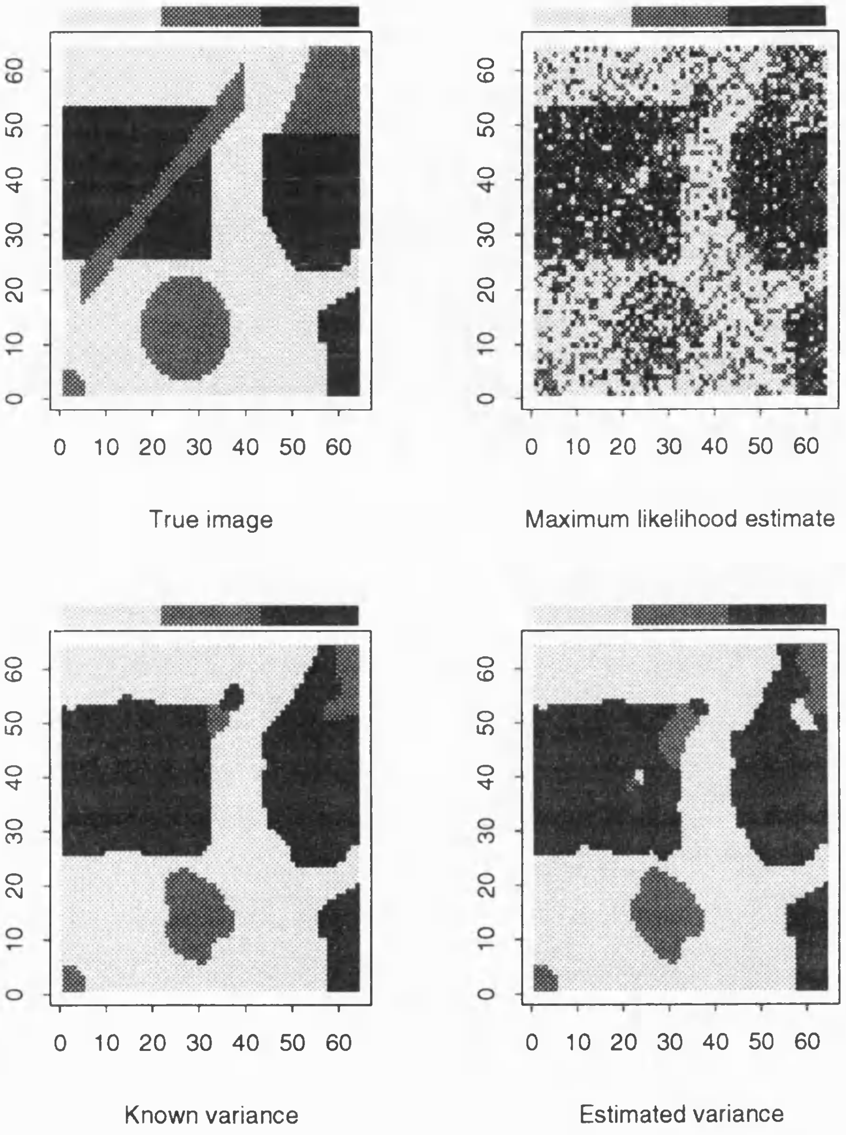


Figure 1.7: *A reconstruction experiment of a image with $c = 3$ colours showing the effect of estimating the variance κ*

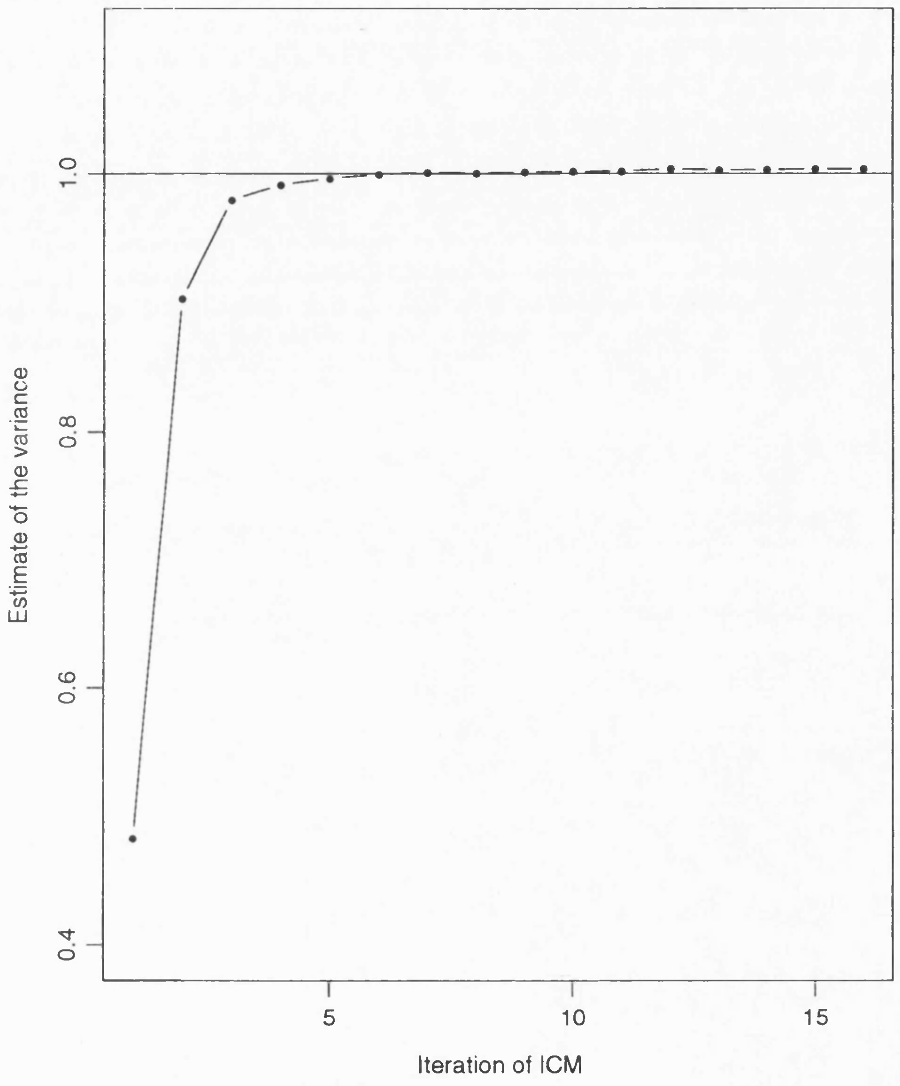


Figure 1.8: *The estimate of the variance κ*

by the horizontal line. The final estimate of κ is slightly above 1.0.

We end this section by noting that the method we have illustrated for estimating κ can be used in conjunction with the simulated annealing algorithm. For both ICM and simulated annealing little is known about the convergence properties of the resulting procedures. However, we have never come across a case when incorporating parameter estimation has affected the convergence of ICM.

Chapter 2

Edge Processes in Statistical Image Reconstruction

2.1 Introduction

Our aim in statistical image reconstruction is to produce a reconstruction of a *true* image from a record or signal. We assume that the true image is a discretized version of an underlying *real* image according to a given pixel grid, upon which the signal is observed and the reconstruction is attempted. Generally in this chapter we assume that all the pixels in the grid are square and of equal size, but, as we shall see, it is not necessary to make this assumption. In Chapter 1 we saw how the general Bayesian paradigm is used in statistical image reconstruction, and we review the basic idea here.

In Section 1.2 we explained how we can write down the likelihood of the record, given an image. The likelihood can be combined with a prior distribution on the set of all possible images by means of Bayes's theorem to give a posterior distribution upon which the reconstruction is based. For the reconstruction we need to produce a summary of the posterior distribution. In this part of the thesis we shall concentrate almost exclusively on the summary known as the maximum *a posteriori* (MAP) estimate, namely the image that maximizes the posterior distribution. In Section 1.3 we discussed the form of possible prior distributions $\Pr(x)$, where x is a vector specifying the *pixel process*, namely the values taken by all pixels of the image. The aim was to model the commonly held belief that nearby pixels take similar values. We formalized the notion of 'nearby' by introducing the concept of neighbourhood; all pairs of horizontally or vertically adjacent pixels were defined to be first-order neighbours, and all pairs of diagonally adjacent pixels were defined to be second-order neighbours. This concept of neighbourhood enabled us to define prior distributions $\Pr(x)$ on the pixel process that are locally dependent Markov random fields, or Gibbs distributions. In this chapter we do not consider second-order neighbours.

The most probable images under $\Pr(x)$ are those of constant intensity; the rougher the image, the less probable it is, where roughness depends upon all pairs of discrepant neighbouring pixels (and, with grey-level images, the size of the discrepancy) in the image. However, it is not unreasonable to assume that the underlying real image (and hence the true image) is made up of regions over each of which its behaviour is not subject to abrupt change, but from one region to another large differences in behaviour may occur. Consider, for example, a satellite picture of an area of the earth's surface. In the picture there are some sharp transitions (*e.g.* from one field to another) and some smooth changes (*e.g.* from one end of a field to another). Geman and Geman[12] use edges in their set-up to allow for images containing separate objects, but for which the objects are not completely uniform. Their modification attempts to model the boundaries in the real image by introducing edges that are either present or absent at each of a set of edge sites that are determined by the pixels. The state of all the edge sites defines the discretized *edge process*. Geman and Geman[12] think of an image as comprising both the pixel process and the edge process. In this chapter only we shall refer to the set of such images as $\mathcal{X}$. In the presence of an edge process the neighbourhood structure of the pixel process is changed according to the rule that says if two pixels are separated by an edge they are no longer neighbours. Thus, by not including all pairs of horizontally or vertically adjacent pixels in the penalty term of the pixel process, we can attempt to model images that contain separate objects, but for which the objects are not completely uniform.

In Section 2.2 we outline in more detail the approach of Geman and Geman[12]. We discuss the edge process in detail. This can be completely specified by assigning penalties to the six possible ways (up to rotation) in which four edge sites meeting at a point can be occupied by edges. Many authors have assigned these penalties in an *ad hoc* fashion. Silverman, Jennison, Stander and Brown (Silverman *et al.*)[39] approach the problem in a more systematic way and use various geometrical insights to develop a method of penalizing the discretized edge process that has genuine meaning in terms of the properties of the underlying boundary pattern of the real image, namely the total edge length and the overall complexity of the boundary. In Section 2.3 we discuss this approach further and illustrate these two properties by means of simple examples. We show for these examples how the penalties of Silverman *et al.*[39] perform better than those suggested by other authors such as Geman and Geman[12]. In Section 2.4 we briefly outline the way in which Silverman *et al.*[39] derive their edge penalties and describe a minor modification that proves useful when image reconstruction is considered.

The present author was responsible for the majority of the work of Section VI of Silverman, Jennison, Stander and Brown[39] in which a reconstruction experiment on an image of $c = 3$ colours on a 64×64 grid of square pixels is reported. Here, in Section 2.5, we present that experiment in greater detail, along with another based on an image with $g = 64$ grey-levels. These two experiments allow us to compare the edge penalties proposed by Silverman *et al.*[39] with those proposed by Geman and Geman[12]. We also discuss how reconstructions that employ edge processes compare with those that do not. In Section 2.6 we present our conclusions.

2.2 The approach of Geman and Geman

We have seen that Geman and Geman[12] use edges in their reconstruction algorithm to allow for images containing separate objects that are not completely uniform. We now describe the approach adopted in [12] in more detail; we use the notation of that paper as far as possible. We represent a possible true image by $x = (f, e)$, where f is the vector of pixel values and thus defines the pixel process, and e denotes the vector giving the state of all the edge sites and thus defines the edge process. We shall write $X = (F, E)$ for the associated random variable. Geman and Geman[12] now represent the joint distribution for $X = (F, E)$ as

$$\Pr(X = x) \propto \exp\left\{-U^{(F,E)}(f, e)\right\},$$

where the energy $U^{(F,E)}(f, e)$ is the sum of the two terms $U^{(F\,|\,E)}(f\,|\,e)$ and $U^{(E)}(e)$, both of which can be computed using local information only. The term $U^{(F\,|\,E)}(f\,|\,e)$ takes the same form as the energy function for the prior distribution used in Chapter 1, as defined in expression (1.8), except that the sum is over those cliques that remain, given the edge process. In other words, if d is the edge site between pixels r and s, the sum only includes the pixel cliques that involve pixels r and s if $e_d = 0$, *i.e.* if no edge is present at d. If $e_d = 1$, *i.e.* if an edge is present at d, then there is no contribution to the sum from the pixel clique that involves pixels r and s. The

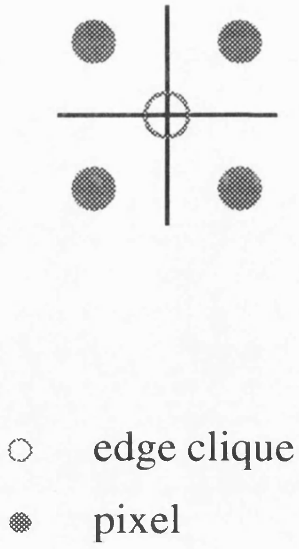


Figure 2.1: *An edge clique*

penalty for the edge process is such that

$$U^{(E)}(e) = \sum_{c \in C^{(E)}} V_c^{(E)}(e), \tag{2.1}$$

where $C^{(E)}$ is the set of edge cliques, which is as yet unspecified. By definition an edge clique is a set of edge sites all of whose elements are neighbours. Therefore, to specify the set of edge cliques $C^{(E)}$, we need to define what it means for two edge sites to be neighbours. Two edge sites are said to be *neighbours* if they meet at a point. Thus an edge clique comprises the four edge sites meeting at a point, or any subset thereof. We illustrate an edge clique in Figure 2.1. As usual, each $V_c^{(E)}(e)$ depends only on the elements of e that are in edge clique c.

We take $V_c^{(E)}(e)$ to be non-zero only when c is an edge clique comprising all four edge sites. Such a clique of size four can be occupied by edges in six distinct ways up to rotation. We illustrate the six possible ways in Figure 2.2[1]. We may think of $V_c^{(E)}(e)$ as the penalty for the way in which edge clique c is occupied by edges. We shall write V_j for the penalty given to an edge configuration of type j, as shown in Figure 2.2. Geman and Geman[12] assign the values of V_j as follows: if *no edges* are present in a clique, $V_0 = 0$; if one edge is present giving

[1] We are grateful to Dr Glenn Stone and Guy Nason for producing this figure.

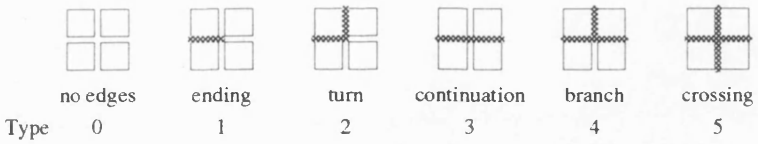


Figure 2.2: *The six possible ways in which edge sites can be occupied*

		Geman and Geman		Murray		Marroquin	
j	Type	V_j	Rank	V_j	Rank	V_j	Rank
0	no edges	0.0		0.0		0.0	
1	ending	2.7	1	2.2	1	2.0	1
2	turn	1.8	3	1.2	4	0.8	4
3	continuation	0.9	5	1.2	4	0.25	5
4	branch	1.8	3	1.4	3	1.2	3
5	crossing	2.7	1	2.2	1	2.0	1

Table 2.1: *The edge penalties used by various authors*

an *ending*, $V_1 = 2.7$; if two edges are present in the form of a *turn*, $V_2 = 1.8$; if two edges are present in the form of a *continuation*, $V_3 = 0.9$; if three edges are present giving a *branch*, $V_4 = 1.8$; if all four edges are present giving a *crossing*, $V_5 = 2.7$. These penalties seem to have been selected in a somewhat arbitrary fashion by reference to experiments. Other authors seem to select their penalties in a similar way. Wright[46] uses the penalties proposed by Murray, Kashko and Buxton[30]. These are said to work better than the penalties proposed by Geman and Geman[12], although both papers consider images comprising regions with boundaries that are only horizontal and vertical lines with respect to the pixel grid. Marroquin[29] uses different penalties again. We summarize all these penalties in Table 2.1. There seems to be general agreement that endings and crossings should receive the highest penalty, and that branches should receive the next highest penalty. Next come turns, and continuations are the least heavily penalized by all these authors. The fact that here turns always receive a penalty at least as large as continuations will be discussed further in Section 2.3 and Section 2.5. We note that with the penalties proposed by Geman and Geman[12] turns receive the same penalty as branches, whereas with the penalties proposed by Murray[30] turns receive the same penalties as continuations. Silverman *et al.*[39] approach the problem of choosing these penalties in a more systematic way, and use various elementary geometrical insights to develop a method of penalizing the discretized edge process that has genuine meaning in terms of the properties of the underlying real boundary pattern. We shall discuss their approach in Section 2.3. Another relevant article, which complements Silverman *et al.*[39] by providing among other things more details of some of the arguments only outlined in that paper, is Jennison and Silverman[20].

2.3 The approach of Silverman *et al.* to the specification of edge penalties

In this section we discuss the approach of Silverman *et al.*[39] to the specification of edge penalties. Silverman *et al.*[39] think of the edge process as being a discretized version of the boundaries of a real image, and aim to select their edge penalties so that the overall edge penalty, namely $\sum_{c \in C^{(E)}} V_c^{(E)}(e)$, depends, as far as possible, upon the boundaries of the real image and not upon the discretization. Their approach is based on two criteria concerning the boundaries of the real image, the total edge length and overall complexity of the boundary. In Section 2.3.1 we examine these two criteria further.

2.3.1 The two criteria of Silverman *et al.*

The first criterion is that the overall edge penalty should, as far as possible, be representative of the total edge length in the real image and, therefore, be independent of the discretization. In particular, the edge penalty should not depend upon the angle at which the pixel grid is placed on the real image. We illustrate this by means of an example given in Figure 2.3. We assume in this illustration that the real image is a square as shown in the first picture of Figure 2.3. In the second picture of Figure 2.3 we assume that the square has been observed in such a way that the pixel grid lies directly on top. The discretized edge process corresponding to this image is then exactly those pixel edges that correspond to the edges of the square. We shall refer to the edge process shown in the second picture of Figure 2.3 as (i). However, it may be the case that the square has been observed in such a way that the pixel grid does not lie directly on top but has been rotated through some angle. We show an example of this in the third picture of Figure 2.3. This time the edge process, discretized by consideration of the dual pixel process as described in Section 2.4.1, is shown in the fourth picture of Figure 2.3. We shall refer to that

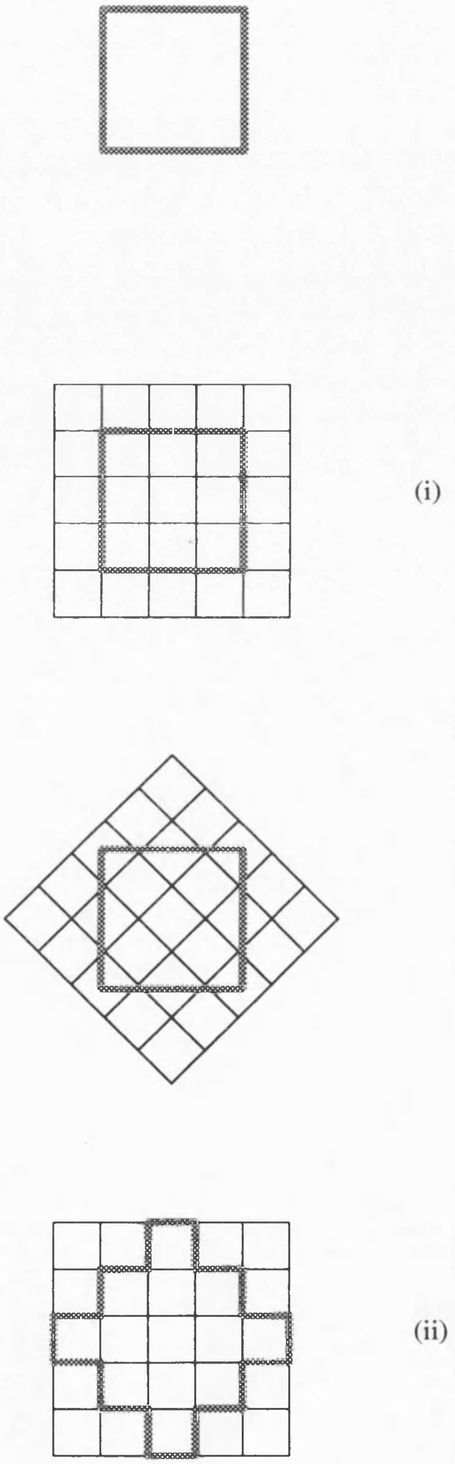


Figure 2.3: *The overall penalty should not be affected by the rotation of the pixel grid*

edge process as (ii). Edge process (i) comprises eight continuations and four turns and would receive a penalty of 14.4 (14.4, 5.2) if we were to use the penalties proposed by Geman and Geman[12] (Murray[30], Marroquin[29]). Edge process (ii) comprises twenty turns and would receive a penalty of 36.0 (24.0, 16.0). Thus, with these penalties the same real image receives very different overall penalties depending upon the discretization. We shall see below that the penalties proposed by Silverman *et al.*[39] give very similar penalties for the edge processes (i) and (ii).

The second criterion is that the overall edge penalty should, as far as possible, depend upon the complexity of the scene. Again, we illustrate this criterion with an example given in Figure 2.4. We see from Figure 2.4 that the first edge process comprises four regions and has total edge length equal to the length of 12 pixels. The second edge process also has total edge length equal to 12 pixels, but is far less complicated having only one region. The first edge process comprises four turns, four branches and a crossing and would receive a total penalty of 17.1 (12.6, 10.0). The second edge process comprises four turns and eight continuations and would receive a total penalty of 14.4 (14.4, 5.2). Although the overall penalty for the first edge process is greater than the overall penalty for the second edge process, the difference does not seem to reflect the far greater complexity of the first edge process in a systematic way. The penalties proposed by Silverman *et al.*[39] attempt to take into account the complexity of the scene by making $\sum_{c \in C^{(E)}} V_c^{(E)}(e)$ relate to the number of regions, as well as the length of the boundary in the real image before discretization. The aim is to have

$$\sum_{c \in C^{(E)}} V_c^{(E)}(e) = (\beta \times \text{total boundary length}) + (\rho \times r) \tag{2.2}$$

where β is the desired penalty per unit length of edge, ρ is the desired penalty per region of the pattern, and r is the number of regions in the underlying real image before it has been discretised. The penalties proposed by Silverman *et al.*[39] for square pixels of gauge h are given in Table 2.2. We set $h = 1$ from now on in this section. In Section 2.4.1 we shall review Silverman *et al.*[39] and briefly outline how these penalties are obtained. We note for now that continuations are more heavily penalized than turns, whereas with the other authors the contrary was true.

Finally in this section we return to our two examples. For the first example, as illustrated in Figure 2.3, the value of $\sum_{c \in C^{(E)}} V_c^{(E)}(e)$ obtained using the penalties proposed by Silverman

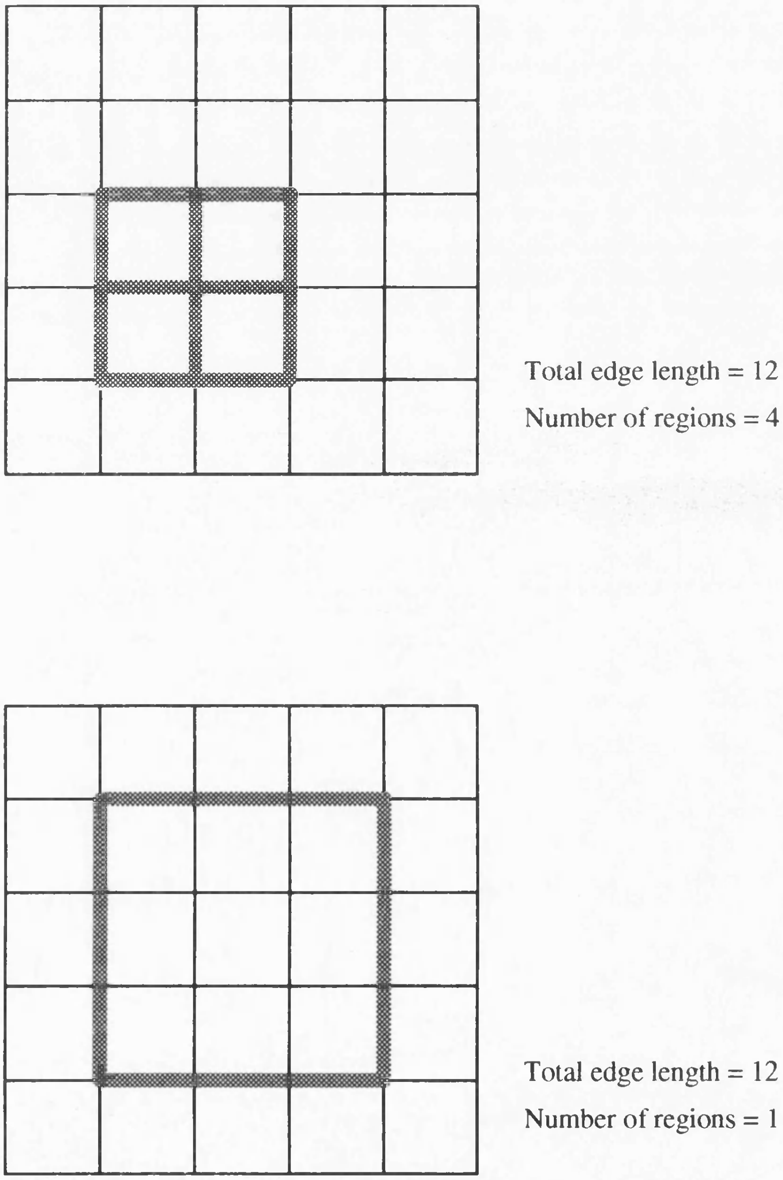


Figure 2.4: *The overall penalty should depend upon the complexity of the image*

j	Type	V_j
0	no lines	0
1	ending	$0.412h\beta+0.5\rho$
2	turn	$0.670h\beta$
3	continuation	$0.948h\beta$
4	branch	$1.4h\beta+0.5\rho$
5	crossing	$1.94h\beta+\rho$

Table 2.2: *The edge penalties proposed by Silverman* et al.

et al.[39] is 10.264β for edge process (i), whereas it is 13.400β for edge process (ii). These two values are similar. For the second example, as illustrated in Figure 2.4, the value of $\sum_{c\in C^{(E)}} V_c^{(E)}(e)$ obtained using the penalties proposed by Silverman *et al.*[39] is 10.22β+3ρ for the edge process that gives four regions, and 10.264β for the edge process that gives one region. Ideally, we would like these quantities to be $10.22\beta+4\rho$ and $10.264\beta+\rho$, for, in this way, they would almost achieve the aim quantified in equation (2.2). We shall explain this short fall of ρ in Section 2.4.1, where we propose a slight modification to overcome it.

2.4 The edge penalties of Silverman *et al.*

In Section 2.4.1 we present a review of the paper by Silverman *et al.*[39] and outline how the penalties for the edge cliques proposed there and given here in Table 2.2 are obtained. In Section 2.4.2 we discuss a possible relationship between ρ and β.

2.4.1 Review of Silverman *et al.*

We have seen in Section 2.3 that the general idea on which the work is based is that the edge process is a discretization of the boundaries of the real image. Accordingly, Silverman *et al.*[39] aim to have $\sum_{c\in C^{(E)}} V_c^{(E)}(e) = (\beta\times\text{total edge length})+(\rho\times r)$, as given in equation (2.2). Hence, they assume that $V_j = (\beta\times b_j)+(\rho\times r_j)$, for $j = 1, \ldots, 5$, where b_j and r_j are to be chosen so that equation (2.2) holds, at least approximately.

Silverman *et al.*[39] first consider how to find b_2 and b_3: turns and continuations have r_2 and r_3 set equal to zero as such clique configurations are not involved with the production of regions. They approach the task of finding b_2 and b_3 by looking at the penalty for a very simple pattern consisting of an infinitely long straight line placed at an angle θ to one of the edge directions of the lattice; without loss of generality $0 \leq \theta \leq \pi/4$. The discretization of this line (by consideration of the dual pixel process, as we shall describe shortly) will give a stepped pattern. When $0 \leq \theta \leq \pi/2$, over a long distance L in the x direction, the number n_x of horizontal segments will be asymptotically Lh^{-1}, and the number n_y of vertical segments will be asymptotically $Lh^{-1}\tan\theta$. The number of continuations in the discretization of the line is $n_x - n_y$ and the number of turns is $2n_y$. The total length of the underlying boundary is $L\sec\theta$ and hence the penalty for unit length of the underlying boundary is, for large L,

$$c(\theta) = h^{-1}\{V_3 + (2V_2 - V_3)\tan\theta\}.$$

Unfortunately, it is impossible to make $c(\theta)$ constant for all θ; this would be the ideal situation. A natural index of how far $c(\theta)$ falls short of ideal is given by the ratio

$$I(\alpha) = \frac{\max_{0\leq\theta\leq\frac{\pi}{4}} c(\theta)}{\min_{0\leq\theta\leq\frac{\pi}{4}} c(\theta)},$$

where $\alpha = V_2/V_3 = b_2/b_3$. We note that $I(\alpha) \geq 1$. The authors show that $I(\alpha)$ is minimized by setting $2\alpha - 1 = \tan(\pi/8)$, which implies that $\alpha = (1 + \tan(\pi/8))/2 = 1/\sqrt{2}$, in which case $I(\alpha) = \sec(\pi/8) = 1.082$. Geman and Geman[12] used $\alpha = 2$, giving $I(\alpha) = 2.83$. Silverman *et al.*[39] then set $b_2 = kh/\sqrt{2}$ and $b_3 = kh$, where k takes the value 0.948. This ensures that

$$(2\pi)^{-1}\int_0^{2\pi} c(\theta)\,d\theta = \beta.$$

Hence, while $c(\theta)/\beta$ is only exactly 1 for certain values of θ, it will be the case that $c(\theta)/\beta$ lies between 0.948 and 1.027 for all θ and furthermore that the average value of $c(\theta)$ over uniformly distributed θ is precisely β.

Next, the authors consider how to assign r_1, r_2 and r_5. Assume that the original process is observed on a window W in the plane and at least one boundary intersects the window edge. Let n_f be the number of regions (faces), n_v be the number of vertices and n_e be the number of edges (sections between vertices) in the pattern. Assume that the pixel size is sufficiently small relative to the scale of the regions in the pattern that each region is represented by a single connected set of pixels in the discretized image. The Euler-Poincaré formula gives

$$n_f = 1 + n_e - n_v.$$

Now, both n_e and n_v can be found by counting the number of branches and crossings in the pattern, provided that points where an edge meets the border of W count as branches. It is immediate that

$$n_v = \#(\text{branches}) + \#(\text{crossings}),$$

where #(branches) means the number of branches. Also

$$n_e = \frac{3}{2}\#(\text{branches}) + \frac{4}{2}\#(\text{ crossings}),$$

since each branch contributes 3 edge ends and each crossing contributes 4 edge ends, and each end is counted twice. Hence

$$n_f = 1 + \frac{1}{2}\#(\text{branches}) + \#(\text{crossings}), \tag{2.3}$$

and so one should set $r_4 = 1/2$ and $r_5 = 1$. This gives a penalty of $\rho/2$ for each branch point and ρ for each crossing. If the edge process gives rise to regions that are not simply connected, the right hand side of equation (2.3) must be increased by 1 for each connected set of edges that does not intersect the border of W. We saw an example of this in Figure 2.4 of Section 2.3.1. The two edge processes in that figure give rise to regions that are not simply connected. The penalties assigned to them by Silverman *et al.*[39] would be $10.22\beta + 3\rho$ and 10.264β. If, however, the right hand side of equation (2.3) were increased by 1, then these penalties would be increased to $10.22\beta + 4\rho$ and $10.264\beta + \rho$, and would thus reflect properly the number of regions. This extra penalty of ρ for each isolated connected set of edges cannot be calculated from local properties, and thus cannot be included in a reconstruction algorithm that operates entirely by local updating. Such an algorithm might, however, be extended to investigate the complete removal of a small connected set of edges in the later stages of reconstruction. When we consider a reconstruction experiment in Section 2.5, we adopt such an extension to the algorithm. This extension leads us to an inequality between ρ and β, as we shall discuss in Section 2.4.2.

Next, Silverman *et al.*[39] move on to discuss irregular and uneven pixel arrays. We shall not examine this section of the paper in detail here, except to state two important general definitions. First, given any pixellation, the *dual* of that pixellation is constructed by placing a point in each cell of the original pixel array, and joining points if their corresponding pixels have some edge in common; secondly, an edge segment is *present* in the edge process if and only if the corresponding dual edge is intersected by the boundary of the true image.

In the penultimate section of their paper Silverman *et al.*[39] obtain b_1, b_4 and b_5. First, they re-derive b_2 and b_3. They do this by consideration of the projection penalty, as follows. First, note that the dual of a square lattice is a square lattice. Label a square in the dual lattice $ABCD$ clockwise. For a straight continuation to be present in the original clique the line l must cross AD and BC, say, (see Fig. 9 of [39]). The *projection penalty* is E[(projection of AB and DC on to l) / 2] where the expectation is taken over random lines conditional on their crossing AD and BC. The sense in which a line is random is given in Jennison and Silverman[20], page 111. The rationale behind this definition is that both AB and DC will be edges of the irregular strip formed by the union of those dual squares intersected by l, and that the total length of the two edges of this strip is approximately twice that of l. The authors compute the projection penalty for a continuation to be $kh = 0.95h = b_3$, as before. Similarly, the projection penalty for a turn configuration is shown to be $kh / \sqrt{2} = 0.67h = b_2$, as before. The authors now consider how to obtain b_1, b_4 and b_5. Again they use the projection penalty appropriately defined. For b_1 (ending) the projection penalty is modified appropriately (see page 227 of Jennison and Silverman[20]) and the expectation is taken over random lines that cross the side AD, say, of the dual square $ABCD$ and terminate in the square itself. The end of the line is considered to be distributed uniformly along its length inside the square. For b_4 (branch), note that a branch arises when three lines meet. Jennison and Silverman[20] show some examples in their Figure 16.17. They also explain the appropriate form of the projection penalty, which now comprises three terms. This time the expectation is with respect to a uniform distribution of the point of intersection of the three lines in the plane and of the orientation of these three lines, conditional on sides DA, AB and BC but not CD being intersected. However, if the lines do not meet at acute angles the associated edge process can be far more complex as is demonstrated in [20] by Figure 16.18. Accordingly, two special cases are considered: three angles of $2\pi / 3$ and angles of $\pi / 2$, $\pi / 2$ and π between the lines. The value $b_4 = 1.4h$ gives a compromise between the two results that they obtained for these angles. For b_5 (crossing), the projection penalty is further modified, and this time comprises four terms. The expectation is with respect to a uniform distribution of the point of intersection over the interior of the dual square $ABCD$ and a uniform distribution of the orientation of the set of four lines, conditional on actually producing a crossing in the edge process. In Figure 16.19(b) of Jennison and Silverman[20] we see an example of when the meeting of four lines produces two adjacent branches, rather than a crossing, in the edge process. Jennison and Silverman[20] consider the case of four lines meeting at right angles and producing a crossing in the edge process, and use numerical integration to obtain $b_5 = 1.94h$.

All that remains now is to assign r_1. Silverman *et al.*[39] set $r_1 = 1/2$ for reasons described in their Section IIIc. There, these authors argue that a pattern made up of disjoint regions cannot have a configuration of edges containing any endings. This suggests that V_1 should be infinite. However, such a penalty may lead to algorithmic difficulties in using the model in practice, violates the theory of Markov random fields which says that all configurations have strictly positive probability (see Section 1.3.2), and is excessively dogmatic. It seems more satisfactory to ascribe a relatively large value to V_1. However, there is no advantage in setting λ much greater than $\rho/2$ since a clever reconstruction algorithm can build a small loop of edges onto a loose end at a penalty $\rho/2$ for the branch plus the penalty for the edge length involved.

The final section of the paper describes a reconstruction experiment performed to compare the proposed penalties to those suggested by Geman and Geman[12]. This experiment will be discussed in greater detail in Section 2.5.

2.4.2 A possible relationship between ρ and β

Although there is no obvious connection between ρ and β, we may obtain a possible relationship between these two parameters, in the form of an inequality, by a simple geometric argument. Assume that the boundary of the real image comprises a circle of radius $r = mh > 0$, where m is an integer, that does not intersect the border of the window W. If we use the penalties proposed by Silverman *et al.*[39], such a circle will receive a penalty approximately equal to $2\pi r\beta$. Hence, as the radius r decreases, the penalty decreases. Assume that our reconstruction algorithm is capable of assigning an extra penalty of ρ, as discussed in Section 2.4.1, to circles whose radius is equal to h, but not to bigger circles. Then circles of radius $2h$ would still have a penalty $4h\pi\beta$, whereas circles of radius h would now have a penalty $2h\pi\beta + \rho$. However, it seems reasonable in practice that circles of radius h should receive a higher penalty than circles of radius $2h$, and hence we have the inequality

$$2h\pi\beta + \rho > 4h\pi\beta.$$

In other words, $\rho > 2h\pi\beta$. We shall see that our choices of ρ and β is Section 2.5 obey this inequality.

2.5 Reconstruction experiments to compare Silverman *et al.*'s edge penalties with Geman and Geman's

In this section, we describe two reconstruction experiments performed to compare the edge penalties proposed by Silverman *et al.*[39] with those suggested by Geman and Geman[12]. We begin by setting up the reconstruction experiment as the minimization of a certain penalty function, which comprises three terms. The second of these in effect penalizes the roughness of the image within the regions defined by the edge process, and is proportional to a smoothing parameter γ. In every other chapter we refer to this smoothing parameter as β. In this chapter, for consistency with Jennison and Silverman[20] and Silverman *et al.*[39], β is used to refer to the desired penalty per unit length of edge. In Section 2.5.1 we use some insight about the behaviour of the edge process near the boundary of the window W to give an inequality involving γ and V_1, V_3 and V_4. In Section 2.5.2 we describe an experiment based on an image of $c = 3$ unordered colours comprising 64×64 (4096 in total) pixels. In Section 2.5.3 we describe a second experiment based on an image of $g = 64$ ordered grey-level comprising 32×32 (1024 in total) pixels. In both cases the record at pixel s, namely y_s, can be thought of as an independent observation from a $\mathcal{N}(f_s, \kappa)$ distribution, where f_s is the colour or grey-level of the original image at pixel s and κ is a known variance. We have seen that the prior distribution for $X = (F, E)$ is defined as

$$\Pr(F = f, E = e) \propto \exp\left\{-U^{(F,E)}(f, e)\right\},$$

where

$$U^{(F,E)}(f, e) = U^{(F|E)}(f \mid e) + U^{(E)}(e)$$

$$= \sum_{c \in C^{(F|E)}} V_c^{(F|E)}(f \mid e) + \sum_{c \in C^{(E)}} V_c^{(E)}(e).$$

This prior distribution can be combined by means of Bayes's theorem with the likelihood of the record y given the pixels f of an image x (see equation (1.5)) to obtain the posterior probability

$$\Pr(x \mid y) \propto \exp\left\{ - \left(\frac{1}{2\kappa} \sum_{s \in S} (y_s - f_s)^2 + \sum_{c \in C^{(F|E)}} V_c^{(F|E)}(f \mid e) + \sum_{c \in C^{(E)}} V_c^{(E)}(e) \right) \right\},$$

where S is the set of pixels.

To be consistent wth the approach adopted by Geman and Geman[12] we attempt to find the MAP estimate of x. This is the image $x = (f, e)$ that maximizes $\Pr(x \mid y)$, or, equivalently, minimizes the following penalty function, over images $x = (f, e)$:

$$\frac{1}{2\kappa} \sum_{s \in S} (y_s - f_s)^2 + \sum_{c \in C^{(F|E)}} V_c^{(F|E)}(f \mid e) + \sum_{c \in C^{(E)}} V_c^{(E)}(e). \tag{2.4}$$

The first term is a penalty for the infidelity between the record and the reconstruction. The second term penalizes the roughness of the reconstruction given the edge process; in other words it penalizes the roughness of the image within the regions defined by the edge process. The third term penalizes the edge process itself.

We have already discussed the term $\sum_{c \in C^{(E)}} V_c^{(E)}(e)$ of the penalty function given in (2.4). All that remains is to specify $V_c^{(F|E)}(f \mid e)$, where c is a pixel clique $\{r, s\}$, say. First, let us consider images comprising c unordered colours. Geman and Geman[12] set

$$\begin{aligned} V_c^{(F|E)}(f \mid e) &= V_{\{r,s\}}^{(F|E)}(f_r, f_s \mid e) \\ &= \begin{cases} 0 & \text{if the edge between pixels } r \text{ and } s \text{ is present,} \\ -\gamma/2 & \text{if the edge between pixels } r \text{ and } s \text{ is absent and } f_r = f_s, \\ \gamma/2 & \text{if the edge between pixels } r \text{ and } s \text{ is absent and } f_r \neq f_s. \end{cases} \end{aligned}$$

Thus, they give a 'reward' for neighbouring pixels of the same colour. The definition that we adopt is more in keeping with the penalty philosophy:

$$V_c^{(F|E)}(f \mid e) = V_{\{r,s\}}^{(F|E)}(f_r, f_s \mid e)$$

$$= \begin{cases} \gamma & \text{if the edge between pixels } r \text{ and } s \text{ is absent and } f_r \neq f_s, \\ 0 & \text{otherwise.} \end{cases}$$

Our approach removes the anomaly that an isolated pixel with all its edges present in, for example, a region of constant background intensity can have its intensity changed to the background intensity and its edges removed simply by increasing the smoothing parameter γ. In the absence of an edge process this gives exactly the same penalty (see equation (1.9)) as we used for images comprising unordered colours in Chapter 1.

We now consider images comprising g grey-levels. We set

$$\begin{aligned} V_c^{(F|E)}(f \mid e) &= V_{\{r,s\}}^{(F|E)}(f_r, f_s \mid e) \\ &= \begin{cases} \gamma \phi_\alpha(|f_r - f_s|) & \text{if the edge between pixels } r \text{ and } s \text{ is absent,} \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{2.5}$$

where, from equation (1.10),

$$\phi_\alpha(u) = 1 - \frac{1}{1 + \alpha u^2} = \frac{1}{1 + (\alpha u^2)^{-1}},$$

as employed by Geman and McClure[13] and discussed in Section 1.3.3.

We employ the method of simulated annealing followed by ICM (see Section 1.6.2) to minimize the penalty function (2.4). Although Geman and Geman rely upon simulated annealing only, ICM can only reduce the penalty function to a value corresponding to a local minimum and so we feel that it is legitimate to include it in our comparison. The maximum likelihood classifier was used as the initial estimate for the pixel process; the initial estimate for the edge process comprised no edges.

2.5.1 Guidelines for the choice of the parameter γ

In our reconstruction experiments there are no edge sites on the border of the window W, although an edge meeting the border was penalized as a branch, as explained in Section 2.4 here, or in Section IIIB of Silverman *et al.*[39]. It turns out that the behaviour of the edge process near the boundary can provide some guidance for the choice of the parameter γ. It may happen that the edge process stops one edge site, between pixels r and s say, short of the boundary, although the pixel process has been suitably reconstructed in such a way that $f_r \neq f_s$.

Thus, to change this unsatisfactory situation, we would like to extend the edge process by one edge site without changing the pixel process. This requires making the appropriate part of the penalty for the unsatisfactory situation (namely, the penalty for an ending plus the penalty due to the pixel process) greater than that for the satisfactory situation (namely, the penalty for a continuation plus the penalty for a branch, with no contribution due to the pixel process). This gives us the inequality

$$\gamma h(f_r, f_s) + V_1 > V_3 + V_4,$$

where $h(f_r, f_s) = 1$ when the image comprises colours, and $h(f_r, f_s) = \phi_\alpha(|f_r - f_s|)$ when the image comprises grey levels. Thus, in the former case we require $\gamma > (V_3 + V_4 - V_1)$, whereas in the latter case we require $\gamma > (V_3 + V_4 - V_1)(1 + (\alpha d^2)^{-1})$, where $d = |f_r - f_s|$. We note that for the edge penalties proposed by Geman and Geman[12] we have $V_3 + V_4 - V_1 = 0$, while for those proposed by Silverman *et al.*[39] we have $V_3 + V_4 - V_1 = 1.936h\beta$. So, since $\gamma > 0$, our requirement is always satisfied when we use the edge penalties proposed by Geman and Geman[12]; with the penalties proposed by Silverman *et al.*[39] we require $\gamma > 1.936h\beta$ in the case of colour images and $\gamma > 1.936h\beta(1 + (\alpha d^2)^{-1})$ in the case of grey-level images. In the latter case we make the assumption that an edge should appear between two pixels whose grey-levels differ by at least 10. Under this assumption it is sufficient to require that $\gamma > 1.936h\beta(1 + (100\alpha)^{-1})$.

2.5.2 An experiment on a 64×64 image with $c = 3$ colours

In this section we discuss an experiment that we have performed on an image comprising 64×64 pixels with $c = 3$ colours. This work has been published in slightly less detail as Section VI of Silverman, Jennison, Stander and Brown[39]. The present author was responsible for the majority of the work of that section. Here, we begin by showing the true image in the top picture of Figure 2.5. The image is made up of disjoint regions separated by edges, and we shall consider the reconstruction of edges as being of some interest in its own right. At each pixel s, a record y_s was generated by adding white noise with variance $\kappa = 0.5$. We show the maximum likelihood estimate in the bottom picture of Figure 2.5; 1185 out of 4096 pixels (28.9%) are misclassified.

In order to be consistent with the work reported in Geman and Geman[12] we adopt

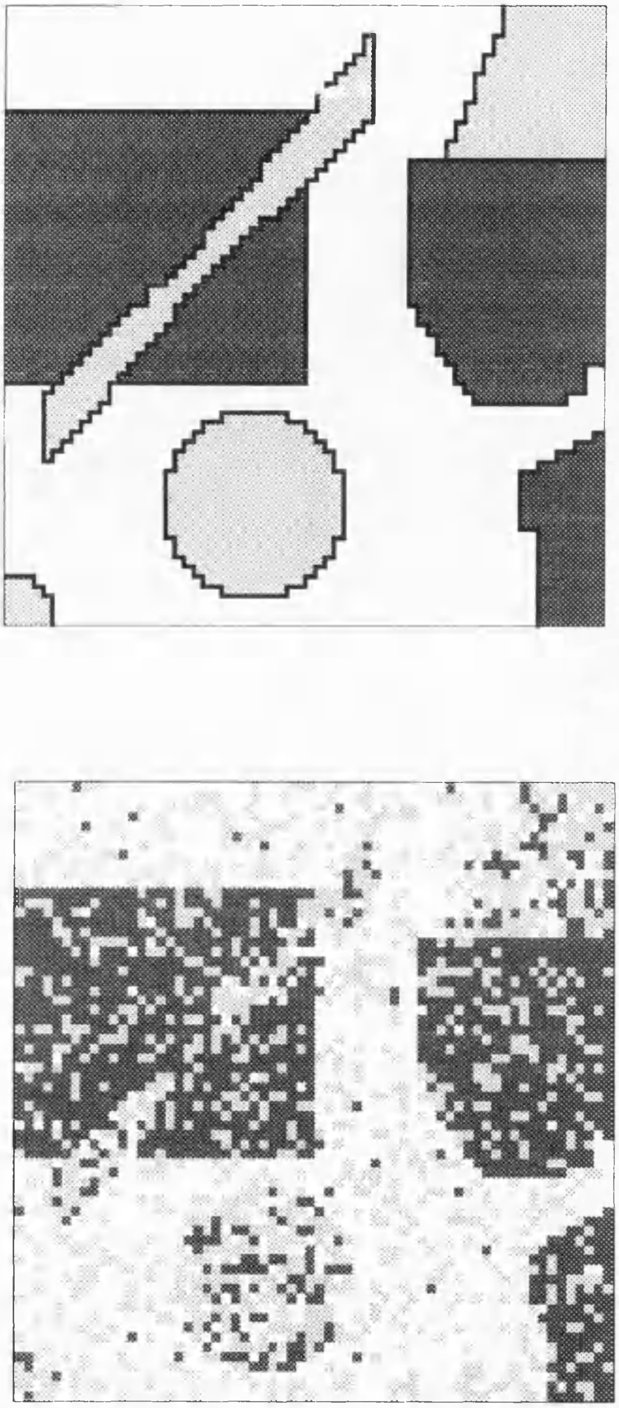


Figure 2.5: *An experiment on a 64×64 image with $c = 3$ colours: the true image and the maximum likelihood estimate*

their temperature schedule for simulated annealing and perform 250 iterations at temperature $\tau(t) = 3 / \log(1 + t)$, where t is the number of the iteration, followed by ICM to convergence. At each pixel the algorithm updates the colour (or grey-level) of the pixel together with its four edges, and thus this updating mechanism is very computationally intensive. However, the only information required for such an update is available locally and so an array of processors, each with access only to this local information, can be employed. Other less computationally intensive ways of updating, such as dealing with the pixel process and the edge process separately, were also considered. These methods performed less well than the one eventually adopted. The basic reason for this was that, in general, in order to make a small change (often on the boundary of a region) to a reconstruction that reduces the penalty, such methods produce an intermediate reconstruction with a very much higher penalty, and thus a very much lower probability. Although simulated annealing allows increases in the penalty function as well as decreases, such intermediate reconstructions are often so improbable that the original desired small change very rarely takes place. Updating the colour (or grey-level) of the pixel together with its four edges allows such small changes to occur directly avoiding any intermediate step.

Two types of penalties $V_c^{(E)}(e)$ were considered. The first type was that used by Geman and Geman[12], except that their penalties were allowed to be multiplied by a scale factor δ. Thus, if V represents the vector of penalties $(V_0, V_1, V_2, V_3, V_4, V_5)$, V now equals $(0, 2.7, 1.8, 0.9, 1.8, 2.7)\delta$. The second type of edge process penalties consisted of those penalties of Silverman *et al.*[39] given in Table 2.2, which depend upon the parameters β, desired penalty per unit length of edge, and ρ, desired penalty per region of the pattern.

First, to assess the edge penalties proposed by Geman and Geman[12], experiments with many different values of the parameter γ and the scale factor δ were performed in an attempt to find the combination that performed best according to some criterion, such as pixel misclassification rate, based on the true image. (Of course, we could not proceed in this way in practice as the true image is unknown.) Reconstructions were only considered if they contained actual edges arranged in a reasonable way. We point out that as $\delta \rightarrow \infty$ in the case of the edge penalties proposed by Geman and Geman[12], or as $\rho \rightarrow \infty$ and $\beta \rightarrow \infty$ in the case of the edge penalties proposed by Silverman *et al.*[39], the overall penalty for the edge process becomes so large that no edge process will appear in the reconstruction process.

The top picture of Figure 2.6 ($\gamma = 2.0$ and $\delta = 1.0$) shows the best reconstruction achieved in terms of pixel misclassification rate. There are 150 (3.7%) misclassified pixels; 263 out of

445 edges have been correctly reconstructed, although there are 146 spurious edges. The best reconstruction in terms of edges ($\gamma = 3.0$ and $\delta = 1.0$) had 152 (3.7%) misclassified pixels, with 267 out of 445 edges correctly reconstructed, and only 141 spurious edges. The second picture of Figure 2.6 is the best reconstruction in terms of both pixel misclassification rate and edges obtained by using the edge penalties of Silverman *et al.*[39]. The implementation included the modification discussed in Section 2.4 of charging ρ for identifiable small connected sets of edges. The parameter γ was set to 3.5, the desired penalty per region of the pattern ρ was set to 10.0, and the penalty per unit length of edge β was set to 1.5. The number of misclassified pixels is 110 (2.7%), the number of correctly reconstructed edges is 307, and the number of spurious edges is 141. Thus, there is a considerable improvement over the first picture of Figure 2.6. We note that the inequality presented in Section 2.4.2 is satisfied ($10.0 = \rho > 2\pi\beta = 9.4$), as is the inequality presented in Section 2.5.1 ($3.5 = \gamma > 1.936\beta = 2.9$). Experiments conducted with different values of the parameters ρ and β indicated that here the improvement is not enormously sensitive to their precise choice. If we turn away from these fairly crude numerical summaries to examine the pictures themselves, we see that the second picture of Figure 2.6 gives a better treatment of boundaries at orientations away from the horizontal and vertical than the first picture, although lines whose orientations are exactly horizontal or vertical are perhaps less well treated. Other experiments that we performed on different colour images with different noise confirmed these observations. Table 2.3 helps us to understand why this is the case. With Silverman *et al.*[39]'s penalties V_2 (turn) is less than V_3 (continuation), whereas this is not so with Geman and Geman[12]'s.

We now make a comparison between the reconstruction obtained with the restriction that edges are present and the reconstruction obtained without this restriction. Again we employ 250 iterations of simulated annealing at temperature $\tau(t) = 3 / \log(1 + t)$, where t is the number of the iteration, followed by ICM to convergence. In this case in order to calculate the number of edges that could be said to have been correctly reconstructed, an edge was said to be present between a pixel and its horizontal or vertical neighbour if these two pixels were coloured differently. The best reconstruction in terms of pixel misclassification rate (again $\gamma = 2.0$) had 137 (3.3%) misclassified pixels, and 248 out of 445 edges correctly reconstructed, although there were 106 spurious edges. The best reconstruction in terms of edges ($\gamma = 3.0$) had 229 (5.6%) misclassified pixels, and 282 out of 445 edges were correctly reconstructed, although there are 71 spurious edges. Accordingly, we see that if we remove the restriction that

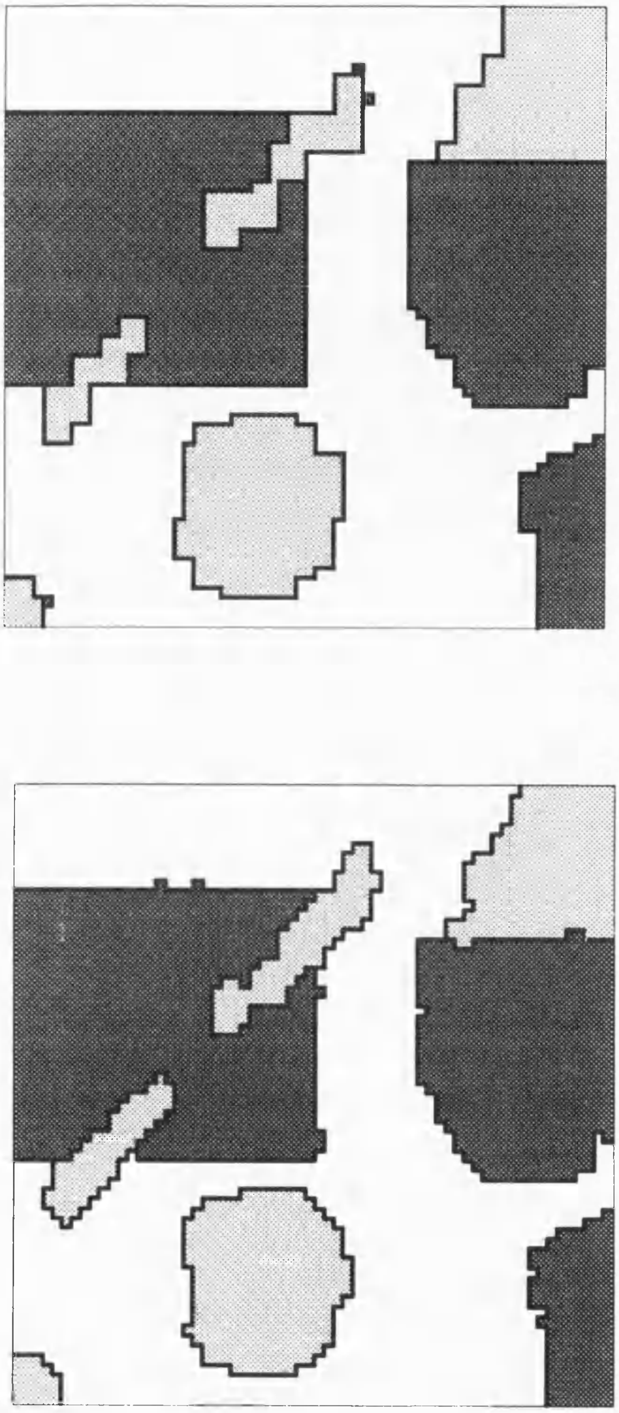


Figure 2.6: *An experiment on a* 64 × 64 *image with c* = 3 *colours: the reconstruction using the edge penalties proposed by Geman and Geman and the reconstruction using the edge penalties proposed by Silverman* et al.

		Geman and Geman		Silverman *et al.*	
j	Type	V_j	Rank	V_j	Rank
0	no edges	0.0		0.0	
1	ending	2.7	1	5.6	3
2	turn	1.8	3	1.0	5
3	continuation	0.9	5	1.4	4
4	branch	1.8	3	7.1	2
5	crossing	2.7	1	12.9	1

Table 2.3: *A comparison between the two types of edge penalties used*

the reconstruction must contain actual edges, reconstructions can be obtained that are better in terms of pixel misclassification rate than obtained with the Geman and Geman[12] penalties, but not better than obtained with the Silverman *et al.*[39] penalties. As far as reconstructing the edge process is concerned, the reconstruction with $\gamma = 3.0$ is better than that achieved by using Geman and Geman[12]'s penalties. However, the $\gamma = 3.0$ reconstruction is not better than that achieved by using Silverman *et al.*[39]'s penalties in terms of the number of correctly restored edges.

In conclusion, the results of this experiment on an image with $c = 3$ colours seem to suggest that the edge penalties proposed by Silverman *et al.*[39] perform considerably better that those proposed by Geman and Geman[12]. It seems that from the results of this experiment and others on colour images that it may be advantageous to employ an edge process in the reconstruction algorithm with Silverman *et al.*[39]'s edge penalties, but not with Geman and Geman[12]'s.

2.5.3 An experiment on an image with $g = 64$ grey-levels

We now report an experiment that we have performed on an image with $g = 64$ grey-levels. The true image is show in the top two pictures of Figure 2.7. We have considered the image before in Section 1.7. There are $32 \times 32 = 1024$ pixels, and four distinct regions based on the three grey-levels, 15, 30 and 45. The histogram gives the number of pixels taking these grey-levels. The true image is corrupted by the addition of independent normal noise with mean 0.0

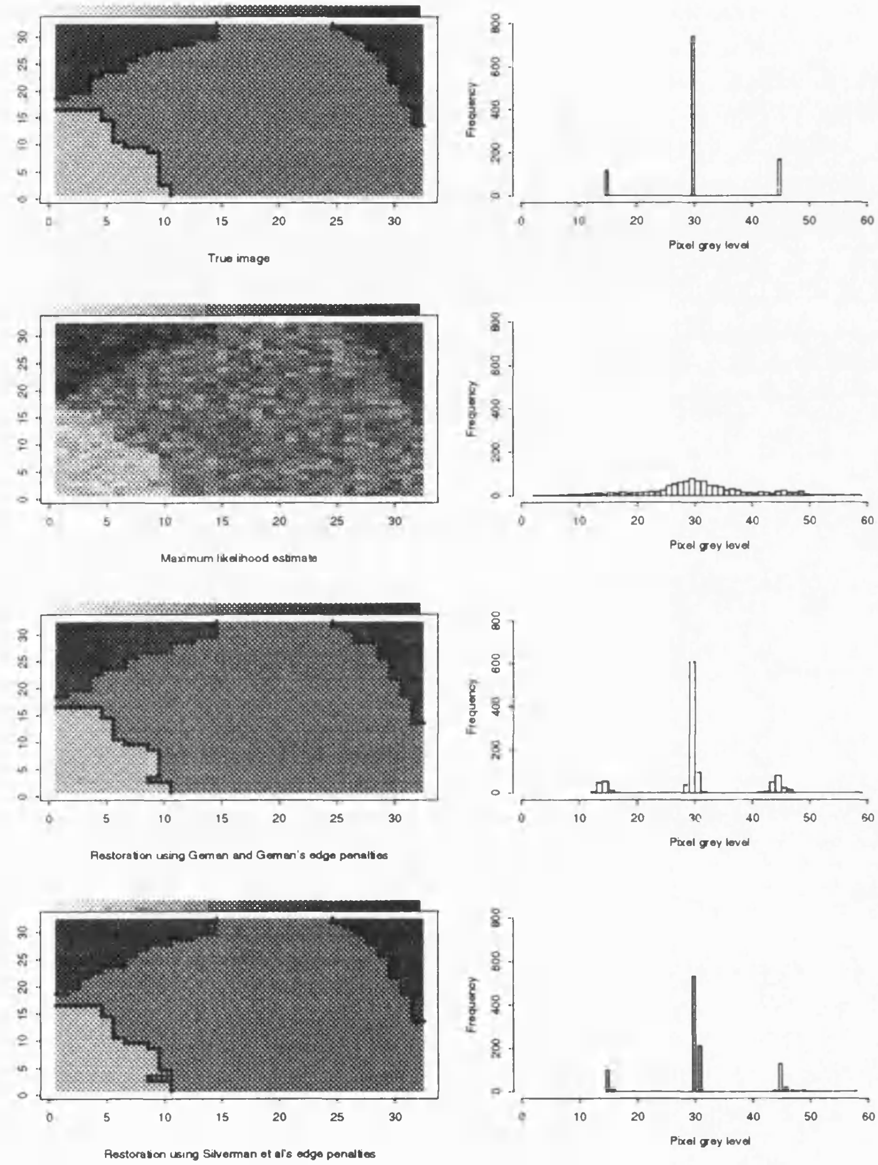


Figure 2.7: *An experiment on a* 32×32 *image with* $g = 64$ *grey-levels*

and variance $\kappa = 20.0$. We present the maximum likelihood estimate and associated histogram in the second of the two pictures; there are 923 (90.1%) misclassified pixels. (As explained in Section 1.7, with grey-level images the number of misclassified pixels does not provide us with an accurate assessment of the similarity of a given image to the true image. This is the reason that we also present histograms.)

Again we produce reconstructions by using the simulated annealing algorithm followed by ICM to convergence. This time 50 iterations are used with a straight line temperature schedule, as employed by Geman and Reynolds[14] in the context of grey-level images, with initial temperature 0.3 and final temperature 0.05. The same updating mechanism was employed here as in Section 2.5.2. We do not perform experiments with many different values of the parameters in order to find the 'best' reconstruction; we merely use parameters similar to those we employed before.

The third pair of pictures in Figure 2.7 shows a reconstruction using the edge penalties proposed by Geman and Geman[12] with $\delta = 1.0$, $\gamma = 2.5$ and $\alpha = 0.075$. There are 286 (27.9%) misclassified pixels; 71 out of 81 edges have been correctly reconstructed, although there are 12 spurious edges.

The fourth pair of pictures in Figure 2.7 shows a reconstruction using the edge penalties proposed by Silverman *et al.*[39] with $\rho = 10.0$ and $\beta = 1.5$. Here $\gamma = 3.5$ and $\alpha = 0.075$. There are 267 (26.1%) misclassified pixels; 73 out of 81 edges have been correctly reconstructed, although this time there are 14 spurious edges. Examination of the histograms seems to suggest that the reconstruction using Silverman *et al.*[39]'s penalties is slightly better than the reconstructions using Geman and Geman[12]'s penalties. However, there is little to choose between the two. We note that the inequality presented in Section 2.4.2 is satisfied ($2.5 = \rho > 2\pi\beta = 9.4$), as is the inequality presented in Section 2.5.1 ($3.5 = \gamma > 1.936\beta(1 + (100\alpha)^{-1}) = 3.3$).

A very good reconstruction was obtained without an edge process with $\gamma = 2.5$ and $\alpha = 0.075$. The straight line temperature schedule employed above followed by ICM to convergence was used. In order to calculate the number of edges that could be said to have been correctly reconstructed, an edge was said to be present between a pixel and its horizontal and vertical neighbours if the grey-levels of these two pixels differ by at least 10. There were 286 (27.9%) misclassified pixels; 73 out of 81 edges were correctly reconstructed, and there were only 10 spurious edges. This reconstruction gives a better treatment of the edge process, and nearly as good a treatment of the pixel process, as the reconstructions obtained using an explicit edge process.

Although we have not tried to select the parameters that define the edge process in an 'optimal' way (impossible to do in practice anyway as the true image is unknown), it appears from this example, and other examples not reported here, that employing an explicit edge process with a grey-level image may not improve the reconstructions to a large degree, if at all. It seems likely that the reason for this is the fact that the function ϕ_α which defined the prior distribution on the pixel process (see equation (2.5)) is carefully designed to preserve edges that may occur between pixels while allowing small variations within regions (see Section 1.3.3). Thus, although in this case no attempt is made to model the boundaries of the real image explicitly, a type of implicit edge process is provided that works well.

2.6 Conclusions

In this chapter we have seen that we can assign a penalty to the edge process of the image by specifying penalties for the six possible ways (up to rotation) in which four edge sites meeting at a point can be occupied. We have described the systematic approach of Silverman *et al.*[39] to the specification of these edge penalties. In order to see whether these penalties lead to better reconstructions than can be obtained by employing those previously used by Geman and Geman[12] (or by employing no edge process), we perform two reconstruction experiments. These provide some evidence to suggest that the edge penalties proposed by Silverman *et al.*[39] will perform well (and should be used) in the case of images comprising unordered colours. However, it seems that with grey-level images little if anything is to be gained by employing an edge process no matter what penalties are used, provided a suitable prior distribution for the pixel process (without an explicit edge process) is employed.

Chapter 3

On Choosing the Smoothing Parameter used in the Exact Maximum *A Posteriori* Estimation for Binary Images

3.1 Introduction

In the Bayesian approach to image analysis a prior distribution Pr (x) over allowable images x is specified, as we discussed in Chapter 1. If we let y denote the record or degraded version of the true image, we can combine the likelihood $l(y \mid x)$ of any image x with the prior Pr (x) to form a posterior or *a posteriori* distribution Pr $(x \mid y)$. The maximum *a posteriori* or MAP estimate of the true image is that x that maximizes this posterior distribution. We shall often refer to this x as $\hat{x}$ in this chapter. For a general image it is not feasible to compute $\hat{x}$ exactly. However, for binary images ($c = 2$ colours), Greig, Porteous and Seheult show in [3] and [17] how $\hat{x}$ may be found exactly by means of reformulating the problem as one of finding the maximum flow through a certain capacitated network. Jubb[24] expanded on their work to produce an efficient implementation of their algorithm. We are grateful to Mike Jubb for making available the basic FORTRAN programs for this implementation.

In this chapter we build on the above mentioned work of Greig *et al.* and Jubb. In Section 3.2 we review the general Bayesian set-up. After this we describe the work of Greig *et al.* and present their fluid flow formulation. In Section 3.3 we discuss the form of the record assumed by this fluid flow approach and introduce two types of degradation mechanism, the addition of normal noise and the binary channel. In Section 3.4 we show that the approach of maximizing the posterior distribution is equivalent to minimizing a certain penalty function. When the prior distribution and the likelihood take certain forms this penalty function represents a trade off between the infidelity of an image x to the record y and the roughness of the image x, controlled by an unspecified smoothing parameter β. We discuss the specific form of this penalty function in the case of the two degradation mechanisms of interest. Also in Section 3.4 we introduce six binary images that we use for our experiments. In Section 3.5 we review the work of Jubb[24]. We introduce the notion of partitioning the image and present some results about its effectiveness. In Section 3.6 we show how the fluid flow approach allows us to produce a sequence of MAP estimates for increasing values of the smoothing parameter β.

The key result presented in Section 3.6 provides us with a method of choosing the smoothing parameter β by eye. An automatic way of choosing β would also be desirable, and we briefly described some such methods for general images in Section 1.8. In Section 3.7 we outline a method for choosing β when the degradation mechanism is the binary channel due to Frigessi and Piccioni[11]. In Section 3.8 we investigate another proposal for choosing β due to Seheult[35] which we apply to the six images corrupted by both types of degradation mechanism. In Section 3.9 we comment upon two further ways of choosing β in the case of binary images.

In Section 3.10 we attempt to motivate the work of Chapter 4 on simulated annealing by using the penalty function with β chosen by means of the method described in Section 3.8 to compare quantitatively the exact MAP estimate for a binary image with estimates produced by simulated annealing. We also consider the performance of Besag[3]'s ICM algorithm. Finally, in Section 3.11 we present our conclusions.

3.2 Formulation and review of the paper by Greig, Porteous and Seheult

In this section we present a thorough review of the work of Greig *et al.* in [3] and [17]. We begin by outlining the general set-up.

In the binary image x, each pixel, i say, can be one of two unordered colours, called white and black, and coded as $x_i = 0$ and $x_i = 1$, respectively. It is assumed that there are n pixels in the image, labelled $1, \dots, n$. The record at pixel i is denoted y_i, where again $i = 1, \dots, n$. Greig *et al.* [17] assume both that the records $y_1, \dots, y_n$ are conditionally independent given x, and that each has known conditional densıty function $f(y_i \mid x_i)$. Thus, the likelihood function for x may be written as

$$l(y \mid x) = \prod_{i=1}^{n} f(y_i \mid x_i) = \prod_{i=1}^{n} f(y_i \mid 1)^{x_i} f(y_i \mid 0)^{1-x_i},$$

where $y = (y_1, \dots, y_n)$ is the vector of records. A remark about the second part of this assumption is given in Section 3.3.

Next, they model the prior distribution $\Pr(x)$ as a pairwise interaction Markov random field of the form

$$\Pr(x) \propto \exp\left\{ \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} \; [x_i x_j + (1-x_i)(1-x_j)] \right\} \tag{3.1}$$

where $\beta_{ii} = 0$ and $\beta_{ij} = \beta_{ji} \geq 0$; in the case of strict inequality, pixels i and j are said to be neighbours. Given the likelihood $l(y \mid x)$ and the prior distribution $\Pr(x)$ we may obtain the posterior distribution $\Pr(x \mid y)$ in the usual way by means of Bayes's theorem:

$$\Pr(x \mid y) \propto l(y \mid x) \Pr(x), \tag{3.2}$$

where the constant of proportionality does not depend upon x. Our interest in this chapter lies in finding that x, $\hat{x}$ say, that maximizes the posterior distribution $\Pr(x \mid y)$. This image $\hat{x}$ is referred to as the maximum *a posteriori* or MAP estimate. Instead of maximizing $\Pr(x \mid y)$, we can equivalently maximize $\log \Pr(x \mid y)$, and, because of the form of $l(y \mid x)$ and $\Pr(x)$, this seems a very sensible thing to do, computationally speaking. Thus, apart from an additive constant independent of x, $\log \Pr(x \mid y)$ can be written as $L(x \mid y)$ where

$$L(x \mid y) = \sum_{i=1}^{n} x_i \lambda_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} \, [x_i x_j + (1-x_i)(1-x_j)] \tag{3.3}$$

and where

$$\lambda_i = \log \left\{ \frac{f(y_i \mid x_i = 1)}{f(y_i \mid x_i = 0)} \right\}, \tag{3.4}$$

is a log-likelihood ratio at pixel i. Hence, our maximization problem may equivalently be viewed as the maximization of $L(x \mid y)$, or the minimization of $-L(x \mid y)$.

In Section 3.4 we shall see how the maximization of $\log \Pr(x \mid y)$ is equivalent to the minimization of a penalty function which differs from $-\log \Pr(x \mid y)$ by an additive constant independent of x. Two examples are given in which the penalty function takes the form of a trade off between the infidelity of the image x to the record y and the roughness of the image x. The balance of this trade off is controlled by a smoothing parameter β which is related to the β_{ij} s of the prior distribution given in expression (3.1) as follows. If pixels i and j are first-order neighbours then $\beta_{ij} = \beta$, whereas if pixels i and j are second-order neighbours, $\beta_{ij} = D\beta$, where D is a downweight. In this chapter we set $D = 0$ and so confine ourselves to first-order models. We have seen that the MAP estimate is that image $\hat{x}$ which maximizes L. Of course, $\hat{x}$ could theoretically be found by direct search over all 2^n possible values of L, but this is computationally infeasible even for quite small n.

In the discussion of Besag[3], Greig *et al.* show that the maximum of $L(x \mid y)$ and hence the maximum *a posteriori* estimate of the binary image can be found using the labelling algorithm of Ford and Fulkerson[10] for finding the maximum fluid flow in a certain capacitated network. We now summarize the derivation of this fluid flow approach of Greig *et al.*[17], and briefly describe the results obtained there using this algorithm.

Consider a capacitated network comprising $n + 2$ vertices, one for the sink, s, another for the source, t, and the remaining n for each of the n pixels. There is a directed edge (s, i) from s to pixel i with capacity $c_{si} = \lambda_i$, if $\lambda_i > 0$; otherwise, there is a directed edge (i, t) from i to t with capacity $c_{it} = -\lambda_i$. Thus, initially pixel i is connected to the source if and only if its maximum likelihood classification is $x_i = 1$ (black). There is an undirected edge (i, j) between two internal vertices (pixels) i and j with capacity $c_{ij} = \beta_{ij}$ as appropriate if the corresponding pixels are neighbours.

For any binary image $x = (x_1, \ldots, x_n)$, define the two sets $B(x)$ and $W(x)$ as

$$\begin{aligned} B(x) &= \{s\} \cup \{i : x_i = 1\} \\ W(x) &= \{i : x_i = 0\} \cup \{t\}. \end{aligned}$$

These two sets give a partition of the network vertices. We now put

$$C(x) = \sum_{k \in B(x)} \sum_{l \in W(x)} c_{kl}.$$

Here the set of edges with a vertex in $B(x)$ and a vertex in $W(x)$ is called a *cut* and $C(x)$ is called the *capacity* of the cut.

It can easily be shown that

$$C(x) = \sum_{i=1}^{n} x_i \max(0, -\lambda_i) + \sum_{i=1}^{n}(1 - x_i) \max(0, \lambda_i) + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}(x_i - x_j)^2.$$

Since

$$x_i \max(0, -\lambda_i) - x_i \max(0, \lambda_i) = -x_i\lambda_i,$$

regardless of the sign of λ_i, and $x_i^2 = x_i$ for all i, we have that $C(x)$ differs from $-L(x \mid y)$ (see equation (3.3)) by a term which does not depend on x. As we noted above, $-L(x \mid y)$ is, apart from an additive constant, the penalty function that we wish to minimize and that will be discussed further in Section 3.4. For consistency with other chapters, we present the value of the penalty function itself rather than the value of $-L(x \mid y)$ when we compare the exact MAP estimate given by this fluid flow approach with that given by simulated annealing and ICM in Section 3.10.

The *max-flow min-cut* theorem (Ford and Fulkerson[10], Theorem 5.1) states that, for any network with a single source and sink, the maximum feasible flow from the source to the sink equals the minimum cut value for any of the cuts of the network. Let F denote the amount of flow from source to sink for any feasible flow pattern. The value of any cut provides an upper bound to F, and the smallest of the cut values is equal to the maximum value of F. Therefore, if a cut can be found in the original network equal to the value of F currently attained by the solution procedure, the current flow pattern must be optimal. Equivalently, optimality has been attained whenever there exists a cut in the current network whose value is zero with respect to the remaining flow capacities. Accordingly, the minimum of $C(x)$ is the maximum flow through the network from source to sink subject to the edge capacities. A corresponding cut is called a *minimum cut*. Thus, in order to maximize $L(x \mid y)$, all that is necessary is to find the minimum cut. In the MAP estimate pixels are black if they are on the source side of the minimum cut and white otherwise.

In the experiments carried out by Greig *et al.* and reported in the discussion of Besag[3], β_{ij} was taken to be β for all internal connections between neighbours. These authors tentatively conclude that relative to ICM, a smaller value of β may be more appropriate for MAP estimation, and that for a given noise level, simulated annealing 'converges' to MAP more rapidly for smaller values of β. In their 1989 paper, where each pixel is assumed to have eight neighbours and $D = 1$, Greig *et al.*[17] state that the availability of exact estimates allows the assessment of the performance of other algorithms. They conclude that MAP estimation is very sensitive to change in the specification of the prior (specification of β), whereas ICM is generally robust to any such change, presumably because it operates locally rather than globally. They also discuss different temperature schedules for simulated annealing, namely logarithmic schedules of the form $\tau(t) = C/\log(1+t)$ and geometric schedules of the form $\tau(t) = A\rho^{t-1}$, where t is the number of the iteration. They state that the opportunity for the simulated annealing algorithm to get trapped in a local maximum of $\Pr(x \mid y)$ increases with increasing β, especially if the temperature is allowed to decrease too rapidly. Thus, they conclude that simulated annealing, applied with practical schedules, does not necessarily produce a good approximation to a MAP estimate. Experimental results suggest that good approximations are more likely for smaller values of β, and that, as β increases, the global properties of the prior distribution very rapidly dominate the likelihood contribution to the posterior distribution. We consider ICM and simulated annealing briefly in Section 3.10, and more thoroughly in Chapter 1 and Chapter 4.

Greig *et al.*[17] state that any attempt to incorporate edge processes (see Chapter 2), or to preserve certain global aspects of the true image will in general render the network method inapplicable. Moreover, although the multi-colour problem can be treated as a generalized minimum cut problem, there is no corresponding network formulation. Finally, they suggest a variant of the basic algorithm: partition the image into connected sub-images and then calculate the MAP estimate for each sub-image separately. This can be interpreted as finding the maximum flow through the network, but under the imposed constraints that no flow is allowed across sub-image boundaries. Next, relax some of these additional constraints to amalgamate sub-images and continue doing this until the MAP estimate of the complete image results. Such a modification was implemented by the authors and resulted in up to a twelve-fold reduction in CPU time. We discuss partitioning in detail in Section 3.5.

3.3 The record

We have seen in Section 3.2 that, following the usual approach as discussed in Chapter 1, Greig *et al.*[17] assume both that the records $y_1, \ldots, y_n$ are conditionally independent given x, and that each has known conditional density function $f(y_i \,|\, x_i)$. The second of these assumptions is, however, not necessary for the fluid-flow formulation. For note that $L(x \,|\, y)$ depends on the record only through λ_i, as defined in equation (3.4). Hence, all that is required regarding the distribution of the record is the ability to write down a likelihood ratio at each pixel. Knowledge of the explicit form of f is not necessary. This also means that the situation where f varies from pixel to pixel, or from colour to colour, is allowed.

In this chapter we assume that all the pixels have been affected in the same way. We consider two types of degradation mechanism: the first is the addition of normal noise of known variance κ to each pixel independent of all the other pixels so that the record at pixel i, y_i, has a $\mathcal{N}(x_i, \kappa)$ distribution, where x_i is the true, but unknown colour of pixel i; the second is corruption by a binary channel, in which each pixel switches colour with known probability ε. The first case, which we shall usually refer to as 'normal noise', is discussed here in Section 3.4.1. The second case, which we shall usually refer to as the 'binary channel', is discussed here in Section 3.4.2 and Section 3.7.

In Figure 3.1 we give an example of the degradation due to normal noise. We consider a 256×256 image of a part of Scotland [1]. At each pixel i the record is

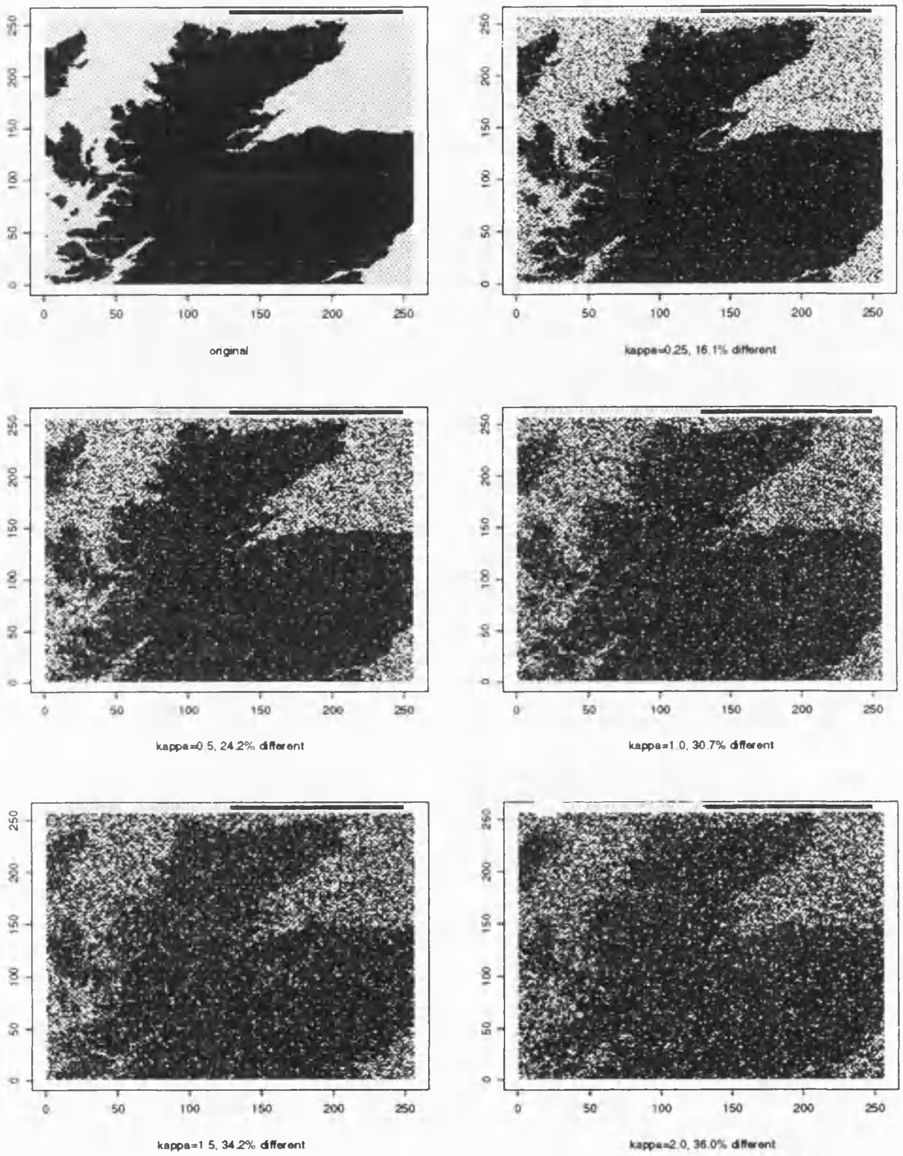
$$y_i \sim \mathcal{N}(x_i, \kappa).$$

We present the maximum likelihood based on y, the vector of records. This is the reconstruction that maximizes $\Pr(x \,|\, y)$ when $\beta = 0$, *i.e.* when there is no spatial information. We shall refer to it as $\hat{x}(0)$. Hence, $\hat{x}(0)$ is the reconstruction that maximizes the likelihood term $l(y \,|\, x)$, and it can be shown that in the case of normal noise

$$\hat{x}_i(0) = \begin{cases} 1 & \text{if } y_i \geq \frac{1}{2} \\ 0 & \text{if } y_i < \frac{1}{2}. \end{cases}$$

We also give in the figure the percentage of pixels in $\hat{x}(0)$ that are different from the original

[1] We are grateful to Art Owen and Mike Jubb for supplying the image of a part of Scotland.



original

kappa=0.25, 16.1% different

kappa=0.5, 24.2% different

kappa=1.0, 30.7% different

kappa=1.5, 34.2% different

kappa=2.0, 36.0% different

Figure 3.1: *A part of Scotland corrupted with normal noise of various variances,* κ

image. For $\kappa = 0.25$ only 16.1% of pixels in $\hat{x}(0)$ are different and the original image is clearly visible, whereas for $\kappa = 2.0$, 36.0% of the pixels are different and it is almost impossible to distinguish the original image.

In Figure 3.2 we present a similar example, but for the binary channel. This time we present the record y itself and the percentage of pixels that have changed, a figure that should be about 100ε. The image with $\varepsilon = 0.1$ is very similar to the original image, whereas with the image with $\varepsilon = 0.4$ it is almost impossible to distinguish the original image. The image with $\varepsilon = 0.5$ is such that at each pixel the colour 0 or the colour 1 is chosen with probability 0.5 and hence has no dependence on the original image.

3.4 MAP estimation as the minimization of a penalty

We have seen that our interest lies in the maximum *a posteriori* estimate. This is the image $\hat{x}$ that maximizes the posterior probability $\Pr(x \mid y)$, where the record y is fixed. We have also seen that $\Pr(x \mid y) \propto l(y \mid x) \Pr(x)$, where the multiplicative constant of proportionality does not depend upon x. Thus, maximizing $\Pr(x \mid y)$ is equivalent to maximizing $l(y \mid x) \Pr(x)$. This, in turn, is equivalent to maximizing $\log l(y \mid x) + \log \Pr(x)$, or minimizing

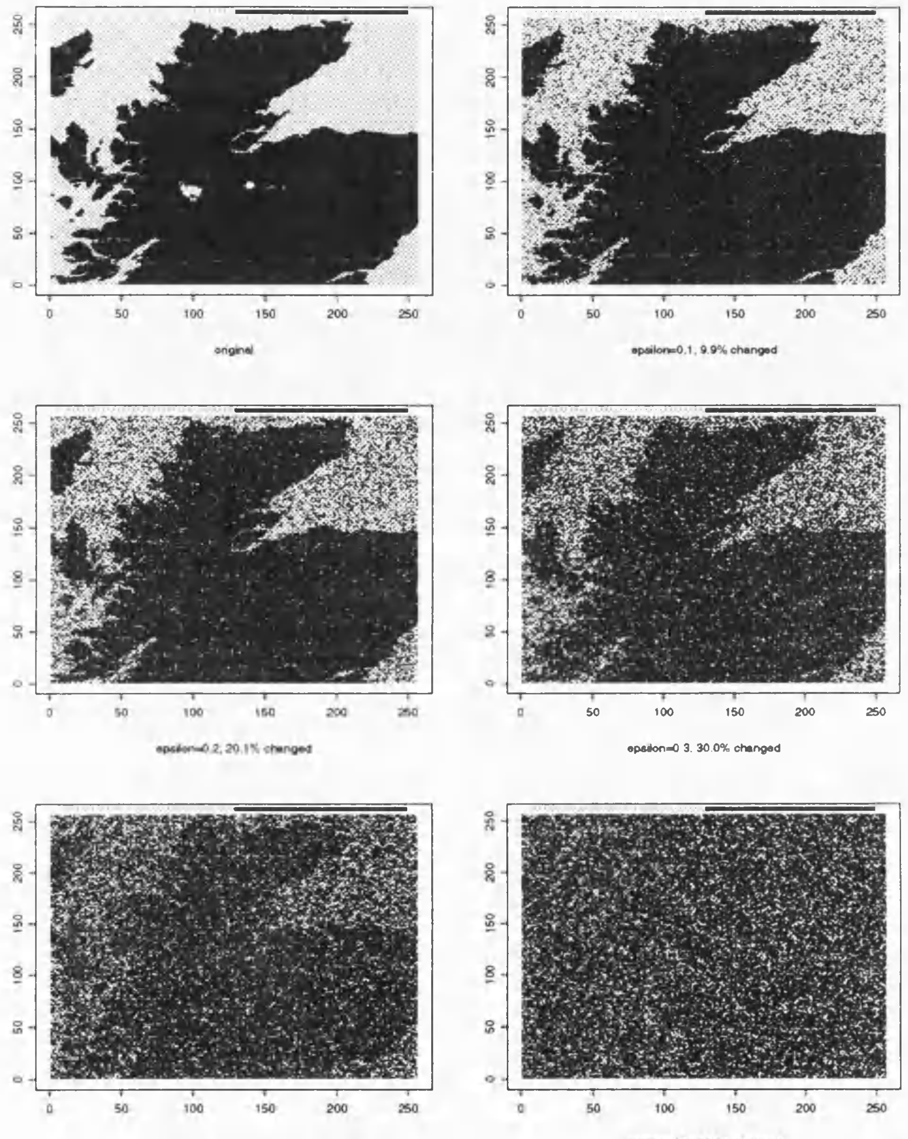
$$-\log l(y \mid x) - \log \Pr(x), \tag{3.5}$$

or any quantity that differs from (3.5) by an additive constant that does not depend upon x. We shall refer to such a quantity as the *penalty function*, which we aim to minimize over images x. Throughout this chapter, we take

$$-\log \Pr(x) = \beta \left(v^{(1)}(x) + D\, v^{(2)}(x) \right), \tag{3.6}$$

plus some additive constant that we may ignore, where $v^{(1)}(x)$ is the number of discrepant first-order pairs in the image, $v^{(2)}(x)$ is the number of discrepant second-order pairs, and D is a downweight. The quantity given in equation (3.6) can be thought of as a measure of roughness of the image x.

We now turn our attention to $-\log l(y \mid x)$. Throughout, we make the same assumption as



original

epsilon=0.1, 9.9% changed

epsilon=0.2, 20.1% changed

epsilon=0 3, 30.0% changed

epsilon=0 4, 40.0% changed

epsilon=0.5, 50.0% changed

Figure 3.2: *A part of Scotland corrupted by the binary channel with various values of ε*

Greig *et al.* [17], as described in Section 3.2, and thus we obtain that

$$l(y \mid x) = \prod_{i=1}^{n} f(y_i \mid x_i).$$

In this chapter we consider two cases for f, that of normal noise and the binary channel. These are discussed below in Section 3.4.1 and Section 3.4.2, respectively. In these sections we show how $-\log l(y \mid x)$ can be thought of as the quantity measuring the *infidelity* of the image to the record x.

3.4.1 Normal noise

We briefly review this common situation here, for the sake of completeness, and to enable comparison with the case of the binary channel discussed in Section 3.4.2. The distribution of the record y_i at pixel i, coloured x_i, is assumed to be normal with mean x_i and known variance $\kappa > 0$, independent of all other pixels. In other words

$$f(y_i \mid x_i) = \frac{1}{\sqrt{2\pi\kappa}} \exp\left\{-\frac{1}{2\kappa}(y_i - x_i)^2\right\}. \tag{3.7}$$

We saw an example of this type of degradation for various values of κ in Figure 3.1 of Section 3.3. From equation (3.7), we immediately obtain

$$-\log f(y_i \mid x_i) = \frac{1}{2}\log(2\pi\kappa) + \frac{1}{2\kappa}(y_i - x_i)^2,$$

and so

$$-\log\, l(y \mid x) = \frac{n}{2}\log(2\pi\kappa) + \frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2.$$

The first term of this expression does not depend upon x (or y) and can be disregarded as far as the minimization is concerned leaving

$$\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2. \tag{3.8}$$

This is clearly a measure of the infidelity of the estimate x to the record y. Moreover, as the variance κ increases, this measure of infidelity decreases. Thus, the greater κ is known to be, the less weight is given to the infidelity term in the penalty function, which, in this case of independent, additive normal noise, is

$$\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left(v^{(1)}(x) + Dv^{(2)}(x)\right). \tag{3.9}$$

Accordingly, this penalty function represents a trade off between the infidelity of the image x to the record y on the one hand, and roughness of the image x on the other. The balance of this trade off is controlled by the (unknown) smoothing parameter $\beta \geq 0$: the higher the value of β, the greater the weight given to the second term of (3.9), and the smoother the image x that minimizes the penalty function (3.9).

For ease of comparison with expression (3.14) of Section 3.4.2 on the binary channel, we can multiply expression (3.9) by the known quantity 2κ to obtain the new penalty function

$$\sum_{i=1}^{n}(y_i - x_i)^2 + 2\kappa\beta\left(v^{(1)}(x) + Dv^{(2)}(x)\right). \tag{3.10}$$

It can be shown that this new penalty function takes the form

$$||y - x||^2 + \lambda_{\mathcal{N}}\Phi(x) \tag{3.11}$$

where $||z||^2 = z^T z = \sum_{i=1}^{n} z_i^2$ and $\Phi(x)$ is a quadratic in the vector x: *i.e.* $\Phi(x) = x^T Cx$. It is now not hard to establish that C is a non-negative definite matrix. The term $\Phi(x)$ may be thought of as a roughness penalty and $\lambda_{\mathcal{N}} = 2\kappa\beta$ may be thought of as a smoothing parameter. This is the form of the penalty function considered in the recent paper by Thompson, Brown, Kay and Titterington[43], which we discussed in Section 1.8. We shall see this penalty function again in equation (3.15) of Section 3.4.2.

3.4.2 Binary channel

Our motivation for considering a binary channel is the paper by Frigessi and Piccioni[11]. In that paper the authors assume that each pixel is wrongly recorded with some fixed probability $0 \leq \varepsilon \leq 1$, independent of all other pixels. This model is known in communication theory as the memoryless binary symmetric channel. The likelihood at each pixel can be expressed as

$$f(y_i \mid x_i) = \begin{cases} \varepsilon & \text{if } y_i \neq x_i \\ 1 - \varepsilon & \text{if } y_i = x_i. \end{cases}$$

In our work, unlike in [11], we assume that ε is known. Immediately, we may disregard the two extreme cases of $\varepsilon = 0$ and $\varepsilon = 1$. Further, we may restrict ε to the interval (0, 1/2], without any loss of generality. For say we know $\varepsilon > 1/2$. We can replace y_i by $1 - y_i$ and in effect replace ε by $\varepsilon' = 1 - \varepsilon$, where ε' is in the interval (0, 1/2). We saw an example of this type of degradation for various values of ε in Figure 3.2 of Section 3.3.

Let us now consider the term $-\log l(y \mid x)$ in detail. Under our assumptions this negative log-likelihood equals $-\sum_{i=1}^{n} \log f(y_i \mid x_i)$. Now

$$-\log f(y_i \mid x_i) = \begin{cases} -\log \varepsilon & \text{if } y_i \neq x_i \\ -\log(1-\varepsilon) & \text{if } y_i = x_i. \end{cases}$$

Let n_d be the number of entries of y_i that are different from x_i, and n_s for the number of entries that are the same. The number n_d can be thought of as the Hamming distance between x and y, as it is the number of pixels at which the true image and the record differ, and $n_s = n - n_d$, when there are n pixels in total. This gives

$$\begin{aligned} l(y \mid x) &= -\sum_{i=1}^{n} \log f(y_i \mid x_i) \\ &= -n_s \log(1-\varepsilon) - n_d \log \varepsilon \\ &= -(n - n_d) \log(1-\varepsilon) - n_d \log \varepsilon \\ &= -n \log(1-\varepsilon) + n_d \log\left(\frac{1-\varepsilon}{\varepsilon}\right), \end{aligned}$$

the first term of which does not depend upon x (or y) and can be disregarded as far as the minimization is concerned. This leaves

$$n_d \log\left(\frac{1-\varepsilon}{\varepsilon}\right), \tag{3.12}$$

which is again a measure of the infidelity of the estimate x to the record y. For comparison with equation (3.8) in section 3.4.1, we can write n_d as $\sum_{i=1}^{n}(y_i - x_i)^2$ and change the order of the two terms in (3.12) to get

$$\log\left(\frac{1-\varepsilon}{\varepsilon}\right) \sum_{i=1}^{n} (y_i - x_i)^2.$$

Since we have

$$\begin{aligned}\frac{d}{d\varepsilon}\log\left(\frac{1-\varepsilon}{\varepsilon}\right) &= -\frac{1}{\varepsilon(1-\varepsilon)} \\ &< 0,\end{aligned}$$

for ε in the interval (0, 1), we obtain that

$$\log\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

is a decreasing function of ε. Thus, the larger ε is known to be, the less weight is given to the infidelity term in the penalty function, which, in this binary case, is

$$\log\left(\frac{1-\varepsilon}{\varepsilon}\right)\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left(v^{(1)}(x) + Dv^{(2)}(x)\right). \tag{3.13}$$

We again remark that this penalty function represents a trade off between infidelity and roughness controlled by the (unknown) smoothing parameter β.

Again, for ease of comparison with expression (3.10) of Section 3.4.1 we can multiply equation (3.13) by the known quantity

$$\frac{1}{\log\left(\frac{1-\varepsilon}{\varepsilon}\right)},$$

which hence forth we shall denote by η, to obtain the new penalty function

$$\sum_{i=1}^{n}(y_i - x_i)^2 + \eta\beta\ \left(v^{(1)}(x) + Dv^{(2)}(x)\right). \tag{3.14}$$

We finish by remarking that, as in Section 3.4.1, the new penalty function (3.14) takes the same form as equation (3.11):

$$||y - x||^2 + \lambda_b\Phi(x). \tag{3.15}$$

Here the smoothing parameter is $\lambda_b = \eta\beta$.

3.4.3 The images used

Throughout this chapter we base our experiments on six binary images. Four of these images, of size 64 × 64, were generated by means of 100 iterations of the Gibbs sampler as described in Section 1.4.1. Although for each case the initial image was produced by assigning a colour to each pixel at random, the theory given in Geman and Geman[12] (see Section 1.4.1) tells us that for a large (finite) number of iterations (raster scans) the realization obtained is (almost) independent of the initial image. These images had β set to 0.2, 0.6, 1.0 and 1.2. For all four images we used a nearest neighbourhood structure ($D = 0.0$). Ripley[33] points out that in this case when we have two colours and the first-order neighbourhood graph, the probability model, whose conditional distribution is given by

$$\begin{aligned}\Pr\left[\text{value of pixel } i \text{ is } 0 \mid \text{neighbours}\right] &\propto \exp\{\beta\#(\text{neighbours coloured } 0)\}\\ \Pr\left[\text{value of pixel } i \text{ is } 1 \mid \text{neighbours}\right] &\propto \exp\{\beta\#(\text{neighbours coloured } 1)\},\end{aligned}$$

reduces to the well-known Ising model of statistical physics. This is known to have a critical value of β, β_c, where

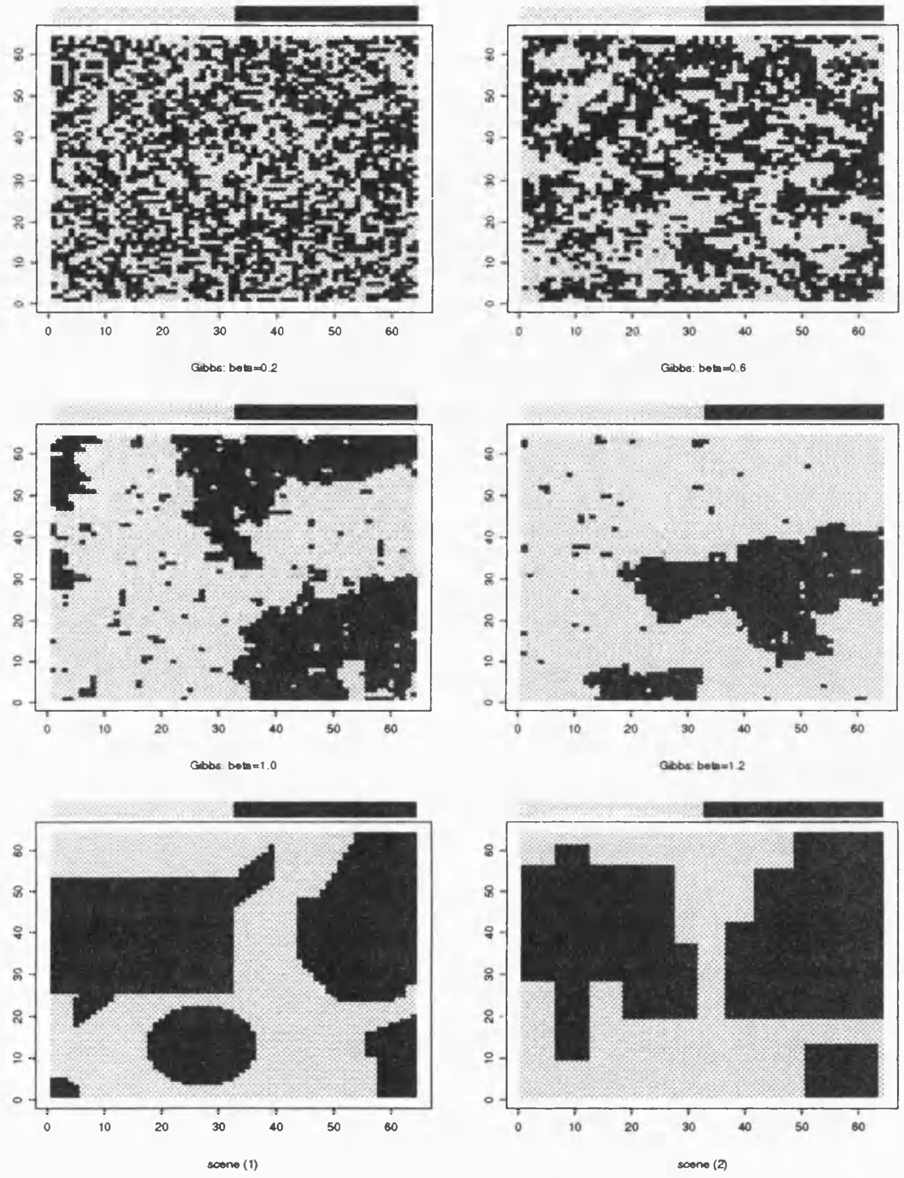
$$\beta_c = \sinh^{-1} 1 \approx 0.88, \tag{3.16}$$

such that, asymptotically as $MN \to \infty$, where M and N are the dimensions of the image, for $\beta < \beta_c$ there are no infinite patches of one type whereas for $\beta > \beta_c$, there will always be such infinite patches. The four images are shown in Figure 3.3 and some details about them are given in Table 3.1.

We considered two further binary images: both are 64 × 64 and 'hand-drawn'. These two images are also shown in Figure 3.3 and details of them are also presented in Table 3.1.

3.5 A review of the work of Jubb and some extensions

After describing the Ford Fulkerson labelling algorithm for maximizing the flow through a network, Jubb[24] formulates and implements some very clever and effective improvements in the algorithm when used for MAP estimation. We shall refer to Jubb's algorithm as the *modified* algorithm. These modifications make use of the particular structure of the network used in the imaging problem.


Gibbs: beta=0.2

Gibbs: beta=0.6

Gibbs: beta=1.0

Gibbs: beta=1.2

scene (1)

scene (2)

Figure 3.3: *The six images used in our experiments*

Image	Size (Total)	Number of pixels taking the value	
		0 (%)	1 (%)
Gibbs $\beta = 0.2$	64×64 (4096)	2007 (49)	2089 (51)
Gibbs $\beta = 0.6$	64×64 (4096)	1984 (48)	2112 (52)
Gibbs $\beta = 1.0$	64×64 (4096)	2517 (61)	1579 (39)
Gibbs $\beta = 1.2$	64×64 (4096)	3075 (75)	1021 (25)
(1)	64×64 (4096)	1956 (48)	2140 (52)
(2)	64×64 (4096)	1787 (44)	2309 (56)

Table 3.1: *Details of the images used*

3.5.1 Partitioning

Greig *et al.*[17] report achieving a 12 fold reduction in CPU time from 3000 seconds to 250 seconds by adopting the following scheme:

- partition the original image into sub-regions;
- employ the original algorithm to find the MAP estimate for each sub-region;
- employ the original algorithm to find the MAP estimate for an amalgamation of these sub-region reconstructions using the estimates already found as the starting point for the Ford-Fulkerson algorithm;
- repeat this until the original image is reached.

They state that this scheme can be interpreted as finding the maximum flow through the network, but under the imposed constraints that no flow is allowed across subimage boundaries. Next these constraints are relaxed. Corresponding subimages are then amalgamated to form a new set of larger subimages. This procedure continues until the MAP estimate for the complete image is obtained. Greig *et al.*[17] base their experiments on an image of size 88 × 100, and they consider a 16×16 array of roughly equal-sized rectangular subimages, followed by an 8×8 array of subimages, a 4x4 array, a 2x2 array and finally the full image itself, as their partitioning. We shall refer to this partitioning as a (16,8,4,2) partition. Greig *et al.*[17] state that this choice of partitioning is unlikely to be optimal and that any sensible choice of partitioning will lead to a

	Partitioning used	Partitioning not used
Modified algorithm used	19.6	15.1
Modified algorithm not used	73.0	770.0

Table 3.2: *Seconds of CPU time for a* 64 × 64 *binary image (Jubb, Table 3.1)*

substantial reduction in CPU time. We investigate this claim in Section 3.5.2 below. However, there we consider the effect of partitioning on the modified algorithm. Another example of the reduction in CPU time that can be achieved in the case of the original algorithm can be seen in Table 3.2. This is Table 3.1 of Jubb[24]. The image under consideration has 64 × 64 pixels and is corrupted by independent, additive normal noise with variance $\kappa = 0.25$. Jubb[24] sets $\beta = 1.0$ and $D = 1/\sqrt{2}$, and uses a (16,4) partition: the flow is maximized first in separate regions each of size 4 pixels by 4 pixels, then in separate regions each of size 16 pixels by 16 pixels, and finally in the whole image. He reports an 11 fold reduction in CPU time. However, that table indicates that partitioning has a detrimental effect in the case when Jubb's modified algorithm is used. We shall discuss this further in Section 3.5.2.

3.5.2 Partitioning and the modified algorithm

Jubb[24] presents two numerical examples of the effectiveness of the modified algorithm and of partitioning. We have already met one of these examples in Section 3.5.1. Jubb[24] concludes from Table 3.2 and his other example that the faster reconstructions were obtained using the modified algorithm alone, and that combining the partitioned version with his modified algorithm has little effect on the CPU time. We attempted to investigate further the effect of partitioning on the modified algorithm. We consider using all possible combinations of these 4 partitions: 16, 8, 4 and 2. There are $2^4 = 16$ possibilities as outlined in Table 3.3. In our investigation of partitioning we use only the 'hand-drawn' image, and we proceed by considering the following four reconstruction problems:

1. Image (1), normal noise with variance $\kappa = 0.5, \beta = 1.25$;

2. Image (2), normal noise with variance $\kappa = 1.0, \beta = 2.0$;

3. Image (1), binary channel with $\varepsilon = 0.2, \beta = 1.2$;

Partition	Sub-region size	Number	Partition	Sub-region size	Number
(16,8,4,2)	4,8,16,32	(i)	(8,4,2)	8,16,32	(ix)
(16,8,4)	4,8,16	(ii)	(8,4)	8,16	(x)
(16,8,2)	4,8,32	(iii)	(8,2)	8,32	(xi)
(16,8)	4,8	(iv)	(8)	8	(xii)
(16,4,2)	4,16,32	(v)	(4,2)	16,32	(xiii)
(16,4)	4,16	(vi)	(4)	16	(xiv)
(16,2)	4,32	(vii)	(2)	32	(xv)
(16)	4	(viii)	None	(64)	(xvi)

Table 3.3: *The possible partitions and their numbers*

4. Image (2), binary channel with $\varepsilon = 0.4$, $\beta = 0.3$.

We recall that both image (1) and image (2) are of size 64×64. They are reproduced in Figure 3.3 and further details are given in Section 3.4.3.

Each image was corrupted and reconstructed 10 times. Sixteen different reconstruction algorithms, corresponding to the above 16 partitions, were considered. For each reconstruction algorithm the average time for the 10 reconstructions was found. This was done by placing the reconstruction part of the program in a separate subroutine `RESTORE`, compiling the `FORTRAN` program with the `-p` flag, and then using the `unix` command `prof`, standing for profile, to find exactly the time spent in the subroutine `RESTORE`. The results of this investigation are presented in Table 3.4. We point out that the times given in Table 3.2 and in Table 3.4 can not be directly compared as they were run on different Sun-4 machines. We now make some observations from Table 3.4.

First, the best two partitions to use seem to be number (viii), in which only a division into 16×16 sub-images each of size 4 pixels by 4 pixels is employed, and number (xvi), in which no partitioning is employed and the complete image is processed without consideration of any sub-images. In three out of the four cases number (viii) is better than number (xvi) by at least half a second, whereas in the fourth case, that of image (1) and normal noise, number (xvi) is better, but this time the margin is only about 0.05 of a second.

	Time (rank)							
	Normal				Binary			
	(1)		(2)		(1)		(2)	
Partition	$(\kappa, \beta) = (0.5, 1.25)$		$(\kappa, \beta) = (1.0, 2.0)$		$(\varepsilon, \beta) = (0.2, 1.2)$		$(\varepsilon, \beta) = (0.4, 0.3)$	
Number	$\lambda_{\mathcal{N}} = 1.25$		$\lambda_{\mathcal{N}} = 4.0$		$\lambda_b = 0.87$		$\lambda_b = 0.74$	
(i)	22.73	(13)	65.98	(9)	22.47	(14)	27.59	(16)
(ii)	19.84	(5)	62.96	(5)	19.61	(8)	25.94	(13)
(iii)	20.91	(7)	68.78	(14)	22.35	(12)	25.48	(12)
(iv)	19.00	(3)	55.29	(3)	17.65	(4)	23.35	(7)
(v)	23.09	(14)	68.91	(15)	22.42	(13)	26.06	(14)
(vi)	20.82	(6)	64.79	(8)	19.14	(6)	24.34	(9)
(vii)	20.96	(8)	67.24	(12)	20.06	(9)	22.87	(6)
(viii)	18.60	(2)	54.48	(1)	15.76	(1)	18.62	(1)
(ix)	23.63	(16)	66.02	(10)	22.88	(15)	27.48	(15)
(x)	21.44	(9)	64.71	(6)	17.86	(5)	23.76	(8)
(xi)	22.19	(12)	66.92	(11)	22.00	(11)	24.95	(11)
(xii)	19.74	(4)	58.29	(4)	17.54	(3)	22.11	(4)
(xiii)	23.54	(15)	70.08	(16)	23.67	(16)	24.69	(10)
(xiv)	21.94	(11)	64.78	(7)	19.45	(7)	22.79	(5)
(xv)	21.86	(10)	67.58	(13)	21.56	(10)	21.70	(3)
(xvi)	18.55	(1)	55.24	(2)	16.93	(2)	20.34	(2)

Table 3.4: *The average CPU times in seconds (and their ranks) for different types of partitioning*

Secondly, the average CPU times for partitions that involve a 2 partition is almost always very large, these times usually having high ranks. A similar comment can be made for partitions involving a 4 partition. Thus it seems that considering these large subimages (*i.e.* here images of 32 pixels by 32 pixels, or 16 pixels by 16 pixels) has a detrimental effect on the average CPU time.

In conclusion, we recommend that either a partition dividing the image into 16×16 sub-images each of size 4 pixels by 4 pixels, or no partitioning at all should be used in the case of a 64×64 image. On balance, as any saving achieved by the partition just described is very small and as such a partitioning adds to the complexity of the program, we feel that partitioning should be avoided. We have not conducted such an investigation for images of other sizes, but we feel that the same general conclusions will apply: a partition that divides the image into many small sub-images may help to reduce CPU time although not by a large amount, whereas a partition that divides the image into a few large sub-images will not be helpful.

3.6 Getting reconstructions for increasing β

Let $\hat{x}(\beta)$ be the reconstruction that minimizes the penalty function given in equation (3.9). If we use the MAP estimation technique of Greig *et al.*[17] described above to find $\hat{x}(\beta)$, and if $\beta_2 > \beta_1$, we can obtain $\hat{x}(\beta_2)$ from $\hat{x}(\beta_1)$ in a simple way, the reason for which is as follows. The smoothing parameter β only appears in the network formulation in the capacities of the arcs between the nodes that represent the n pixels, and only pixels that are neighbours have their nodes connected. Thus, an increase in β from β_1 to β_2 only affects the network by increasing the capacities of these arcs. Accordingly, any feasible flow through the network when $\beta = \beta_1$ remains feasible when $\beta = \beta_2 > \beta_1$. Hence, given $\hat{x}(\beta_1)$, we can obtain $\hat{x}(\beta_2)$ by first increasing the pixel-to-pixel capacities in the network formulation by $\beta_2 - \beta_1$ (or $D(\beta_2 - \beta_1)$ for second-order neighbours), and then running the Ford Fulkerson algorithm with the flows associated with $\hat{x}(\beta_1)$ as the initial solution. In this way it is possible to get the reconstructions for a sequence of increasing β, say $\beta = 0.1, 0.2, \ldots, 1.9, 2.0$, without having to do each minimization from the beginning separately. We shall make use of this very convenient observation in Section 3.8. Indeed, this observation is the key to our method of choosing the smoothing parameter β.

We now give an example produced by means of the feature that we have just described to show the effect of smoothing, and to motivate the work of Section 3.8. Again we consider the

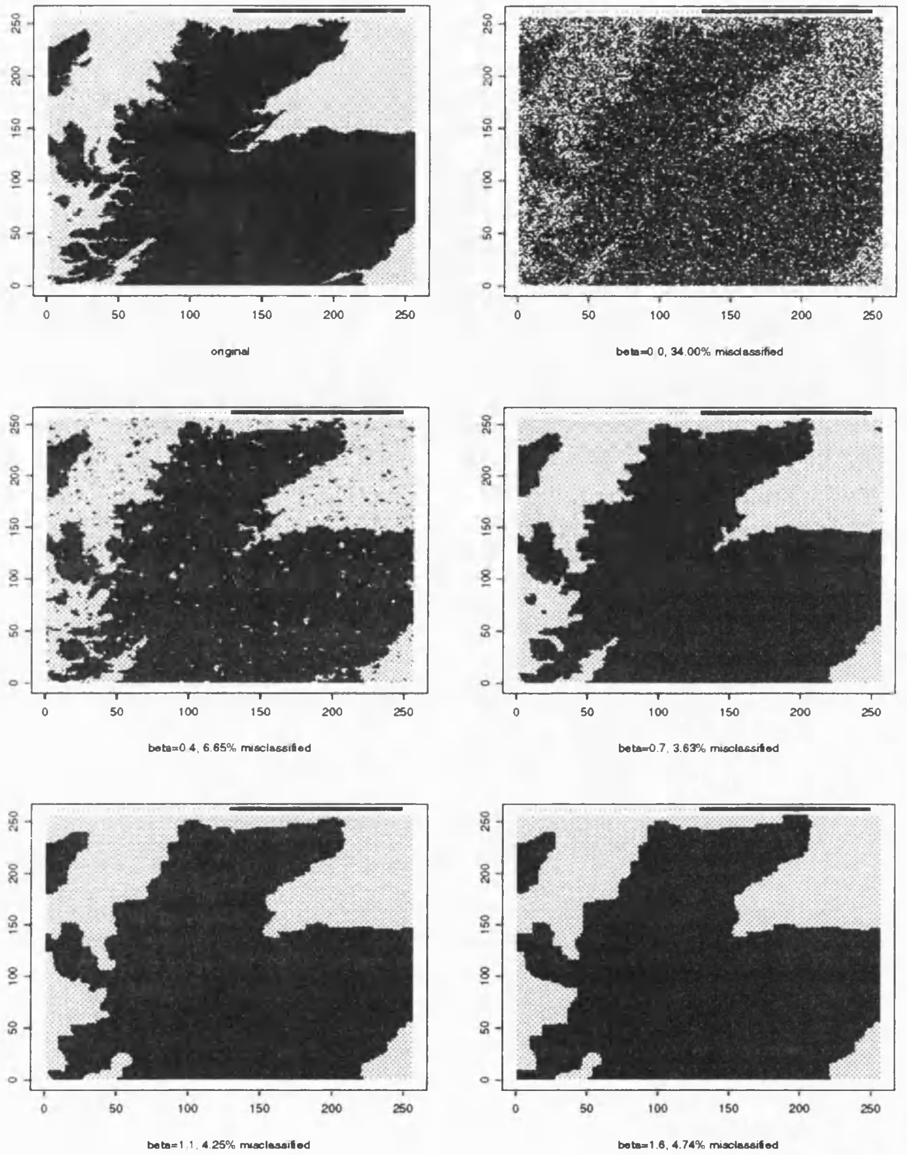
256×256 image of a part of Scotland. That image is corrupted by the addition of normal noise with variance $\kappa = 1.5$. The original image and the maximum likelihood estimator ($\beta = 0$) are shown in Figure 3.4. Figure 3.4 also shows, as examples of the effect of different amounts of smoothing, the results of the exact MAP reconstruction for some βs from the above sequence. For all βs in the sequence a nearest neighbourhood system ($D = 0.0$) was used. The reconstruction with $\beta = 0.4$ suffers from speckle error and is clearly undersmoothed. The reconstruction with $\beta = 0.7$ is very good although some of the detail of the coastline is missing. The reconstructions with $\beta = 1.1$ and $\beta = 1.6$ are still good but they are clearly oversmoothed. Since the true image is known we can also consider the percentage of pixels that are misclassified. Figure 3.5 is a graph of the percentage of misclassified pixels for the above values of β that are at least 0.3. We exclude $\beta < 0.3$ from Figure 3.5 as the reconstructions for such βs are very poor. It can be seen that the best β in terms of the number of misclassified pixels is indeed $\beta = 0.7$!

In general, however, the original image is unknown. One way of selecting the best β may be to produce reconstructions for a large number of βs and to choose the best image by eye. We have implemented a suite of programs which displays reconstructions for increasing β in real time, provided that the image is not too big. With these programs the user can sit at the console and watch as the various reconstructions are produced. The reconstruction that appears best can then be selected.

The above way of (subjectively) selecting the best β by eye may not be appropriate in all cases. Moreover, it is important to try to have an automatic method of (objectively) selecting the smoothing parameter β. Several such methods have been suggested. In Section 3.7 we briefly outline one of these due to Frigessi and Piccioni[11]. In Section 3.8 we investigate a new method for choosing the smoothing parameter β due to Seheult[35]. This method is seen to be relatively successful in certain cases. In Section 3.9 we discuss other methods that have been suggested, and we give examples to illustrate why we prefer the method of Section 3.8.

3.7 A review of the approach of Frigessi and Piccioni

Frigessi and Piccioni[11] consider the case of a binary channel, where each pixel changes colour with unknown probability ε, independently of the others. They assume that ε is unknown and they propose a method for finding estimates of both ε and β which they show are consistent if



original

beta=0.0, 34.00% misclassified

beta=0.4, 6.65% misclassified

beta=0.7, 3.63% misclassified

beta=1.1, 4.25% misclassified

beta=1.6, 4.74% misclassified

Figure 3.4: *A part of Scotland corrupted with normal noise, κ = 1.5, and reconstructed with various values of β*

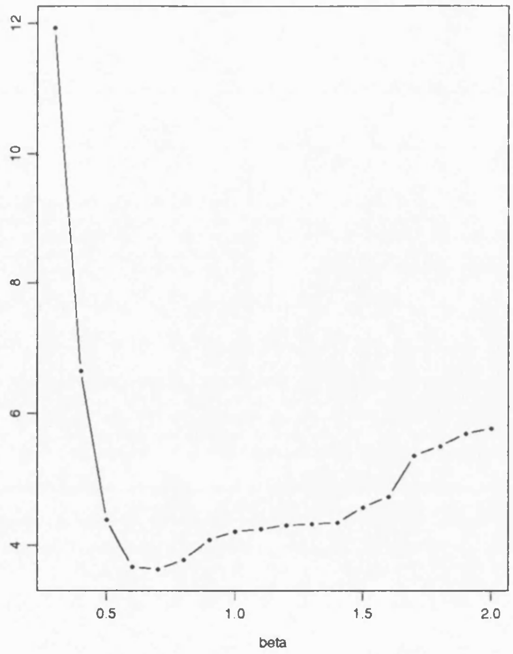


Figure 3.5: *The percentage of misclassified pixels for various values of β*

the region is regarded as having a free boundary, and easily computable. We briefly outline this method below. (The reader should note that Frigessi and Piccioni[11] set up the Ising model in such a way that their β is half our β. For consistency we convert their βs to our β throughout.) They report numerical experiments which we describe in Section 3.7.1.

The method proposed by Frigessi and Piccioni[11] is an extension of one derived from the theory of Time Series. First, for the one-dimensional case, they consider the finite lattice $\Lambda_n = \{-n, \cdots, n\}$. Next they derive estimators $(\hat{\beta}_n(Y_n), \hat{\varepsilon}_n(Y_n))$ for (β, ε) which are based on the lag-1 and lag-2 sample correlations of the data Y_n. They show that this sequence $(\hat{\beta}_n, \hat{\varepsilon}_n)$ converges, $\Pr_{\beta\varepsilon}$ almost everywhere, to (β, ε) as $n \to \infty$, for all $(\beta, \varepsilon) \in \Theta$, where Θ is the set

$$\{(\beta, \varepsilon) : \beta > 0, 0 \leq \varepsilon < 1/2\}.$$

The authors now extend the above result to a two-dimensional lattice Λ. The formulae obtained in the two-dimensional case are similar to those obtained in the one-dimensional case, except that the expression for $\hat{\beta}_\Lambda$ involves the inverse of a function that the authors call ϕ and that is strictly decreasing for positive values of its argument. A graph of this function ϕ is given in [11]; in practice ϕ is inverted numerically. We briefly describe the numerical experiments performed by the authors in Section 3.7.1.

3.7.1 Numerical experiments

Frigessi and Piccioni[11] investigate numerically the behaviour of their $(\hat{\beta}, \hat{\varepsilon})$. They consider 6 images of 128×128 pixels synthesized by applying the Gibbs sampler algorithm, as described in Section 1.4.1, with 100 raster scans, with β set to 0.2, 0.6, 1.0, 1.2, 1.6 and 2.0. Although they do not state what their initial image was, the theory tells us that asymptotically the realization obtained is independent of the initial image. The values of ε considered for the binary channel are 0.25, 0.1, 0.2, 0.3 and 0.4, although they do not consider all values of ε for all images.

The authors produce reconstructions by running the Gibbs sampler on the posterior distribution with both the estimated values $(\hat{\beta}, \hat{\varepsilon})$ and the true values (β, ε) for comparison, rather than by using the exact MAP technique of Greig *et al.*[17]. Again they use 100 raster scans. The authors present misclassification rates, μ and $\hat{\mu}$, *i.e.* the percentage of pixels wrongly assigned in each of the above two reconstructions.

The above results are presented in their Table 1. We reproduce a modified version of that table as our Table 3.5. We recall that for the model that we consider the critical value of β is

$$\beta_c = \sinh^{-1} 1 \approx 0.88.$$

The critical value is defined and discussed in Section 3.4.3: the basic idea is that for infinite images for $\beta < \beta_c$ there are no infinite patches of one type, whereas for $\beta > \beta_c$ there will always be such infinite patches. Frigessi and Piccioni[11] remark that, whereas $\hat{\varepsilon}$ shows a good precision for all values of the parameters considered, the quality of $\hat{\beta}$ drastically decreases as β increases. We find it curious that μ, the misclassification rate for reconstructions obtained using the true parameter values β and ε, is always higher than $\hat{\mu}$, the misclassification rate for reconstructions obtained using the estimated parameters $\hat{\beta}$ and $\hat{\varepsilon}$. We further remark that for images generated by the Gibbs sampler with β less than the critical value (*i.e.* $\beta = 0.2$ and $\beta = 0.6$) the corresponding value of $\hat{\beta}$ (and $\hat{\lambda}_b$) is in all cases larger that β (and λ_b), suggesting that better reconstructions are obtained in this region of β-space by so-called oversmoothing, whereas for images generated by the Gibbs sampler with β greater than the critical value (*i.e.* $\beta = 1.0$, 1.2, 1.6 and 2.0) the corresponding value of $\hat{\beta}$ (and $\hat{\lambda}_b$) is in all

β	ε	$\lambda_b = \eta\beta$	$\hat{\beta}$	$\hat{\varepsilon}$	$\hat{\lambda}_b = \hat{\eta}\hat{\beta}$	μ	$\hat{\mu}$
0.2	0.1	0.09	0.32	0.19	0.22		
	0.2	0.14	0.33	0.30	0.38		
0.6	0.1	0.27	0.64	0.13	0.33	10.11	10.09
	0.2	0.43	0.67	0.24	0.57	21.00	20.03
	0.3	0.71	0.64	0.34	0.93	32.00	30.30
1.0	0.1	0.46	0.89	0.10	0.40	5.04	5.00
	0.2	0.72	0.89	0.20	0.64	7.76	7.59
	0.3	1.18	0.90	0.31	1.09	11.37	10.99
1.2	0.1	0.55	0.92	0.10	0.41	3.23	2.84
	0.2	0.87	0.90	0.19	0.62	4.91	4.39
	0.3	1.42	0.90	0.34	1.35	7.30	7.30
	0.4	2.96	1.50	0.42	4.65	13.90	12.90
1.6	0.1	0.73	0.95	0.09	0.41	1.80	1.30
	0.2	1.15	0.95	0.19	0.65	3.10	2.60
	0.3	1.89	0.92	0.30	1.06	4.86	4.56
2.0	0.25	1.82	0.99	0.25	0.89	3.00	2.97

Table 3.5: *Table 1 of Frigessi and Piccioni (modified)*

cases but one less that β (and λ_b), suggesting that here so-called undersmoothing yields better reconstructions. (An alternative explanation may be that the Gibbs sampler algorithm that generates the original image was not run long enough to produce a genuine realization from the appropriate distribution.) We shall see a similar situation occurring in our work in both the case when the degradation mechanism is the addition of normal noise (see Section 3.8.2) and the case when it is a binary channel (see Section 3.8.3). Other features that are common to our work and that are reported in [11] are that within each β, μ and $\hat{\mu}$ increase with increasing ε (as expected), and that, for each fixed ε, μ and $\hat{\mu}$ decrease with increasing β. The latter remark suggests that smoother images are easier to deal with than rougher ones.

The results of Table 3.5 are based on only one application of the degradation process and reconstruction. We proceed in a slightly different way, which we discuss in Section 3.7.2.

3.7.2 Our approach

In our work we consider 10 different degradations, the seed used for each case being different. Because of the increased computational burden that this imposes, we consider 64×64 images, and only 4 values of β, namely 0.2, 0.6, 1.0 and 1.2. In addition to the images generated by the Gibbs sampler, we consider two 'hand-drawn' images. Reproductions and details of the images are given in Section 3.4.3.

We consider two types of degradation mechanism. The first is the addition of independent Gaussian noise with variance κ set to 0.25, 0.5 and 1.0; the second is the binary channel with ε set to 0.1, 0.2, 0.3 and 0.4. Unlike in the work of Frigessi and Piccioni[11], we assume that ε for the binary channel, and κ for the Gaussian noise are known. In Section 3.8 we outline a different method for choosing the smoothing parameter β that relies upon the key feature of the fluid flow approach to finding the exact MAP estimate outlined in Section 3.6, and we present the results of an investigation of this method.

3.8 A method for choosing the smoothing parameter β

In this section we investigate one of three suggestions of Seheult[35] for selecting the smoothing parameter β. (We briefly discuss the other two suggestions in Section 3.9.) The basic idea is to choose the reconstruction corresponding to the value of β that maximizes a function, g, defined as

$$g(\beta) = \prod_i \left\{ \sum_{z_i \in \{0,1\}} \Pr(y_i \mid z_i) \Pr(z_i \mid \hat{x}_{\partial i}(\beta); \beta) \right\} \tag{3.17}$$

where i runs over all the pixels in the image and where $\hat{x}(\beta)$ is the exact MAP reconstruction with smoothing parameter β. The set of neighbours of pixel i is denoted ∂i, and, accordingly, $\hat{x}_{\partial i}(\beta)$ represents the colours of the pixels that are the neighbours of pixel i in the exact MAP reconstruction. We use the notation $\Pr(\cdot\,; \beta)$ to indicate that the distribution depends upon the unknown parameter β.

Seheult[35] provided some justification for using the function given in equation (3.17). First, they reflect that the posterior distribution of an image x, given record y, is

$$\Pr(x \mid y; \beta) = \frac{\Pr(y \mid x) \Pr(x; \beta)}{\Pr(y; \beta)} \tag{3.18}$$

where

$$\Pr(y; \beta) = \sum_x \Pr(y \mid x) \Pr(x; \beta) \tag{3.19}$$

They think of $\Pr(y; \beta)$ as a likelihood for β given the record y. Unfortunately, in this context $\Pr(y; \beta)$ as given in equation (3.19) is computationally infeasible, and so a way of approximating it must be found.

The approximation suggested by Seheult[35] is based on a *pseudo-likelihood* approach, as advocated by Besag in [2] and [3]. We briefly outline the derivation of this approximation. Consider $\Pr(y; \beta)$ and approximate it as follows:

$$\begin{aligned}
\Pr(y; \beta) &\approx \prod_i \Pr(y_i; \beta) \quad (\textit{pseudo} \text{ step}) \\
&= \prod_i \left\{ \sum_{x_{\partial i}} \Pr(y_i \mid x_{\partial i}; \beta) \Pr(x_{\partial i}; \beta) \right\} \\
&\approx \prod_i \Pr(y_i \mid \hat{x}_{\partial i}(\beta); \beta) \quad (\text{approximate } \Pr(x_{\partial i}; \beta) \text{ as 1 at } \hat{x}_{\partial i}(\beta) \text{ and 0 otherwise}) \\
&= \prod_i \left\{ \sum_{z_i \in \{0,1\}} \Pr(y_i \mid z_i) \Pr(z_i \mid \hat{x}_{\partial i}(\beta); \beta) \right\}
\end{aligned}$$

$$= g(\beta),$$

as in equation (3.17).

Our aim now is to maximize $g(\beta)$ with respect to β. Equivalently, we can minimize $-h(\beta)$, where

$$h(\beta) = \log g(\beta)$$

and indeed we do just this as the computation of $\log g(\beta)$ is easier and more reliable than the direct computation of $g(\beta)$, as defined in (3.17). Moreover, if we judge the quality of the reconstruction of a degraded known image by the percentage of misclassified pixels, $p_{\text{mis}}(\beta)$ say, our aim would be to find the value of the smoothing parameter β that minimizes $p_{\text{mis}}(\beta)$ (which, of course, is not known in practice). Hence, it is the *minimization* of $-h(\beta)$ that we consider.

To provide some further motivation for studying $-h(\beta)$, we produce in Figure 3.6 plots of $p_{\text{mis}}(\beta)$ and $-h(\beta)$, as functions of β, for two examples: the 'hand-drawn' image (1) degraded by the addition of normal noise with variance 0.5, and the image generated by the Gibbs sampler with $\beta = 1.0$ degraded by the binary channel with $\varepsilon = 0.3$. The unbroken vertical line marks the value of β that minimizes $p_{\text{mis}}(\beta)$ (in Section 3.8.1 we shall refer to this β as β_o), whereas the broken vertical line marks the value of β that minimizes $-h(\beta)$ (in Section 3.8.1 we shall refer to this β as β_h). For the plots we restrict the range of β to a neighbourhood of the minimizer of $-h(\beta)$, and we plot the values of $p_{\text{mis}}(\beta)$ and $-h(\beta)$ at every 0.01 in that range of β. It is clear that the graphs of $(\beta, p_{\text{mis}}(\beta))$ and $(\beta, -h(\beta))$ have a similar shape. To provide some quantification of this observation we computed the value of Spearman's rank correlation coefficient r_S between $p_{\text{mis}}(\beta)$ and $-h(\beta)$ for each example. (This nonparametric statistic is invariant to monotone transformations of the data.) For the case of additive normal noise $r_S = 0.8373$ based on the 201 points displayed in the graphs, whereas for the case of the binary channel $r_S = 0.6013$ based on the 151 points displayed. Both these values of r_S are certainly significant at the 1% level, one-tailed test: we do such a test as we are interested in testing the null hypothesis that there is no relationship between p_{mis} and $-h$ against the alternative that p_{mis} increases as $-h$ increases. We point out that the graphs of $-h(\beta)$ against β are fairly typical examples of the behaviour of the function h, although larger images tend to produce smoother curves.

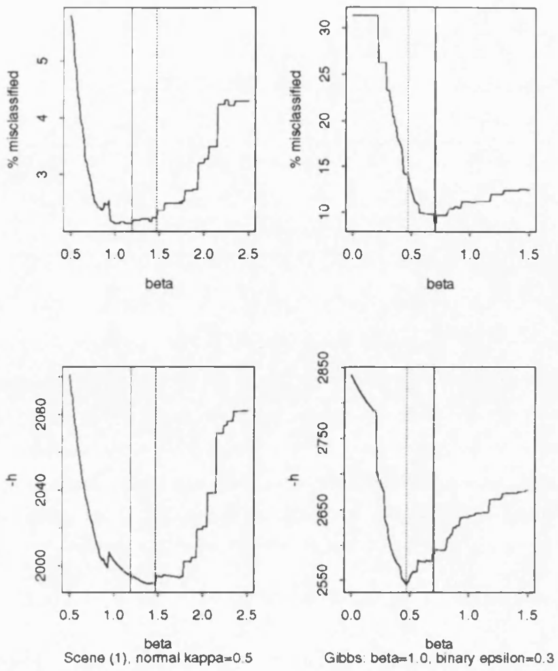


Figure 3.6: *How percentage of misclassified pixels, $p_{\text{mis}}(\beta)$, and $-h$ depend on β*

3.8.1 Experiments

Our experiments are based on the six images that were described in Section 3.4.3 and presented in Figure 3.6. For each of our six images we apply the degradation mechanisms, described in Section 3.7.2. Next we produce reconstructions for values of β in the set $\{0.0, 0.01, 0.02, \dots, 1.98, 1.99, 2.00\}$ making use of the key remarks we made in Section 3.6 about obtaining reconstructions for increasing β. We then record the value of β in this set that minimizes $-h(\beta)$. We shall refer to this β as β_h, and to the algorithm that produces a reconstruction using β_h as *algorithm-h*.

We also record the value of β in this set that corresponds to the reconstruction with the smallest number of misclassified pixels. This 'optimal' β is denoted β_o, and we shall refer to the algorithm that produces a reconstruction using β_o as *algorithm-o*. Of course, in practice β_o is unavailable. Clearly the β that minimizes $-h$ and the β that minimizes the number of misclassified pixels need not lie in the set $\{0.0, 0.01, 0.02, \dots, 1.98, 1.99, 2.00\}$. However, we feel that this approach of searching over increasing βs is a reasonable one to adopt due to its computational feasibility, as outlined in Section 3.6. Moreover, the set that we have chosen is

	Variance								
	$\kappa = 0.25$			$\kappa = 0.5$			$\kappa = 1.0$		
	with			with			with		
Image	β_o	β_h	$\Delta\%$	β_o	β_h	$\Delta\%$	β_o	β_h	$\Delta\%$
$\beta = 0.2$	15.5	18.5	19	23.5	28.2	20	30.6	35.6	16
$\beta = 0.6$	12.8	13.6	6	20.0	21.0	5	27.1	28.1	4
$\beta = 1.0$	4.6	4.9	5	6.6	6.9	5	8.4	9.0	7
$\beta = 1.2$	2.4	2.6	6	3.6	3.9	10	4.5	4.8	8
(1)	1.0	1.1	3	1.9	2.0	7	3.9	4.3	11
(2)	0.3	0.4	22	1.3	1.5	12	2.8	3.0	8

Table 3.6: *Percentage of misclassified pixels: normal noise ($\lambda_{\mathcal{N}} = 2\kappa\beta$)*

quite large, and the difference between consecutive βs in it is small. Hence, for each image, with the particular record, we have produced β_h and β_o. We repeat the procedure for 10 different records (each with the same value of the κ or ε, as appropriate). Our hope, so far based on the approximation of $\Pr(y; \beta)$ given by $g(\beta)$ and on experimental evidence such as that presented in Figure 3.6, is that (the reconstruction produced using) β_h will be similar to (the reconstruction produced using) β_o. We now discuss the results obtained: we examine the normal noise case in Section 3.8.2 and the binary channel case in Section 3.8.3. In both cases we present the results averaged over the 10 different records.

3.8.2 Results: normal noise

To begin our description of the results for the normal noise case, we consider the performance of the two algorithms, algorithm-o and algorithm-h, in terms of the number of misclassified pixels. The results themselves are given in Table 3.6. We begin our analysis of Table 3.6 by making two obvious comments. First, for each value of κ considered, algorithm-o misclassified a smaller percentage of pixels than algorithm-h. Secondly, for each algorithm, the percentage of misclassified pixels increases with increasing variance κ. For those images generated by the Gibbs sampler with values of β below the critical value (see equation (3.16)), namely $\beta = 0.2$ and $\beta = 0.6$, both algorithms perform quite badly: in all cases at least 12% of pixels are misclassified, and that figure increases to well over 30% when $\kappa = 1.0$ and algorithm-h is used. However, for the images with β greater than the critical value, namely $\beta = 1.0$ and $\beta = 1.2$, both algorithms perform very well: always less than 10% of the pixels are misclassified. We note that for both algorithms and for all values of κ, the average number of misclassified pixels decreases with increasing β for those images generated by the Gibbs sampler. This phenomenon was observed also in the case of the binary channel, see Section 3.8.3. Even better results are obtained for the 'hand-drawn' images: in all cases less than 5% of the pixels are misclassified. As we have stated in Section 3.8.1, algorithm-o is not applicable in practice. However, it seems that on average algorithm-h does not perform very much worse. To facilitate comparison we give the approximate value of $\Delta\%$, the increase in the average number of misclassified pixels when algorithm-h is used as opposed to algorithm-o, expressed as a percentage of the average number of pixels misclassified by algorithm-o. Although there is not a clear pattern, $\Delta\%$ is only greater than 20% in one case, and it is often less than 10%. Accordingly, algorithm-h seems to perform well in the case of additive normal noise, at least for the values of κ considered.

We now move on to discuss the values of the smoothing parameter selected by the two algorithms. We recall that β_o is the value of the smoothing parameter selected by algorithm-o, whereas β_h is the value selected by algorithm-h. We present the values for the case of normal noise in Table 3.7. A discussion of the values found is, however, not easy as clear patterns do not emerge. First, for the images generated by the Gibbs sampler and for each value of the variance κ, we consider how the behaviour of β_o and β_h depends upon the value of β used to generate the image. For each value of κ, β_o (and $2\kappa\beta_o$) increases as the β used for the images generated by the Gibbs sampler increases. Similarly, for each κ, β_h (and $2\kappa\beta_h$) increase as that β increases. Secondly, for a given image, we discuss how the behaviour of β_o and $2\kappa\beta_o$, and β_h and $2\kappa\beta_h$ depends upon κ. It can be observed that, for each image, β_o decreases as κ increases, whereas $2\kappa\beta_o$ increases as κ increases. The behaviour of β_h and $2\kappa\beta_h$ with κ is less clear. For the images generated by the Gibbs sampler with $\beta = 0.6$ and $\beta = 1.0$ and the 'hand-drawn' images, β_h decreases as κ increases, whereas for each image except the image generated with $\beta = 0.2$, $2\kappa\beta_h$ increases as κ increases. Thirdly, we turn our attention to the relationship between the β used for the images generated by the Gibbs sampler and β_o. We observe a phenomenon similar to that described in Section 3.7.1, where we discussed the work of Frigessi and Piccioni[11]. For values of β less than the critical value β_c, $\beta_o > \beta$ (and $2\kappa\beta_o > 2\kappa\beta$) for all values of κ with one exception. That occurs when $\kappa = 1.0$ and with the image generated with $\beta = 0.6$,

	Variance											
	$\kappa = 0.25$						$\kappa = 0.5$					
Image	$2\kappa\beta$	β_o	$2\kappa\beta_o$	β_h	$2\kappa\beta_h$	δ%	$2\kappa\beta$	β_o	$2\kappa\beta_o$	β_h	$2\kappa\beta_h$	δ%
$\beta = 0.2$	0.1	0.64	0.32	0.12	0.06	-81	0.2	0.60	0.60	0.14	0.14	-77
$\beta = 0.6$	0.3	0.80	0.40	0.56	0.28	-30	0.6	0.65	0.65	0.44	0.44	-32
$\beta = 1.0$	0.5	0.88	0.44	0.94	0.47	7	1.0	0.76	0.76	0.80	0.80	5
$\beta = 1.2$	0.6	1.06	0.53	1.10	0.55	4	1.2	0.89	0.89	1.15	1.15	29
(1)		1.74	0.87	1.80	0.90	3		1.27	1.27	1.21	1.21	-5
(2)		1.86	0.93	1.90	0.95	2		1.49	1.49	1.50	1.50	1

	Variance					
	$\kappa = 1.0$					
Image	$2\kappa\beta$	β_o	$2\kappa\beta_o$	β_h	$2\kappa\beta_h$	δ%
$\beta = 0.2$	0.4	0.48	0.96	0.06	0.12	-88
$\beta = 0.6$	1.2	0.52	1.04	0.34	0.68	-35
$\beta = 1.0$	2.0	0.68	1.36	0.77	1.54	13
$\beta = 1.2$	2.4	0.88	1.76	0.84	1.68	-5
(1)		0.88	1.76	0.90	1.80	2
(2)		1.02	2.04	0.96	1.92	-6

Table 3.7: *Values of the smoothing parameter: normal noise ($\lambda_{\mathcal{N}} = 2\kappa\beta$, $\delta = (\beta_h - \beta_o)/\beta_o$)*

when β_o is 0.52. As in Section 3.7.1, it seems that better reconstructions are obtained by so-called oversmoothing for values of β less than β_c. On the other hand, for values of β greater than the critical value, $\beta_o < \beta$ (and $2\kappa\beta_o < 2\kappa\beta$) for all values of κ. Hence, it seems that better reconstructions are obtained by so-called undersmoothing for values of β greater than β_c. (Again, an alternative explanation may be that the Gibbs sampler algorithm that generates the original image was not run long enough to produce a genuine realization from the appropriate distribution.) We also point out that β_h is less than β in all cases. Finally, we examine the relationship between β_o and β_h. To do this we present the value of $\delta = (\beta_h - \beta_o)/\beta_o$ as a percentage. It is difficult to say much about δ. However, δ is always negative for the images generated with βs that are less than the critical value, and positive for the images generated with βs that are greater than β_c, except in the case when $\kappa = 1.0$ and $\beta = 1.2$. In brief, we can state that $|\delta|$ seems very large for all values of κ for the image generated with $\beta = 0.2$, quite large for $\beta = 0.6$, and small (under 10%) for the images generated with $\beta = 1.0$ and $\beta = 1.2$, with only a couple of exceptions. The value of $|\delta|$ seems to be small for the 'hand-drawn' images, but there is no obvious pattern to the sign of δ.

3.8.3 Results: binary channel

Again we start by considering the performance of the two algorithms in terms of the number of misclassified pixels. Our results for the case of the binary channel are given in Table 3.8. Again we see immediately from Table 3.8 that, for each value of ε, algorithm-o misclassified a smaller percentage of pixels than algorithm-h. We also see that for each algorithm, the number of misclassified pixels increases with increasing ε. A more detailed examination of Table 3.8 causes us to make comments that are very similar to those made in Section 3.8.2. For the images with $\beta = 0.2$ and $\beta = 0.6$, both algorithms again perform quite badly: in all cases at least about 10% of pixels are misclassified and that figure increases to well over 40% when $\varepsilon = 0.4$ and algorithm-h is used. On the other hand, for the images with $\beta = 1.0$ and $\beta = 1.2$, both algorithms perform well: when $\varepsilon = 0.1$ and $\varepsilon = 0.2$ the average number of misclassified pixels is less than 8%, when $\varepsilon = 0.3$ the average number never exceeds 11%, and even when $\varepsilon = 0.4$ the average number is less than 21%. Again for both algorithms and for all values of ε, the average number of misclassified pixels decreases with increasing β for those images generated by the Gibbs sampler. This phenomenon can also be observed for the binary case in the work of Frigessi and Piccioni[11]. We also saw it in Section 3.8.2, for the normal noise. Again, even better results are

	Probability of change									
	$\varepsilon = 0.1, \eta = 0.46$					$\varepsilon = 0.2, \eta = 0.72$				
	with			μ	$\hat{\mu}$	with			μ	$\hat{\mu}$
Image	β_o	β_h	$\Delta\%$	from [11]		β_o	β_h	$\Delta\%$	from [11]	
$\beta = 0.2$	10.0	13.4	34			20.1	29.3	46		
$\beta = 0.6$	9.9	11.3	15	10.1	10.1	18.8	22.6	20	21.0	20.0
$\beta = 1.0$	4.6	5.1	9	5.0	5.0	7.0	7.9	13	7.7	7.6
$\beta = 1.2$	2.7	3.3	26	3.2	2.8	3.6	4.6	30	4.9	4.4
(1)	0.9	0.9	3			2.3	2.4	7		
(2)	0.6	0.6	2			1.5	2.0	32		
	$\varepsilon = 0.3, \eta = 1.18$					$\varepsilon = 0.4, \eta = 2.47$				
	with			μ	$\hat{\mu}$	with			μ	$\hat{\mu}$
Image	β_o	β_h	$\Delta\%$	from [11]		β_o	β_h	$\Delta\%$	from [11]	
$\beta = 0.2$	30.2	39.0	29			40.1	45.0	12		
$\beta = 0.6$	27.8	29.6	7	32.0	30.3	38.8	40.3	4		
$\beta = 1.0$	9.7	10.7	11	11.4	11.0	14.2	20.5	45		
$\beta = 1.2$	4.9	5.7	16	7.3	7.3	8.7	15.9	83	13.9	12.9
(1)	5.0	5.3	6			11.5	13.3	16		
(2)	4.4	5.1	16			12.1	15.8	30		

Table 3.8: *Percentage of misclassified pixels: binary channel ($\lambda_b = \eta\beta$, $\eta = 1 / \log((1 - \varepsilon) / \varepsilon)$)*

obtained for the 'hand-drawn' images: for $\varepsilon = 0.1$ the average number of misclassified pixels is less than 1%, and even when $\varepsilon = 0.4$ this average number is still less than 16%. Unfortunately, algorithm-o is not applicable in practice. However, it seems that on average algorithm-h does not perform much worse, although perhaps here the performance of the latter algorithm is not as good as it was in the case of normal noise. Again there is not a clear pattern in the values of $\Delta\%$, but sometimes $\Delta\%$ can be quite large, reaching 83% on one occasion. Nevertheless, algorithm-h seems to perform reasonably well, at least for the values of ε considered. We include in Table 3.8, where possible, the percentage of misclassified pixels μ and $\hat{\mu}$ obtained by Frigessi and Piccioni [11] and reproduced in our Table 3.5. We discuss μ and $\hat{\mu}$ in Section 3.7. The results are not directly comparable for three reasons. First, Frigessi and Piccioni[11] only consider one realization of the degradation process. Secondly, the images used in [11] are different from the images that we used. Indeed, their images are 128×128, whereas ours are 64×64. Thirdly, our methodology is quite different. Frigessi and Piccioni[11] estimate both β and ε as $\hat{\beta}$ and $\hat{\varepsilon}$, and then use these values in order to reconstruct the true image by means of the Gibbs sampler. They simulate 100 images from the distribution $\Pr(x \mid y; \hat{\beta}, \hat{\varepsilon})$ and select at each pixel the colour that occurs most often in these simulations. This method, known as MPM, was discussed in Section 1.4, where it was seen to be the preferred method (compared to MAP and ICM) when the quantity of interest is the number of misclassified pixels. Moreover, their method of reconstruction by means of the Gibbs sampler should yield good reconstructions for images produced initially by the Gibbs sampler. In our approach, we assume ε is known and estimate β by β_h. Our reconstruction $\hat{x}(\beta_h)$ is then that x that minimizes a certain penalty function, as given in equation (3.13), or equivalently any monotone transformation of it. Thus our problem can be thought of as only depending on one parameter. The same cannot be said for the approach of Frigessi and Piccioni[11] because sampling from

$$\Pr(x \mid y; \hat{\beta}, \hat{\varepsilon}) \propto \exp\left\{\frac{1}{\hat{\eta}}\|y - x\|^2 + \hat{\beta}\Phi(x)\right\}$$

is different from sampling from

$$\Pr_2(x \mid y; \hat{\beta}, \hat{\varepsilon}) \propto \exp\left\{\|y - x\|^2 + \hat{\eta}\hat{\beta}\Phi(x)\right\},$$

where $\hat{\eta} = 1/\log((1-\hat{\varepsilon})/\hat{\varepsilon})$, $\|x - y\|^2 = \sum_{i=1}^{n}(x_i - y_i)^2$ and $\Phi(x) = v^{(1)}(x) + D\,v^{(2)}(x)$, as in Section 3.4. However, it is interesting to note that there is little difference between our results

	Probability of change											
	$\varepsilon = 0.1, \eta = 0.46$						$\varepsilon = 0.2, \eta = 0.72$					
Image	$\eta\beta$	β_o	$\eta\beta_o$	β_h	$\eta\beta_h$	$\delta\%$	$\eta\beta$	β_o	$\eta\beta_o$	β_h	$\eta\beta_h$	$\delta\%$
$\beta = 0.2$	0.09	0.62	0.28	0.55	0.25	-11	0.14	0.65	0.47	0.35	0.25	-46
$\beta = 0.6$	0.27	0.81	0.37	0.55	0.25	-32	0.43	0.69	0.50	0.49	0.35	-29
$\beta = 1.0$	0.46	0.81	0.37	0.88	0.40	9	0.72	0.73	0.53	0.85	0.61	16
$\beta = 1.2$	0.55	0.90	0.41	1.10	0.50	22	0.87	0.80	0.58	1.10	0.79	38
(1)		1.80	0.82	1.63	0.74	-9		1.12	0.87	1.28	0.92	14
(2)		1.89	0.86	1.82	0.83	-4		1.15	0.83	1.43	1.03	24
	$\varepsilon = 0.3, \eta = 1.18$						$\varepsilon = 0.4, \eta = 2.47$					
Image	$\eta\beta$	β_o	$\eta\beta_o$	β_h	$\eta\beta_h$	$\delta\%$	$\eta\beta$	β_o	$\eta\beta_o$	β_h	$\eta\beta_h$	$\delta\%$
$\beta = 0.2$	0.24	0.44	0.52	0.21	0.25	-52	0.49	0.24	0.59	0.10	0.25	-58
$\beta = 0.6$	0.71	0.46	0.54	0.34	0.40	-26	1.48	0.25	0.62	0.19	0.47	-24
$\beta = 1.0$	1.18	0.54	0.64	0.70	0.83	30	2.47	0.28	0.69	0.42	1.04	50
$\beta = 1.2$	1.42	0.72	0.85	0.70	0.83	-3	2.96	0.30	0.74	0.40	0.99	33
(1)		0.76	0.90	0.73	0.86	-4		0.39	0.96	0.37	0.91	-5
(2)		0.75	0.89	0.83	0.98	11		0.31	0.76	0.32	0.81	3

Table 3.9: *Values of the smoothing parameter: binary channel ($\lambda_b = \eta\beta$, $\delta = (\beta_h - \beta_o) / \beta_o$)*

and those of [11].

As we did for additive normal noise in Section 3.8.2, we now discuss the values of the smoothing parameter selected by the two algorithms. We recall that β_o is the value of the smoothing parameter selected by algorithm-o, whereas β_h is the value selected by algorithm-h, and we present the values for the case of the binary channel in Table 3.9. A discussion of the values found is, however, even more difficult here than it was in the case of additive normal noise, although the conclusions that we reach are broadly similar. We do, however, try to follow the format that we used in Section 3.8.2. First, for the images generated by the Gibbs sampler and for each value of the probability of change ε, we consider how the behaviour of β_o and β_h depends upon the value of β used to generate the image. For each value of ε, β_o (and $\eta\beta_o$) increases as the β used for the images generated by the Gibbs sampler increases.

Similarly, for each ε, β_h (and $\eta\beta_h$) increase as that β increases, except in the case when $\varepsilon = 0.4$ and $\beta = 1.2$. Secondly, for a given image, we discuss how the behaviour of β_o and $\eta\beta_o$, and β_h and $\eta\beta_h$ depends upon ε. It can be observed that, for each image except the one generated with $\beta = 0.2$, β_o decreases as ε increases, whereas $\eta\beta_o$ increases as ε increases for all images except the one generated with $\beta = 1.2$ and image (2). For all images β_h decreases as ε increases, whereas for all image except the 'hand-drawn' images $\eta\beta_h$ increases (or remains the same) as ε increases. In Section 3.7.1, where we discussed the work of Frigessi and Piccioni[11], we saw that their estimate of β, $\hat{\beta}$, remained fairly constant over all values of ε causing $\hat{\eta}\hat{\beta}$ to increase as ε increased. Thirdly, we turn our attention to the relationship between the β used for the images generated by the Gibbs sampler and β_o. Again, we observe a phenomenon similar to that described in [11]. For values of β less than β_c, $\beta_o > \beta$ (and $\eta\beta_o > \eta\beta$) for all values of ε, except $\varepsilon = 0.3$ and $\varepsilon = 0.4$ with the image generated with $\beta = 0.6$. As in Section 3.7.1, it seems that in general better reconstructions are obtained by so-called oversmoothing for values of β less than β_c. On the other hand, for values of β greater than the critical value, $\beta_o < \beta$ (and $\eta\beta_o < \eta\beta$) for all values of ε. Hence, it seems that better reconstructions are obtained by so-called undersmoothing for values of β greater than β_c. (Again, an alternative explanation may be that the Gibbs sampler algorithm that generates the original image was not run long enough to produce a genuine realization from the appropriate distribution.) We also point out that β_h is less than β in all cases except for the image generatc by the Gibbs sampler with $\beta = 0.2$ when $\varepsilon = 0.1$, $\varepsilon = 0.2$ and $\varepsilon = 0.3$. Finally, we examine the relationship between β_o and β_h by presenting the value of $\delta = (\beta_h - \beta_o)/\beta_o$ as a percentage. It is difficult to say much about δ. However, δ is always negative for the images generated with βs that are less than the critical value, and positive for the images generated with βs that are greater than β_c, except in the case when $\varepsilon = 0.3$ and $\beta = 1.2$. There is no clear pattern about $|\delta|$, although $|\delta|$ seems quite low for the 'hand-drawn' images, as it did for the normal noise. An exception to this statement occurs with image (2) and $\varepsilon = 0.2$.

3.8.4 Conclusions

In this section we attempt to summarize as far as possible the results that we obtained from our experiments.

First, we present some conclusions about the behaviour of algorithm-h and algorithm-o based on the percentage of misclassified pixels. Algorithm-h seems to perform quite badly for those images generated by the Gibbs sampler that have $\beta < \beta_c$. However, this algorithm seems to do well for those images generated by the Gibbs sampler that have $\beta > \beta_c$, and very well for the hand drawn images, (1) and (2). Thus, it seems that the smoother the original image, the better algorithm-h performs. These remarks apply in both the case of normal noise and the case of the binary channel. For normal noise, algorithm-h seems to perform only slightly worse that algorithm-o, whereas for the binary channel the difference between the two algorithms is more noticeable.

Secondly, we present some conclusions about the behaviour of the two algorithms based on the value of the smoothing parameter β, or λ (where we mean $\lambda_{\mathcal{N}}$ in the case of normal noise and λ_b in the case of the binary channel). Our conclusions here are general impressions, as there are some places where they do not hold. We found that β_o and β_h increase as the parameter β used for the Gibbs sampler increases. For the case of normal noise β_o decreases as κ increases (and the corresponding $\lambda_{\mathcal{N}}$ increases as κ increases). There is no such pattern for β_h. We can make a similar statement in the case of the binary channel, but here our conclusions apply to both β_h and β_o, and the corresponding λ_b s. We found that both β_h and β_o decrease as ε increases (and the corresponding λ_b increases as ε increases). We also saw that for both degradation mechanisms the images generated by the Gibbs sampler with parameter β had $\beta_o > \beta$ when $\beta < \beta_c$, and $\beta_o < \beta$ when $\beta > \beta_c$. We also point out that $\beta_h < \beta$ in almost all cases: and very often we have that

$$\beta < \beta_c \quad \Rightarrow \quad \beta_h < \beta < \beta_o$$

$$\beta > \beta_c \quad \Rightarrow \quad \beta_o < \beta_h < \beta.$$

Finally, although it is difficult to make any comparison between our results and those of Frigessi and Piccioni[11] for the binary channel case, it seems that the results obtained are quite similar. However, a full comparison between our method and the method of [11] should be the subject of further work.

3.9 Other suggestions for choosing the smoothing parameter

Seheult[35] suggested two other methods for choosing the smoothing parameter β. In Section 3.9.1 we outline these two methods and try to give some justification for them. We also state some theoretical results of only limited value. In Section 3.9.2 we briefly present some problems that we have found with these two methods.

3.9.1 Two possible methods

Both of the other methods suggested by Seheult[35] are based on approximating the prior distribution $\Pr(x; \beta)$ as follows:

$$\Pr(x; \beta) \approx \prod_{i=1}^{n} \Pr(x_i \mid x_{\partial i}; \beta). \tag{3.20}$$

Next, under the assumption that, given x, the records are independent, we have

$$\Pr(y \mid x) = \prod_{i=1}^{n} \Pr(y_i \mid x_i). \tag{3.21}$$

We can combine expression (3.20) with expression (3.21) to give an approximation to the numerator of equation (3.18):

$$\Pr(y \mid x) \Pr(x; \beta) \approx \prod_{i=1}^{n} \Pr(y_i \mid x_i) \Pr(x_i \mid x_{\partial i}; \beta).$$

Now replace the unknown x by its MAP estimate $\hat{x}(\beta)$ to get a new approximation of equation (3.18), which we shall call $g_2(\beta)$:

$$\begin{aligned} \Pr(y \mid x) \Pr(x; \beta) &\approx \prod_{i=1}^{n} \Pr(y_i \mid \hat{x}_i(\beta)) \Pr(\hat{x}_i(\beta) \mid \hat{x}_{\partial i}(\beta); \beta) \\ &= g_2(\beta). \end{aligned} \tag{3.22}$$

We have seen that the denominator of equation (3.18), as defined in equation (3.19), can be approximated by $g(\beta)$, which is defined in equation (3.17). Accordingly, we now have

$$\Pr(x \mid y; \beta) \approx \frac{g_2(\beta)}{g(\beta)}.$$

As before, take logs to get

$$\begin{aligned} \log \Pr(x \mid y; \beta) &\approx h_2(\beta) - h(\beta) \\ &= h_3(\beta) \end{aligned}$$

where $h(\beta) = \log g(\beta)$, $h_2(\beta) = \log g_2(\beta)$, and $h_3(\beta) = h_2(\beta) - h(\beta)$. Seheult[35] suggests plotting $h_3(\beta)$ as a function of β and choosing the β that maximizes it.

Seheult[35] also suggests doing the same with $g_2(\beta)$ (or $h_2(\beta)$). We offer a justification for doing this based upon the method of Section 5.1.2 of Besag[3]. This method applies to general images and not just binary images. First, we outline a modification of that method in which estimates of the image x are produced by the method of Iterated Conditional Modes (ICM), as described in Section 1.6.2.

1. For some value of β, call it β_{old}, carry out a single cycle of ICM to produce an estimate of x, $x(\beta_{\text{old}})$ say, that maximizes (approximately)

$$\prod_{i=1}^{n} \Pr(y_i \mid x_i) \Pr(x_i \mid x_{\partial i}; \beta_{\text{old}}). \tag{3.23}$$

2. Select β_{new} to maximize

$$\prod_{i=1}^{n} \Pr(y_i \mid x_i(\beta_{\text{old}})) \Pr(x_i(\beta_{\text{old}}) \mid x_{\partial i}(\beta_{\text{old}}); \beta) \tag{3.24}$$

as a function of β.

3. Set β_{old} to β_{new}.

4. Go to Step 1.

We make the following observations:

- In Step 1 we often take the initial value of β_{old} to be 0. For both MAP and ICM the estimate $x(0)$ that is produced by this value of β maximizes $\Pr(y \mid x)$ and is sometimes referred to as the maximum likelihood estimate. It uses no spatial information.
- Expression (3.23) is in fact an approximation to what happens in ICM. For each cycle of ICM all the pixels are visited in turn (according to a raster scan, for example) and at pixel i we select x_i to maximize

$$\Pr(y_i \mid x_i) \Pr(x_i \mid x_{\partial i}; \beta_{\text{old}}).$$

  We then update x before moving on to the next pixel.

- The maximization of expression (3.24) as a function of β in Step 2 is equivalent to the maximization of

  $$\prod_{i=1}^{n} \Pr(x_i(\beta_{\text{old}}) \mid x_{\partial i}(\beta_{\text{old}}); \beta),$$

  as advocated by Besag[3] in his Section 5.1.2, as the first term of (3.24) does not depend upon β.

The idea of Seheult[35] makes use of the similarity between expression (3.23) and expression (3.24). He combines Step 1 and Step 2 in such a way as to reduces the problem to that of a simple maximization over $\beta \geq 0$ by replacing $x(\beta_{\text{old}})$ in (3.24) by $\hat{x}(\beta)$, the MAP estimate for the parameter β, to get

$$\prod_{i=1}^{n} \Pr(y_i \mid \hat{x}_i(\beta)) \Pr(\hat{x}_i(\beta) \mid \hat{x}_{\partial i}(\beta); \beta),$$

our $g_2(\beta)$.

Because of their complexity it seems difficult, if not impossible, to perform any meaningful analysis on the quantities $h(\beta)$, $h_2(\beta)$ and $h_3(\beta)$. We know from consideration of the penalty function (see equations (3.10) and (3.14)) that $\hat{x}(\beta)$ becomes smoother as β increases and eventually tends to an image of one colour, c say, which is either 0 or 1. In this case we can show that

$$\left.\begin{array}{l} h(\beta) \rightarrow \\ h_2(\beta) \nearrow \end{array}\right\} -\frac{n}{2}\log(2\pi\kappa) - \frac{1}{2\kappa}\sum_i (y_i - c)^2 \text{ as } \beta \rightarrow \infty, \tag{3.25}$$

for normal noise, and

$$\left.\begin{array}{l} h(\beta) \rightarrow \\ h_2(\beta) \nearrow \end{array}\right\} \#(y_i = c)\log\varepsilon + \#(y_i \neq c)\log(1-\varepsilon) \text{ as } \beta \rightarrow \infty, \tag{3.26}$$

for the binary channel, where $\#(y_i = c)$ means the total number of pixels whose record takes the value c, and that

$$h_3(\beta) \nearrow 0 \text{ as } \beta \rightarrow \infty, \tag{3.27}$$

for both normal noise and the binary channel.

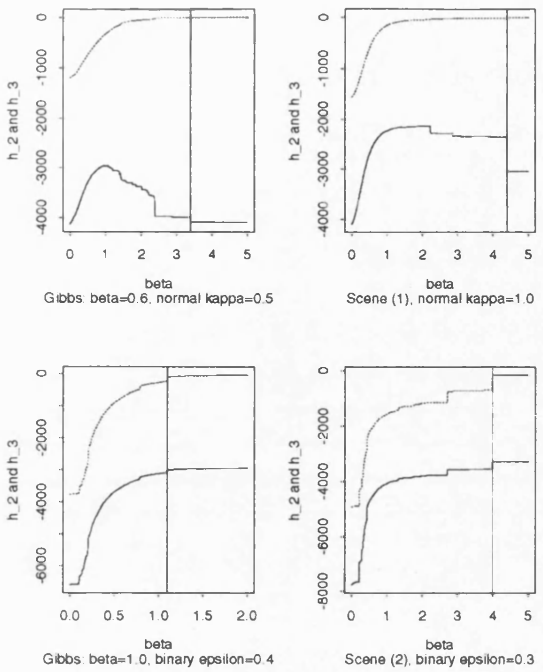


Figure 3.7: *h_2 (unbroken line) and h_3 (broken line) as a function of β*

Since images of one colour are of no interest and occur for values of the smoothing parameter β that are much higher than would ever be considered for reconstructions, such analysis serves very little purpose except to provide some check of the computer programs.

3.9.2 Some examples

We tried computing $h_2(\beta)$ and $h_3(\beta)$ for many examples and, in this section, we briefly describe some problems that we have found with using these two functions. We illustrate our claims with four examples: in the first we consider the image generated by the Gibbs sampler with $\beta = 0.6$, and corrupted by normal noise with variance $\kappa = 0.5$; in the second we consider image (1) corrupted by normal noise with variance $\kappa = 1.0$; in the third we consider the image generated by the Gibbs sampler with $\beta = 1.0$, and corrupted by the binary channel with $\varepsilon = 0.4$; finally in the fourth we consider image (2) corrupted by the binary channel with $\varepsilon = 0.3$. We present the graphs in Figure 3.7. In all four graphs the vertical line indicates the smallest value of β for which an image of one colour is achieved. The analysis of Section 3.9.1 tells us that both curves are monotonically increasing after this value of β. In fact the increase is very slow and is hardly noticeable from the graphs. We use an unbroken line for h_2 and a broken line for h_3.

The problem with $h_3(\beta)$ is clearly illustrated in all four graphs: it is monotonically increasing in β. Indeed in our experience we have never found a case when $h_3(\beta)$ is not monotonic increasing. Thus, the curve h_3 cannot be used for choosing the smoothing parameter. In the examples the function $h_2(\beta)$ is seen to be monotonically increasing in the case of the binary channel: we have never found an example in the case of the binary channel when the function h_2 is not monotonically increasing. Our experience with normal noise is almost the same in that for many cases the function h_2 is monotonically increasing. However, sometimes it does display a maximum, and the two examples with normal noise illustrate this. The value of β at such a maximum tends to produce reconstructions that are oversmoothed. However, it is felt that further investigation of h_2 may be fruitful. Moreover, in practice it may be worth checking to see whether h_2 has a maximum. If it does, the exact MAP reconstruction at that maximum, $\hat{x}\,(\beta_{h_2})$ say, should be considered along with $\hat{x}\,(\beta_h)$.

3.9.3 Further comments

There are of course many other ways of choosing the smoothing parameter. Some of these were described in Section 1.8. It is hoped that the comparison of some of these methods with our algorithm-h in the particular case of the binary image will be the subject of further research by the author.

3.10 Comparing the exact MAP estimate with the simulated annealing reconstruction

Finally in this chapter we motivate the work of Chapter 4 by comparing—in one example only—the exact MAP estimate of the image with estimates produced by simulated annealing and ICM. The example that we use is the image of a part of Scotland example that we first considered in Section 3.6.

Jubb[24] compares the approximate MAP reconstruction given by simulated annealing with the exact MAP estimate. He does not, however, give any values for the penalty function (3.9), but states the number of pixels that differ between the reconstructions. Here we give the value of the penalty function and show that for the Scotland example simulated annealing does not find the minimum of the penalty function (3.9), with $D = 0.0$ (*i.e.* first-order model for the prior

Algorithm		Misclassified pixels		Penalty function	
Type	β	Number	%	Value	% increase
Maximum likelihood estimate	0	22281	34.00		
Exact MAP	0.7	2377	3.63	34284.70	
Annealing: geometric	0.7	2363	3.61	34335.07	0.15
Annealing: logarithmic	0.7	2631	4.01	34495.37	0.61
ICM	0.7	7599	11.60	37752.50	10.11

Table 3.10: *Reconstructions of a part of Scotland by different algorithms*

distribution).

Throughout this section we set $\beta = 0.7$, as this is the value that minimized the function $-h(\beta)$ over the set $\{0.0, 0.1, 0.2, \ldots, 1.8, 1.9, 2.0\}$, and that gave the best reconstruction when the exact MAP algorithm was used from the point of view of both a subjective judgement and the number of misclassified pixels. In Table 3.10 we give the value of the penalty function (3.9) achieved by the exact MAP estimate. This value of 34284.70 is the global minimum of the penalty function. For the reconstructions obtained by simulated annealing we used 250 sweeps of simulated annealing followed by ICM to convergence. We considered two temperature schedules: a logarithmic schedule suggested by Geman and Geman[12] with

$$\tau(t) = \frac{3}{\log(1+t)},$$

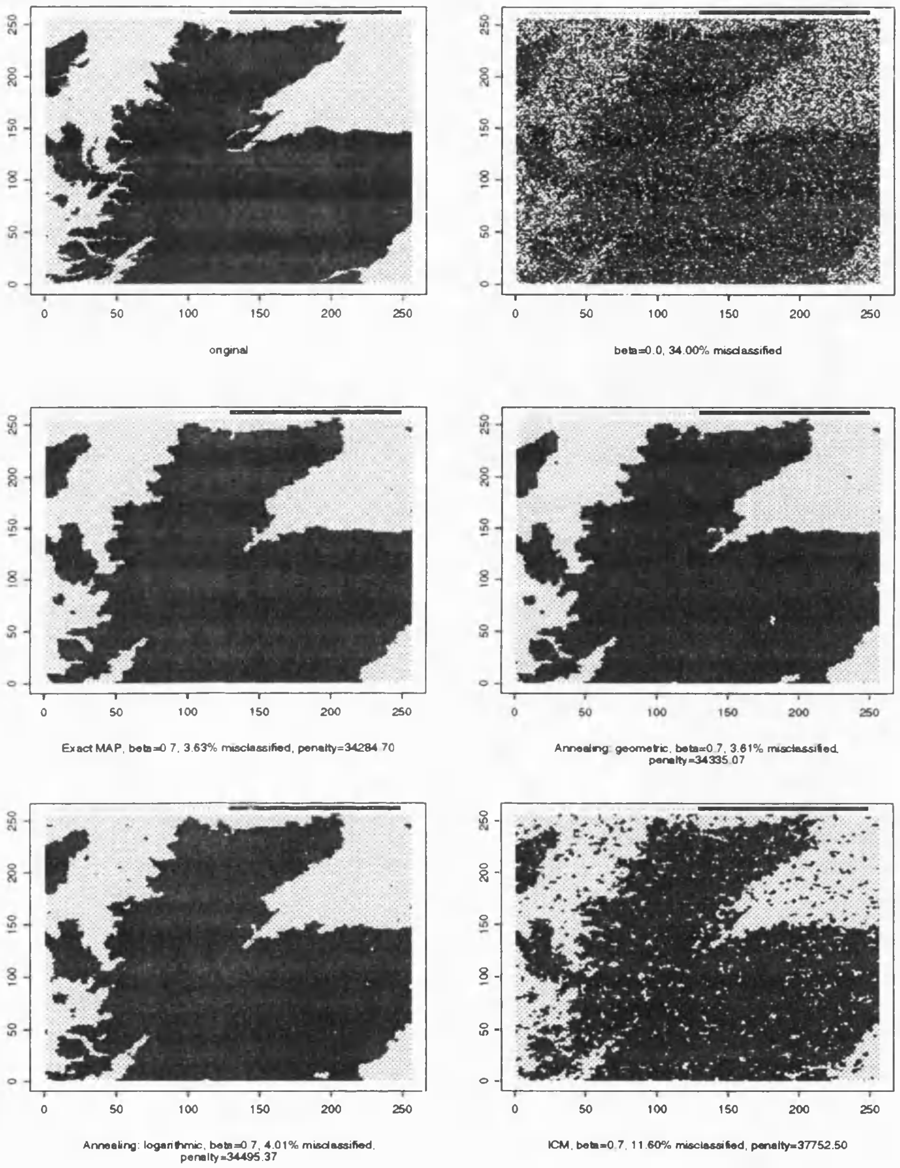
where $\tau(t)$ is the temperature used for the tth iteration, and a geometric schedule which takes the form

$$\tau(t) = A\rho^{t-1},$$

where the constants A and ρ are selected so that the initial temperature, $\tau(1)$, is the same as the initial temperature for the logarithmic temperature schedule, namely $3 / \log(2) = 4.32$, while the final temperature, $\tau(250)$, is 0.01. For comparison we point out the final temperature for the logarithmic schedule is considerably higher at 0.543. We also consider the reconstruction given by ICM alone. The value of the penalty function achieved and the number and percentage of misclassified pixels for both annealing examples and the ICM example are also given in Table 3.10. The number and percentage of misclassified pixels in the reconstruction that uses no spatial information, the maximum likelihood estimate, is also given. From Table 3.10 we see that both annealing examples fail to find the exact MAP estimate. The geometric temperature schedule does better than the logarithmic schedule producing a reconstruction that has only slightly fewer misclassified pixels and only a little higher penalty function than the exact MAP reconstruction. The ICM algorithm does substantially worse than both annealing algorithms, both in terms of the number of misclassified pixels and in terms of the value of the penalty function. To illustrate these points further we reproduce in Figure 3.8 the reconstructions themselves along with the original image and the maximum likelihood estimate. The two images produced by simulated annealing are not very different from that produced by exact MAP: the image produced by the geometric schedule is especially similar and provides us with a very satisfactory reconstruction; the image produced by the logarithmic schedule is not as good as that produced by the geometric schedule in that it has a high level of speckle error. The ICM reconstruction is clearly unsatisfactory for this example. We should point out that it is not always the case that for fixed β as the penalty function decreases so the number of misclassified pixels decreases. The question of how to assess the quality of a reconstruction and how to represent this assessment mathematically are not of concern in this chapter, but is discussed briefly in Section 1.7.

3.11 Conclusions

In this chapter we have extended the work of Greig *et al.* in [2] and [17] and of Jubb in [24] on exact maximum *a posteriori* estimation for binary images. Throughout we considered two types of degradation mechanism, the addition of normal noise and the corruption by a binary channel. We showed that the approach of maximizing the posterior distribution is equivalent to minimizing a certain penalty function. This penalty function represents a trade off between the infidelity of an image x to the record y and the roughness of the image x, controlled by an unspecified smoothing parameter β. After introducing the notion of partitioning, we showed that a partition that divides the image into many small sub-images may help to reduce CPU time although not by a large amount, whereas a partition that divides the image into a few large sub-images will not be helpful.



original

beta=0.0, 34.00% misclassified

Exact MAP, beta=0.7, 3.63% misclassified, penalty=34284.70

Annealing: geometric, beta=0.7, 3.61% misclassified, penalty=34335.07

Annealing: logarithmic, beta=0.7, 4.01% misclassified, penalty=34495.37

ICM, beta=0.7, 11.60% misclassified, penalty=37752.50

Figure 3.8: *A part of Scotland corrupted with normal noise, κ = 1.5, and reconstructed by simulated annealing and ICM with $\beta = 0.7$*

We showed how a simple observation from the fluid flow formulation allows us to produce a sequence of MAP estimates for increasing values of the smoothing parameter β. This provided us with a method of choosing β by eye. We also described and investigated several automatic ways that have been suggested for choosing the smoothing parameter. One of these gave reasonably good results, both for normal noise and the binary channel in experiments conducted with six test images.

Finally, we demonstrated how the exact MAP estimate can be used to assess the performance of other approximate algorithms for finding $\hat{x}$. In a very small experiment we saw that neither ICM nor simulated annealing succeeded in finding $\hat{x}$.

Chapter 4

Simulated Annealing and Image Reconstruction

4.1 Introduction

We have seen in Section 1.5 that the Bayesian approach to image reconstruction leads us to attempt to find the image $x \in \mathcal{X}$, the set of all possible images, that minimizes the penalty function (1.15). This penalty function, which is often referred to as energy, takes the form

$$\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left(\sum_{[i,j]}\phi(|x_i - x_j|) + D\sum_{<i,j>}\phi(|x_i - x_j|)\right),$$

where the first term is a measure of the fidelity of the reconstruction x to the data y, the second term is a measure of roughness of x, $\beta \geq 0$ is a type of smoothing parameter, $\sum_{[i,j]}$ indicates summation over horizontal and vertical neighbours, $\sum_{<i,j>}$ indicates summation over diagonal neighbours, and D is a downweight. In this chapter we take $D = 0.0$, corresponding to a first-order model. Often the penalty function is referred to as the posterior energy, $E(x)$, say.

We consider two classes of images: images made up of unordered colours and grey-level images (see Section 1.3.3). In the case when we are considering images made up of unordered colours we take $\phi(u) = I(u \neq 0)$, where I is the indicator function. In this way all discrepancies incur the same penalty. The resulting penalty function can be written as

$$\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left(v^{(1)}(x) + D\,v^{(2)}(x)\right), \qquad (4.1)$$

where $\nu^{(1)}(x)$ is the number of discrepant first-order pairs in the image and $\nu^{(2)}(x)$ is the number of discrepant second-order pairs. For ordered grey-level images we take $\phi = \phi_\alpha$, where the function ϕ_α was defined as

$$\phi_\alpha(u) \quad = \quad 1 - \frac{1}{1+\alpha u^2} \quad = \quad \frac{1}{1+(\alpha u^2)^{-1}}$$

in equation (1.10). When $u = |x_i - x_j|$ this ϕ_α can be thought of as a penalty for the discrepancy of the grey-levels taken by pixel i and pixel j. Further discussion about this family of functions can be found in Section 1.3.3. The resulting penalty function can be written as

$$\frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left(\sum_{[i,j]}\phi_\alpha(|x_i - x_j|) + D\sum_{<i,j>}\phi_\alpha(|x_i - x_j|)\right). \tag{4.2}$$

The exact minimization of the appropriate penalty function can, in theory, be achieved by a direct search over all c^n possible images, where c is the number of colours or grey-levels in the image and n is the number of pixels. In practice, however, for even moderate values of c and n such a search is not computationally feasible, and other techniques have to be employed.

Besag[3] advocates using the method of iterated conditional modes (ICM) to carry out the minimization. This algorithm requires an initial x and produces a sequence of images such that the penalty function for these images decreases monotonically. It converges to an image that corresponds to a local minimum of the penalty function and that depends upon the initial x. We discussed ICM in Section 1.6.2 and gave some examples of this deterministic algorithm in action in Section 1.7 and Section 1.8. In Figure 1.5 we presented a graph of the behaviour of the penalty function when the ICM algorithm is employed. This graph clearly showed the monotonicity.

Geman and Geman[12] advocate using a technique known as simulated annealing to perform the minimization of the penalty function. The aim of simulated annealing is to produce an image that corresponds to not a local, but a global minimum of the penalty function. Again a sequence of images is constructed, but this time the penalty function does not necessarily decrease monotonically; images that increase it are allowed. The idea is that in this way escapes from local minima can occur, and, under certain circumstances, this algorithm theoretically converges to an image which corresponds to a global minimum of the penalty function and which does not depend upon the initial x. In Section 1.6.1 we introduced the simulated annealing algorithm as it is used in image reconstruction. In Section 1.7 we gave an example of this stochastic algorithm in action. In that example we followed the simulated annealing algorithm by ICM. This could only result in a further reduction of the penalty function. Figure 1.6 showed a graph of the behaviour of the penalty function when simulated annealing followed by ICM is employed. There the penalty function was seen to be non-monotonic.

In Section 4.2 we discuss the simulated annealing algorithm in greater detail, describe how it is implemented by Geman and Geman[12] in the case of image reconstruction, and review the main theoretical results of [12]. We also consider an alternative, but very similar formulation and discuss some results due to Hajek[18].

The problem with such theoretical work on simulated annealing is that most of the results are of an asymptotic nature. In effect, this means that they hold provided that the number of iterations of simulated annealing is infinite. In practice, only a small finite number of iterations can be employed. Geman and Geman[12] show some examples of reconstructions that have been obtained in this way. Although the reconstructions presented are good in a qualitative sense, no indication of how good they are in a quantitative sense is given. In Chapter 3 we saw that it is possible to find the exact minimum of the penalty function (4.1) for binary images ($c = 2$) using a fluid flow approach. In Section 3.10 we compared quantitatively—in one example only—the global (exact) minimum obtained using this approach with other local minima produced by the simulated annealing and ICM algorithms. We found for this example that neither simulated annealing nor ICM found the global minimum of the penalty function.

Variations on the basic simulated annealing algorithm may be expected to produce better results in a quantitative sense. In Section 4.3 we introduce three common sense variations of the algorithm outlined by Geman and Geman[12]. We show by means of a simulation experiment that one of the variations out-performs the others, especially when the number of iterations used in the annealing algorithm is large. In other sections we restrict our attention to this variation. In Section 4.4 we discuss some practical temperature schedules and show how the performance of the simulated annealing algorithm is sensitive to the first and last temperatures of the temperature schedule, but comparatively insensitive to the type of schedule used. The investigation up to the end of Section 4.4 is based upon images made up of colours whose reconstruction requires the minimization of the penalty function (4.1). In Section 4.5 we move on to grey-level images whose reconstruction requires the minimization of the penalty function (4.2). We consider a further variation on the simulated annealing algorithm, the aim of which is to reduce the heavy computation required when dealing with grey-level images. Using this further modification we examine practical schedules for the simulated annealing algorithm applied to such images. Finally, in Section 4.6 we present our conclusions and make some suggestions for further work.

4.2 The simulated annealing algorithm

In this section we discuss the simulated annealing algorithm itself. In Section 4.2.1 we outline the basic idea behind the algorithm. In Section 4.2.2 we describe how this idea is implemented by Geman and Geman[12] in the case of image analysis and we discuss their theoretical results. In Section 4.2.3 we consider another approach to simulated annealing and outline some associated theoretical results.

4.2.1 Basic idea

The idea behind simulated annealing is simple and is outlined in many places, *e.g.* Ripley[33]. It is based on an analogy with the chemical process of annealing. The general problem that the simulated annealing algorithm addresses is the minimization of a certain (positive) energy function $E(x)$ over a large, but finite set of possible configurations $\mathcal{X}$. Here, for the sake of simplicity, we shall assume that there is only one configuration, $\hat{x}$, that achieves this minimization. If we define a probability measure P on the set of all xs by

$$P(x) \propto \exp\{-E(x)\}$$

then our task becomes that of finding the $\hat{x}$ that maximizes $P(x)$. In essence we have returned to the Bayesian framework used in image reconstruction in which we try to find the image $\hat{x}$ that maximizes an appropriate posterior distribution. We saw in Section 1.4 that for the model that we consider this posterior distribution is a Gibbs distribution. Now define a further probability measure P_λ on $\mathcal{X}$ by

$$P_\lambda(x) \propto P(x)^\lambda,$$

where $\lambda > 0$, and note that as $\lambda \longrightarrow \infty$, P_λ increasingly concentrates on $\hat{x}$. (We also point out that in general the maxima of P are the maxima of P_λ for all $\lambda > 0$.) In particular, if we take a series of samples x_λ from P_λ as $\lambda \longrightarrow \infty$, we would expect $x_\lambda \longrightarrow \hat{x}$ in some sense. If we set $\lambda = 1/\tau$ we can talk of decreasing *temperature* τ to zero, instead of increasing λ to ∞. The basic problem with simulated annealing is, however, that the lower the value of τ, the harder it is to sample from the corresponding probability measure. In Section 4.2.2 we begin to consider how we go about such sampling.

4.2.2 Simulated annealing and image reconstruction

First, we define the distribution $\pi_\tau(x)$ as

$$\pi_\tau(x) \propto P(x)^{1/\tau},$$

for fixed τ. Since, in our approach to image reconstruction, $P(x)$ is a Gibbs distribution, $\pi_\tau(x)$ is also a Gibbs distribution. Geman and Geman[12] propose a method for sampling from such a distribution that they refer to as the Gibbs sampler which we have already discussed in Section 1.4.1.

Geman and Geman[12] let $\{n_t,\ t \geq 1$ and t an integer$\}$ be a sequence of pixels of the image that contains every pixel infinitely often (*e.g.* we can consider the set of all pixels arranged in order according to a raster scan repeated an infinite number of times). They define a Markov chain $X(t)$, indexed by discrete time t, whose values are images representing successive reconstructions (differing by only one pixel, as we shall see), by explaining its evolution at time t form $X(t-1)$ to $X(t)$. First, $X(t)$ may differ from $X(t-1)$ only at pixel n_t, and we shall refer to the value of the random variable $X(t)$ at pixel n_t as $X_{n_t}(t)$. Then

$$\Pr\left(X_{n_t}(t) = x_{n_t}\right) \propto \pi_\tau(x_1, \ldots, x_{n_t}, \ldots, x_n), \tag{4.3}$$

where $\{x_1, \ldots, x_n\} \setminus \{x_{n_t}\}$ are fixed. In other words, in order to move from $X(t-1)$ to $X(t)$ we visit pixel n_t and select a value for $X_{n_t}(t)$ according to the conditional distribution defined from π_τ by (4.3). Since $\pi_\tau(x)$ is a Gibbs distribution, the conditional distribution (4.3) takes a very simple form depending on the record at pixel n_t, namely y_{n_t}, and the value of the image at the pixels that belong to the cliques that involve pixel n_t (*i.e.*, in the case under consideration $x_{\partial n_t}$, where ∂n_t is the set of neighbours of pixel n_t). Thus, the distribution $\Pr(X_{n_t} = x_{n_t})$ can be quickly computed and the value for $X_{n_t}(t)$ easily sampled. Essentially, Theorem A of Geman and Geman[12] tells us that the distribution of $X(t)$ as $t \longrightarrow \infty$ is π_τ, and is thus independent of the the initial value $X(0)$. This can be expressed by stating that

$$\lim_{t\to\infty} \Pr(X(t) = \omega \mid X(0) = \eta) = \pi_\tau(\omega),$$

for all possible images $\omega, \eta \in \mathcal{X}$. The proof of this theorem requires a lemma. This lemma tells us that information about the starting configuration is lost approximately exponentially with the number of complete sweeps.

The idea of simulated annealing is to sample from π_τ while reducing the temperature τ in the hope that eventually samples are drawn from the uniform distribution π^* over the set

$$\Omega^* = \{\omega : \omega \in \mathcal{X} \text{ and } E(\omega) = \min_{\eta \in \mathcal{X}} E(\eta)\}.$$

Geman and Geman[12] quantify this idea in their Theorem B. They begin by making three definitions

$$\begin{aligned} E_{\max} &= \max_{\omega \in \mathcal{X}} E(\omega) \\ E_{\min} &= \min_{\omega \in \mathcal{X}} E(\omega) \\ \Delta &= E_{\max} - E_{\min}. \end{aligned}$$

Next they assume that there exists an integer $t^* \geq n$ such that for every $t = 0, 1, 2, \ldots$ the set of pixels that make up the image is contained in the set $\{n_{t+1}, n_{t+2}, \ldots, n_{t+t^*}\}$ (*e.g.* we can again consider the set of all pixels arranged in order according to a raster scan, and we can set $t^* = n$). They then let $\tau(t)$ be any sequence of temperatures for which

1. $\tau(t) \to 0$ as $t \to \infty$;

2. $\tau(t) \geq n\Delta / \log t$ for all $t \geq t_0$ for some integer $t_0 \geq 2$.

Finally, they state the conclusion, namely that for any initial image η and for all images ω,

$$\lim_{t\to\infty} \Pr(X(t) = \omega \mid X(0) = \eta) = \pi^*(\omega). \tag{4.4}$$

As condition 2 cannot be followed in practice, Geman and Geman[12] suggest using a schedule of the form $C / \log(1 + k)$, where k is the number of full sweeps, and selecting C in such a way that τ decreases from approximately 4.0 to 0.5 over 300 to 1000 sweeps. Geman and

Reynolds[14], working in the context of grey-level images, employ 200 sweeps and drop the temperature linearly from an initial value, which they set equal to 0.3, to a final value $\tau \approx 0$. With this faster schedule than the one dictated by the theory they obtain very good results. It is, however, clear that the simulated annealing algorithm is computationally demanding. However, in the image reconstruction case when all the appropriate distributions are Gibbs distributions, the algorithm can be seen to be highly parallelizable.

4.2.3 An alternative annealing algorithm

In this section we outline an alternative (and more common) approach to simulated annealing. We do not use this approach in our work on image reconstruction. Let us return to the general problem of minimizing a (non-convex) energy function $E(x)$ over a large finite set of configurations $\mathcal{X}$. The form of simulated annealing now under discussion again proceeds by defining a Markov chain $X(t)$, indexed by discrete time t, whose values are configurations in the set $\mathcal{X}$ and whose distribution theoretically converges to the uniform distribution over the set of global minima of the function E. Let us assume that $X(t-1) = x$. At time t and temperature $\tau(t)$ a candidate value for $X(t)$, x' say, is generated by means of some generation mechanism that we shall discuss below. The probability that this x' is accepted as the value of $X(t)$ is given by

$$\Pr(X(t) = x' \mid X(t-1) = x) = \min\left\{1, \exp\left(-\frac{E(x') - E(x)}{\tau(t)}\right)\right\}. \tag{4.5}$$

Thus, a generated configuration x' is accepted with probability 1 if $E(x') \leq E(x)$ (*i.e.* if a decrease in energy results from accepting x'), and with non-zero probability if $E(x') > E(x)$. It can be seen that this acceptance probability depends only upon the difference between the energies $E(x')$ and $E(x)$. In many applications the candidate state x' is generated in such a way that such a difference is easily calculable.

All that remains in this description is to define a possible generation mechanism. To do this, we need to write down a matrix $G(t) = (G_{x,x'}(t))$, where x and x' are any two possible configurations in $\mathcal{X}$. The usual assumptions made about this matrix are that $G(t)$ is such that from any configuration in $\mathcal{X}$ it is possible, in a finite number of steps, to visit any other configuration, and that $G(t)$ is the same for all t.

We now make some remarks about this algorithm. First, if we run the above algorithm for an infinite time with $\tau(t)$ equal to a constant τ for all t, then

$$\lim_{t\to\infty} \Pr(X(t) = \omega \mid X(0) = \eta) \propto \exp\left\{-\frac{E(\omega)}{\tau}\right\},$$

for all $\omega, \eta \in \mathcal{X}$. Secondly, if $\tau(t)$ is reduced sufficiently slowly, again equation (4.4) holds where now $X(t)$ is a Markov chain with transition probabilities given by equation (4.5) and ω and η are arbitrary members of $\mathcal{X}$. The rate at which $\tau(t)$ has to be reduced has been the subject of many papers. The book by Laarhoven and Aarts[26] gives a thorough discussion of the relevant literature (as well as a general review of all the literature on simulated annealing). Many authors set $\tau(t) = \Gamma / \log(1+t)$ and produce a sufficient condition on Γ for (4.4) to hold. Such a condition takes the form $\Gamma \geq \Gamma^*$, where Γ^* is some parameter depending on the structure of the optimization problem. In the discussion in [26] a number of successively smaller values for Γ^* are presented. The paper by Hajek[18] presents a necessary and sufficient condition on $\tau(t)$ for the convergence given by (4.4). To outline this result we need three definitions: first, a configuration x' is said to be *reachable at height L* from a configuration z, if there is a sequence of configurations

$$x' = x_0, x_1, \ldots, x_p = z$$

such that $G_{x_k, x_{k+1}} > 0$ for $k = 0, 1, \ldots, p-1$, and $E(x_k) \leq L$ for $k = 0, 1, \ldots, p$; secondly, state x is said to be a *local minimum* if no state x' with $E(x') < E(x)$ is reachable at height $E(x)$; and thirdly, a local minimum x is said to have *depth* equal to plus infinity if x is a global minimum, and equal to the smallest number $\mathcal{E}$, $\mathcal{E} > 0$, such that some state x' with $E(x') < E(x)$ can be reached at height $E(x) + \mathcal{E}$. Hajek[18] makes the assumption that for any real number $\mathcal{D}$ and any two states $x, x' \in \mathcal{X}$, x is reachable at height $\mathcal{D}$ from x' if and only if x' is reachable at height $\mathcal{D}$ from x. He states that a necessary and sufficient condition for (4.4) to hold is that

$$\sum_{t=1}^{\infty} \exp\left(-\frac{d^*}{\tau(t)}\right) = \infty, \tag{4.6}$$

where d^* is the maximum of the depths of all states which are local but not global minima. If $\tau(t) = \Gamma / \log(t+1)$ then the necessary and sufficient condition (4.6) holds if and only if $\Gamma \geq d^*$. Hajek[18] states that Geman and Geman[12] 'considered a model which is nearly a special case' of the model used in this section. As we have seen in Section 4.2.2, Geman and Geman[12] give a sufficient condition on Γ, namely that $\Gamma \geq \Gamma^*$, some Γ^*, for (4.4) to hold.

However, their value for Γ^* is 'substantially larger than d^*'.

4.2.4 Further comments

We have seen in this section that the asymptotic properties of the annealing algorithm are fairly well understood, even though the conditions on the temperature schedule $\tau(t)$ that are stated for (4.4) to hold involve constants that depend upon the structure of the optimization problem and that are often of such a size that the temperature is still very high even after a large number of iterations. As Geman and Reynolds[14] say (in the context of grey-level images)

> what is important is the *finite-time* behaviour. ... In particular, we have no guarantee of obtaining an actual minimum with a finite amount of computation; in fact it is highly doubtful that we ever achieve the minimum energy ....

The rest of this chapter is devoted to a small study of the finite-time behaviour of the simulated annealing algorithm, as applied to image analysis and described in Section 4.2.2. In Section 4.3 we suggest and explore some variations on the original algorithm that seem appropriate in the finite-time case.

4.3 Four simulated annealing algorithms

In this section we introduce four simulated annealing algorithms in the context of image reconstruction. The second, third and fourth algorithms are variations on the first. We now present the four algorithms.

Original A total of M iterations of simulated annealing are performed and the reconstruction is the image x that results from the final iteration.

Lowest A total of M iterations of simulated annealing are performed. The penalty function is computed after each iteration and the reconstruction is the image x that yields the minimum value over all M iterations.

Original plus ICM The ICM algorithm is applied until convergence to the x produced by the algorithm referred to as **original**. The reconstruction so produced is guaranteed to give a local minimum of the penalty function (see Section 1.6.2).

Lowest plus ICM The ICM algorithm is applied until convergence to the x produced by the algorithm referred to as **lowest**. Again the reconstruction so produced is guaranteed to give a local minimum of the penalty function (see Section 1.6.2).

To assess the performance of these four algorithms in terms of the penalty function (4.1) we conduct a simulation experiment. We base this experiment on a binary image ($c = 2$) of size 32 × 32 pixels (1024 pixels in total). The image is then corrupted by the addition to each pixel of independent normal noise of known variance $\kappa = 0.5$. This corruption yields a record y. We attempt to recover the original image from the record y in the standard way by trying to find the x that minimizes (4.1). For simplicity, and to reduce computations later, we set $D = 0.0$. The smoothing parameter β is as yet unspecified. In Chapter 3 we presented a method for finding β for binary images, such as the one used here. That method involved minimizing a certain function. We performed this minimization over values of β in the set $\{0.0, 0.05, 0.1, \ldots, 1.4, 1.45, 1.5\}$, and found that the minimizing β was equal to 1.0. This β also minimized over the above set the number of misclassified pixels at 40 (3.91%). As we saw in Chapter 3 it is possible in the case of binary images to find the image that gives the global minimum of the penalty function (4.1). This image is referred to as the exact MAP reconstruction. When $\beta = 1.0$ the exact MAP reconstruction gives the value of the global minimum of (4.1) to be 599.65. We shall refer to this quantity as E. (We point out that the value of (4.1) for the original image was 626.35, while for tne maximum likeiihood estimate it was 1115.26.) As a comparison, the reconstruction produced by the ICM technique of Besag[3], which we discussed in Section 1.6.2 yielded a value of the penalty function (4.1) of 623.73, when the initial image was the maximum likelihood estimate or closest mean classifier. We shall refer to this quantity as I. Henceforth, we transform the penalty function (4.1) to

$$\frac{\text{old penalty function} - E}{I - E}, \tag{4.7}$$

where

$$\text{old penalty function} = \frac{1}{2\kappa}\sum_{i=1}^{n}(y_i - x_i)^2 + \beta\left(v^{(1)}(x) + Dv^{(2)}(x)\right),$$

as in expression (4.1). We shall refer to the transformed penalty function (4.7) as the TPF. The exact MAP reconstruction has a value of TPF as given by expression (4.7) of 0.0, whereas the ICM reconstruction has a value of 1.0.

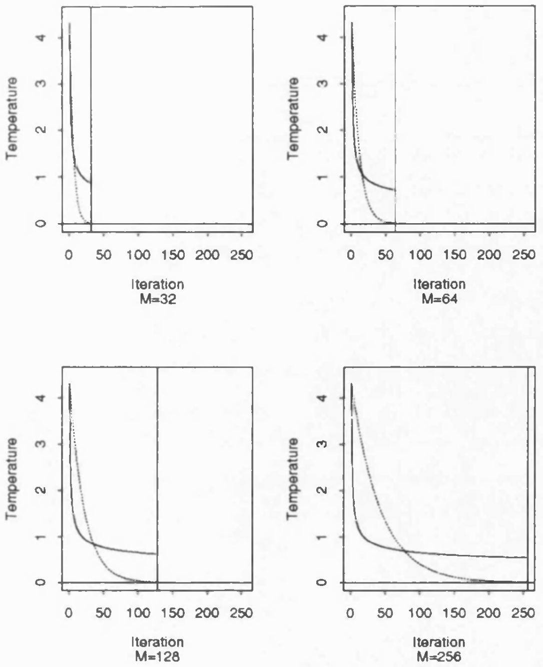


Figure 4.1: *The logarithmic (unbroken like) and geometric (broken line) temperature schedules*

We now attempt to see how well each of the above four algorithms perform. In Section 3.10 we considered two temperature schedules. The first is based on the logarithmic schedule of Geman and Geman[12] in which

$$\tau(m) = \frac{3}{\log(1+m)}, \quad m = 1, \dots, M,$$

where M is the total number of iterations. The second is based on the geometric schedule

$$\tau(m) = A\rho^{m-1}, \quad m = 1, \dots, M,$$

where we select A and ρ so that the initial temperature, $\tau(1)$, is the same as that used for the logarithmic schedule, namely $3 / \log(2) = 4.33$, while the final temperature, $\tau(M)$, is set to a quantity that is only just greater than zero, namely 0.01. We consider the following values of M: $32 = 2^5$, $64 = 2^6$, $128 = 2^7$, $256 = 2^8$ and $512 = 2^9$. Graphs of these temperature schedules are shown in Figure 4.1, for all M except $M = 512$. We note from the graphs that the logarithmic schedule does not change with M; it is merely truncated. The geometric schedule, on the other hand, varies with M in such a way that the final temperature, $\tau(M)$, is 0.01.

Simulated annealing is a stochastic process and in practice the value of TPF produced varies with the seed used in the random number generator. Accordingly, we repeat each algorithm with 100 different seeds.

4.3.1 Some results

We present our results by means of Figure 4.2. For clarity we plot the logarithm of the mean of TPF as given by expression (4.7) for the 100 different seeds. Thus, a value of 0.0 represents an average value equal to that given by ICM alone, while if all 100 values of TPF were equal to its minimum value then a value of negative infinity would result! The area of the circles is proportional to the variance of TPF over the 100 repetitions. We make the following observations:

- For all four algorithms and both temperature schedules the mean value of TPF decreases as M, the number of iterations of simulated annealing, increases.
- For all four algorithms and both temperature schedules the variance of TPF decreases as M, the number of iterations of simulated annealing, increases.
- For all five values of M considered and the logarithmic schedule, the algorithm that performs worst in terms of the mean value of TPF is **original**, then comes **lowest**, followed by **original plus ICM**. The algorithm that performs best is **lowest plus ICM**. The improvement of **lowest** over **original**, and of **lowest plus ICM** over **original plus ICM** is negligible for low values of M, but increases with M, as we might expect. The average increase in TPF due to not considering the lowest value produced by simulated annealing, for both schedules, is discussed in Section 4.3.3. The average increase in TPF due to not using ICM, for the logarithmic schedule, is discussed in Section 4.3.4.
- For all five values of M considered and the geometric schedule, the performance of **original** and **original plus ICM**, and the performance of **lowest** and **lowest plus ICM** were almost (see Section 4.3.4) identical. Thus, the geometric schedule always ended up with a reconstruction that corresponded to a local minimum of TPF, and ICM had no effect. The reason for this is that, in effect, ICM is zero temperature annealing, as explained in Section 1.6.2, and the final annealing temperature with the geometric

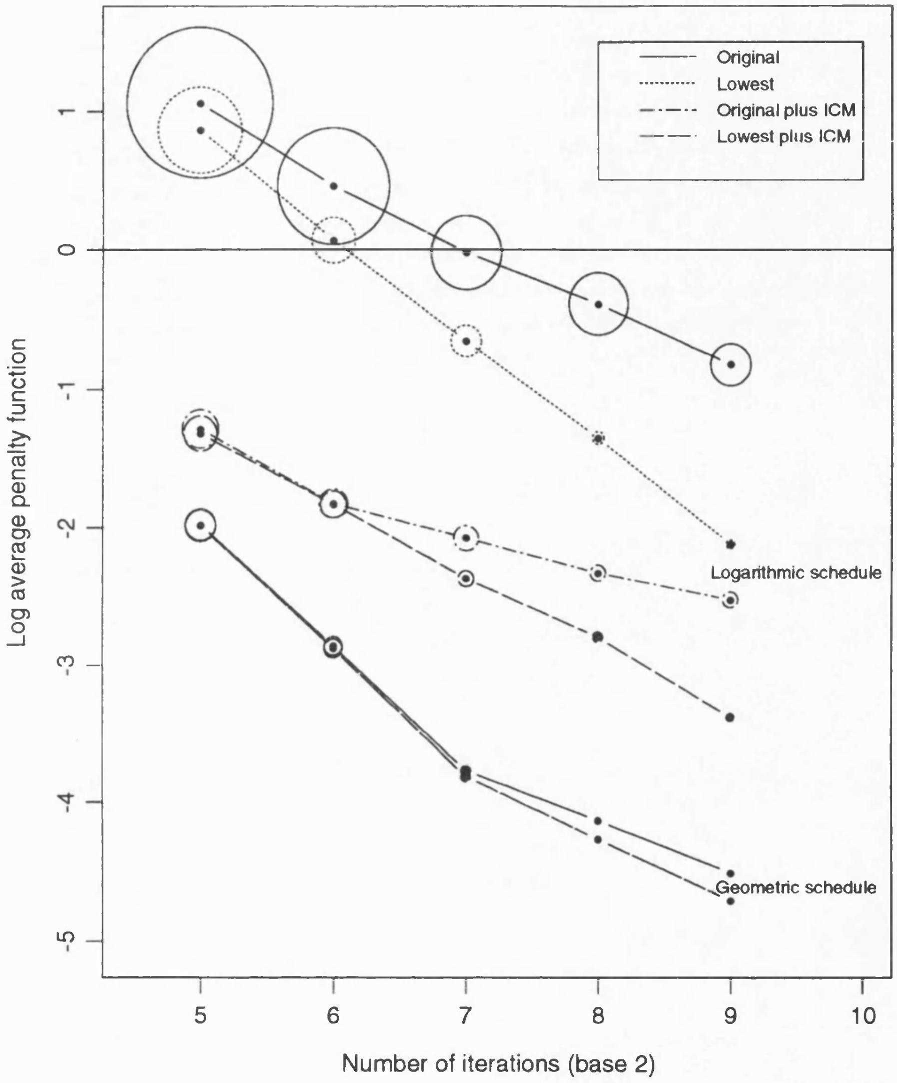


Figure 4.2: *The average performance of the four algorithms for the two different temperature schedules*

schedule is very low. The algorithm **original** performed a little less well than the algorithm **lowest**, as we would expect.

- For M equal to 32, 64 and 128 and the logarithmic schedule, the algorithm that gives the highest variance of TPF is **original**, then comes **lowest**, followed by **original plus ICM**. The algorithm that gives the lowest variance is **lowest plus ICM**. For higher values of M the situation is less clear, although **original** is always most variable and **lowest plus ICM** is always least variable.
- For all five values of M considered and the geometric schedule, the variances of TPF when **original** is used and when **lowest** is used are almost identical.

4.3.2 Density estimates

To illustrate the effect of the four algorithms we present in Figure 4.3 density estimates of the logarithm of the 100 values of TPF. Of course, if the exact MAP is obtained the value of the TPF will be 0.0, and its logarithm is negative infinity. Such points are indicated by the numbers written on the left part of the graph. Assume that the exact MAP is obtained p times out of 100. The number p is printed at a height proportional to p and a density estimate is produced from the remaining $n = 100 - p$ points. This density estimate is scaled so that its total area is $(100 - p) / 100$. The density estimate, $\hat{f}(x)$ say, is produced using the formula

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right),$$

where h is window width. In our case we take K to be the standard normal kernel and we set

$$h = 0.9 A n^{-1/5},$$

where

$$A = \min(\text{standard deviation}, \text{interquartile range} / 1.34).$$

This h is the value recommended in Section 3.4.2 of Silverman[37], where it is said to work well for a wide range of densities. The interested reader is referred to that reference for further details. For each density estimate we mark the lower and upper quartiles of the data from which it is constructed, and shade the area between. The density estimates presented are for the case when $M = 128$. First, we consider the logarithmic temperature schedule. For **original** the density is concentrated around 0.0 (the value achieved by ICM), whereas for **lowest** the density is of approximately the same shape although moved to the left. The density for **original plus ICM** has a very different shape and again the mass is seen to have moved to the left. The density for **lowest plus ICM** is of similar shape to that for **original plus ICM** although a slight shift to the left is visible. These four densities suggest that the effect of using the lowest value found during the annealing when the logarithmic temperature schedule is used is small compared to the effect of employing ICM. We attempt to quantify this a little further in Section 4.3.3 and Section 4.3.4. We now consider the density estimates for the geometric schedules. The first and third, and the second and fourth estimates are based on almost the same data and hence appear identical. Moreover there is very little noticeable difference between the first and second density estimates. If we compare the four density estimates produced for the logarithmic temperature schedule with those produced for the geometric temperature schedule, we notice that for **original** and **lowest** the shapes are very different, but when ICM is employed the shapes are quite similar.

4.3.3 Benefits of using the lowest value found during annealing

In this section we discuss briefly the benefits of using the lowest value found during simulated annealing for both temperature schedules considered. We summarize our results in Table 4.1, where we present the increase in the mean of TPF due to not considering the lowest value produced by simulated annealing, for both schedules, as a percentage. For example, a value of 22% indicated that the average value of TPF produced by **original** is 1.22 times the average produced by **lowest**. In the columns headed 'Without ICM' we see how much worse algorithm **original** does compared to algorithm **lowest**, whereas in the columns headed 'With ICM' we see how much worse algorithm **original plus ICM** does compared to algorithm **lowest plus ICM**. (The addition of ICM did not affect the performance of **original** and **lowest** when the geometric schedule was employed for reasons that were explained in Section 4.3.1.) We note from Table 4.1 that the increase due to not considering the lowest value produced by simulated annealing is very much larger for the logarithmic schedule than for the geometric schedule. For the logarithmic schedule the increase is larger when ICM is not used than when it is used. For all four columns this increase seems to grow monotonically with M, except in the case of the logarithmic schedule with ICM and low values of M. However, as we shall see in Section 4.3.4

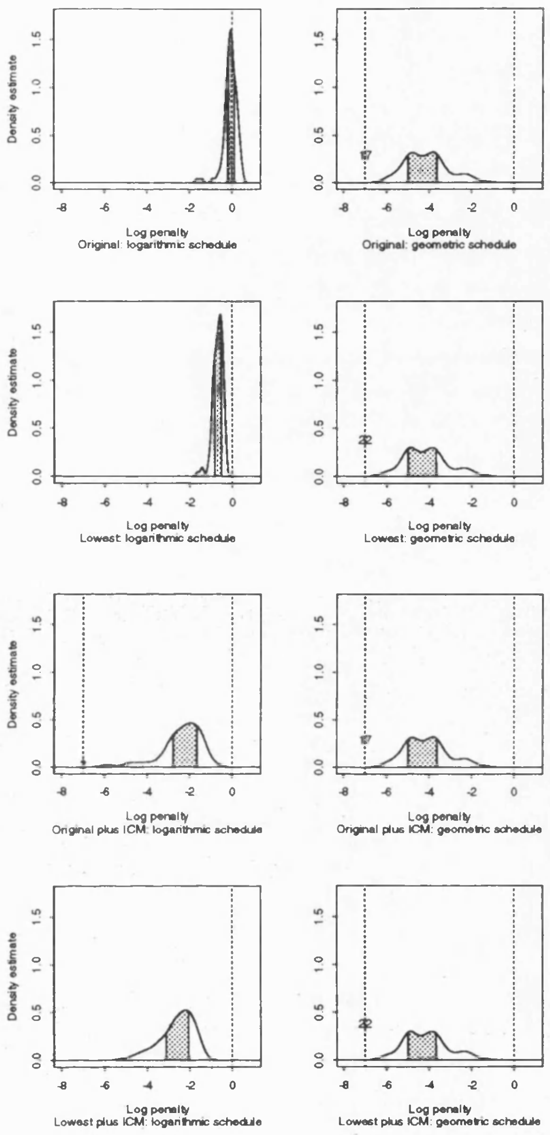


Figure 4.3: *Density estimates for the four algorithms and the two temperature schedules: 128 iterations of simulated annealing*

	Without ICM		With ICM	
M	Logarithmic	Geometric	Logarithmic	(Geometric)
32	22	1	3	(1)
64	48	2	1	(2)
128	89	5	34	(5)
256	162	14	59	(14)
512	267	22	135	(22)

Table 4.1: *Percentage increase in the transformed penalty function when the lowest value found during the annealing is not used (logarithmic and geometric temperature schedule)*

the increase in TPF due to not considering the lowest value produced by simulated annealing is much less than the increase in TPF due to not employing ICM. Nevertheless, as using the lowest value found during the annealing resulted in benefits for both temperature schedules considered, and as the extra computational burden imposed by this modification is minimal, we recommend its adoption.

4.3.4 Benefits of using ICM

In this section we briefly discuss the benefits of using ICM when the logarithmic temperature schedule is employed. We have already seen in Section 4.3.1 that almost no advantage is gained by using ICM with the geometric temperature schedule. This is due to the extremely low temperature of the final iteration and to the fact that ICM can be thought of as zero temperature annealing (see Section 1.6.2 and Section 4.3.1). We summarize our results in Table 4.2, where we present the increase in the mean of TPF as given in expression (4.7) due to not employing ICM, as a percentage. In the columns headed 'Without lowest value' we see how much worse algorithm **original** does compared to algorithm **original plus ICM**, whereas in the columns headed 'With lowest value' we see how much worse algorithm **lowest** does compared to algorithm **lowest plus ICM**. We note from Table 4.2 that the increase due to not employing ICM after the *M* iterations of annealing is very much larger when the starting image for ICM is the one that results from the *M*th iteration of annealing than when the starting point is that image that gives the lowest value of TPF over all *M* iterations. In both cases the increase seems

M	Without lowest Value	With lowest Value
32	943	784
64	881	569
128	684	459
256	601	326
512	450	253

Table 4.2: *Percentage increase in transformed penalty function when ICM is not employed after annealing (logarithmic temperature schedule)*

to grow monotonically with M. As we stated in Section 4.3.3 the increase in TPF due to not considering the lowest value produced by simulated annealing is much less than the increase in TPF due to not employing ICM.

In Table 4.3 we present the mean and variance of the number of iterations of ICM required by each of the algorithms **original plus ICM** and **lowest plus ICM**. We remark that no pixels are changed during the final iteration of ICM. We can clearly see from Table 4.3 that for the geometric schedule very few iterations of ICM are required. In fact, in only very few of the 100 replications does ICM have any effect with this schedule, and in those cases the effect is negligible. With the logarithmic schedule, however, the addition of ICM does have a noticeable effect, which we now consider. We see from Table 4.3 that for both **original plus ICM** and **lowest plus ICM** the number of iterations of ICM required for convergence decreases as M increases. Moreover, for each value of M, the number of iterations of ICM required by **original plus ICM** is greater than the number required by **lowest plus ICM.** We note the interesting fact that with the **lowest plus ICM** algorithm the variance decreases as M increases. No such pattern exists with the **original plus ICM** algorithm. Finally, we remark that the average number of iterations of ICM is low compared with M, the number of iterations of simulated annealing. Thus, the addition of ICM to the annealing part of the algorithm does not substantially increase the overall amount of computation required. Hence we recommend its adoption.

In Table 4.4 we present for the logarithmic schedule the average decrease in TPF when ICM is used. In some sense we can regard this as a measure of how far the reconstruction produced by the annealing part of the algorithm is from a local minimum of TPF. Thus, as M increases

	original plus ICM				**lowest plus ICM**			
	Logarithmic		Geometric		Logarithmic		Geometric	
M	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
32	3.67	0.446	1.00	0.000	3.54	0.473	1.02	0.020
64	3.29	0.471	1.01	0.010	3.05	0.472	1.03	0.029
128	2.97	0.373	1.00	0.000	2.71	0.430	1.07	0.066
256	2.89	0.483	1.00	0.000	2.40	0.303	1.04	0.039
512	2.58	0.347	1.00	0.000	1.10	0.091	1.10	0.091

Table 4.3: *Number of iterations of ICM required (logarithmic and geometric temperature schedule)*

M	**original plus ICM**	**lowest plus ICM**
32	2.61	2.10
64	1.42	0.91
128	0.86	0.43
256	0.58	0.20
512	0.36	0.09

Table 4.4: *Average decrease in transformed penalty function when ICM is used (logarithmic temperature schedule)*

	Logarithmic		Geometric	
M	Mean	Variance	Mean	Variance
32	0.268	0.0159	0.137	0.0134
64	0.160	0.0097	0.056	0.0046
128	0.093	0.0033	0.022	0.0012
256	0.061	0.0014	0.014	0.0005
512	0.034	0.0008	0.009	0.0002

Table 4.5: *Mean and variance of the transformed penalty function for* **lowest plus ICM** *(logarithmic and geometric temperature schedule)*

on average the reconstruction produced by the annealing part gets nearer a local minimum, and for each M on average the reconstruction produced by the annealing part of **lowest plus ICM** is nearer to a local minimum than the reconstruction produced by **original plus ICM.**

4.3.5 The effect of different values of M

The effect of different values of M has already been illustrated in Figure 4.2 and discussed in Section 4.3.1. In this section we examine in more detail the effect of different values of M for the **lowest plus ICM** algorithm. In Table 4.5 we give the mean and the variance of TPF as given in expression (4.7) for the **lowest plus ICM** algorithm. We note again that for both temperature schedules the mean and variance decrease as M increases.

We also present density estimates of the log of TPF in Figure 4.4. For the logarithmic temperature schedule we see that the density moves to the left as M increases, although the exact MAP value is never attained. A similar phenomenon occurs for the geometric schedule. However, there the number of times the exact MAP value is attained increases with M, and becomes quite large (nearly one half of all the realizations in the $M = 512$ case, for example).

4.4 Towards some practical temperature schedules

In this section we investigate further how the performance of the simulated annealing algorithm depends upon the temperature schedule. In the rest of this chapter we restrict our attention to the algorithm **lowest plus ICM.** In Section 4.4.1 we concentrate on the geometric temperature

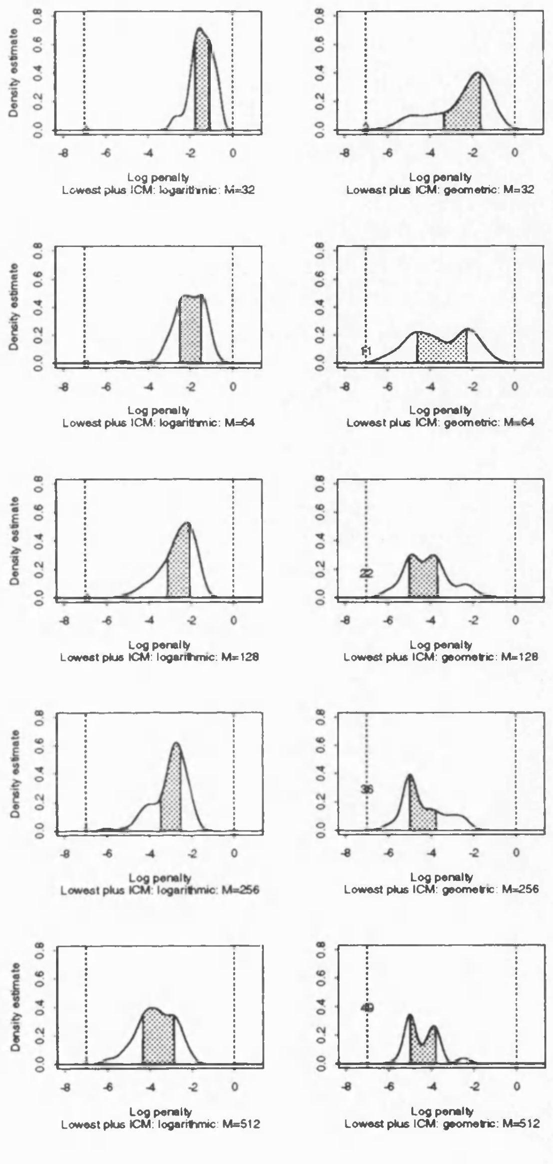


Figure 4.4: *Density estimates for different values of M and the two temperature schedules*

schedule. In particular we investigate how the first and last temperatures of the temperature schedule affect the performance of the algorithm. We discover that good performance requires both the first and last temperatures to be low. In Section 4.4.2 we consider a variety of other temperature schedules and discover that what is important is the choice of the first and last temperature, rather than the choice of the schedule itself. In Section 4.4.3 we turn our attention away from the case in which the image has $c = 2$ colours, to images with more than two colours; in particular we look at the case $c = 5$.

4.4.1 The choice of the first and last temperatures for geometric schedules

In this section we consider only geometric schedules, namely schedules of the form

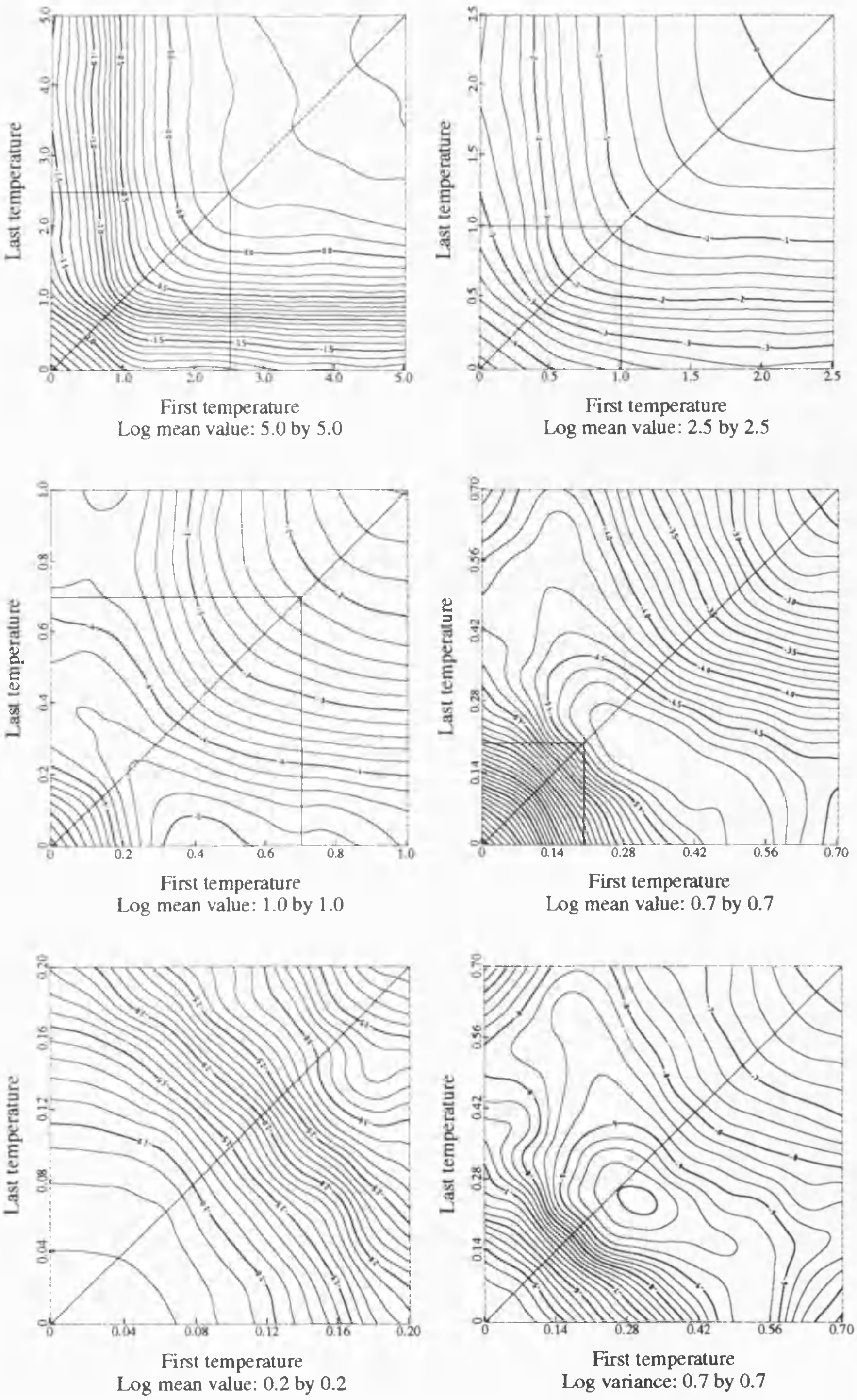
$$\tau(m) = A\rho^{m-1}, \quad m = 1, \ldots, M. \tag{4.8}$$

From now on we set M equal to 128. This schedule has two parameters, A and ρ, as yet unspecified. We prefer to reparametrize this schedule in terms of the first temperature $\tau(1)$, which we shall refer to as f, and the last temperature $\tau(M)$, which we shall refer to as l. We assume that $f > 0$ and $l > 0$ throughout. Thus, we may rewrite (4.8) as

$$\tau(m) = f\left(\frac{l}{f}\right)^{\frac{m-1}{M-1}}.$$

We note that if $f = l$ we have the constant schedule $\tau(m) = f = l$, $\forall m$.

We now investigate how the performance of the simulated annealing algorithm with this schedule depends on the temperatures f and l. In our investigation we do not insist that $f \geq l$. We proceed by computing the average value of TPF as given in expression (4.7) for 25 different seeds at each of 100 (*i.e.* 10×10) different values of f and l. We present our results using the excellent CONICON programs of Sibson[36]. In particular we use an interface to these programs written by Dr Glenn Stone. This interface requires gradient information to be supplied at every point. We compute the gradient information from the values at the points themselves and not separately. This seems to work well for interior points of the region, but to lead to minor problems on the boundaries. There the surfaces that we present may not be representative. This, however, affects substantially neither our investigation nor our conclusions. The axes and the labels are produced by means of a POSTSCRIPT program written by the author. We display f along the horizontal axis and l along the vertical axis. The diagonal line marks the boundary between the schedules with $f > l$ (below the line) and the schedules with $f < l$ (above the line). Schedules that lie on this line have constant temperatures throughout. In all the plots we display the logarithm of the average of the transformed penalty function as this leads to a clearer presentation. We point out that the figures presented on these diagrams point uphill, as is conventional. We recall that for this transformation a value of 0.0 represent an average value equal to that produced by ICM, whereas a value of negative infinity represents 25 reconstructions all of which attain the global minimum of the penalty function. In the first contour plot given in Figure 4.5 we consider $0.0 < f, l \leq 5.0$. We note that for much of the plot the contour lines run parallel to either the f axis or the l axis. In the region in which $f > l$ the mean value of the transformed penalty function seems to depend only on the lower l and not on f, whereas in the region in which $f < l$ the mean value of the transformed penalty function seems only to depend on the lower f and not on l. Exceptions to these observations occur in two regions. The first is in the top right corner of the plot, where the algorithm seems to perform very badly indeed yielding average values greater than 0.0, thus indicating an average performance worse than would be achieved by ICM alone. The second is in the bottom left corner. It seems that the algorithm performs well in this region and that the quality of performance increases as the point (f, l) nears the origin. To investigate this region more fully we magnify the region with $0.0 < f, l \leq 2.5$, as indicated by the box, and we present our results in the second contour plot. Again 100 points of (f, l) are used, and a similar situation seems to result. The main area of interest is once more the bottom right hand corner. This second plot suggests that we should continue the magnification process, and we do this as indicated for $0.0 < f, l \leq 1.0$, $0.0 < f, l \leq 0.7$ and $0.0 < f, l \leq 0.2$. Clearly, the best plot is that for $0.0 < f, l \leq 0.7$ which shows an area of values under -4.6; the $0.0 < f, l \leq 1.0$ plot is lacking in detail, while the $0.0 < f, l \leq 0.2$ plot displays only a side of this area. The final plot that we present is the logarithm of the variance of the 25 realizations at each point $0.0 < f, l \leq 0.7$. This graph has a very similar form to the one showing the logarithm of the average values. Returning to that contour plot, we remark that the area of values below -4.6 is for the most part in the $f < l$ region. We concentrate on the area of values below -4.6 in order to facilitate comparisons with the other schedules that we discuss in Section 4.4.2.


First temperature
Log mean value: 5.0 by 5.0

First temperature
Log mean value: 2.5 by 2.5

First temperature
Log mean value: 1.0 by 1.0

First temperature
Log mean value: 0.7 by 0.7

First temperature
Log mean value: 0.2 by 0.2

First temperature
Log variance: 0.7 by 0.7

Figure 4.5: *Contour plots of the logarithm of the mean value of the transformed penalty function for various values of the first temperature f and the last temperature l using the geometric schedule ($c = 2$)*

4.4.2 Other schedules

We now investigate the behaviour of other schedules in the region $0.0 < f, l \leq 0.7$. We begin by introducing what we shall refer to as schedules of the second kind. Schedules of the second kind, $h(m)$, say, are derived from schedules of the first kind, $g(m)$, say, by means of the relationship $h(m) = l + f - g(M + 1 - m)$. Straight line schedules of the first kind are exactly the same as straight line schedules of the second kind. Examples of these schedules with $f = 0.28$ and $l = 0.16$ are shown in Figure 4.6. These values of f and l are chosen because they seem to give quite reasonable results for all the schedules. In general we consider both monotonically increasing and monotonically decreasing schedules. We now describe the schedules in detail:

1. Straight line schedules

$$g(m) = \frac{l-f}{M-1}(m-1) + f = h(m);$$

2. Geometric schedules of the first kind

$$g(m) = f\left(\frac{l}{f}\right)^{\frac{m-1}{M-1}};$$

3. Geometric schedules of the second kind

$$h(m) = l + f - f\left(\frac{l}{f}\right)^{\frac{M-m}{M-1}};$$

4. Reciprocal schedules of the first kind

$$g(m) = \frac{lf(M-1)}{(lM-f)+(f-l)m};$$

5. Reciprocal schedules of the second kind

$$h(m) = \frac{(f^2M-l^2)+(l^2-f^2)m}{(fM-l)+(l-f)m};$$

6. Logarithmic schedules of the first kind

$$g(m) = \frac{lf(\log(M+1)-\log(2))}{(l\log(M+1)-f\log(2))+(f-l)\log(m+1)};$$

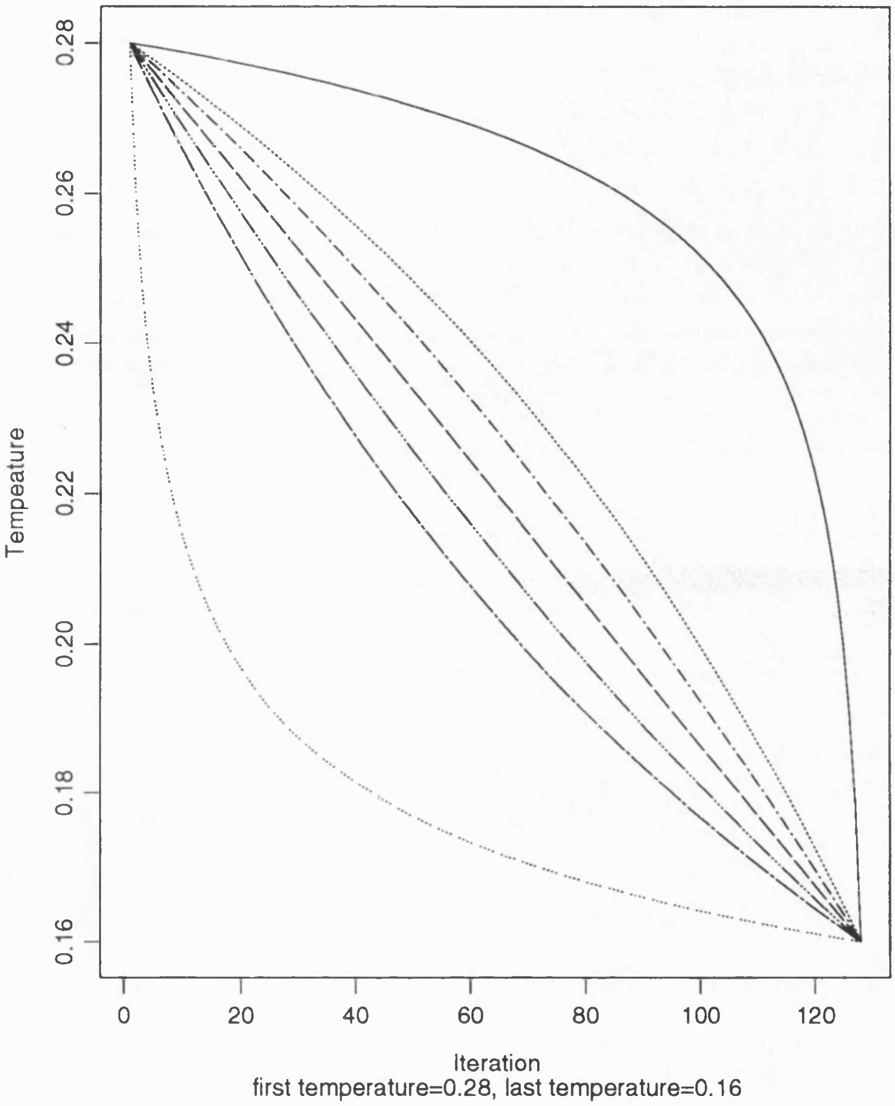


Figure 4.6: *Some examples of the temperature schedules used. From top to bottom: logarithmic of the second kind, reciprocal of the second kind, geometric of the second kind, straight, geometric, reciprocal and logarithmic*

7. Logarithmic schedules of the second kind

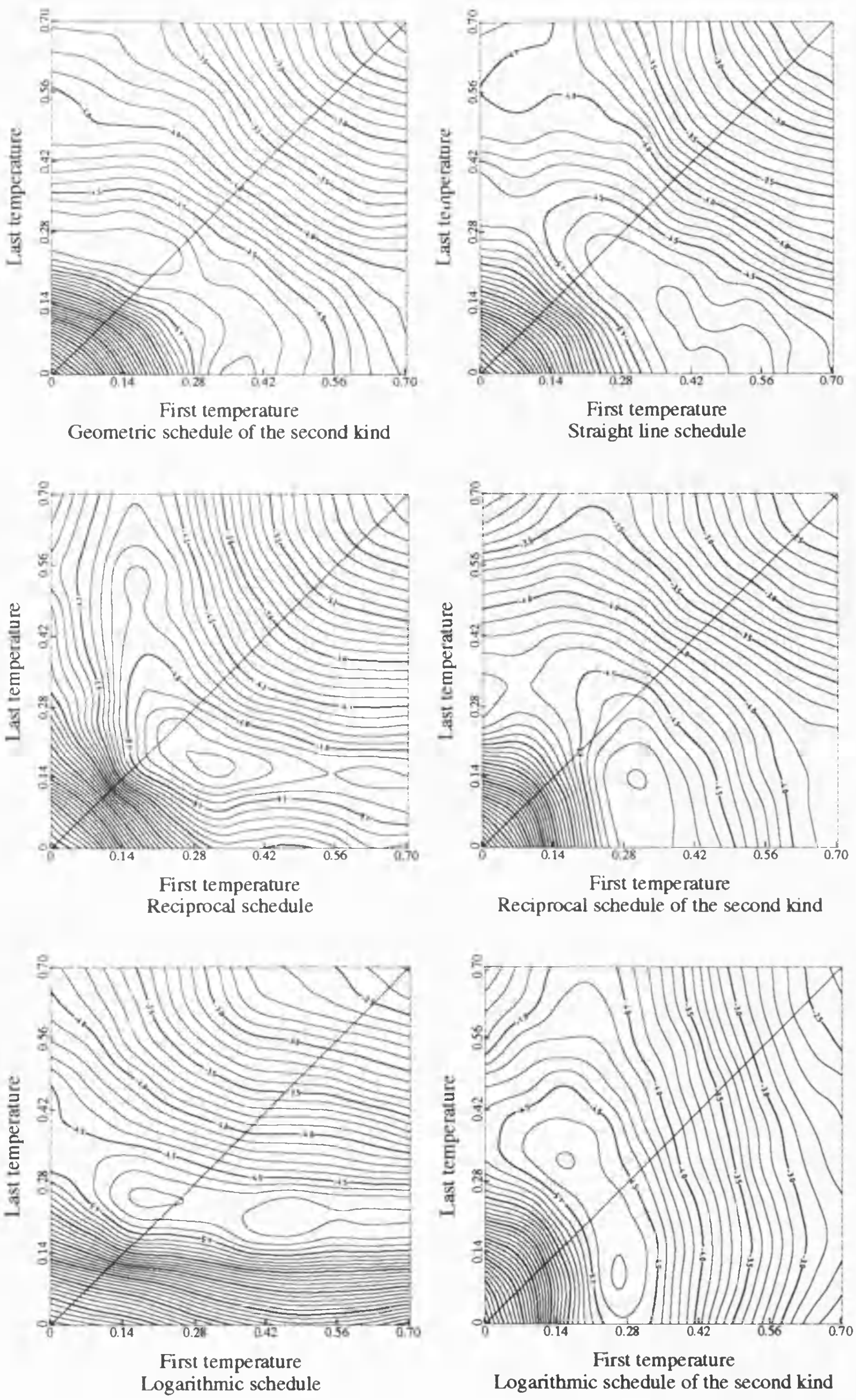
$$h(m) = \frac{(l^2 \log(M+1) - f^2 \log(2)) + (f^2 - l^2)\log(M-m+2)}{(l \log(M+1) - f\log(2)) + (f-l)\log(M-m+2)};$$

8. Constant schedules

$$g(m) = f = l = h(m) > 0.$$

The fourth contour plot of Figure 4.5 shows the logarithm of TPF as given in expression (4.7) for $0.0 < f, l \leq 0.7$ when the geometric schedule is used. This range of first and last temperatures was considered to be the most appropriate. In Figure 4.7 we present similar contour plots for the following six schedules: geometric of the second kind, straight line, reciprocal, reciprocal of the second kind, logarithmic and logarithmic of the second kind. The six contour plots given in Figure 4.7 are in many senses very similar to the contour plot of the geometric schedule over the same region of (f, l) space given in Figure 4.5. If, for example, one considers taking a walk over the surface in the $f > l$ region along a line parallel and fairly close to the line $f = l$, one first descends rapidly before reaching quite a broad area of value less than –4.6. After this 'valley' one ascends once more, but this ascent is less rapid than the previous descent. The line $f = l$ itself corresponds to schedules that have constant temperature. In Figure 4.8 we present the results of a walk along this line by plotting the logarithm of TPF against the constant temperature $\tau > 0.0$ that is used throughout the annealing. Again we see a sharp descent to the minimum followed by a less rapid ascent. A plot such as that shown in Figure 4.8 may provide a good way to find a bound b such that the region $f, l \leq b$ is a good one for further examination.

The reader is now invited to concentrate on the area of each contour plot with values less than –4.6. The size of this region is about the same in all the seven plots just mentioned, although it seems bigger in the case of the straight line schedule and smaller in the case of the logarithmic schedules, although for the first kind of logarithmic schedule this area is long, but thin. The most important point to note from these contour plots is that there is very little effective variation between plots, but considerable variation within plots. In other words, it is the choice of f and l that is important, rather than the choice of the temperature schedule itself. From a practical point of view we recommend the straight line schedule because it combines simplicity with good performance. We also reiterate that the best results seem to occur when f and l are relatively small (*e.g.* $0.0 < f, l \leq 0.7$). Larger values seem to cause the algorithm to



Geometric schedule of the second kind

Straight line schedule

Reciprocal schedule

Reciprocal schedule of the second kind

Logarithmic schedule

Logarithmic schedule of the second kind

Figure 4.7: *Contour plots of the logarithm of the mean value of the transformed penalty function for* $0.0 < f, l \leq 0.7$ *for the six schedules indicated,* $c = 2$

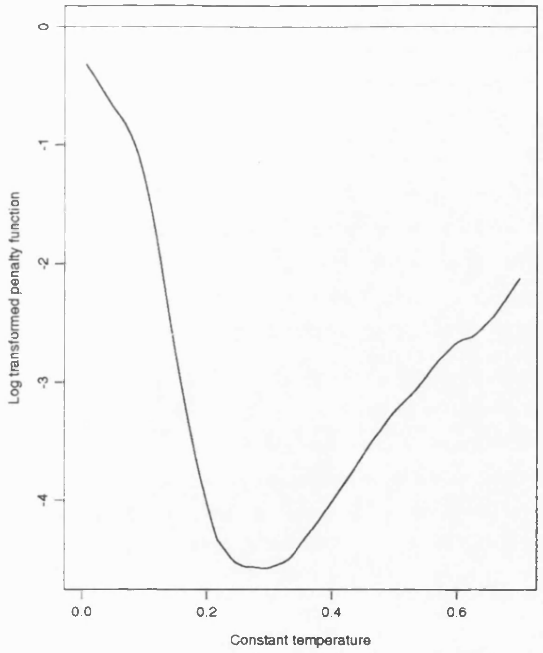


Figure 4.8: *Log of the mean value of the transformed penalty function, $\tau(t)$ = constant, $c = 2$*

perform badly.

4.4.3 Images with more than two colours

So far we have only considered the case when the image has two colours. With such images we were able to make use of the fluid flow approach of Greig, Porteous and Seheult[17] to find the global minimum of the penalty function (4.1). In this section we consider the minimization of (4.1) in the case when $c = 5$, c being the number of (unordered) colours. The image of interest comprises 32×32 pixels. To each pixel we add independent normal noise of variance $\kappa = 1.0$. The maximum likelihood estimate misclassifies 663 (64.75%) pixels. We consider the minimization of (4.1) with the smoothing parameter β equal to 0.5. With $\beta = 0.5$ the value of (4.1) for the original image was 590.43, while for the maximum likelihood estimate it was 766.70. The reconstruction produced by the ICM algorithm starting from the maximum likelihood estimate misclassified 397 (38.77%) pixels and gave a value of (4.1) equal to 554.23. Simulated annealing is now employed to reconstruct the image. We employ 64 iterations and again only consider the algorithm **lowest plus ICM**.

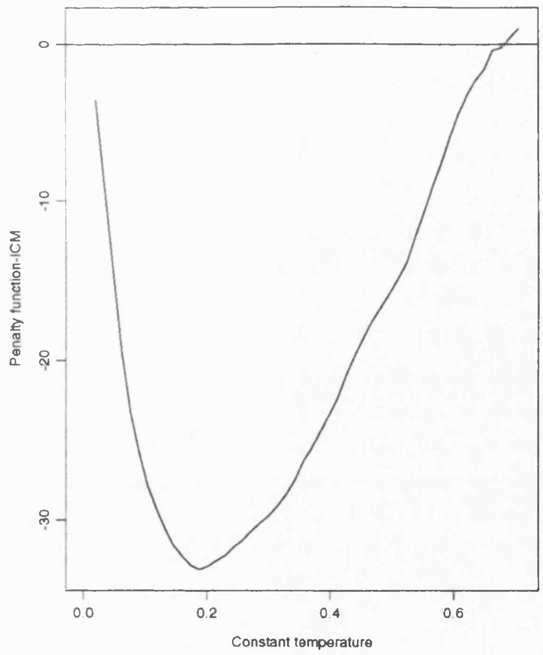
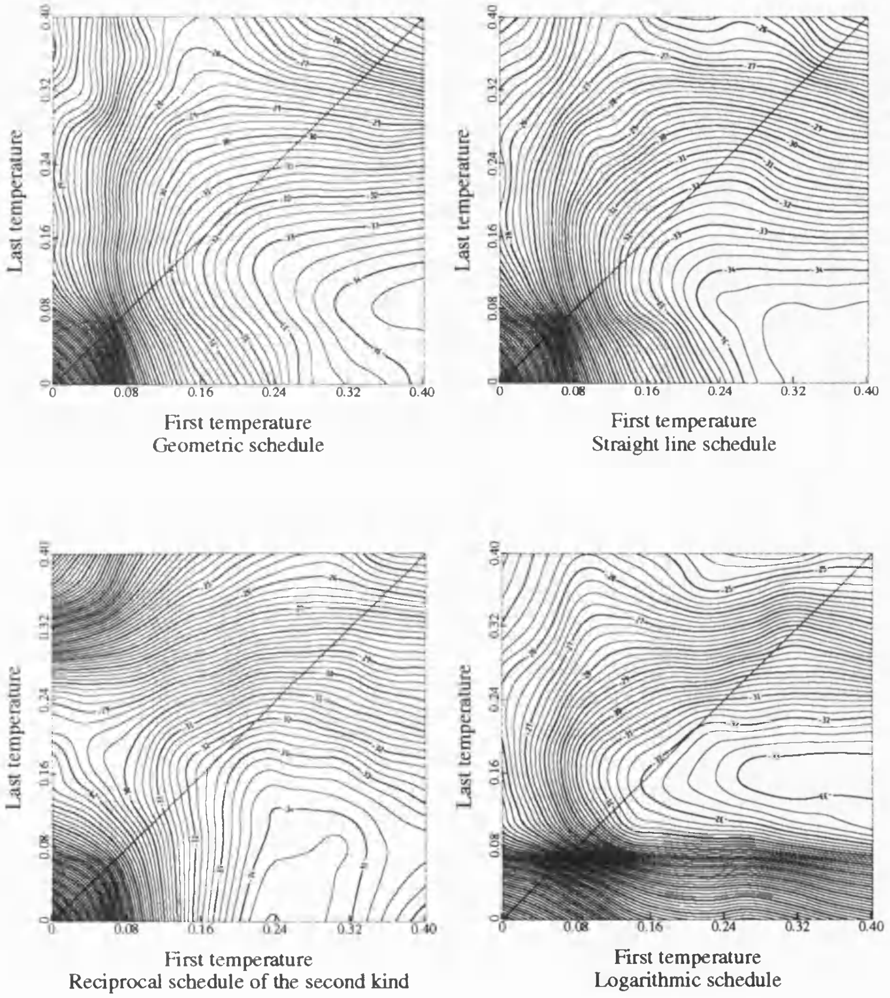


Figure 4.9: *Mean value of the penalty function relative to ICM,* $\tau(t)$ = *constant,* $c = 5$

In order to assess the effects of the various schedules we begin with schedules that take the same value for all iterations, namely constant schedules. In Figure 4.9 we present a plot of the mean value of the penalty function (4.1) minus the value of the penalty function achieved by ICM, for the range of temperatures (0.0,0.7]. For computational reasons we take the average over 10 reconstructions at each temperature considered. The horizontal line indicates the value achieved by the ICM algorithm with initial image the maximum likelihood estimate. We see quite a rapid descent in penalty function to a minimum value that occurs around the temperature 0.2, followed by a somewhat less rapid ascent. This suggests that we should investigate the other (two parameter) schedules in the range $0.0 < f, l \leq 0.4$.

In Figure 4.10 we present contour plots of the mean value of the penalty function for $0.0 < f, l \leq 0.4$ for the geometric schedule, the straight line schedule, the reciprocal schedule and the logarithmic schedule. These contour plots are not dissimilar to those given in Figure 4.5 and Figure 4.7 (and indeed, as we will see in Section 4.5.2, to those given in Figure 4.13). The minimum is achieved by values of f and l in the $f > l$ region. There does not seems to be much effective variation between the schedules: inspection of the area of values less than −32.0,



First temperature
Geometric schedule

First temperature
Straight line schedule

First temperature
Reciprocal schedule of the second kind

First temperature
Logarithmic schedule

Figure 4.10: *Contour plots of the mean value of the penalty function (relative to ICM) for* $0.0 < f, l \leq 0.4$ *for the four schedules indicated,* $c = 5$

say, reveals that this region is very similar in the case of the geometric schedule, straight line schedule and geometric schedule of the second kind, although for the logarithmic schedule it is much smaller and thinner. Of the four schedules presented in Figure 4.10, we judge the logarithmic schedule to be the least good. We recommend the straight line schedule in this case, as well as in the case of binary images, due to its good performance and simplicity. However, it is clear from Figure 4.10 that the most important consideration is again not the choice of schedule but the choice of the first temperature f and the last temperature l.

4.5 Simulated annealing and grey-level images

In this section we discuss simulated annealing as applied to grey-level images. We use the example that we considered in detail in Chapter 1. The reader is referred to Figure 1.4 where the various images that we now discuss are presented. The original image comprises 32×32 pixels. There are 64 possible grey levels and four distinct regions based on the three grey-levels, 15, 30 and 45. We corrupt the original image by the addition of independent normal noise with mean equal to 0.0 and variance κ equal to 20.0 to produce the record y. We attempt to recover the original image from y by the minimization of the penalty function given in (4.2). We take $\alpha = 0.075$ and $\beta = 2.5$. To reduce computation again we only consider a first-order model ($D = 0.0$). The true image has a value of (4.2) equal to 724.20, whereas the value of (4.2) for the maximum likelihood estimate is much higher at 2671.39. The reconstruction produced by ICM has a value of (4.2) equal to 781.03. In Figure 1.4 we present the reconstruction obtained by applying the ICM algorithm to convergence to the reconstruction given by $M = 64$ sweeps of simulating annealing with a straight line temperature schedule with $f = 0.3$ and $l = 0.05$. This reconstruction has a value of (4.2) equal to 718.93 We recall that the iteration of simulated annealing that yielded the lowest value of (4.2) at 721.41 was the final iteration. The version of simulated annealing employed for this example includes a slight modification due to Geman and Reynolds[14]. The aim of this modification is to reduce the computation required. This is particularly important when we are dealing with grey-level images. When updating the value of the estimate of the image at pixel i, instead of sampling from the actual conditional distribution of x_i that puts positive weight on all the 64 grey-levels, the support of the distribution is reduced to the values obtained by taking the union of small intervals about the current value at site i, the current values at the neighbours of i and the data value y_i. We consider an interval of radius five grey-levels about each of these six values. Geman and Reynolds[14] state that this modification yields no apparent change since the true distribution places virtually zero mass on the complement of the reduced support, and that it would be interesting to understand the behaviour of this 'truncated' algorithm from a theoretical viewpoint. In Section 4.5.1 we present a small simulation study to assess the effect of this modification on the value of the penalty function achieved by the simulated annealing algorithm. This study provides us with no evidence against using the truncated algorithm when dealing with grey-level images. In Section 4.5.2 we present an investigation into various temperature schedules similar to the one presented in Section 4.4.

4.5.1 A small simulation study to assess the performance of the truncated algorithm

Here we present the results of a small simulation study designed to compare the performance of the original annealing algorithm with that of the truncated algorithm, which has just been discussed in detail.

We repeat the reconstruction experiment described in Section 4.5 one hundred times for the original algorithm and one hundred times for the truncated algorithm. In Figure 4.11 we present histograms of the values of the penalty function (4.2) achieved. In Figure 4.11 the mean is indicated by the broken vertical line. The mean value of (4.2) for the original algorithm is 718.15 and the variance is 0.1383. The corresponding figures for the truncated version are 718.10 and 0.3542. Accordingly, the mean value achieved by the truncated algorithm is less than the mean value achieved by the original algorithm, although the variances have the opposite order. The range for the original algorithm is [717.88,720.40] and the range for the truncated algorithm is [717.76,720.03]. Examination of the two histograms allows us to remark that 72 realizations from the truncated algorithm take the minimum value obtained by that algorithm, namely 717.76, whereas only 31 realizations from the original algorithm take the minimum value obtained by that algorithm, the slightly higher 717.88. Further comments about Figure 4.11 are difficult to make. However, there is no evidence to suggest that the truncated algorithm performs substantially worse than the original, and certainly some evidence to suggest that it is to be preferred. Moreover, the average number of grey-levels considered at each pixel over the one hundred runs of the algorithm is 16.49, about one quarter of the 64 considered by the original algorithm. This clearly represents a substantial saving in

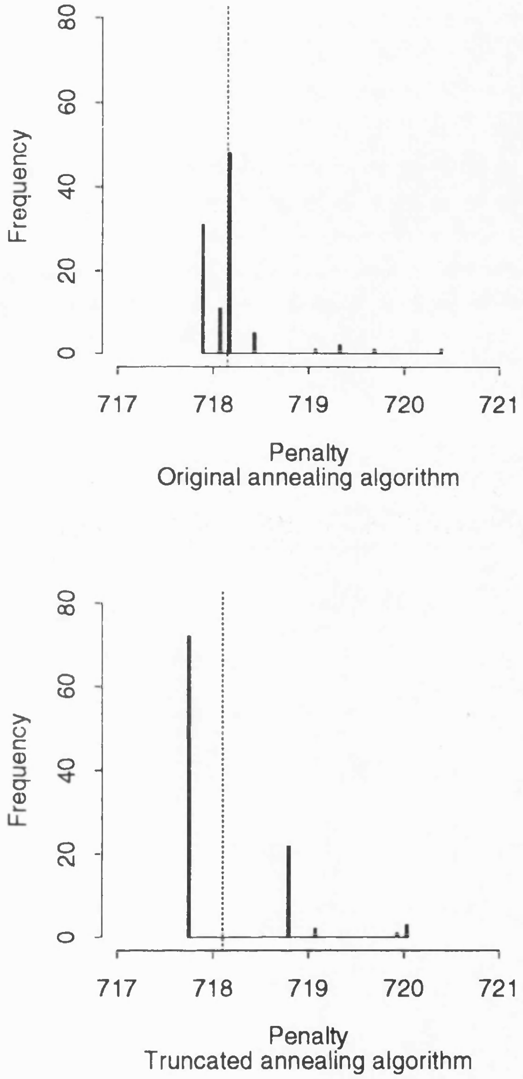


Figure 4.11: *The penalty function achieved by the original algorithm and the truncated algorithm*

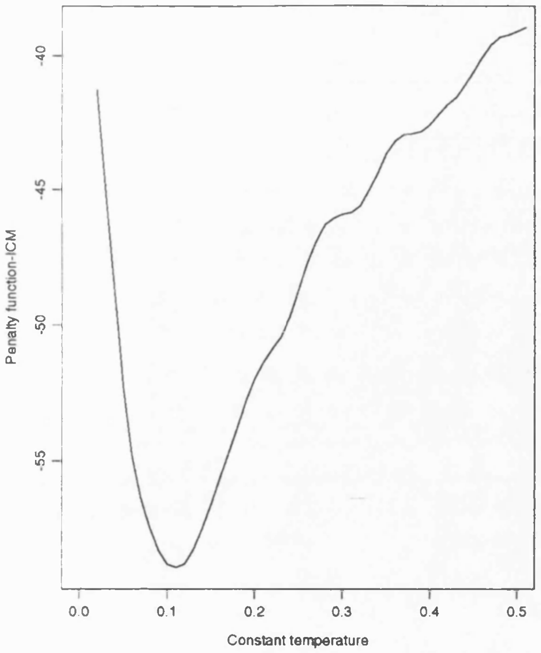


Figure 4.12: *Mean value of the penalty function relative to ICM, $\tau(t)$ = constant, g = 64*

computation. Accordingly, we see no reason not to adopt the truncated algorithm when dealing with grey-level images. Further work on the truncated algorithm would try to quantify the statement in [14] that the true distribution places virtually zero mass on the complement of the reduced support.

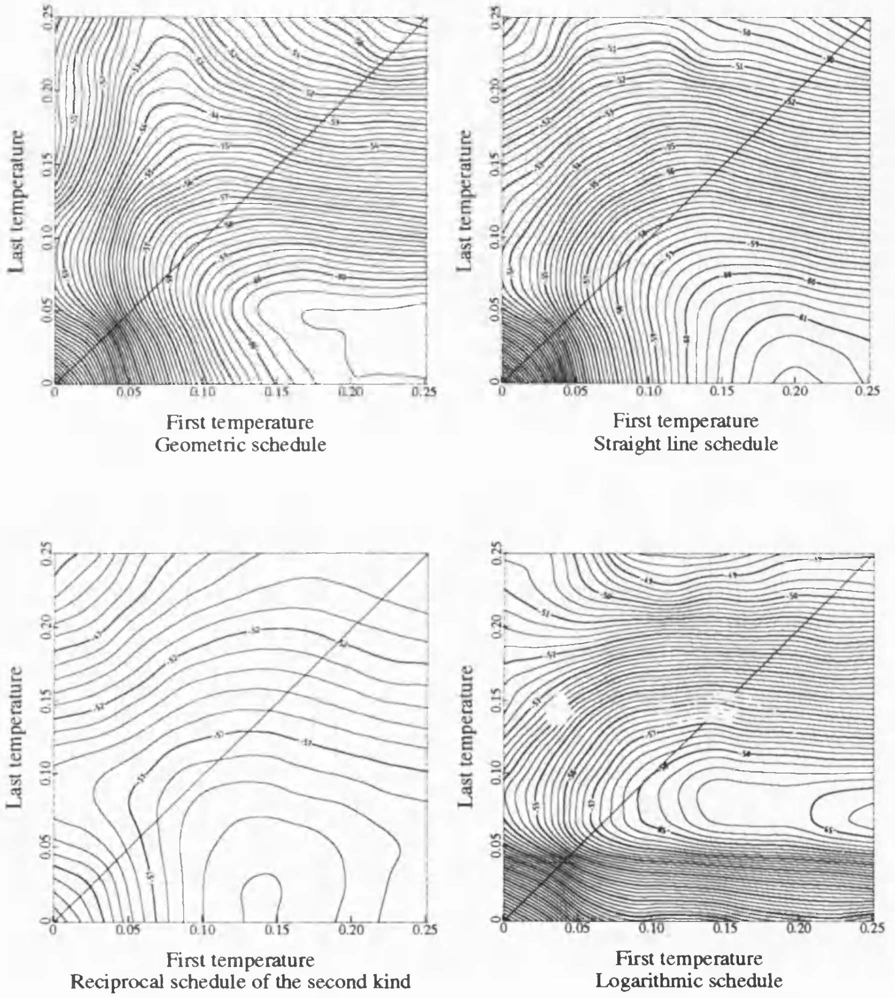
4.5.2 Practical temperature schedules for grey-level images

In this section we investigate the choice of the temperature schedule when we are dealing with a grey-level image. The approach taken is similar to that adopted in Section 4.4, and we consider only algorithm **lowest plus ICM**. We begin our investigation with schedules that take the same value for all iterations, namely constant schedules. In Figure 4.12 we present a plot of the mean value of the penalty function (4.2) minus the value of the penalty function achieved by ICM, for the range of temperatures (0.0,0.5]. For computational reasons we take the average over 10 reconstructions at each temperature considered. Again we see quite a rapid descent in the penalty function to a minimum value that occurs around the temperature 0.1, followed by a less rapid ascent. This suggests that we should investigate the other (two parameter) schedules in the range $0.0 < f, l \leq 0.25$.

In Figure 4.13 we present contour plots of the mean value of the penalty function (relative to ICM) for $0.0 < f, l \leq 0.25$ for the geometric schedule, the straight line schedule, the reciprocal schedule and the logarithmic schedule. These contour plots are not dissimilar to those given in Figure 4.5 and Figure 4.7 for the case when $c = 2$, and Figure 4.10 for the case when $c = 5$. Again the minimum is achieved by values of f and l in the $f > l$ region. There does not seems to be substantial variation between the schedules: inspection of the area of values less than −58.0, say, reveals that this region is very similar in the case of the geometric schedule, straight line schedule and geometric schedule of the second kind, although for the logarithmic schedule it is much smaller and thinner. Indeed, of the four schedules presented in Figure 4.13, the logarithmic schedule seems to be least good. Once more we recommend the straight line schedule due to its good performance and simplicity. However, the most important consideration is again not the choice of schedule but the choice of the first temperature f and the last temperature l.

4.6 Conclusions

In this chapter we have conducted an investigation into the performance of the simulated annealing algorithm as used in the context of image analysis to minimize an appropriate penalty function. After introducing the algorithm and reviewing some of the asymptotic theory, we considered by means of simulations the finite time behaviour of the algorithm. We proposed three variations on the basic simulated annealing algorithm and produced evidence to suggest that one of these variations out-performs the others. We then concentrated on that variation. We discussed practical temperature schedules for binary and multi-colour images, and for grey-level images, by considering the performance in particular examples of many different families of temperature schedules, parameterized by two parameters, the first and the last temperatures. We saw that, while there was not much effective difference between schedules, there was considerable variation within schedules in the sense that performance depended heavily on the first and last temperatures. In particular, it seems that high values of these temperatures give poor results. In general we would recommend the use of a straight line schedule, and we note that the logarithmic schedule, as given by the asymptotic theory on simulated annealing, often performed disappointingly. In the context of grey-level images we examined a further variation



Geometric schedule

Straight line schedule

Reciprocal schedule of the second kind

Logarithmic schedule

Figure 4.13: *Contour plots of the mean value of the penalty function (relative to ICM) for* $0.0 < f, l \leq 0.25$ *for the four schedules indicated,* $g = 64$

on the basic simulated annealing algorithm, known as the truncated algorithm, that reduced the computation required, without noticeably affecting the results.

Further work would involve an attempt to produce some theoretical results along the lines of those given by Geman and Geman[12] for the algorithm **lowest plus ICM** and the truncated algorithm, both in the asymptotic case and the finite time case. We do not expect this task to be easy. A comparison of the performance of the simulated annealing algorithm applied to grey-level images with the approximate grey-level MAP technique described in Chapter 7 of Jubb[24] would be of interest, and it is hoped that this would be the subject of further research by the author.

Chapter 5

Estimating Linear Functionals of a PET Image: Introduction and Theory

The second part of this thesis comprises this chapter and Chapter 6, and considers a topic in positron emission tomography (PET). In this case the image of interest represents the metabolic activity of a cross-section of the brain or other organ and can be thought of as a density f defined on the unit circle. The radioactive tagging of glucose gives rise to emissions of positrons distributed as a Poisson process on the unit circle with intensity f. Each positron that is emitted annihilates with a nearby electron and yields two photons that fly off in opposite directions along a line with uniformly distributed orientation. Ideally, a continuous circular ring of detectors placed around the patient's head makes it possible to detect the photon pair and to give a line l on which the point of emission must have occurred. Thus, the observed data are not drawn from the density f of real interest, but rather from another derived from f by the application of an integral operator. Much work has concentrated on estimating f from these indirectly observed data. However, recently some interest has arisen in estimating linear functionals of the density, rather than f itself. An added complication is that, in practice, the circular ring of detectors is not continuous but comprises a finite number of detectors. In this discrete case, only a tube within which the line l lies is known. In this chapter we consider the problem of estimating linear functionals of a PET image in both the continuous and discrete cases, and in Chapter 6 we present some numerical examples that illustrate the theory developed here.

5.1 Introduction

The main aim of this chapter is to describe in detail the theoretical aspects of the estimation of linear functionals of a PET image. This work relies heavily upon two recent papers by Iain M. Johnstone and Bernard W. Silverman, [21] and [22]. Thus, in this chapter we present a thorough review of these two papers and other related papers, as well as our work. In Chapter 6 we describe some numerical experiments that we have undertaken and that relate directly to the theory reported in this chapter.

In Section 5.2 we outline the positron emission tomography problem in detail, and introduce much of the technical machinery and notation that we shall use later. In addition, we discuss the minimax approach which we shall employ for the estimation of linear functionals, and we review the main techniques and results from [21] and [22], and other relevant papers. In Section 5.3 we introduce the problem of estimating a linear function of the density f both in the idealised case when the ring of detectors is considered to be continuous and in the more realistic discrete case when the ring comprises a finite number N of detectors. We define the loss function, derive the minimax estimator, minimax risk and least favourable function, and establish the remarkable result that the minimax estimator can be found by applying the functional of interest T to a function $\hat{f}$, which does not depend on the functional T and which minimizes a certain penalized least squares form. We investigate the consequence of doubling the number of detectors N, and the effect on the minimax risk of letting n, a measure of the number of emissions, tend to infinity. We also discuss the least favourable function: that is, the function that yields the maximum risk. In Section 5.4, we review other work in this area and in Section 5.5 we discuss some generalizations due to Silverman[38] and other workers. Finally, in Section 5.6 we present our conclusions.

5.2 Background and setting up the model

There are many practical problems where the observed data are not drawn directly from the density f of real interest, but rather from another derived from f by the application of an integral operator. Two examples of this type of problem, those of stereology and positron emission tomography, are considered by Silverman, Jones, Wilson and Nychka[40]. In this chapter our main concern will be the PET case, although the results obtained are readily generalizable, as we shall see in Section 5.5.

Tomography is a non-invasive technique for reconstructing the internal structure of an object of interest, often in a medical context. PET deals with the estimation of the amount and location of a radioactively labelled metabolite on the basis of particle decays indirectly observed outside the body. We follow the set-up of Vardi, Shepp and Kaufman[44], which we have already outlined. The brain, heart or liver is scanned by counting radioactive emissions from tagged glucose. The radioactive tagging of the glucose gives rise to emissions of positrons distributed as a Poisson process in space and time; the spatial intensity of emissions is the same as the distribution of glucose and gives an indication of the organ's metabolic activity. It is convenient to renormalize the emission intensity to be a probability density function $f(r, \theta)$, say, with respect to a normalised Lebesgue measure μ, where $d\mu(r, \theta) = \pi^{-1} r \, dr \, d\theta$ and r and θ are polar coordinates. We discuss μ in detail in Section 5.2.2. Each positron that is emitted annihilates with a nearby electron and yields two photons that fly off in opposite directions along a line with uniformly distributed orientation. A circular ring of detectors placed around the patient's head makes it possible to detect the photon pair and hence, for each emission that is detected, to give a line on which the point of emission must have occurred. We shall refer to this line as the detected line. It is, however, not possible to detect the position of the emission on the line. This set-up is in fact an idealization: in reality we can only identify a tube within which the detected line lies.

Throughout this work, we make the further idealization that the ring of detectors defines a slice of the patient's head which is planar. In this way the problem under consideration is essentially two-dimensional. We make no attempt to extend our results to take into account this third dimension. In the whole of Section 5.2, our interest is in reproducing a picture of the metabolic activity of a cross-section of the brain from the detected lines (or tubes).

5.2.1 Single photon emission computed tomography

Another form of tomography that has received recent attention in the literature is SPECT, or single photon emission computed tomography (see, for example, Geman and McClure[13] and Green[15]). Again the aim is to determine the concentration of a pharmaceutical in a part of the body such as the brain, liver or heart. In both SPECT and PET this concentration is estimated by detecting photon emissions from a dose of the pharmaceutical that has been combined with a radioactive isotope. (The number of emissions observed in a typical experiment using SPECT seems to be smaller than the number observed in a typical experiment using PET.) However, unlike in PET where the detectors take the form of a ring around the organ of interest, in SPECT they are arranged in a linear array of L detectors. This array can be rotated about an axis through the patient to any orientation θ relative to a fixed line. An excellent diagram of this set-up is given in Figure 1 of Geman and McClure[13]. These authors assume that the detector array is positioned at K equally spaced angles θ_k for duration T time units at each angle. Then at each of the K angles, the number of single photons reaching each of the L detectors is recorded. This is the data from which the concentration of the pharmaceutical is estimated. In this thesis we do not consider SPECT any further but confine our attention to PET. We point out, however, that in practice SPECT is more widely used than PET, partly because the machines and the experiment are cheaper.

5.2.2 Setting up the model

Johnstone and Silverman[21] introduce the notion of *brain space B* and *detector space D*. Brain space is the original disc in the plane enclosed by the detector ring, whereas detector space is the space of all possible unordered pairs of points on the detector circle.

Brain space B is considered to be the unit circle and is equipped with a dominating measure μ which is defined to be proportional to Lebesgue measure as follows:

$$d\mu(r, \theta) = \pi^{-1}\, r\, dr\, d\theta$$

for $0 \leq r \leq 1$ and $0 \leq \theta \leq 2\pi$ if polar coordinates are used, and

$$d\mu(x_1, x_2) = \pi^{-1}\, dx_1\, dx_2$$

for $||x|| \leq 1$, where $x = (x_1, x_2)$ in Cartesian coordinates. Note that μ integrates to 1 over the unit circle. We stated in Section 5.2 that it is convenient to renormalize the emission intensity to be a probability density function with respect to μ. This density of interest, f, say, associated with the fixed number n of unobserved independent random variables $X_1, X_2, \ldots, X_n$, where X_i is the position of the ith emission in the brain, is defined on brain space, B.

To parameterize detector space D, Johnstone and Silverman[21] let s be the length of the perpendicular from the origin to the detected line, and ϕ be the orientation of this perpendicular (see their Fig. 2). Thus D is $\{(s, \phi) : 0 \leq s \leq 1, 0 \leq \phi \leq 2\pi\}$. As was the case with brain space, detector space is equipped with a dominating measure λ, defined by

$$d\lambda(s, \phi) = 2\pi^{-2}(1 - s^2)^{1/2}\, ds\, d\phi,$$

which also integrates to 1. Associated with the independent observations $Y_1, Y_2, \ldots, Y_n$, where Y_i corresponds to the i th observed pair (or line), is the density $g = g(s, \phi)$.

The density g on D is related to the density f on B by the linear[1] transformation P, where

$$\begin{aligned} g(s, \phi) &= Pf(s, \phi) \\ &= \frac{1}{2}(1 - s^2)^{-1/2} \int_{-\sqrt{1-s^2}}^{\sqrt{1-s^2}} f(s\cos\phi - t\sin\phi,\ s\sin\phi + t\cos\phi)\, dt. \end{aligned} \tag{5.1}$$

The integral is the so-called *Radon transform* of the density f, namely the line integral of f along the line l with coordinates (s, ϕ) in detector space. As the length of the intersection of this line with B is $2\,(1-s^2)^{1/2}$, $Pf(s, \phi)$ represents the average of f over the part of l that intersects B.

In any particular PET scan, not all the pairs of emitted photons are detected. Johnstone and Silverman[21] examine two reasons for this: the effect of the third dimension and attenuation. The effect of the third dimension is due to the fact that in reality the detectors form a ring of finite thickness $d > 0$, and the orientation of the line of flight of the photons is uniformly distributed in R^3. Johnstone and Silverman[21] assume that the density is constant over the thickness of the cylindrical slab enclosed by the detector ring, and present a formula for $a_{3D}(s, \phi)$, the probability that an emission in the tube defined by (s, ϕ) is actually detected. This quantity increases as s increases, reflecting the fact that emissions in shorter tubes (large s) are more likely to be detected. Silverman, Jones, Wilson and Nychka[40] also discuss the effect of the third dimension and show that, in fact, the three dimensional problem does not tend, in the limit as $d \rightarrow 0$, to the two dimensional problem. Attenuation is defined as the loss of a detection caused by the absorption or scattering of one of the photons in flight. Johnstone and Silverman[21] show that the probability that neither photon due to an emission in the tube defined by (s, ϕ) will be lost is given by $a_A(s, \phi)$. In both cases the probability that the emission will be detected depends only upon the tube (s, ϕ) and not on the emission's position within the tube. In general, if both effects are considered, the probability that any particular detection

[1] We use linear in the sense that $P(\lambda_1 f_1 + \lambda_2 f_2) = \lambda_1 P(f_1) + \lambda_2 P(f_2)$, where λ_1 and λ_2 are scalars and f_1 and f_2 are densities.

will not be lost will be $a_{3D}(s, \phi)\, a_A(s, \phi)$. Thus, the two effects can be combined into a single $a(s, \phi) \in (0, 1]$. It follows that the *observed* detections form a *biased sample* with density in detector space with respect to $d\lambda\,(s, \phi)$:

$$g_a(s, \phi) = P_a f(s, \phi) \propto a(s, \phi)\, Pf(s, \phi). \tag{5.2}$$

A further remark on the incompleteness of sampling will be found in Section 5.2.7.

Our interest throughout Section 5.2 is to produce an estimate of the density f from the indirect observations $Y_1, \ldots, Y_n$. Only in Section 5.3 do we start to consider the estimation of linear functionals of f.

5.2.3 Some proposals for the estimation of a PET image

A very important paper on this subject is Vardi *et al.*[44], the ideas of which we review briefly. These authors use a slightly different notation from the one that we generally use. However, in this section we shall adopt their notation; the connection with our work is clear.

It is assumed that the data arise in histogram form (see our discussion of 'discretization' in Section 5.2.9), so that detector space D is divided into bins (or tubes), indexed in [40] by $t = 1, \ldots, T$, with an observed number of counts n_t in the tth bin. Brain space B, which is assumed to be the unit circle, is also divided into S bins (or pixels), indexed by s. The pixellation used is arbitrary and will be discussed later in this section. The assumption of [44] is that events occur in pixel s according to a Poisson distribution with mean $\lambda(s)$. The aim is to estimate $\{\lambda(s),\ s = 1, \ldots, S\} = \lambda$, say.

Let

$$p(s, t) = \Pr\,[\text{an event in pixel } s \text{ gives rise to a count on tube } t],$$

and set

$$q(s) = \sum_{t=1}^{\mathrm{T}} p(s, t) \leq 1,$$

since an event in pixel s may not be recorded. We assume that the $p(s, t)$ s are known from the geometry of the set-up. Next, define $K(s, t)$ to be the number of events on pixel s counted on tube t, and note that the $K(s, t)$ s are Poisson random variables with mean $\lambda(s)\, p(s, t)$ and are independent of each other. Unfortunately, we only observe the total counts on each tube as

$$N(t) = \sum_{s=1}^{S} K(s, t).$$

These too are Poisson random variables with mean $\sum_{s=1}^{S} \lambda(s) p(s, t)$ and are independent of each other. A natural statistical approach to the estimation of λ is maximum likelihood (ML). If we let

$$L(\lambda) = \Pr[N(t) = n(t),\ t = 1, \dots, T \mid \lambda],$$

where $n(t)$ is the observed number of counts in tube t, it is easy to see that

$$L(\lambda) = \prod_{t=1}^{\mathrm{T}} \exp\{-\sum_{s=1}^{S} \lambda(s) p(s, t)\} \frac{[\sum_{s=1}^{S} \lambda(s) p(s, t)]^{n(t)}}{n(t)!}.$$

Vardi *et al.*[44] show that $\log(L(\lambda))$ is concave and propose the EM (E for *expectation* and M for *maximization*) algorithm, as described in Dempster, Laird and Rubin [8], for finding the maximizing λ. We now briefly describe the EM algorithm in the context of PET. To begin, think of $N(t)$, $t = 1, \dots, T$, as *incomplete* (but observed) data, and $K(s, t)$, $s = 1, \dots, S$ and $t = 1, \dots, T$, as *complete* (but unobserved) data. Now, if we were to observe the $K(s, t)$ s, we would be able to consider $\sum_{t=1}^{\mathrm{T}} K(s, t)$, the total number of counts originating from pixel s, which are Poisson random variables with mean $\lambda(s) \sum_{t=1}^{\mathrm{T}} p(s, t) = \lambda(s) q(s)$ and are independent from each other. In this case the maximum likelihood estimator of λ maximizes

$$\sum_{s=1}^{S} \sum_{t=1}^{\mathrm{T}} \{K(s, t) \log \lambda(s) - \lambda(s) p(s, t)\} \tag{5.3}$$

and is thus

$$\hat{\lambda}(s) = \frac{\sum_{t=1}^{\mathrm{T}} K(s, t)}{q(s)}, \quad s = 1, \dots, S. \tag{5.4}$$

However, as in practice the $K(s, t)$ s are not observed, we proceed using the EM algorithm. We start with any estimate $\hat{\lambda} > 0$, and repeat iterations of the following E and M steps.

E step. Estimate the complete data from the incomplete data by

$$\hat{K}(s, t) \quad = \quad \mathrm{E}[K(s, t) \mid N(t) = n(t),\ \hat{\lambda}^{\mathrm{old}}]$$

$$= \quad n(t)\frac{\hat{\lambda}^{\text{old}}(s)\,p(s,\,t)}{\sum_{r=1}^{S}\hat{\lambda}^{\text{old}}(r)\,p(r,\,t)},$$

since, for independent Poisson variables X and Y with means λ_X and λ_Y,

$$\mathrm{E}\,[X\,|\,X+Y=x+y] = (x+y)\frac{\lambda_X}{(\lambda_X+\lambda_Y)}.$$

M step. Estimate $\hat{\lambda}^{\text{new}}$ by maximum likelihood based on the estimated complete data $\hat{K}(s,\,t)$:

$$\hat{\lambda}^{\text{new}}(s) = \frac{\sum_{t=1}^{\mathrm{T}}\hat{K}(s,\,t)}{q(s)},\ s=1,\ldots,S,$$

as explained above through equations (5.3) and (5.4). We now make some remarks.

1. The E step and the M step can be combined into a single updating step.

2. λ remains positive throughout.

3. General theorems about EM estimation tell us that the likelihood increases at each iteration, and, accordingly, the method will converge to a global maximum of $L(\lambda)$ since log $L(\lambda)$ is concave.

4. In practice this method works well. However, the reconstruction improves as the EM algorithm iterates up to a point, but then things get worse and the solution becomes spiky and inaccurate.

5. No attempt to model the true image has been made.

Remark 4 suggests that some form of smoothing may be appropriate. This may be introduced by incorporating a prior model for λ, as suggested by Remark 5. Suppose that a prior model for λ takes the form

$$p(\lambda) \propto \exp\{-R(\lambda)\},$$

where $R(\lambda)$ is some measure of the roughness of the image, then a standard argument shows that in this case the maximum likelihood approach requires the maximization of log $L(\lambda)-R(\lambda)$, as opposed to log $L(\lambda)$. However, finding the maximum of log $L(\lambda)-R(\lambda)$ is not an easy task: one possible approach is to use the EM algorithm as described above, with an adapted M step:

find $\hat{\lambda}$ to maximize

$$\sum_{s=1}^{S}\sum_{t=1}^{T}\{\hat{K}(s,t)\log\lambda(s)-\lambda(s)p(s,t)\}-R(\lambda), \tag{5.5}$$

whereas before we did not have the roughness term $R(\lambda)$. Accordingly, this new M step is much harder, as it is essentially the same problem that arises in image reconstruction problems approached by maximum *a posteriori* estimation using a Markov random field or Gibbs distribution (see Section 1.3.2) and may require simulated annealing for its solution (see Section 1.6.1). There are of course many possibilities for the choice of $R(\lambda)$. For example, Geman and McClure[13] (working in the context of SPECT) propose

$$R(\lambda)=\beta\left(\sum_{[s_1,s_2]}\phi\left(\frac{\lambda(s_1)-\lambda(s_2)}{\delta}\right)+\sum_{<s_1,s_2>}\frac{1}{\sqrt{2}}\phi\left(\frac{\lambda(s_1)-\lambda(s_2)}{\delta}\right)\right), \tag{5.6}$$

where $[s_1, s_2]$ indicates that s_1 and s_2 are nearest horizontal or vertical neighbours in the square lattice that they employ, $< s_1, s_2 >$ indicates diagonal neighbours, the (smoothing) parameter β is positive and controls the strength of the interaction between a pixel and its neighbours, δ is a constant that can be interpreted as a scale parameter on the range of values of $\lambda(s)$, and the function $\phi(\xi)$ is even and minimized at $\xi = 0$. Thus R itself is minimized by images of constant intensity. In [13] ϕ is (up to an additive constant) defined as

$$\phi(\xi) \quad = \quad 1-\frac{1}{1+\xi^2} \quad = \quad \frac{1}{1+\xi^{-2}}. \tag{5.7}$$

We discussed the choice of the function ϕ in Section 1.3.3. There the set-up was slightly different: we set $\alpha = 1/\delta^2$ and took this parameter into the definition of ϕ itself. We shall meet a different ϕ later in this section in equation (5.11).

To overcome the computation problems caused by the modified M step (5.5), Silverman *et al.*[40] propose the EMS (S for *smoothing*) algorithm. In short, a smoothing step is added after the M step to produce a smoothed version of $\hat{\lambda}$, and in this way prior knowledge about the smoothness of the image is incorporated. In simulation experiments, these authors found that the EMS algorithm, with only a small amount of simple smoothing, always converged to good estimates of λ, in a relatively small number of iterations. Moreover, the limit point was observed to be unique. However, attempts to prove both convergence and uniqueness have so far failed. Nevertheless, such attempts have helped to give understanding of the EMS algorithm and are discussed in Section 5 of Silverman *et al.*[40] and in Nychka[31].

Another improvement provided by Silverman *et al.*[40] concerns the pixellation of brain space B. Vardi *et al.*[44], Geman and McClure[13] and many other workers in the field simply superimpose a square grid of pixels over B, but this approach has computational disadvantages compared with discretizations that take better account of the detector geometry. Silverman *et al.*[40] exploit circular symmetries to propose a discretization that leads to substantial computational savings in both storage and time.

Green, in the discussion to Silverman *et al.*[40], proposes another way to overcome the problems caused by the modified M step (5.5). First, write (5.5) as

$$Q(\lambda \mid \lambda^{\text{old}}) - R(\lambda), \tag{5.8}$$

and recall that the difficult part of our task is to maximize (5.8) over λ. This can be done by solving

$$D^{10}Q(\lambda \mid \lambda^{\text{old}}) - DR(\lambda) = 0, \tag{5.9}$$

for λ, where D denotes the derivative operator and $D^{ij}F(x \mid y)$ means $\partial^{i+j}F(x \mid y) / \partial x^i \partial y^j$. Green, in the discussion to Silverman *et al.*[40], suggests solving

$$D^{10}Q(\lambda \mid \lambda^{\text{old}}) - DR(\lambda^{\text{old}}) = 0 \tag{5.10}$$

in which the gradient of the penalty term is evaluated at the current estimate. He refers to this algorithm as 'one-step-late' (OSL) and points out that solving the system given by equation (5.10) is as trivial as for the unpenalized likelihood problem. In [16], Green states that (5.9) and (5.10) have the same fixed points and so, if the OSL algorithm converges, the limit is a maximum likelihood penalized estimate. What is lost in comparison with the true EM algorithm is the guarantee that the method converges, and in particular that the iteration always increases the penalized log-likelihood. In the context of PET, Green states in [40] that while OSL may not converge if a heavy degree of smoothing is applied (which is rarely necessary in the PET case), it otherwise converges more quickly than the impractical EM procedure. In [16], Green gives some examples of the use of the OSL algorithm: multinomial sample, ridge regression and Poisson additive regression. The latter example, which is relevant to emission tomography, is illustrated in the case of SPECT using real data in Green[15]. In this paper

Green considers

$$y_t \sim \text{Poisson}\left(\sum_s a_{ts}\lambda_s\right) \text{ independently,}$$

where $\{\lambda_s\}$ is a discretized version of isotope concentration as a function of pixel s in the body, $\{y_t\}$ are the recorded counts of particles detected in bins labelled t and $\{a_{ts}\}$ form a matrix of coefficients encoding the physical circumstances under which the data were collected. Green[15] attempts to model the set $\{a_{ts}\}$ in a sophisticated way. We do not concern ourselves with the details here; the interested reader is referred to that paper. Moreover, in that paper, Green gives some diagnostics by means of which he can modify the model. He gives an example of the OSL algorithm applied to real data. He adopts a conventional rectangular grid since, unlike the PET problem, SPECT offers no symmetries to be exploited. He proceeds by running the ordinary algorithm for 16 complete sweeps starting from a 'flat' image and using no prior term. He then calculates residuals and, noticing that they reveal a pronounced pattern, modifies his model by changing the $\{a_{ts}\}$. He runs the OSL algorithm for a further 128 iterations using the log cosh prior obtained by setting the ϕ of equation (5.6) to

$$\phi(\xi) = c_1 \log \cosh(c_2\xi), \tag{5.11}$$

where c_1 and c_2 are chosen to match Geman and McClure[13]'s prior (5.7) in the sense that $\max\phi'$ and $\phi''(0)$ coincide for the two functions, with $\beta = 0.2$ and $\delta = 50.0$. At this point no further changes were perceptible and the residuals give little cause for concern. The resulting reconstruction is considered to be both reasonable and useful.

There are many other ways to estimate a PET image. Vardi *et al.*[44] discuss moment estimates and convolution backprojection, least squares estimators and Stein-type estimators. Such estimates are not of concern to us in this work and we refer the interested reader to [44] for further details. Jones and Silverman[23], however, consider a technique that is closely related to our work. They consider an orthogonal series intensity (or density) estimation approach. For the case of directly observed data, the approach is discussed in Section 2.7 of Silverman[37]. We briefly review that approach. Suppose for the moment that we are trying to estimate a density, f, say, on the unit interval [0, 1], from a directly observed sample $X_1, \dots, X_n$. The basic idea is to express f as a Fourier series with respect to some orthonormal sequence ϕ_ν, where

$\nu \geq 0$:

$$f = \sum_{\nu \geq 0} f_\nu \phi_\nu,$$

where, for each ν,

$$f_\nu = \int_0^1 f(x)\, \phi_\nu(x)\, dx, \tag{5.12}$$

and to estimate the coefficients f_ν. Now suppose that X is a random variable with density f. Then (5.12) can be written

$$f_\nu = \mathbf{E}\,[\phi_\nu(X)],$$

and may be estimated by

$$\hat{f}_\nu = \frac{1}{n} \sum_{i=1}^{n} \phi_\nu(X_i).$$

Silverman[37] shows that the sum $\sum_{\nu \geq 0} \hat{f}_\nu \phi_\nu$ will not be a good estimate of f, but will 'converge' to a sum of delta functions at the observations. Thus, in order to obtain a useful estimate of the density f some smoothing is necessary. The easiest way to apply this is to truncate the expansion $\sum \hat{f}_\nu \phi_\nu$ at some point. To do this, choose an integer K and define the density estimate $\hat{f}$ by

$$\hat{f} = \sum_{\nu=0}^{K} \hat{f}_\nu \phi_\nu, \tag{5.13}$$

where the choice of the cutoff point K determines the amount of smoothing, or, more generally, by

$$\hat{f} = \sum_{\nu \geq 0} \lambda_\nu \hat{f}_\nu \phi_\nu,$$

where the weights λ_ν satisfy $\lambda_\nu \longrightarrow 0$ as $\nu \longrightarrow \infty$, the rate of this convergence controlling the amount of smoothing. Jones and Silverman[23] introduce an estimate of f which defines the Poisson process of the PET set-up based on (5.13). Unfortunately, because of the indirect nature of the problem, the $\hat{f}_\nu$s can not be estimated directly. However, Jones and Silverman[23] set $\hat{f}_\nu = b_\nu^{-1} \hat{g}_\nu$ (compare equation (5.28) of Section 5.2.8), where the b_νs are known and are defined in equation (5.17) of Section 5.2.4, and where the $\hat{g}_\nu$s are easily computable from the (indirectly observed) data. The reader should compare this estimator with the linear minimax estimator as derived and described in Section 5.2.8. Reconstructions for several different values of the smoothing parameter K are presented in [23]. Moreover, an automatic choice of K based upon the mean integrated square error is proposed and found to work well, at least in the example presented. One drawback of this method that emerges from the experiments presented is the presence of negative regions in the reconstructions. The EM and EMS approaches outlined above do not permit such negative regions (see Remark 2 of Section 5.2.3). However, Jones and Silverman[23] report that this orthogonal series intensity estimation approach to PET image reconstruction yields a 30-fold improvement in computer time in comparison with the best EMS procedure of Silverman *et al.*[40]. Another advantage of the approach of [23] is that there is no need to discretize brain space; the truely continuous nature of orthogonal series reconstruction is said to be most appealing. The disadvantage is, however, the difficult with generalizing the approach to cope with more realistic versions of the PET model.

5.2.4 The minimax approach and the singular value decomposition of the Radon transform

Johnstone and Silverman[21] adopt an approach that is somewhat different from those discussed in Section 5.2.3, namely a minimax approach. This is because the main thrust of their interest is not, directly, towards obtaining reconstruction methods but more towards giving a deeper understanding of indirect estimation methods in general. In particular they are concerned with quantifying the ill-posedness of the PET problem. To do so, they calculate theoretically the order of magnitude of the size of a sample of directly observed positron emissions that would be required to be equivalent to a given sample size of the indirectly observed data which is available in practice, in the sense of yielding equally accurate image reconstructions. We shall see examples of the result of such a calculation in Section 5.2.7. They conclude that the amount of information available is still substantial, but is by no means as great as if a sample of direct observations were available. However, before we discuss this in detail, we must set up some more machinery. We will discuss the minimax approach again in Section 5.2.7

The singular value decomposition (SVD) of the normalized Radon transform P defined in (5.1) is of crucial importance to this approach. First, let H be the space of functions on B that are square-integrable with respect to the dominating measure μ and let K be the space of functions on D that are square-integrable with respect to the dominating measure λ. Suppose that the point $X = (X_1, X_2)$ is drawn at random (according to μ) from B. If a direction ϕ is specified by $u_\phi = (\cos\phi, \sin\phi)$, then

$$Pf(s,\phi) = \mathrm{E}\,\{f(X) \mid u_\phi \cdot X = s\}.$$

From this representation it follows at once that P is a bounded operator from H to K with norm 1, and is indeed one-to-one.

Next define the lattice $\mathcal{N}'$ as

$$\mathcal{N}' = \{(j,k) \mid j \geq 0, k \geq 0\}.$$

It can be shown that the set of functions $\{\phi_\nu\}$ is an orthonormal system on H (in the sense that $(\phi_\nu, \phi_{\nu'})_\mu = \int \phi_\nu(x)\,\phi_{\nu'}(x)\,d\mu(x) = \delta_{\nu,\nu'}$, for ν and $\nu' \in \mathcal{N}'$) where

$$\begin{aligned} \phi_\nu(r,\theta) &= \phi_{(j,k)}(r,\theta) \\ &= (j+k+1)^{1/2} Z_{j+k}^{|j-k|}(r)\,e^{i(j-k)\theta} \end{aligned} \qquad (5.14)$$

for $\nu \in \mathcal{N}'$ and $(r,\theta) \in B$, and where Z_m^k denotes the *Zernike polynomial* of degree m and order k, which will be discussed in Section 6.2.3. Similarly, $\{\psi_\nu\}$ is an orthonormal system on K where

$$\begin{aligned} \psi_\nu(s,\phi) &= \psi_{(j,k)}(s,\phi) \\ &= U_{j+k}(s)\,e^{i(j-k)\phi}; \end{aligned} \qquad (5.15)$$

here $\nu \in \mathcal{N}'$, $(s,\phi) \in D$, and $U_m(\cos\theta) = \sin(m+1)\theta / \sin\theta$ are the *Chebyshev polynomials* of the second kind. An arbitrary $f \in H$ and an arbitrary $g \in K$ can be expressed in a way analogous to a Fourier series as

$$\begin{aligned} f(x) &= \sum_{\nu \in \mathcal{N}'} f_\nu\,\phi_\nu(x) \\ g(x) &= \sum_{\nu \in \mathcal{N}'} g_\nu\,\psi_\nu(x) \end{aligned}$$

where the coefficients f_ν and g_ν are given by the appropriate inner products: $f_\nu = (f, \phi_\nu)_\mu =$

$\int_B f(x)\,\phi_\nu^*(x)\,d\mu(x)$ and $g_\nu = (g, \psi_\nu)_\lambda = \int_D g(y)\,\psi_\nu^*(y)\,d\lambda(y)$, the star denoting the complex conjugate. It can be established that, with these inner products, H and K are Hilbert spaces.

It can also be shown that

$$P\phi_\nu = b_\nu \psi_\nu, \tag{5.16}$$

with the *singular values* $b_\nu = b_{(j,k)}$ specified by

$$b_\nu = (j+k+1)^{-1/2}. \tag{5.17}$$

Thus P can be represented by a diagonal matrix $B = \mathrm{diag}\,(b_\nu)$ with respect to the bases $\{\phi_\nu\}$ and $\{\psi_\nu\}$. It is easy to establish that $g_\nu = b_\nu f_\nu$. We also remark that $b_{(0,0)} = 1$.

In addition, because $\phi_{(0,0)} \equiv 1$ and $\psi_{(0,0)} \equiv 1$, $\int \phi_\nu(x)\,d\mu(x) = \int \phi_\nu(x)\,1\,d\mu(x) = (\phi_\nu, \phi_{(0,0)})_\mu = \delta_{\nu,(0,0)}$ and, similarly $\int \psi_\nu(x)\,d\mu(x) = \delta_{\nu,(0,0)}$. Hence, the condition that $\int f\,d\mu = 1$ and $\int g\,d\lambda = 1$ forces $f_{(0,0)} = 1$ and $g_{(0,0)} = 1$. Therefore, we rewrite f and g as

$$\begin{aligned} f(x) &= 1 + \sum_{\nu\neq(0,0)} f_\nu\,\phi_\nu(x) \\ g(x) &= 1 + \sum_{\nu\neq(0,0)} g_\nu\,\psi_\nu(x). \end{aligned}$$

Accordingly, the function f may be represented by the vector $f = (1, f_\nu)$, $\nu \in \mathcal{N}^0$, where $\mathcal{N}^0$ is the set $\mathcal{N}'$ without the element $(0,0)$, and the function g by the vector g which is defined similarly. We remark that if the function $f^{(1)} \in H$ is represented by the vector $f^{(1)} = (1, f_\nu^{(1)})$, $\nu \in \mathcal{N}^0$, and if the function $f^{(2)} \in H$ is represented by the vector $f^{(2)} = (1, f_\nu^{(2)})$, then $(f^{(1)}, f^{(2)})_\mu = \sum_{\nu\in\mathcal{N}'} f_\nu^{(1)} f_\nu^{(2)*} = f^{(1)^{\mathrm{T}}} f^{(2)*}$, where the star denotes the complex conjugate. This vector representation will be very important throughout this work. A similar remark holds for functions in K.

5.2.5 Real densities

Throughout this work, we consider *real* densities f. We adopt where necessary the treatment of Johnstone and Silverman[21]. The complex bases (5.14) and (5.15) are identified with equivalent real orthonormal bases in a standard fashion. For example $f = \sum_{\nu\in\mathcal{N}'} f_\nu \phi_\nu =$

$\sum_{\nu \in \mathcal{N}'} \tilde{f}_\nu \tilde{\phi}_\nu$, where

$$\tilde{\phi}_\nu = \tilde{\phi}_{(j,k)} = \begin{cases} \sqrt{2}\,\mathrm{Re}\,(\phi_{(j,k)}) & \text{if } j > k \\ \phi_{(j,j)} & \text{if } j = k \\ \sqrt{2}\,\mathrm{Im}\,(\phi_{(j,k)}) & \text{if } j < k\,. \end{cases}$$

Following the treatment given in [21], we suppress the tildes and use whichever basis is convenient. We shall discuss this again in Section 6.2.6 when we consider the actual computation of the least favourable function.

5.2.6 Linear estimators

An estimator $\hat{f}$ based on observations $Y_1, \ldots, Y_n$ is called *linear* if there exists a weight function $w(x, y)$ such that $\int w(x, y)\, d\mu(x) = 1$ for all y in the space of observations, and

$$\hat{f}(x) = n^{-1} \sum_{i=1}^{n} w(x, Y_i) \qquad (5.18)$$

for all x in B. Let $\mathcal{T}_{LI}(n)$ be the class of all linear estimators of f based on the indirect observations $Y_1, \ldots, Y_n$, subject to the additional condition $\iint w(x, y)^2\, d\mu(x)\, d\lambda(y) < \infty$, and for comparison let $\mathcal{T}_{LD}(n)$ be the class of all linear estimators of f based on the direct observations $X_1, \ldots, X_n$ (these cannot be observed in practice), subject to the additional condition $\iint w(x, x')^2\, d\mu(x)\, d\mu(x') < \infty$. Finally, let $\mathcal{F}$ be a class to which the densities on B are restricted. Johnstone and Silverman[21] define $\mathcal{F}$ to be the set

$$\{f \in H : f_{(0,0)} = 1,\ f^{\mathrm{T}} A f \leq 1 + C^2\} \qquad (5.19)$$

where the diagonal matrix $A = \mathrm{diag}(1, a_\nu^2)$, $\nu \in \mathcal{N}^0$. In the PET case these authors set

$$\begin{aligned} a_\nu &= a_{(j,k)} \\ &= (j+1)^a (k+1)^a, \end{aligned} \qquad (5.20)$$

for some $a > 1/2$, and point out that the class consists of functions whose $2a$th derivatives exist and satisfy a weighted square-integrability condition. This set is an ellipsoid in H and it is assumed that the a_ν and the C, the constant that governs the size of the ellipsoid, are chosen to ensure that all members of $\mathcal{F}$ are nonnegative. In the PET case with a_ν given by (5.20) this is achieved if

$$C \leq 2^{a-\frac{1}{2}}. \tag{5.21}$$

We now make two remarks. First, if $a = 1.0\,(1.5)$ the upper bound given in (5.21) is $\sqrt{2.0}\,(2.0)$. Secondly, equation (5.20) holds for $\nu = (0,0)$: in this case $a_{(0,0)} = 1$. Similarly, we write B to be the diagonal matrix diag$(1, b_\nu)$, $\nu \in \mathcal{N}^0$, where the b_νs were defined above. The case when $b_\nu = 1$, $\forall \nu \in \mathcal{N}^0$ corresponds to the direct observation case: that of having observations on brain space. Such a B is used in [21] to allow comparisons between the direct and indirect cases.

If we let ν and $\pi \in \mathcal{N}'$ and write

$$w_{\nu\pi} = \int\!\!\int w(x,y)\,\phi_\nu(x)\,\psi_\pi(y)\,d\mu(x)\,d\lambda(y),$$

then Johnstone and Silverman[21] explain that, because of the condition $\int\!\int w^2\,d\mu\,d\lambda < \infty$, standard functional analysis gives that, in the L^2 sense,

$$w(x,y) = \sum_{\nu \in \mathcal{N}'} \sum_{\pi \in \mathcal{N}'} w_{\nu\pi}\,\phi_\nu(x)\,\psi_\pi(y). \tag{5.22}$$

We shall make use of this form in Section 5.2.8.

5.2.7 Loss functions and equivalent sample size

Next, Johnstone and Silverman[21] define $M(\hat{f}; f)$ to be some measure of accuracy of an estimator $\hat{f}$ of f. In particular they take $M(\hat{f}; f)$ to be the mean integrated square error

$$M(\hat{f}; f) = \mathbf{E}\,[\int_B (\hat{f} - f)^2\,d\mu]. \tag{5.23}$$

By standard calculations $M(\hat{f}; f)$ can be put in 'variance + squared bias' form as

$$M(\hat{f}; f) = \int \left[\mathrm{Var}_f[\hat{f}(x)] + \{\mathbf{E}_f[\hat{f}(x)] - f(x)\}^2\right] d\mu(x) \tag{5.24}$$

where the suffix f indicates that the mean and variance are calculated for data drawn from f in the direct case and Pf in the indirect case. The *surrogate mean integrated square error* $M^*(\hat{f}; f)$ is obtained by replacing the variance term in (5.24) by the corresponding term calculated for the uniform density on brain space

$$M^*(\hat{f}; f) = \int \left[\mathrm{Var}_1[\hat{f}(x)] + \{\mathrm{E}_f[\hat{f}(x)] - f(x)\}^2\right] d\mu(x) \tag{5.25}$$

The surrogate mean integrated square error is used because of its simplicity compared to (5.24). Proposition 2.1 of [21] gives an important relation between the surrogate and the true mean integrated square error for linear estimators: if f is bounded above and below away from zero, *i.e.* if

$$0 < \inf_B f \leq \sup_B f < \infty, \tag{5.26}$$

then, for all $\hat{f}$ in $\mathcal{T}_{LD}(n)$ or in $\mathcal{T}_{LI}(n)$

$$\inf_B f(x) \leq \frac{M(\hat{f}; f)}{M^*(\hat{f}; f)} \leq \sup_B f(x).$$

This proposition means that the ratio of surrogate to true mean integrated square error will be bounded above and below away from zero uniformly on $\mathcal{F}$, so that order of magnitude statements made for one mean integrated square error will also be true for the other. We now outline the proof of this proposition given in the Appendix of [21].

First, a standard argument establishes that

$$\inf_B f \leq \frac{\mathrm{Var}_f[\hat{f}(x)]}{\mathrm{Var}_1[\hat{f}(x)]} \leq \sup_B f$$

in the direct case, and

$$\inf_D g \leq \frac{\mathrm{Var}_f[\hat{f}(x)]}{\mathrm{Var}_1[\hat{f}(x)]} \leq \sup_D g$$

in the indirect case, where $g = Pf$. We use a similar argument later in the proof of Proposition 3. It is easy to show that $\sup_B f \geq 1$, $\inf_B f \leq 1$, $\sup_D g \geq 1$ and $\inf_D g \leq 1$. This immediately gives

$$\inf_B f \leq \frac{M(\hat{f}; f)}{M^*(\hat{f}; f)} \leq \sup_B f$$

in the direct case, and

$$\inf_D g \leq \frac{M(\hat{f}; f)}{M^*(\hat{f}; f)} \leq \sup_D g$$

in the indirect case. Finally, since $\sup_D g \leq \sup_B f$ and $\inf_B f \leq \inf_D g$, the proposition follows.

Suppose that one has a sample from a density f and an estimator $\hat{f}$ based on that sample. Then an assessment of the accuracy of $\hat{f}$ that does not depend on a particular known f is given by the maximum risk

$$R(\hat{f}) = \sup_{f \in \mathcal{F}} M(\hat{f}; f). \tag{5.27}$$

Because interest lies in the experiment itself rather than any particular estimator, the authors consider the minimum of $R(\hat{f})$ over suitable classes of estimators $\hat{f}$. They set

$$r_{LI}(n) = \inf_{\hat{f} \in \mathcal{T}_{LI}(n)} R(\hat{f})$$

and, for comparison,

$$r_{LD}(n) = \inf_{\hat{f} \in \mathcal{T}_{LD}(n)} R(\hat{f}).$$

These *minimax risks* quantify the information about the unknown density inherent in indirect and direct data sets of size n, in a manner that is independent of the method of estimation. Comparing their relative values gives an indication of how much information is lost because data can only be observed indirectly in practice. Johnstone and Silverman[21] set $p = 2a$ and present the following theorem in which estimators are not restricted to be linear:

Theorem 2 (Johnstone and Silverman[21], Theorem 3.1) *For fixed* $p \geq 1$ *and* $0 < C < 2^{(p-1)/2}$, *we obtain*

$$r_D(n) \approx (\log n / n)^{p/(p+1)}$$

and

$$r_I(n) \approx (1/n)^{p/(p+2)},$$

where $a_n \approx b_n$ *means that the sequences* $\{a_n\}$ *and* $\{b_n\}$ *satisfy* $\inf_n(a_n / b_n) > 0$ *and*

$\sup_n(a_n / b_n) < \infty$.

In effect this theorem tells us that the indirect nature reduces somewhat the rate at which the minimax risk converges to zero. For the case in which estimators are restricted to be linear the authors give a more precise version of their Theorem 3.1 in which the exact large sample behaviour of the minimax risks is stated. The proof of this more precise theorem starts by obtaining exact expressions for these minimax risks in terms of the constants a_v and b_v (see, for example, Lemma 2). Then, for the PET case, it justifies integral approximations that make it possible to give the stated exact large sample behaviour. The constants in these expressions are complicated but tractable. It is then possible to derive the numerical *equivalent direct sample size* $m(n)$ to a given indirect sample of size n, *i.e.* the size of the sample from f itself that would give the same amount of information as the given indirect sample, under the given smoothness assumptions, so that, for linear estimators for example, $r_{LD}(m) = r_{LI}(n)$. In fact for general estimators

$$m(n) \approx n^{(p+1)/(p+2)} \log n.$$

In the PET case with linear estimators, under the assumption that f has square integrable first derivatives ($p = 1$) and using the surrogate mean integrated square error, the equivalent sample size to an indirect sample of size 10^7 is 193,000, whereas the equivalent sample size to an indirect sample of size 10^8 is 1,030,000. Other similar results are given in their Table 2, from which it can be concluded that the more smoothness that is assumed, the less information is lost. Johnstone and Silverman[21] also demonstrate, using Fano's lemma of information theory as developed by Ibragimov and Hasminskii, that the restriction to *linear* minimax estimators does not affect the minimax rates, and hence derive their Theorem 3.1. They also show that, under mild assumptions, the incompleteness of sampling, as demonstrated in equation (5.2), has no effect on the minimax rates found in Theorem 3.1.

5.2.8 The linear minimax estimator

Recall that we observe $Y_1, \dots, Y_n$, where Y_i corresponds to the ith observed pair (*i.e.* straight line). From this fixed number n of observations, we define Z_π for $\pi \in \mathcal{N}'$ as

$$Z_\pi = \frac{1}{n} \sum_{i=1}^{n} \psi_\pi(Y_i),$$

and we observe that $Z_{(0,0)} = 1$. Hence, using equation (5.22), we obtain

$$\begin{aligned}\hat{f}(x) &= \sum_{\nu\in\mathcal{N}'}\sum_{\pi\in\mathcal{N}'} w_{\nu\pi}\left(\frac{1}{n}\sum_{i=1}^{n}\psi_\pi(Y_i)\right)\phi_\nu(x)\\ &= \sum_{\nu\in\mathcal{N}'}\phi_\nu(x)\left(\sum_{\pi\in\mathcal{N}'} w_{\nu\pi}Z_\pi\right).\end{aligned}$$

Since the coefficient of $\phi_{(0,0)}(x)$ is 1, we must have

$$\sum_{\pi\in\mathcal{N}'} w_{(0,0)\pi}Z_\pi = 1$$

and this can be achieved by setting $w_{(0,0)\pi} = \delta_{(0,0)\pi}$, for all $\pi \in \mathcal{N}'$. Therefore, we now focus on linear estimators of f given by

$$\hat{f} = WZ$$

where $\hat{f}_{(0,0)} = 1$, W is the infinite matrix $(w_{\nu\pi})$, whose first column is defined as above, and Z is the vector Z_ν, ν and π belonging to the set $\mathcal{N}'$.

Johnstone and Silverman[21] present two lemmas: the first lemma gives a matrix form for (5.25), the surrogate mean integrated square error of the linear estimator $\hat{f}$; the second lemma provides an expression for the surrogate linear minimax risk and gives the general form of the minimax estimator. We reproduce these lemmas. We require one further definition: the vector

$$e_\nu = (\delta_{\nu\pi} : \pi \in \mathcal{N}').$$

Lemma 1 (Johnstone and Silverman[21], Lemma 4.1) *With the above definitions*

$$M^*(\hat{f}; f) = n^{-1}\mathrm{tr}\, W(I - e_0e_0^{\mathrm{T}})W + f^{\mathrm{T}}(I - WB)^{\mathrm{T}}(I - WB)f.$$

Lemma 2 (Johnstone and Silverman[21], Lemma 4.2) *With the above definitions*

$$\inf_{\hat{f}\in\mathcal{T}_{LI}(n)}\ \sup_{f\in\mathcal{F}} M^*(\hat{f}; f) = n^{-1}\sum_{\nu\in\mathcal{N}'} b_\nu^{-2}(1 - a_\nu\gamma^{1/2})_+,$$

where γ is chosen to ensure that

$$n^{-1} \sum_{v \in \mathcal{N}^0} b_v^{-2} a_v^2 (\gamma^{-1/2} a_v^{-1} - 1)_+ = C^2.$$

The minimax estimator is given by setting

$$w_{(0,0)\pi} = \delta_{(0,0)\pi}$$

as above, where $\pi \in \mathcal{N}'$, and

$$w_{v\pi} = \delta_{v\pi}\, b_v^{-1}(1 - \gamma^{1/2} a_v)_+$$

where $v \in \mathcal{N}^0$ and $\pi \in \mathcal{N}'$.

We make a remark about these estimates. First, the matrix W is diagonal. This means that

$$\begin{aligned} \hat{f}(x) &= 1 + \sum_{v \in \mathcal{N}^0} w_{vv} Z_v \phi_v(x) \\ &= 1 + \sum_{v \in \mathcal{N}^0} b_v^{-1}(1 - \gamma^{1/2} a_v)_+ Z_v \phi_v(x) \\ &= 1 + \sum_{v \in \mathcal{N}^0} \hat{f}_v \phi_v(x), \end{aligned}$$

where $\hat{f}_v = b_v^{-1}(1 - \gamma^{1/2} a_v)_+ Z_v$, $v \in \mathcal{N}^0$. Now since

$$\mathbf{E}[Z_v] = g_v = b_v f_v, \tag{5.28}$$

a possible estimate for f_v would be $b_v^{-1} Z_v$. However, this estimate includes the additional factor $(1 - \gamma^{1/2} a_v)_+$. It is by means of this factor that smoothing is introduced. Large values of a_v, which are brought about by large values of a, result in $(1 - \gamma^{1/2} a_v)_+$ being equal to zero and the removal of high frequency components in the density estimate $\hat{f}(x)$. We remark that $\hat{f}(x)$ is not necessarily positive. The reader should compare this minimax estimator of f with the one employed in Jones and Silverman[23] and discussed in Section 5.2.3. In addition, the reader is invited to compare this discussion about smoothing with that given for the discrete case in 5.2.12.

In essence our problem is that of estimating an array f_v given observations

$$Z_v = b_v f_v + \text{error},$$

where the error has zero expected value, subject to the restriction that the f_v fall in some ellipsoid $\mathcal{F}$. In this case there is a single observation for each parameter to be estimated. Further work (from the point of view of not only estimating the density f, but also estimating a linear functional $T(f)$, say) may consider the case when $\mathcal{F}$ is a hyperrectangle. An important related paper here is Donoho, Liu and MacGibbon[9] in which the following problem is considered. Suppose we are given

$$y_i = \theta_i + \varepsilon_i, \quad i = 0, 1, 2, \ldots,$$

where the ε_i are iid $\mathcal{N}(0, \sigma)$ and θ_i are unknown, but it is known that $\theta = (\theta_i)$ lies in Θ, a compact subset of l_2, *i.e.* $\sum_i |\theta_i|^2 < \infty$. Donoho *et al.*[9] consider the case when Θ is a hyperrectangle

$$\Theta(\tau) = \{\theta : |\theta_i| \leq \tau_i\},$$

where $\tau_i \to 0$ as $i \to \infty$, and also the case when Θ is an ellipsoid

$$\{\theta : \sum_i a_i \theta_i^2 \leq 1\},$$

or more generally an l_p-body

$$\Theta_p(a) = \{\theta : \sum_i a_i |\theta_i|^p \leq 1\}.$$

They wish to estimate θ with small squared error loss $||\hat{\theta} - \theta||^2 = \sum(\hat{\theta}_i - \theta_i)^2$, and they use the minimax principle to evaluate estimates.

5.2.9 Discretization

So far we have considered the idealized case where detector space D is assumed to be continuous. Under this assumption for PET it was possible to know the exact line along which an emission took place. However, in reality, the ring of detectors is divided into a finite number N of separate detectors, as is depicted in Johnstone and Silverman[22]'s FIG. 1. These N

detectors give rise to $r(N) = N(N-1)/2$ tubes, each tube being defined by a pair of distinct detectors. A detected emission is recorded by one of these tubes, and so the exact position of the line along which the emission took place is no longer known: only the tube is known.

To model this situation we shall refer to the discretized detector space as D_N, and assume that the detector space is divided into $r(N)$ bins: $D_1, \ldots, D_{r(N)}$. Thus, if f is the density on brain space B, and $g = Pf$ is the density on detector space D, then $Q_N g$ is the density (probabilities) on D_N where $Q_N g$ is defined as

$$(Q_N g)_j = \frac{1}{\lambda(D_j)} \int_{D_j} g \, d\lambda, \; j = 1, \ldots, r(N), \tag{5.29}$$

where λ is the dominating measure of D. In effect a continuous density is turned into a histogram with $r(N)$ bins by the discretization map $Q_N : G \to G_N$, where G_N is the finite dimensional space of vectors of length $r(N)$. The vector $Q_N g$ gives the averages of g over each of the bins D_j. We observe a Poisson number $\mathcal{N}$ of observations, $Y_1, Y_2, \ldots, Y_{\mathcal{N}}$. The mean of the random variable $\mathcal{N}$ is n. Let n_j be the number of these observations falling into bin D_j. The n_j are independent Poisson random variables with means $n\lambda(D_j)(Q_N g)_j$, $j = 1, \ldots, r(N)$.

We remark that in this chapter n has two slightly different uses. In the continuous case it is the number of observations, whereas in the discrete case it is the expected number of observations. Johnstone and Silverman[22] refer to n as used in the discrete case as 'an integer giving an index of the amount of data collected'.

5.2.10 Key assumptions

With the same orthonormal bases $\{\phi_\nu\}$ and $\{\psi_\nu\}$ for B and D that we used above, we again have the singular value decomposition as stated in equation (5.16), namely that $P\phi_\nu = b_\nu \psi_\nu$, where the singular values b_ν are defined in equation (5.17).

In addition, this time we make the *matching SVD assumption*: given any ν_1 and ν_2, the vectors $Q_N \psi_{\nu_1}$ and $Q_N \psi_{\nu_2}$ are either parallel or orthogonal on the space G_N.

Although this *is* a restrictive assumption, it holds exactly in the cases of *density estimation from binned data* ([22]: Section 1, Example 1, and Section 4.1) and of *deconvolution* ([22]: Section 1, Example 2, and Section 4.2), or approximately in the cases of *The Wicksell 'unfolding' problem of stereology* ([22]: Section 1, Example 3, and Section 5, especially 5.2 and 5.4 where the approximate nature is discussed) and of PET ([22]: Section 1, Example 4, and Section 6, especially 6.2 and 6.3). (As far as surrogate risk behaviour is concerned condition (5.26) holds in the two former cases. However, the lower bound of (5.26) is not available for the Wicksell operator, although the ratio of surrogate to true mean integrated squared error can be bounded above and below away from zero uniformly over $\mathcal{F}$ at least for diagonal linear estimators. In the PET case condition (5.26) holds provided that C satisfies condition (5.21), the condition for $\mathcal{F}$ to contain only nonnegative densities.) We shall discuss this further for the PET case in Section 5.2.12.

Because of the matching SVD theorem, we can construct an orthonormal basis $\{\chi_p\}$ of G_N such that for each ν there exists $[\nu]$ for which, for certain constants γ_ν,

$$Q_N \psi_\nu = \gamma_\nu \chi_{[\nu]}. \tag{5.30}$$

Under the matching SVD assumption, for any fixed N we define an equivalence relation on the set of subscripts $\{\nu\}$ by

$$\nu_1 \sim \nu_2 \text{ if and only if } [\nu_1] = [\nu_2]$$

where $[\nu]$ is defined in equation (5.30) above, so that the equivalence classes correspond to sets of ψ_ν that are mapped by Q_N to multiples of the same basis vector in G_N. The $[\nu]$ can be considered as being equivalence classes under $\sim$, and in this way an infinite dimensional vector space is mapped to a finite dimensional vector space.

5.2.11 More notation and definitions

In Section 5.2.9 we assumed that we observed a Poisson number of observations $\mathcal{N}$ with mean n. The number of observations falling into bin D_j was defined to be n_j. The n_j s were seen to be independent Poisson variables with means $n\lambda(D_j)(Q_N g)_j$, for $j = 1, \ldots, r(N)$. We assume now that the observed data consists of an $r(N)$-vector Z of normalised bin counts of G_N such that

$$Z_j = \frac{n_j}{n\lambda(D_j)}$$

for $j = 1, \ldots, r(N)$. The Z_j are are independent Poisson counts with intensities $(Q_N g)_j$, for some $g = \sum_\nu g_\nu \psi_\nu$ in G.

We now express the $r(N)$-vector of observations Z in terms of the $r(N)$ 'new' basis vectors $\chi_{[\nu]}$: $Z = \sum \tilde{Z}_{[\nu]} \chi_{[\nu]}$. We let Γ be the matrix representation of Q_N relative to the basis

$\{\psi_v\}$ and $\{\chi_{[v]}\}$ respectively: *i.e.* $\Gamma_{[v]\rho} = \gamma_\rho$ if $\rho \in [v]$ and zero otherwise.

Consider now the equivalence class $[v]$. There are $r(N)$ such classes: let us consider the kth class. We order the vs in this class in some way (according to $\prec$ or $\preceq$ say) and we write v_i^k to be the ith value in the kth class, $k = 1, \ldots, r(N)$. Thus we have $v_1^k \sim v_2^k \sim v_3^k \sim \cdots$, $[v_1^k] = [v_2^k] = [v_3^k] = \cdots$, and $v_1^k \prec v_2^k \preceq v_3^k \preceq \cdots$. We take the first value v_1^k to be the representative of the kth equivalence class. Johnstone and Silverman[22] construct a set of such representatives $\mathcal{L}_N = \{v_1^k, k = 1, \ldots, r(N)\}$ of these equivalence classes by selecting from each $[v]$ a v that maximises $b_v\gamma_v$ over this equivalence class. They state that this v essentially corresponds to the direction in which least energy is lost under the mapping $Q_N P$. Table 5.1 may help to clarify the situation.

Equivalence class	B, detector space, $\{g\}$	$\mathcal{L}_N$	G_N
1	$\psi_{v_1^1}, \psi_{v_2^1}, \psi_{v_3^1}, \cdots$	v_1^1	$\chi_{[v_1^1]}$
2	$\psi_{v_1^2}, \psi_{v_2^2}, \psi_{v_3^2}, \cdots$	v_1^2	$\chi_{[v_1^2]}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	$\psi_{v_1^k}, \psi_{v_2^k}, \psi_{v_3^k}, \cdots$	v_1^k	$\chi_{[v_1^k]}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$r(N)$	$\psi_{v_1^{r(N)}}, \psi_{v_2^{r(N)}}, \psi_{v_3^{r(N)}}, \cdots$	$v_1^{r(N)}$	$\chi_{[v_1^{r(N)}]}$

Table 5.1: *Table of equivalence classes and corresponding basis vectors for* G_N

Next Johnstone and Silverman[22] assume that $0 \in \mathcal{L}_N$. Because of the restriction $f_{(0,0)} = 1$, this zero frequency plays a special role and so the set $\mathcal{L}_N^0 = \mathcal{L}_N \backslash \{0\}$ is introduced. They now define

$$\mathcal{F}^L = \operatorname{span}\{\phi_v : v \in \mathcal{L}_N^0\}$$

(L stands for low frequencies and this set depends on N) and

$$\mathcal{F}^H = \{f = \sum f_v \phi_v : f_v = 0 \text{ for } v \in \mathcal{L}_N\}$$

(H stands for high frequencies and again this set depends on N), so that

$$\mathcal{F} = \operatorname{span}\{\phi_0\} \oplus \mathcal{F}^L \oplus \mathcal{F}^H.$$

The ellipsoid $\mathcal{F}_0^L$ is defined as $\mathcal{F}_0^L = (\phi_0 + \mathcal{F}^L) \cap \mathcal{F}_0$, where

$$\mathcal{F}_0 = \{f = \sum_\nu f_\nu \phi_\nu \;:\; f_0 = 1,\; \sum_{\nu \in \mathcal{N}^0} a_\nu^2 f_\nu^2 \le C^2\},$$

as in equation (5.19).

As with the case where there is no discretization, the aim of [22] is to estimate f. The approach taken in Johnstone and Silverman[22] is to consider only 'low frequency' functions $\mathcal{F}^L$. Let $\mathcal{T}_N$ be the class of linear functions from G_N to $\mathcal{F}^L$, and write T_{jk} for the matrix representation of T in $\mathcal{T}_N$ relative to the bases $\{\chi_{[\nu]}\}$ and $\{\phi_\nu : \nu \in \mathcal{L}_N^0\}$. The estimator corresponding to $T \in \mathcal{T}_N$ may be written as

$$\hat{T}(x) = \phi_0 + \sum_{\nu \in \mathcal{L}_N^0} \phi_\nu(x) \sum_{[\pi]} T_{\nu[\pi]} \tilde{Z}_{[\pi]}, \tag{5.31}$$

which should be compared to equation (5.28) above.

5.2.12 Some results for the discrete case

In this case, the form of the mean integrated square error corresponding to equation (5.24) is

$$M(T; f) = \int \left[\operatorname{Var}_{Pf}[\hat{T}(x)] + \{\mathrm{E}_{Pf}[\hat{T}(x)] - f(x)\}^2\right] d\mu(x).$$

If we define the *surrogate* mean integrated square error $M^*(T; f)$ as

$$M^*(T; f) = \int \left[\operatorname{Var}_{P1}[\hat{T}(x)] + \{\mathrm{E}_{Pf}[\hat{T}(x)] - f(x)\}^2\right] d\mu(x),$$

where 1 is the uniform distribution on B (and hence $P1 = 1$ is the uniform distribution on D) we can establish that, if f (and hence g) is bounded above and away from zero, then for all linear estimators T,

$$\inf_D g \le \frac{M(T; f)}{M^*(T; f)} \le \sup_D g.$$

This result follows easily from equation (9) of [22] which tells us that

$$\mathrm{Var}_g[\hat{T}(x)] = \frac{1}{n}\int t^2\,(x, j(y))\, g(y)\, d\lambda\,(y), \tag{5.32}$$

where

$$t\,(x, j) = \sum_{\nu\in\mathcal{L}_N^0} \phi_\nu(x) \sum_{[\pi]} T_{\nu[\pi]}\,\chi_{[\pi]j}$$

and $j\,(y)$ indicates the bin into which y falls. Johnstone and Silverman[22] explain that the expression (5.32) is obtained by showing that

$$\hat{T}(x) = \phi_0(x) + \frac{1}{n}\sum_{i=1}^{\mathcal{N}} t\,(x, J_i),$$

where we have seen that $\mathcal{N} = \sum_{j=1}^{r(N)} n_j$ is a Poisson random variable with mean n, and establishing that

$$\mathrm{Var}_g[\sum_{i=1}^{\mathcal{N}} t\,(x, J_i)] = n\int t^2(x, j(y))\, g(y)\, d\lambda\,(y).$$

Again, as we saw in Section 5.2.7, order of magnitude statements made for one mean integrated square error will also be true for the other.

We can write down $M^*(T; f)$ in matrix notation as

$$M^*(T; f) = n^{-1}\mathrm{tr}\, T^{\mathrm{T}}T + f^{\mathrm{T}}(I - T\Gamma B)^{\mathrm{T}}(I - T\Gamma B)\, f. \tag{5.33}$$

After defining

$$\mathcal{M}(\mathcal{T}; \mathcal{F}_0) = \inf_{T\in\mathcal{T}}\ \sup_{f\in\mathcal{F}_0} M^*(T; f),$$

the authors state a lemma, corresponding to the above Lemma 2:

Lemma 3 (Johnstone and Silverman[22], Lemma 1) *With the above definitions, we have*

$$\mathcal{M}(\mathcal{T}_N; \mathcal{F}_0^L) = n^{-1}\sum_{\nu\in\mathcal{L}_N^0} b_\nu^{-2}\gamma_\nu^{-2}\,(1 - a_\nu\alpha^{1/2})_+,$$

where α is chosen to ensure that

$$n^{-1} \sum_{v \in \mathcal{L}_N^0} b_v^{-2} \gamma_v^{-2} a_v^2 (\alpha^{-1/2} a_v^{-1} - 1)_+ = C^2.$$

The minimax estimator is given by setting

$$T^*_{v\pi} = \delta_{v\pi} b_v^{-1} \gamma_v^{-1} (1 - \alpha^{1/2} a_v)_+$$

for all v in $\mathcal{L}_N^0$ and π in $\mathcal{L}_N$.

We remark that T^* is diagonal. The authors now state and prove the following theorem:

Theorem 3 (Johnstone and Silverman[22], Theorem 1) *Given any v in $\mathcal{L}_N$, define*

$$S_N(v) = \sum_{\rho \in [v] \backslash v} a_\rho^{-2}.$$

Then

$$\mathcal{M}(\mathcal{T}_N; \mathcal{F}_0^L) \leq \mathcal{M}(\mathcal{T}_N; \mathcal{F}_0) \leq \sup_{f \in \mathcal{F}_0} M^*(T^*; f) \leq \{\mathcal{M}(\mathcal{T}_N; \mathcal{F}_0^L)^{1/2} + \varepsilon(N)^{1/2}\}^2,$$

where

$$\varepsilon(N) = C^2 \{ \max_{v \in \mathcal{L}_N^0} S_N(v) + \max_{\rho \notin \mathcal{L}_N} a_\rho^{-2} \}.$$

In the theorem $\varepsilon(N)$, which also depends on n, can be thought of as a high frequency error caused by throwing away high frequency components.

We finish this section with a brief discussion about smoothing similar to the one made at the end of Section 5.2.8. First,

$$\begin{aligned}
\hat{T}(x) &= 1 + \sum_{v \in \mathcal{L}_N^0} \left(\sum_{[\pi]} T_{v[\pi]} \tilde{Z}_{[\pi]} \right) \phi_v(x) \\
&= 1 + \sum_{v \in \mathcal{L}_N^0} b_v^{-1} \gamma_v^{-1} (1 - \alpha^{1/2} a_v)_+ \tilde{Z}_{[v]} \phi_v(x) \\
&= 1 + \sum_{v \in \mathcal{L}_N^0} \hat{f}_v \phi_v(x),
\end{aligned}$$

where $\hat{f}_\nu = b_\nu^{-1}\gamma_\nu^{-1}(1-\alpha^{1/2}a_\nu)_+\tilde{Z}_{[\nu]}$, $\nu \in \mathcal{L}_N^0$. Now, it is not difficult to show that

$$\mathrm{E}\,[\tilde{Z}_{[\nu]}] = \sum_{[\nu']=\nu} b_{\nu'}\gamma_{\nu'}f_{\nu'},\ \nu \in \mathcal{L}_N^0,$$

see equation (5) of [22]. (The analogous result for the continuous case is given in equation (5.28).) If we ignore all the terms in the sum except for the one with $\nu' \in \mathcal{L}_N^0$, we have

$$\mathrm{E}\,[\tilde{Z}_{[\nu]}] \approx b_\nu\gamma_\nu f_\nu,\ \nu \in \mathcal{L}_N^0.$$

Hence, a possible estimator for f_ν would be $b_\nu^{-1}\gamma_\nu^{-1}\tilde{Z}_{[\nu]}$. However, as in Section 5.2.8, a smoothing factor $(1-\alpha^{1/2}a_\nu)_+$ is introduced. This factor has the same effect here as it did in the continuous case in Section 5.2.8.

This time our problem is in essence that of estimating an array f_ν given observations

$$\tilde{Z}_{[\nu]} = \sum_{[\nu']=\nu} b_{\nu'}\gamma_{\nu'}f_{\nu'} + \text{error}$$

where the error has zero expected value, subject to the restriction that the $f_{\nu'}$ fall in some ellipsoid $\mathcal{F}$. In this case there is a whole set of parameters to be estimated from a single observation. Setting f_ν to zero except for $\nu \in \mathcal{L}_N^0$ puts us back in the case when there is one parameter to be estimated from each observation. Further work would consider in detail ways in which the full set of possible estimators can be called into play, so that the information available can be used to give estimates of all the coefficients f_ν, not just the low-frequency ones. For now see Section 5.3.7, especially equations (5.49) and (5.50), where such an estimate is presented. This estimate minimizes a type of penalized least squares form given in equation (5.51).

5.2.13 Applying these results to the PET case

First, it is necessary to state the equivalence classes needed for the SVD property (5.16). The set $\mathcal{N}_{N-1}$ is defined to be $\{(j,k) : j,k \geq 0, j+k \leq N-2\}$. This set contains $N(N-1)/2$ members, including (0, 0), and is a possible set $\mathcal{L}_N$ defined in Section 5.2.11. Next, given any $\nu_0 = (j_0, k_0)$ in $\mathcal{N}_{N-1}$, define $j_1 = N-1-k_0$ and $k_1 = N-1-j_0$. Then the equivalence class of ν_0 consists of all indices of the form $(j_0 + r_j N, k_0 + r_k N)$ or $(j_1 + r_j N, k_1 + r_k N)$ for nonnegative integers r_j

and r_k. In addition, all pairs (j, k) with

$$j + k \equiv N - 1 \pmod{N}$$

are added to the equivalence class of (0, 0).

Finally (using $[\cdot]$ to denote integer part) we have

$$\gamma_{(j,k)} = \begin{cases} 0 & j + k \equiv N - 1 \pmod{N} \\ -1^{[(j+k)/N]} \operatorname{sinc}\{(2j+1)\pi/2N\} \operatorname{sinc}\{(2k+1)\pi/2N\} & \text{otherwise,} \end{cases}$$

where $\operatorname{sinc}\theta = \sin\theta/\theta$.

Johnstone and Silverman[22] investigate in a detailed fashion the minimax risk in the PET case. Their conclusions depend upon the relationship between N and n, and there are three different cases to consider: the subcritical case, the critical case and the supercritical case. The interested reader is referred to Section (6.4) of [22] for further details.

5.3 The estimation of linear functionals

We now come to the important new work of this chapter, the estimation of linear functionals. Our treatment here is specific to the PET case. The results have, however, been generalized by using sophisticated techniques of functional analysis by Silverman[38] and others. We discuss such work in Section 5.5.

Let the density f, defined on brain space B, lie in the Hilbert space H. We now turn our attention to the estimation of linear functionals[2] of f. Let T be a linear functional. If

$$f = 1 + \sum_{\nu \in \mathcal{N}^0} f_\nu \phi_\nu,$$

then the quantity of interest is

$$T(f) = T(1) + \sum_{\nu \in \mathcal{N}^0} f_\nu T(\phi_\nu).$$

In vector notation this becomes $T(1) + \xi^{\mathrm{T}} f$, where $\xi = (T(\phi_\nu))$ and $f = (f_\nu)$, for $\nu \in \mathcal{N}^0$.

[2] A *linear functional* is a linear operator T with domain in a vector space X and range in the scalar field K of X; thus, $T : \mathcal{D}(T) \to K$, where $K = \mathbf{R}$ if X is real and $K = \mathbf{C}$ if X is complex.

5.3.1 Restriction on the linear functionals to be estimated

We shall see later, for example in Section 5.3.5 for the continuous case and Sections 5.3.7 and 5.3.10 for the discrete case, that we need to impose some form of restriction on the class of linear functionals considered. In fact we insist that

$$\sum_{v} \frac{|\xi_v|^2}{a_v^2} < \infty \tag{5.34}$$

where a_v was defined in equation (5.20) as

$$\begin{aligned} a_v &= a_{(j,k)} \\ &= (j+1)^a (k+1)^a \end{aligned}$$

for some $a > 1/2$. We now give two examples of such functionals.

Example 1. In this example we present a proposition that gives a sufficient, but not a necessary condition for a linear functional T to satisfy inequality (5.34). First, we state the definition of a bounded linear functional. A *bounded linear functional* T is a bounded linear operator with range in the scalar field of the normed space X in which the domain $\mathcal{D}(T)$ lies. That is, there exists a real number c such that for all $f \in \mathcal{D}(T)$, $|T(f)| \leq c\,||f||$. Furthermore, the *norm of* T, $||T||$, is defined as

$$||T|| = \sup_{f \in \mathcal{D}(T)\backslash\{0\}} \frac{|T(f)|}{||f||} = \sup_{f \in \mathcal{D}(T),\, ||f||=1} |T(f)|.$$

We shall henceforth refer to $||\cdot||$ as the supremum norm. Now, if T is a bounded linear functional it is not hard to show that

$$||T|| = \sqrt{\sum_{v} |\xi_v|^2} < \infty. \tag{5.35}$$

Proposition 1 *If T is a bounded linear functional, then T satisfies the inequality (5.34).*

Proof. First, note that $a_v = a_{(j,k)} = (j+1)^a (k+1)^a \geq 1$, $\forall v$, with strict inequality for $v \neq 0$. Thus, $|\xi_v|^2 / a_v^2 \leq |\xi_v|^2$, $\forall v$. Immediately, we have

$$\sum_{v} \frac{|\xi_v|^2}{a_v^2} < \sum_{v} |\xi_v|^2 < \infty,$$

from (5.35) since T is a bounded linear functional. □

Our second example shows that there are functionals satisfying (5.34) that are not bounded.

Example 2. In this example we consider the case when $a > 1$. Let x be a fixed point in the unit circle. Consider the functional $T_x(f) = f(x)$, evaluation of the density f at the point x. We shall establish in Section 6.2.2 that this is not a bounded linear functional with respect to the supremum norm. However, we have the following proposition.

Proposition 2 *For $a > 1$, the functional $T_x(f) = f(x)$ satisfies the inequality (5.34). In other words $\sum_\nu \frac{|\phi_\nu(x)|^2}{a_\nu^2} < \infty$.*

Proof. We have that

$$\begin{aligned} |\phi_\nu(x)| &= |\phi_{(j,k)}(r,\theta)| \\ &= |(j+k+1)^{1/2} Z_{j+k}^{|j-k|}(r)\, e^{i(j-k)\theta}| \\ &= (j+k+1)^{1/2}\, |Z_{j+k}^{|j-k|}(r)| \\ &\leq (j+k+1)^{1/2} \end{aligned}$$

since $|Z_{j+k}^{|j-k|}(r)| \leq 1$. Hence

$$\frac{|\phi_\nu(x)|^2}{a_\nu^2} \leq \frac{j+k+1}{(j+1)^{2a}(k+1)^{2a}}.$$

Now

$$\sum_{j,k} \frac{j+k+1}{(j+1)^{2a}(k+1)^{2a}}$$

can be shown to converge if $a > 1$, and so

$$\sum_\nu \frac{|\phi_\nu(x)|^2}{a_\nu^2} < \infty,$$

as required. □

We end this section by noting that

$$\sqrt{\sum_\nu \frac{|\xi_\nu|^2}{a_\nu^2}}$$

can itself be thought of as a norm for the functional T, $||T||_2$ say.

5.3.2 The continuous case

First of all we are interested in the case when there is no discretization, *i.e.* the case when the ring of detectors is considered to be continuous and the line along which every emission occurs can be observed exactly. In Section 5.3.7 we consider the discrete case, and many of the techniques that we use in dealing with the continuous case are also employed there.

We now attempt to estimate the quantity $T(1)+\xi^{\mathrm{T}}f$ by the linear estimator $\mu+\tau^{\mathrm{T}}Z$ where the infinite dimensional vector (Z_v), $v \in \mathcal{N}^0$ is as defined in Section 5.2.8 above. In fact, we shall see in Section 5.3.4 below that the minimax estimator of μ is $T(1)$ and therefore the problem simplifies to the minimax estimation of $\xi^{\mathrm{T}}f$ by $\tau^{\mathrm{T}}Z$. For now we shall refer to $\mu+\tau^{\mathrm{T}}Z$ as $\widehat{T(f)}$.

5.3.3 Loss function

Here we follow the methodology and we use the notation of Section 5.2.7. Define the loss when $T(f)$ is estimated by $\widehat{T(f)}$, namely

$$M(\widehat{T(f)}; T(f))$$

to be the mean square error, specifically

$$\mathbf{E}_{Pf}[(\widehat{T(f)} - T(f))^2],$$

when the underlying density on brain space B is f and the data upon which the calculation of the variance is based are drawn from Pf. (On this occasion we do not integrate as $\widehat{T(f)}$ and $T(f)$ are scalars.) This quantity can be shown, by a simple argument, to be equal to

$$\mathrm{Var}_{Pf}[\widehat{T(f)}] + (\mathbf{E}_{Pf}[\widehat{T(f)}] - T(f))^2,$$

again 'variance + squared bias' . As a simplification we replace $\mathrm{Var}_{Pf}[\widehat{T(f)}]$ by $\mathrm{Var}_1[\widehat{T(f)}]$, where $1 = P1$ is the uniform density on D to get the *surrogate mean square error* $M^*(\widehat{T(f)}; T(f))$. We now state a proposition

Proposition 3 *Suppose that f (and hence $g = Pf$) is bounded above and below away from zero. Then*

$$\inf_B f \le \inf_D g \le \frac{M(\widehat{T(f)}; T(f))}{M^*(\widehat{T(f)}; T(f))} \le \sup_D g \le \sup_B f.$$

Proof. We need only establish the inner pair of inequalities as the outer pair follows immediately from the fact that $\sup_D g \le \sup_B f$ and $\inf_B f \le \inf_D g$, as we saw in Section 5.2.7. First recall the following two relationships:

$$\begin{aligned} M(\widehat{T(f)}; T(f)) &= \mathrm{Var}_{Pf}[\widehat{T(f)}] + (\mathrm{E}_{Pf}[\widehat{T(f)}] - T(f))^2 \\ M^*(\widehat{T(f)}; T(f)) &= \mathrm{Var}_1[\widehat{T(f)}] + (\mathrm{E}_{Pf}[\widehat{T(f)}] - T(f))^2. \end{aligned} \tag{5.36}$$

In order to establish that $\inf_D g \le M / M^* \le \sup_D g$, it is sufficient to show that

$$\inf_D g \le \frac{\mathrm{Var}_{Pf}[\widehat{T(f)}]}{\mathrm{Var}_1[\widehat{T(f)}]} \le \sup_D g,$$

since the required inequalities follow from the fact that $\sup_D g \ge 1$ and $\inf_D g \le 1$.

It is easy to establish that

$$\widehat{T(f)} = \mu + \frac{1}{n}\sum_{i=1}^{n} \nu\,(Y_i),$$

where Y_i corresponds to the i th observed pair and $\nu\,(Y) = \sum_{\pi \in \mathcal{N}^0} \tau_\pi \psi_\pi(Y)$. Hence,

$$\mathrm{Var}_{Pf}[\widehat{T(f)}] = \frac{\mathrm{Var}\,[\nu\,(Y)]}{n},$$

where $Y \sim Pf$, and similarly

$$\mathrm{Var}_1[\widehat{T(f)}] = \frac{\mathrm{Var}\,[\nu\,(\Xi)]}{n},$$

where $\Xi \sim P1 = 1$. So we must show that

$$\inf_D g \le \frac{\mathrm{Var}\,[\nu\,(Y)]}{\mathrm{Var}\,[\nu\,(\Xi)]} \le \sup_D g.$$

We employ an argument similar to the one used in the Appendix of [21]. For the right hand inequality we argue as follows:

$$\begin{aligned}\mathrm{Var}\,[\nu(Y)] &\le \mathbf{E}\,[(\nu(Y)-\mathbf{E}\,[\nu(\Xi)])^2] \\ &= \int_D (\nu(\xi)-\mathbf{E}\,[\nu(\Xi)])^2 g(\xi)\,d\lambda(\xi) \\ &\le \sup_D g \int_D (\nu(\xi)-\mathbf{E}\,[\nu(\Xi)])^2\,d\lambda(\xi) \\ &= \sup_D g\,\mathrm{Var}\,[\nu(\Xi)],\end{aligned}$$

and, for the left hand inequality:

$$\begin{aligned}\mathrm{Var}\,[\nu(\Xi)] &\le \mathbf{E}\,[(\nu(\Xi)-\mathbf{E}\,[\nu(Y)])^2] \\ &= \int_D (\nu(\xi)-\mathbf{E}\,[\nu(Y)])^2\,d\lambda(\xi) \\ &= \int_D (\nu(\xi)-\mathbf{E}\,[\nu(Y)])^2 \frac{1}{g(\xi)} g(\xi)\,d\lambda(\xi) \\ &\le \sup_D \left(\frac{1}{g}\right) \int_D (\nu(\xi)-\mathbf{E}\,[\nu(Y)])^2 g(\xi)\,d\lambda(\xi) \\ &= \frac{1}{\inf_D g}\mathrm{Var}\,[\nu(Y)].\end{aligned}$$

The divisions are valid since $\inf_D g > 0$ by hypothesis. This completes the proof. □

The quantity $\mathrm{Var}_1[\widehat{T(f)}] = \mathrm{Var}_1[\mu+\tau^{\mathrm{T}}Z] = \mathrm{Var}_1[\tau^{\mathrm{T}}Z] = \tau^{\mathrm{T}}\mathrm{Var}_1[Z]\tau = (1/n)\tau^{\mathrm{T}}\tau$. Moreover, $\mathbf{E}_{Pf}[\widehat{T(f)}]-T(f) = \mathbf{E}_{Pf}[\mu+\tau^{\mathrm{T}}Z]-(T(1)+\xi^{\mathrm{T}}f) = (\mu-T(1))+\tau^{\mathrm{T}}\mathbf{E}_f[Z]-\xi^{\mathrm{T}}f = (\mu-T(1))+\tau^{\mathrm{T}}Bf-\xi^{\mathrm{T}}f = (\mu-T(1))+(\xi-B\tau)^{\mathrm{T}}f$, a measure of bias. If we combine these two results we find that the surrogate mean square error can be written as

$$M^*(\widehat{T(f)};T(f)) = \frac{1}{n}\tau^{\mathrm{T}}\tau + \Big((\mu-T(1))+(\xi-B\tau)^{\mathrm{T}}f\Big)^2 \tag{5.37}$$

where, for example, $\tau = (\tau_\nu)$, $\nu \in \mathcal{N}^0$.

As before we adopt a minimax approach. For clarity of notation we rewrite the loss function

$$M^*(\widehat{T(f)};T(f))$$

as

$$M^*(\mu, \tau; T(1), f),$$

and we seek

$$\inf_{\mu,\tau} \sup_{f \in \mathcal{F}} M^*(\mu, \tau; T(1), f),$$

where $\mathcal{F}$ is the ellipsoid defined in equation (5.19) above, viewed from the point of view of the vector f where $f = (f_\nu)$, $\nu \in \mathcal{N}^0$:

$$\mathcal{F} = \{f : f^{\mathrm{T}} A f \leq C^2\},$$

where the matrix A is redefined accordingly.

5.3.4 Minimax estimation of $T(1)$ and resulting simplifications

In this section we show that the minimax estimator of μ is $T(1)$, as we would hope. First, we recall from equation (5.37) and the definition of $M^*(\mu, \tau; T(1), f)$ above that the surrogate mean square error takes the form

$$M^*(\mu, \tau; T(1), f) = \frac{1}{n}\tau^{\mathrm{T}}\tau + \Big((\mu - T(1)) + (\xi - B\tau)^{\mathrm{T}} f\Big)^2.$$

Now reparametrize by setting $(\mu - T(1))$ equal to K to get

$$M^*(K, \tau; f) = \frac{1}{n}\tau^{\mathrm{T}}\tau + (K + (\xi - B\tau)^{\mathrm{T}} f)^2.$$

This gives

$$\inf_{K,\tau} \sup_{f \in \mathcal{F}} M^*(K, \tau; f) = \inf_{K,\tau} \left\{ \frac{1}{n}\tau^{\mathrm{T}}\tau + \sup_{f \in \mathcal{F}} \left[(K + (\xi - B\tau)^{\mathrm{T}} f)^2\right] \right\}.$$

We now show that the minimax estimator of K is 0. First we remark that, because of the square and the fact that $f \in \mathcal{F}$ if and only if $-f \in \mathcal{F}$, we can restrict our attention to the infimum over $K \geq 0$. For arbitrary $K \geq 0$ consider

$$\sup_{f \in \mathcal{F}} \left[(K + (\xi - B\tau)^{\mathrm{T}} f)^2\right] = (K + (\xi - B\tau)^{\mathrm{T}} f^*)^2, \tag{5.38}$$

say, where $f^* \in \mathcal{F}$. Now if $(\xi - B\tau)^{\mathrm{T}} f^* < 0$, we can replace f^* in (5.38) by $-f^* \in \mathcal{F}$ to get

$$(K - (\xi - B\tau)^{\mathrm{T}} f^*)^2.$$

We note

$$(K - (\xi - B\tau)^{\mathrm{T}} f^*)^2 \geq (K + (\xi - B\tau)^{\mathrm{T}} f^*)^2$$

with strict equality in the case $K > 0$. This is a contradiction to the fact that the supremum is attained at f^* for non-zero K. So we must have $(\xi - B\tau)^{\mathrm{T}} f^* \geq 0$, except when $K = 0$.

Now assume that

$$\inf_{K \geq 0, \tau} \left\{ \frac{1}{n} \tau^{\mathrm{T}} \tau + \sup_{f \in \mathcal{F}} \left[(K + (\xi - B\tau)^{\mathrm{T}} f)^2 \right] \right\} =$$

$$\inf_{K \geq 0, \tau} \left\{ \frac{1}{n} \tau^{\mathrm{T}} \tau + (K + (\xi - B\tau)^{\mathrm{T}} f^*)^2 \right\} =$$

$$\frac{1}{n} \tau^{*\mathrm{T}} \tau^* + (K^* + (\xi - B\tau^*)^{\mathrm{T}} f^*)^2,$$

where the infimum is achieved at $K^* \geq 0$ and τ^*. If $K^* > 0$, we have from the above argument that

$$(\xi - B\tau^*)^{\mathrm{T}} f^* \geq 0.$$

However, in this case it is easy to see that a lower value of the infimum could be achieved by taking $K^* = 0$. Thus, we conclude that $K^* = 0$.

This argument allows us to conclude that the minimax estimator of μ, namely μ^*, is $T(1)$. Thus

$$M^*(\mu^*, \tau; T(1), f) = \frac{1}{n} \tau^{\mathrm{T}} \tau + ((\xi - B\tau)^{\mathrm{T}} f)^2,$$

and so from now on we consider

$$M^*(\tau; f) = \frac{1}{n} \tau^{\mathrm{T}} \tau + ((\xi - B\tau)^{\mathrm{T}} f)^2. \tag{5.39}$$

5.3.5 The minimax estimator and the minimax risk

We are now in a position to present the main theorem of this section.

Theorem 4 *The minimax estimator $\hat{\tau}$ is given by*

$$\hat{\tau} = \left(\frac{C^2 \frac{b_\nu \xi_\nu}{a_\nu^2}}{\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2}} \right)_{\nu \in \mathcal{N}^0} .$$

This yields a minimax risk of

$$M^*(\hat{\tau}; f_{LF}) = \frac{C^2}{n} \sum_{\nu \in \mathcal{N}^0} \frac{\xi_\nu^2}{a_\nu^2} \frac{1}{\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2}}, \tag{5.40}$$

where f_{LF} is the least favourable function and takes the form

$$f_{LF}(\hat{\tau}) = \pm C \left(\frac{\frac{\xi_\nu}{a_\nu^2} \Big/ \left(\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2} \right)}{\sqrt{\sum_{\nu' \in \mathcal{N}^0} \left\{ \left(\frac{\xi_{\nu'}}{a_{\nu'}} \right)^2 \Big/ \left(\frac{1}{n} + C^2 \frac{b_{\nu'}^2}{a_{\nu'}^2} \right)^2 \right\}}} \right)_{\nu \in \mathcal{N}^0} . \tag{5.41}$$

Proof. First we consider the sup over $f \in \mathcal{F}$.

$$\begin{aligned} \sup_{f \in \mathcal{F}} M^*(\tau; f) &= \frac{1}{n} \tau^{\mathrm{T}} \tau + \sup_{f : f^{\mathrm{T}} A f \leq C^2} ((\xi - B\tau)^{\mathrm{T}} f)^2 \\ &= \frac{1}{n} \tau^{\mathrm{T}} \tau + \sup_{f : f^{\mathrm{T}} A f = C^2} ((\xi - B\tau)^{\mathrm{T}} f)^2, \end{aligned}$$

the sup occurring on the boundary of $\mathcal{F}$ because of the following argument. Say, for a contradiction that the supremum is achieved by a vector f^* such that $f^{*\mathrm{T}} A f^* = D^2 < C^2$. The value of the supremum is, of course, $((\xi - B\tau)^{\mathrm{T}} f^*)^2$. Now consider the vector

$$f^\dagger = \frac{C}{D} f^*,$$

and note that this vector lies on the boundary of the ellipsoid since

$$f^{\dagger \mathrm{T}} A f^\dagger = \frac{C^2}{D^2} f^{*\mathrm{T}} A f^*$$

$$= C^2.$$

Note that

$$\begin{aligned}((\xi - B\tau)^{\mathrm{T}} f^{\dagger})^2 &= ((\xi - B\tau)^{\mathrm{T}} \frac{C}{D} f^*)^2 \\ &= \frac{C^2}{D^2}((\xi - B\tau)^{\mathrm{T}} f^*)^2 \\ &> ((\xi - B\tau)^{\mathrm{T}} f^*)^2,\end{aligned}$$

since $C^2 / D^2 > 1$. This gives the required contradiction since the last expression is the value of the supremum. Next we set $y = A^{1/2} f$ to obtain

$$\begin{aligned}\sup_{f \in \mathcal{F}} M^*(\tau; f) &= \frac{1}{n}\tau^{\mathrm{T}}\tau + \sup_{y:y^{\mathrm{T}}y=C^2} ((\xi - B\tau)^{\mathrm{T}} A^{-1/2} y)^2 \\ &= \frac{1}{n}\tau^{\mathrm{T}}\tau + \sup_{y:y^{\mathrm{T}}y=C^2} (d^{\mathrm{T}} y)^2 \\ &= \frac{1}{n}\tau^{\mathrm{T}}\tau + \sup_{y:||y||^2=C^2} (d^{\mathrm{T}} y)^2,\end{aligned}$$

where $d = A^{-1/2}(\xi - B\tau)$. Now, by the Cauchy-Schwarz inequality for inner products ([1], page 294), $(d^{\mathrm{T}}y)^2 = (d \cdot y)^2 \leq ||d||^2 ||y||^2 = ||d||^2 C^2$, with equality if and only if $y = \lambda d$, where $\lambda = C / ||d||$ to ensure that $||y||^2 = C^2$. By substituting back for d in $y = Cd / ||d||$, we immediately get an expression for the least favourable f, f_{LF}, as a function of (the as yet unknown) τ:

$$f_{LF}(\tau) = \pm \frac{CA^{-1}(\xi - B\tau)}{||A^{-1/2}(\xi - B\tau)||}.$$

Hence

$$\begin{aligned}\sup_{f \in \mathcal{F}} M^*(\tau; f) &= \frac{1}{n}\tau^{\mathrm{T}}\tau + C^2(\xi - B\tau)^{\mathrm{T}} A^{-1}(\xi - B\tau) \\ &= \frac{1}{n}\tau^{\mathrm{T}}\tau + C^2[\xi A^{-1}\xi - 2\tau^{\mathrm{T}} B A^{-1}\xi + \tau^{\mathrm{T}} B A^{-1} B\,\tau].\end{aligned}$$

Next we consider the inf over τ. First we differentiate with respect to τ:

$$\frac{\partial}{\partial\tau}\sup_{f\in\mathcal{F}} M^*(\tau;f_{LF}) \;=\; \frac{2}{n}\tau + C^2[-2BA^{-1}\xi + 2BA^{-1}B\,\tau].$$

Now we set $\partial/\partial\tau\sup_{f\in\mathcal{F}} M^*(\hat\tau;f_{LF}) = 0$ to get $(C^2BA^{-1}B + I/n)\,\hat\tau = C^2BA^{-1}\xi$, and hence

$$\hat\tau = (C^2BA^{-1}B + I/n)^{-1}C^2BA^{-1}\xi,$$

which can easily be shown to yield a minimum. Fortunately, the matrix $C^2BA^{-1}B + I/n$ is diagonal and therefore its inverse is easy to find.

Substituting for the diagonal matrices A and B, and for the vector ξ, we get

$$\hat\tau = \left(\frac{C^2\frac{b_\nu\xi_\nu}{a_\nu^2}}{\frac{1}{n}+C^2\frac{b_\nu^2}{a_\nu^2}}\right)_{\nu\in\mathcal{N}^0}.$$

Moreover, the least favourable function $f_{LF}(\hat\tau)$ can now be easily evaluated as

$$f_{LF}(\hat\tau) = \pm C\left(\frac{\frac{\xi_\nu}{a_\nu^2}\Big/\left(\frac{1}{n} + C^2\frac{b_\nu^2}{a_\nu^2}\right)}{\sqrt{\sum_{\nu'\in\mathcal{N}^0}\left\{\left(\frac{\xi_{\nu'}}{a_{\nu'}}\right)^2\Big/\left(\frac{1}{n}+C^2\frac{b_{\nu'}^2}{a_{\nu'}^2}\right)^2\right\}}}\right)_{\nu\in\mathcal{N}^0}.$$

(The symmetry of the least favourable function is discussed in Section 5.3.11.) Next, a long but standard calculation enables us to evaluate $\inf_\tau\sup_{f\in\mathcal{F}} M^*(\tau;f) = M^*(\hat\tau;f_{LF})$ as:

$$M^*(\hat\tau;f_{LF}) = \frac{C^2}{n}\sum_{\nu\in\mathcal{N}^0}\frac{\xi_\nu^2}{a_\nu^2}\,\frac{1}{\frac{1}{n}+C^2\frac{b_\nu^2}{a_\nu^2}}, \tag{5.42}$$

equation (5.40) above. Thus the proof is complete. □

In the above proof we must, however, check that $||d|| = ||d(\hat\tau)||$ is finite. From this, the finite nature of the minimax risk, for example, follows. An easy computation gives that

$$||d||^2 = \frac{1}{n}\sum_{\nu'\in\mathcal{N}^0}\left\{\left(\frac{\xi_{\nu'}}{a_{\nu'}}\right)^2\Big/\left(\frac{1}{n}+C^2\frac{b_{\nu'}^2}{a_{\nu'}^2}\right)^2\right\},$$

the expression which appears in the denominator of equation (5.41). Hence we have $||d||$ is finite if and only if

$$\sum_{\nu \in \mathcal{N}^0} \left\{ \left(\frac{\xi_\nu}{a_\nu}\right)^2 / \left(\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2}\right)^2 \right\}$$

is finite. Now

$$\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2} \geq \frac{1}{n},$$

and hence

$$\frac{1}{\left(\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2}\right)^2} \leq n^2.$$

Therefore,

$$\frac{\left(\frac{|\xi_\nu|}{a_\nu}\right)^2}{\left(\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2}\right)^2} \leq n^2 \left(\frac{|\xi_\nu|}{a_\nu}\right)^2.$$

The finite nature of $||d||$ now follows from (5.34).

Finally, we remark that the minimax risk is a decreasing function of n, the number of emissions, as we should hope. To see this just rewrite equation (5.40) for the minimax risk $M^*(\hat{\tau}; f_{LF})$ as

$$C^2 \sum_{\nu \in \mathcal{N}^0} \frac{\xi_\nu^2}{a_\nu^2} \frac{1}{1 + n\, C^2 \frac{b_\nu^2}{a_\nu^2}},$$

and note that $M^*(\hat{\tau}; f_{LF})$ decreases, as n increases.

5.3.6 An important observation

Our estimate $\widehat{T(f)} = \hat{\tau}^{\mathrm{T}} Z$ (we drop the $T(1)$ term) can easily be rewritten as

$$\widehat{T(f)} = \left(\frac{C^2 \frac{b_\nu}{a_\nu^2} Z_\nu}{\frac{1}{n} + C^2 \frac{b_\nu^2}{a_\nu^2}} \right)^{\mathrm{T}} \xi$$

$$= \hat{f}^{\mathrm{T}}\xi$$

$$= T(\hat{f}),$$

where

$$\hat{f} = \left(\frac{C^2 \frac{b_v}{a_v^2} Z_v}{\frac{1}{n} + C^2 \frac{b_v^2}{a_v^2}} \right)_{v \in \mathcal{N}^0} . \tag{5.43}$$

Moreover, the vector $\hat{f}$ (and hence the function $\hat{f}$) can be seen not to depend upon the functional T. Also, the following lemma holds

Lemma 4 *$\hat{f}$ minimizes*

$$\{Z - Bf\}^{\mathrm{T}}\{Z - Bf\} + \frac{1}{n\,C^2} f^{\mathrm{T}} A f$$

where in fact $Bf = \mathrm{E}\,[Z]$.

Proof. First, recall that A and B are diagonal matrices. A simple calculation yields that

$$BZ - B^2\hat{f} - \frac{1}{n\,C^2} A\hat{f} = 0.$$

Now set $f = \hat{f} + g$.

$$\{Z - Bf\}^{\mathrm{T}}\{Z - Bf\} + \frac{1}{n\,C^2} f^{\mathrm{T}} A f =$$

$$\{Z - B\hat{f} - Bg\}^{\mathrm{T}}\{Z - B\hat{f} - Bg\} + \frac{1}{n\,C^2}(\hat{f} + g)^{\mathrm{T}} A (\hat{f} + g) =$$

$$\{Z - B\hat{f}\}^{\mathrm{T}}\{Z - B\hat{f}\} + \frac{1}{n\,C^2}\hat{f}^{\mathrm{T}} A \hat{f} + g^{\mathrm{T}} B^2 g + \frac{1}{n\,C^2} g^{\mathrm{T}} A g -$$

$$2\,(BZ - B^2\hat{f} - \frac{1}{n\,C^2} A\hat{f})^{\mathrm{T}} g.$$

The last term disappears, leaving us with

$$\{Z - Bf\}^{\mathrm{T}}\{Z - Bf\} + \frac{1}{n\,C^2} f^{\mathrm{T}} A f =$$

$$\{Z - B\hat{f}\}^{\mathrm{T}}\{Z - B\hat{f}\} + \frac{1}{n\,C^2}\hat{f}^{\mathrm{T}} A \hat{f} +$$

$$g^{\mathrm{T}}B^2g+\frac{1}{nC^2}g^{\mathrm{T}}Ag,$$

a quantity that is minimized by setting $g = 0$. This completes the proof. (An alternative proof can be obtained by differentiation with respect to f.) □

We make two remarks about Lemma 4. First, $\hat{f}$ can be seen to minimize a penalized least squares form. Secondly, the penalty term can be interpreted as follows. Because

$$a_v = a_{(j,k)} = (j+1)^a(k+1)^a,$$

the high frequency terms in f make a large contribution to the term $f^{\mathrm{T}}Af$, and so this penalty term can be thought of as penalizing rough fs. Moreover, the higher the value of C^2 the less the effect of the penalty term. This ties in with C^2 controlling the size of the ellipsoid that defines the class $\mathcal{F}$.

5.3.7 The discrete case

One difference between the continuous case and discrete case is the fact that the vector of data Z is no longer infinitely long, but has length $r(N)$. We shall write it as $\tilde{Z}$. Again the idea is to estimate $T(1)+\xi^{\mathrm{T}}f$ by $\mu+\tau^{\mathrm{T}}\tilde{Z}$, where τ this time is a vector of length $r(N)$. Almost exactly the same analysis goes through as described in Section 5.3.5.

We use the same definitions here as we did in Section 5.3.3 for the mean square error. This definition leads to the two expressions given in equation (5.36) for $M(\widehat{T(f)};T(f))$ and $M^*(\widehat{T(f)};T(f))$, the surrogate mean square error. Again we can state a proposition that means that order of magnitude statements made about one error will also be true for the other. This proposition is identical in form to Proposition 3, although its proof is a little different.

Proposition 4 *Suppose that f (and hence g = Pf) is bounded above and below away from zero. Then*

$$\inf_B f \leq \inf_D g \leq \frac{M(\widehat{T(f)};T(f))}{M^*(\widehat{T(f)};T(f))} \leq \sup_D g \leq \sup_B f$$

Proof. Again, we consider only the inner pair of inequalities. As we have seen in the proof of Proposition 3 for example, the result will follow if we can show

$$\inf_D g \leq \frac{\mathrm{Var}_{Pf}[\widehat{T(f)}]}{\mathrm{Var}_1[\widehat{T(f)}]} \leq \sup_D g.$$

The idea of the proof comes from [22] and was outlined in Section 5.2.12. The estimator

$$\widehat{T(f)} = \mu + \sum_{[\pi]} \tau_{[\pi]} \tilde{Z}_{[\pi]}$$

can be shown to equal

$$\mu + \frac{1}{n} \sum_{i=1}^{\mathcal{N}} t(J_i),$$

where $t(j) = \sum_{[\pi]} \tau_{[\pi]} \chi_{[\pi]j}$, $J_i = j(Y_i)$ is a random variable indicating the bin into which Y_i falls, and $\mathcal{N} = \sum_{j=1}^{r(N)} n_j$ is a random variable with mean n. Thus, $\mathrm{Var}_{Pf}[\widehat{T(f)}] = (1/n^2)\,\mathrm{Var}_{Pf}[\sum_{i=1}^{\mathcal{N}} t(J_i)]$. It can be shown that $\mathrm{Var}_{Pf}[\sum_{i=1}^{\mathcal{N}} t(J_i)] = n \int_D t^2(j(y))\, g(y)\, d\lambda(y)$, where $g = Pf$. Immediately, we have

$$\begin{aligned} \mathrm{Var}_{Pf}[\widehat{T(f)}] &= \frac{1}{n} \int_D t^2(j(y))\, g(y)\, d\lambda(y) \\ &\leq \sup_D g \, \frac{1}{n} \int_D t^2(j(y))\, d\lambda(y) \\ &= \sup_D g \, \mathrm{Var}_1[\widehat{T(f)}]. \end{aligned}$$

Also,

$$\begin{aligned} \mathrm{Var}_1[\widehat{T(f)}] &= \frac{1}{n} \int_D t^2(j(y))\, d\lambda(y) \\ &= \frac{1}{n} \int_D t^2(j(y))\, \frac{1}{g(y)} g(y)\, d\lambda(y) \\ &\leq \sup_D \left(\frac{1}{g}\right) \frac{1}{n} \int_D t^2(j(y))\, g(y)\, d\lambda(y) \\ &= \frac{1}{\inf_D g} \mathrm{Var}_f[\widehat{T(f)}]. \end{aligned}$$

The divisions are valid since $\inf_D g > 0$ by hypothesis. This completes the proof. □

We now recall a few definitions: A and B are the diagonal matrices

$$A = \mathrm{diag}\,(a^2_{v^1_2}, a^2_{v^1_3}, \ldots, a^2_{v^2_1}, a^2_{v^2_2}, a^2_{v^2_3}, \ldots, \cdots, a^2_{v^{r(N)}_1}, a^2_{v^{r(N)}_2}, a^2_{v^{r(N)}_3}, \ldots)$$

and

$$B = \operatorname{diag}(b_{\nu_2^1}, b_{\nu_3^1}, \ldots, b_{\nu_1^2}, b_{\nu_2^2}, b_{\nu_3^2}, \ldots, \cdots, b_{\nu_1^{r(N)}}, b_{\nu_2^{r(N)}}, b_{\nu_3^{r(N)}}, \ldots).$$

The matrix Γ, which was discussed in Section 5.2.11, has $r(N)$ rows:

$$\begin{pmatrix} \gamma_{\nu_2^1}, \gamma_{\nu_3^1}, \ldots & 0, 0, 0, \ldots & \cdots & 0, 0, 0, \ldots \\ 0, 0, \ldots & \gamma_{\nu_1^2}, \gamma_{\nu_2^2}, \gamma_{\nu_3^2}, \ldots & \cdots & 0, 0, 0, \ldots \\ \vdots & \vdots & \vdots & \vdots \\ 0, 0, \ldots & 0, 0, 0, \ldots & \cdots & \gamma_{\nu_1^{r(N)}}, \gamma_{\nu_2^{r(N)}}, \gamma_{\nu_3^{r(N)}}, \ldots \end{pmatrix}.$$

Since $\mathbf{E}[\tau^{\mathrm{T}}\tilde{Z}] = \tau^{\mathrm{T}}\mathbf{E}[\tilde{Z}] = \tau^{\mathrm{T}}\Gamma B f$ (see equation (5) of [22]), we can now formulate an expression for $M^*(\tau; f)$, analogous to equation (5.37):

$$M^*(\tau; f) = \frac{1}{n}\tau^{\mathrm{T}}\tau + ((\xi - B\Gamma^{\mathrm{T}}\tau)^{\mathrm{T}} f)^2, \tag{5.44}$$

using the result given on page 7 of Johnstone and Silverman[22] that $\mathrm{Var}_1[\tilde{Z}_{[\nu_1]}, \tilde{Z}_{[\nu_2]}] = n^{-1}\,\delta_{[\nu_1]\,[\nu_2]}$. We now introduce three quantities which we will find very useful for writing down the results of this section. For $k = 1, \ldots, r(N)$, we let

$$\begin{aligned} \alpha_k &= \sum_l \frac{\xi^2_{\nu_l^k}}{a^2_{\nu_l^k}} \\ \beta_k &= \sum_l \frac{b^2_{\nu_l^k}\gamma^2_{\nu_l^k}}{a^2_{\nu_l^k}} \\ \gamma_k &= \sum_l \frac{b_{\nu_l^k}\gamma_{\nu_l^k}\xi_{\nu_l^k}}{a^2_{\nu_l^k}}. \end{aligned} \tag{5.45}$$

In Section 5.3.10 we establish that these three quantities are finite. In the case of α_k and γ_k this involves using the inequality given in (5.34) that imposes a restriction on the class of linear functionals considered. We point out that l runs over $2, \ldots$ when $k = 1$, and $1, 2, \ldots$ for $k = 2, \ldots, r(N)$.

An argument similar to the proof of the above theorem yields

$$\hat{\tau} = \left(\frac{C^2\gamma_k}{\frac{1}{n} + C^2\beta_k} \right)_{k=1,\ldots,r(N)}. \tag{5.46}$$

This $\hat{\tau}$ yields a minimax risk $M^*(\hat{\tau}; f_{LF})$ of

$$\frac{C^2}{n}\sum_{k=1}^{r(N)} \frac{\alpha_k}{\frac{1}{n} + C^2\beta_k} + C^4 \sum_{k=1}^{r(N)} \left(\frac{\alpha_k\beta_k - \gamma_k^2}{\frac{1}{n} + C^2\beta_k} \right), \tag{5.47}$$

and a least favourable function f_{LF} where

$$f_{LF}(\hat{\tau}) = \pm \frac{CA^{-1}(\xi - B\Gamma^{\mathrm{T}}\hat{\tau})}{\|A^{-1/2}(\xi - B\Gamma^{\mathrm{T}}\hat{\tau})\|}, \tag{5.48}$$

the explicit form of which is quite complicated. (The symmetry of the least favourable function is discussed in Section 5.3.11.)

The second term of equation (5.47) can be shown to be positive by means of the Cauchy-Schwartz inequality. Moreover, by writing equation (5.47) as

$$C^2 \sum_{k=1}^{r(N)} \left\{ \alpha_k - C^2 \frac{\gamma_k^2}{\frac{1}{n} + C^2\beta_k} \right\},$$

we can easily see that the minimax risk $M^*(\hat{\tau}; f_{LF})$ is a decreasing function of the expected number of observations n, as we would hope. The reader is invited to compare the first term of expression (5.47) with the minimax risk for the continuous case given in (5.40). It can be shown that the expression given in (5.47) tends to the expression given in (5.40) from above as $N \rightarrow \infty$.

Again it is possible to write $\widehat{T(f)} = T(\hat{f})$ where the function $\hat{f}$ is independent of the functional T and is represented by the vector $\hat{f}$. First, we must define the quantity $\rho(m_1, m_2)$ where $m_1 = 1, \ldots, r(N)$. Here $m_2 = 2, \ldots$ if $m_1 = 1$, and $m_2 = 1, 2, \ldots$ if $m_1 > 1$, and the definition is:

$$\rho(m_1, m_2) = \frac{C^2 \frac{b_{\nu_{m_2}^{m_1}} \gamma_{\nu_{m_2}^{m_1}}}{a^2_{\nu_{m_2}^{m_1}}} \tilde{Z}_{m_1}}{\frac{1}{n} + C^2\beta_{m_1}}. \tag{5.49}$$

Then $\hat{f}$ takes the following form

$$(\rho(1,2), \rho(1,3), \ldots, \rho(2,1), \rho(2,2), \ldots, \cdots, \rho(r(N),1), \rho(r(N),2), \ldots). \tag{5.50}$$

We can now state a result about $\hat{f}$ that is analogous to Lemma 4: $\hat{f}$ minimizes

$$\{\tilde{Z} - \Gamma B f\}^{\mathrm{T}}\{\tilde{Z} - \Gamma B f\} + \frac{1}{n\,C^2} f^{\mathrm{T}} A f, \tag{5.51}$$

where in fact $\Gamma B f = \mathrm{E}\,[\tilde{Z}]$. The proof, which follows the form of the one used above, relies on the fact that

$$B\Gamma^{\mathrm{T}}\tilde{Z} - B\,\Gamma^{\mathrm{T}}\Gamma\,B\hat{f} - \frac{1}{n\,C^2} A\hat{f} = 0,$$

which can be established by a simple, but tedious calculation. A similar remark can be made here as was made after the proof of Lemma 4 in Section 5.3.6 about the interpretation of this penalized least squares form and about the role of C^2. We note also that this $\hat{f}$ provides an estimate for all the coefficients f_ν, and not just the low frequency ones; see Section 5.2.12.

5.3.8 The limit of the minimax risks as $n \to \infty$

It is interesting to consider the limit of the minimax risk as the expected number of observations $n \to \infty$. In the continuous case the minimax risk (5.40) clearly tends to zero as $n \to \infty$. In the case when N is finite, the same is true for the first term of equation (5.47). However, the second term tends to

$$C^2 \sum_{k=1}^{r(N)} \left(\alpha_k - \frac{\gamma_k^2}{\beta_k}\right), \tag{5.52}$$

a constant which can be shown to be greater that 0 (again by using the Cauchy-Schwartz inequality). We shall discuss this further with the aid of a numerical example in Section 6.3.2.

5.3.9 Doubling the number of detectors

In this section we consider the effect of doubling the number of detectors N from M to $2M$. We assume that the expected number of emissions n is fixed, although possibly infinite, throughout this section. First, however, we must introduce some further notation. Let $Z_{r(M)}$ be the vector of normalized bin counts when there are M detectors. We write $Z_{r(M)} = \sum_{[\nu]} \tilde{Z}_{r(M)[\nu]}\,\chi_{[\nu]}$. Hence $Z_{r(M)} = X_{r(M)}\tilde{Z}_{r(M)}$ where $X_{r(M)}$ is the $r(M) \times r(M)$ orthogonal matrix ($X^{-1} = X^{\mathrm{T}}$) representing the $r(M)$ orthonormal vectors $\chi_{[\nu]}$ with respect to the standard basis of the space $\mathbf{R}^{r(M)}$. Clearly, the vector of bin counts $Z_{r(M)}$ can be obtained from the vector of bin counts $Z_{r(2M)}$ in a simple way by multiplying $Z_{r(2M)}$ by an $r(M) \times r(2M)$ matrix $\mathcal{A}$:

$$Z_{r(M)} = \mathcal{A}\, Z_{r(2M)}$$

since the vector $Z_{r(2M)}$ contains all the information of $Z_{r(M)}$ split up (and additional information that does not contribute to $Z_{r(M)}$, as will become clear below). Consider, for example the case when $M = 4$. In this case we shall refer to the detectors as $1, 2, 3, 4$. Now split each detector into two, giving the following 8 detectors: $1_1, 1_2, 2_1, 2_2, 3_1, 3_2, 4_1, 4_2$. When $N = 4$ there are $N(N-1)/2 = 4.3/2 = 6$ bins (or tubes) defined by these detectors, which we shall refer to as $(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)$, whereas, when $N = 8$ there are $8.7/2 = 28$ bins. Table 5.2, in which (i, j) denotes the bin defined by the detectors i and j, $i \neq j$, shows the relationship between the (information given by the) bins in the $N = 8$ case and the bins in the $N = 4$ case.

Bins when $N = 2M$	Bins when $N = M$
$(1_1, 2_1)(1_1, 2_2)(1_2, 2_1)(1_2, 2_2)$	$(1, 2)$
$(1_1, 3_1)(1_1, 3_2)(1_2, 3_1)(1_2, 3_2)$	$(1, 3)$
$(1_1, 4_1)(1_1, 4_2)(1_2, 4_1)(1_2, 4_2)$	$(1, 4)$
$(2_1, 3_1)(2_1, 3_2)(2_2, 3_1)(2_2, 3_2)$	$(2, 3)$
$(2_1, 4_1)(2_1, 4_2)(2_2, 4_1)(2_2, 4_2)$	$(2, 4)$
$(3_1, 4_1)(3_1, 4_2)(3_2, 4_1)(3_2, 4_2)$	$(3, 4)$

Table 5.2: *An example of the relationship between the bins when the number of detectors N is doubled from M to 2M,* $M = 4$

The information in the following 4 (= M) detectors is not used in reconstructing $Z_{r(4)}$ from $Z_{r(8)}$: $(1_1, 1_2), (2_1, 2_2), (3_1, 3_2), (4_1, 4_2)$, each of these 4 tubes corresponding to a single detector in the $N = 4$ case. This gives us that the $r(8) \times r(4)$ matrix $\mathcal{A}$ has 4 (= M) completely zero columns. In general we have

$$\begin{aligned} \tilde{Z}_{r(M)} &= X^{-1}_{r(M)} Z_{r(M)} \\ &= X^{\mathrm{T}}_{r(M)} Z_{r(M)} \end{aligned}$$

$$= X^{\mathrm{T}}_{r(M)} \mathcal{A} Z_{r(2M)}$$

$$= X^{\mathrm{T}}_{r(M)} \mathcal{A} X_{r(2M)} \tilde{Z}_{r(2M)}$$

$$= D \tilde{Z}_{r(2M)} \tag{5.53}$$

where D is an $r(M) \times r(2M)$ matrix of rank $r(M)$.

Let $\mathcal{T}_{r(N)}$ be the class of estimators that we consider in this chapter, where $\mathcal{T}_{r(N)} = \{\tau^{\mathrm{T}}_{r(N)} \tilde{Z}_{r(N)}\}$. It is easy to see that

$$\mathcal{T}_{r(M)} \subset \mathcal{T}_{r(2M)},$$

for

$$\mathcal{T}_{r(M)} = \{\tau^{\mathrm{T}}_{r(M)} \tilde{Z}_{r(M)}\} = \{\tau^{\mathrm{T}}_{r(M)} D \tilde{Z}_{r(2M)}\} \subset \{\tau^{\mathrm{T}}_{r(2M)} \tilde{Z}_{r(2M)}\}.$$

Hence,

$$\inf_{\widehat{T(f)} \in \mathcal{T}_{r(2M)}} \sup_{f \in \mathcal{F}} M^*(\widehat{T(f)}; T(f)) < \inf_{\widehat{T(f)} \in \mathcal{T}_{r(M)}} \sup_{f \in \mathcal{F}} M^*(\widehat{T(f)}; T(f)),$$

where $\mathcal{F}$ is some set in which the function f is restricted. Thus, the minimax risk for $N = 2M$ detectors is less than the minimax risk for $N = M$ detectors.

The above monotonicity property does not, however, hold in general, *i.e.* the minimax risk for $N = m_1$ detectors is not necessarily less than the minimax risk for $N = m_2$ detectors, if $m_1 > m_2$. We illustrate this feature by means of the Figure 5.1 which shows a graph of the logarithm of the limit of the minimax risk as $n \to \infty$ (as discussed in Section 5.3.8) for the functional

$$T(f) = \frac{\int_{\mathrm{Disc}} f \, d\mu}{\int_{\mathrm{Disc}} d\mu} \tag{5.54}$$

for various numbers of detectors N, where the disc that is considered has centre the origin of the unit circle and radius 0.1. We shall meet this functional again in detail in Chapter 6. In this work we consider only even values of N, as do Johnstone and Silverman in [22] (see page 26). In the example presented in Figure 5.1 $a = 1.0$ and $C = \sqrt{2}$. We indicate by dots on the graph the logarithm of limit of the minimax risk when $N = 4, 8, 16, 32, 64, 128$. The overall

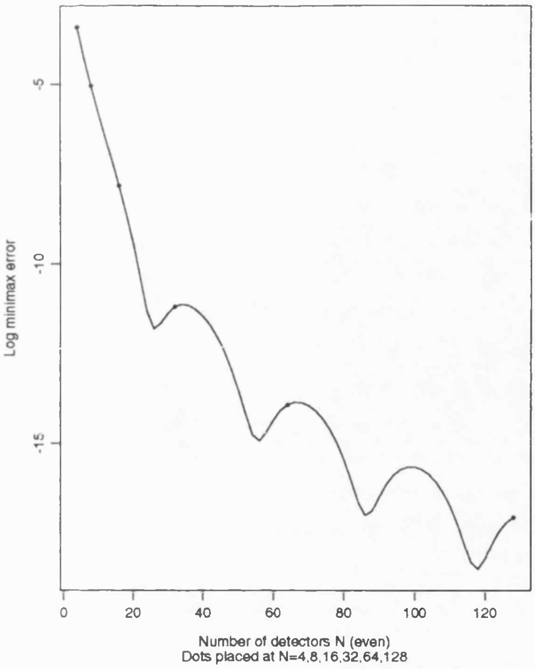


Figure 5.1: *The limit of the minimax error as* $n \rightarrow \infty$ *as a function of* N

appearance is of a function that is not monotonic. However, examination of the dots only shows the monotonicity discussed in this section.

5.3.10 More on α_k, β_k and γ_k

The first part of this section is devoted to an attempt to establish the finite nature of the three quantities defined in (5.45) of Section 5.3.7: α_k, β_k and γ_k, $k = 1, \dots, r(N)$. The second part outlines an attempt to consider the minimax risk as a power series in $1/n$. Unfortunately, this attempt is seen to be unsuccessful.

First, we establish the finiteness of α_k, β_k and γ_k, $k = 1, \dots, r(N)$.

The quantity α_k is easy to deal with. For

$$\begin{aligned} |\alpha_k| &= \left| \sum_l \frac{\xi^2_{v^k_l}}{a^2_{v^k_l}} \right| \\ &\leq \sum_l \frac{|\xi_{v^k_l}|^2}{a^2_{v^k_l}} \end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{\nu} \frac{|\xi_\nu|^2}{a_\nu^2} \\
&< \infty
\end{aligned}
$$

by inequality (5.34).

Before we can establish the finiteness of β_k, we must consider γ_ν^2. From the definition we immediately have that

$$
\begin{aligned}
\gamma_\nu^2 &= \gamma_{(j,k)}^2 \\
&= \operatorname{sinc}^2\{(2j+1)\pi/2N\}\operatorname{sinc}^2\{(2k+1)\pi/2N\}.
\end{aligned}
$$

We can easily get an upper bound on $\operatorname{sinc}^2\theta$, for real θ, by noting that

$$
\operatorname{sinc}^2\theta = \sin^2\theta/\theta^2 \leq 1/\theta^2.
$$

Hence,

$$
\begin{aligned}
\gamma_\nu^2 &= \gamma_{(j,k)}^2 \\
&\leq \frac{(2N)^2}{\pi^2(2j+1)^2}\frac{(2N)^2}{\pi^2(2k+1)^2} \\
&= \frac{16N^4}{\pi^4(2j+1)^2(2k+1)^2}.
\end{aligned}
\tag{5.55}
$$

Now

$$
\begin{aligned}
|\beta_k| &= \beta_k \\
&= \sum_{l} \frac{b_{\nu_l^k}^2 \gamma_{\nu_l^k}^2}{a_{\nu_l^k}^2} \\
&\leq \sum_{\nu} \frac{b_\nu^2 \gamma_\nu^2}{a_\nu^2} \\
&\leq \frac{16N^4}{\pi^4} \sum_{(j,k)} \frac{b_{(j,k)}^2}{a_{(j,k)}^2 (2j+1)^2 (2k+1)^2}
\end{aligned}
$$

$$= \frac{16N^4}{\pi^4}\sum_{(j,k)}\frac{1}{(j+k+1)\,(j+1)^{2a}\,(k+1)^{2a}\,(2j+1)^2\,(2k+1)^2}, \qquad (5.56)$$

where the second inequality follows from inequality (5.55) and the last equality follows from the definitions (5.17) and (5.20). The final bound given in (5.56) is a quantity that certainly converges for positive values of a.

Finally, we consider γ_k. We argue as follows:

$$\begin{aligned}
|\gamma_k| &= \left|\sum_l \frac{b_{v_l^k}\gamma_{v_l^k}\xi_{v_l^k}}{a^2_{v_l^k}}\right| \\
&\leq \sum_l \frac{b_{v_l^k}\,|\gamma_{v_l^k}|\,|\xi_{v_l^k}|}{a^2_{v_l^k}} \\
&\leq \sum_v \frac{b_v\,|\gamma_v|\,|\xi_v|}{a_v^2} \\
&\leq \frac{4N^2}{\pi^2}\sum_{(j,k)}\frac{b_{(j,k)}\,|\xi_{(j,k)}|}{a^2_{(j,k)}\,(2j+1)(2k+1)} \quad \text{by inequality (5.55)} \\
&\leq \frac{4N^2}{\pi^2}\sum_{(j,k)}\frac{|\xi_{(j,k)}|}{a^2_{(j,k)}} \quad \text{since } b_{(j,k)} < 1 \\
&= \frac{4N^2}{\pi^2}\sum_v \frac{|\xi_v|}{a_v^2}
\end{aligned}$$

Hence, if we can establish that

$$\sum_v \frac{|\xi_v|}{a_v^2} \qquad (5.57)$$

is finite, we have the desired result. To do this we split the sum given in expression (5.57) into two parts:

$$\begin{aligned}
\sum_v \frac{|\xi_v|}{a_v^2} &= \sum_{v:|\xi_v|\leq 1}\frac{|\xi_v|}{a_v^2} + \sum_{v:|\xi_v|>1}\frac{|\xi_v|}{a_v^2} \\
&\leq \sum_{v:|\xi_v|\leq 1}\frac{1}{a_v^2} + \sum_{v:|\xi_v|>1}\frac{|\xi_v|^2}{a_v^2} \\
&\leq \sum_v \frac{1}{a_v^2} + \sum_v \frac{|\xi_v|^2}{a_v^2}
\end{aligned}$$

$$= \sum_{(j,k)} \frac{1}{(j+1)^{2a}(k+1)^{2a}} + \sum_{v} \frac{|\xi_v|^2}{a_v^2},$$

the first term of which can be shown to be finite if $a > 1/2$ and the second term of which is finite by the assumption given in equation (5.34).

We now move on to the second part of this section in which we consider the minimax risk as a power series in $1/n$. However, we shall see that the the power series that we produce are meaningless as the coefficients grow extremely rapidly. This analysis does, however, provide us with some confirmation that the limit as $n \rightarrow \infty$ of (5.47) is the expression given by equation (5.52). We proceed in a standard way using the fact that

$$(1+x)^{-1} = \sum_{k=0}^{\infty}(-1)^k x^k, \ |x| < 1. \tag{5.58}$$

First, we consider the power series for the minimax risk when there are a finite number N of detectors, as given in equation (5.47).

We proceed by considering the expansion of

$$\left(1 + \frac{1}{C^2\beta_k n}\right)^{-1}, \ k = 1, \ldots, r(N)$$

by means of (5.58). It is easy to see that the expansion is valid if $n > 1/C^2\beta_k$, $k = 1, \ldots, r(N)$. Hence, for fixed N, we only have a finite number of conditions to check, and we can easily ensure that they all hold by taking $n > \max\{1/C^2\beta_k, \ k = 1, \ldots, r(N)\}$. The power series then becomes

$$\sum_{l=0}^{\infty}(-1)^l \left[\sum_{k=1}^{r(N)} \frac{\gamma_k^2}{\beta_k^2}\left(\frac{1}{C^2\beta_k}\right)^l\right] \frac{1}{n^{l+1}} + C^2 \sum_{k=1}^{r(N)}(\alpha_k - \frac{\gamma_k^2}{\beta_k}). \tag{5.59}$$

It should be pointed out at this stage that, at least for the example considered in Section 6.3.2 below, the coefficients of $1/n^{l+1}$ in the square brackets of expression (5.59)

$$\sum_{k=1}^{r(N)} \frac{\gamma_k^2}{\beta_k^2}\left(\frac{1}{C^2\beta_k}\right)^l$$

increase very rapidly (although remaining finite for fixed N, as the sum is of a finite number of terms) with l as Table 5.3 of approximate coefficients when the functional under consideration is given by equation (5.54) shows. The constant term, *i.e.* the term that is independent of n,

N	constant	$\frac{1}{n}$	$\frac{1}{n^2}$	$\frac{1}{n^3}$
32	6.8×10^{-6}	250	-2.2×10^{8}	4.9×10^{14}
64	4.4×10^{-7}	260	-6.2×10^{9}	4.5×10^{17}
128	1.9×10^{-8}	270	-1.2×10^{11}	2.9×10^{20}
256	7.2×10^{-10}	300	-2.4×10^{12}	1.4×10^{23}
512	2.3×10^{-11}	350	-7.6×10^{13}	1.2×10^{26}
1024	1.5×10^{-13}	430	-5.7×10^{15}	4.3×10^{29}

Table 5.3: *Coefficients for different values of N*

is the value of the minimax risk, when $n = \infty$, as given by equation (5.52) and discussed in Section 5.3.8.

The power series for the continuous case can be written as

$$\sum_{l=0}^{\infty}(-1)^l \left[\sum_{v:\frac{a_v^2}{C^2b_v^2}<n} \frac{\xi_v^2}{b_v^2}\left(\frac{a_v^2}{C^2b_v^2}\right)^l\right]\frac{1}{n^{l+1}} + \frac{C^2}{n}\sum_{v:\frac{a_v^2}{C^2b_v^2}\ge n}\frac{\xi_v^2}{a_v^2}\frac{1}{\frac{1}{n}+C^2\frac{b_v^2}{a_v^2}}, \qquad (5.60)$$

ensuring convergence of the expansions used in the first term. The second term of equation (5.60) can be rewritten as

$$C^2\sum_{v:\frac{a_v^2}{C^2b_v^2}\ge n}\frac{\xi_v^2}{a_v^2}\frac{1}{1+C^2\frac{b_v^2 n}{a_v^2}} < C^2\sum_{v:\frac{a_v^2}{C^2b_v^2}\ge n}\frac{|\xi_v|^2}{a_v^2}. \qquad (5.61)$$

Since we assume that $\sum_v |\xi_v|^2 / a_v^2$ converges then the second term of equation (5.60) is bounded above by the tail of a convergent series, and so can be made arbitrarily small by taking n sufficiently large. Hence, for large n, $M^*(\hat{t}; f_{LF})$ may be thought of as the following power series, which has a zero constant term:

$$\sum_{l=0}^{\infty}(-1)^l \left[\sum_{v:\frac{a_v^2}{C^2b_v^2}<n} \frac{\xi_v^2}{b_v^2}\left(\frac{a_v^2}{C^2b_v^2}\right)^l \right] \frac{1}{n^{l+1}}.$$

It should be noted that as $n \to \infty$

$$\sum_{v:\frac{a_v^2}{C^2b_v^2}<n} \frac{\xi_v^2}{b_v^2}\left(\frac{a_v^2}{C^2b_v^2}\right)^l$$

diverges, although the minimax risk itself (5.40) tends to zero. We shall consider these power series expansions again in Section 6.3.2.

5.3.11 Radial symmetry of the least favourable function

The least favourable function $f_{LF}(r,\theta)$, represented by the vector f_{LF}, takes the form

$$f_{LF}(r,\theta) = \sum_v (f_{LF})_v\, \phi_v(r,\theta).$$

We say that this function is *radially symmetric* if it is independent of θ, and we state the following proposition:

Proposition 5 *If* $(f_{LF})_{(j,k)} = 0$ *whenever* $j \neq k$ *then the least favourable function* f_{LF} *is radially symmetric.*

Proof. It is easy to see that

$$\begin{aligned}\sum_v (f_{LF})_v\, \phi_v(r,\theta) &= \sum_{(j,k)\in\mathcal{N}'} (f_{LF})_{(j,k)}\, \phi_{(j,k)}(r,\theta) \\ &= \sum_{j\geq 0} (f_{LF})_{(j,j)}\, \phi_{(j,j)}(r,\theta)\end{aligned}$$

if $(f_{LF})_{(j,k)} = 0$ when $j \neq k$. Since

$$\phi_{(j,k)}(r,\theta) = (j+k+1)^{1/2}\, Z_{j+k}^{|j-k|}(r)\, e^{i(j-k)\theta},$$

then

$$\phi_{(j,j)}(r,\theta) = (2j+1)^{1/2} Z_{2j}^{0}(r),$$

which is independent of θ. □

The next proposition gives us a sufficient condition for $(f_{LF})_{(j,k)} = 0$ whenever $j \neq k$, in terms of *all* the $\xi_{(j,k)}$, $j \neq k$, where $\xi_{(j,k)} = T(\phi_{(j,k)})$.

Proposition 6 *In both the discrete case (N finite) and the continuous case (N infinite),*

$$\xi_{(j,k)} = 0 \textit{ whenever } j \neq k \Rightarrow (f_{LF})_{(j,k)} = 0 \textit{ whenever } j \neq k.$$

Proof. The proof in the continuous case is easy since $(f_{LF})_{(j,k)} \propto \xi_{(j,k)}$ by equation (5.41), and $\xi_{(j,k)} = 0$ by hypothesis. The discrete case is, however, a little more difficult. Let j and k, $j \neq k$, both be fixed and consider $v = (j, k)$. Let $\mathcal{E}$ be the equivalence class which contains v. A simple piece of algebra from equation (5.48) shows that

$$(f_{LF})_{(j,k)} \propto \xi_{(j,k)} - b_{(j,k)}\,\gamma_{(j,k)}\,f(\mathcal{E})$$

where

$$f(\mathcal{E}) = \frac{C^2 \sum_{v\in\mathcal{E}} \frac{b_v \gamma_v \xi_v}{a_v^2}}{\frac{1}{n} + C^2 \sum_{v\in\mathcal{E}} \frac{b_v^2\gamma_v^2}{a_v^2}} = \frac{C^2\gamma_{\mathcal{E}}}{\frac{1}{n} + C^2\beta_{\mathcal{E}}}$$

(compare equation (5.46)). By hypothesis $\xi_{(j,k)} = 0$ since $j \neq k$. Hence $(f_{LF})_{(j,k)} \propto \gamma_{(j,k)}\,f(\mathcal{E})$. First, $\gamma_{(j,k)} = 0$ if $j + k \equiv N - 1 \pmod N$, and so we can exclude this case from possible values of (j, k), leaving us with only two possibilities: Section 5.2.13 tells us that *either* $j = j_0 + r_j N$ and $k = k_0 + r_k N$ for some $(j_0, k_0) \in \mathcal{N}_{N-1}$ and for some $r_j, r_k \geq 0$, *or* $j = N - 1 - k_0 + r_j N$ and $k = N - 1 - j_0 + r_k N$ again for some $(j_0, k_0) \in \mathcal{N}_{N-1}$ and for some $r_j, r_k \geq 0$. Given the (j_0, k_0) that defines the equivalence class $\mathcal{E}$, we must now show that it is not possible for any pair j', k', such that $(j', k') \in \mathcal{E}$, to be equal. We proceed by contradiction. Again there are two cases to consider. First, assume that $j' = k'$ where $(j', k') = (j_0 + r_j'N, k_0 + r_k'N)$ for some $r_j', r_k' \geq 0$. Then immediately j_0 and k_0 differ by a multiple of N and so can not be in $\mathcal{N}_{N-1}$. Secondly, assume that $j' = k'$ where $(j', k') = (N - 1 - k_0 + r_j'N, N - 1 - j_0 + r_k'N)$ and observe that exactly the same reasoning applies. In both cases we obtain the required contradiction.

Therefore, $\xi_v = 0$, $\forall v \in \mathcal{E}$ and so the proof is complete. □

We shall see in Sections 6.2.4 and 6.2.5 some general examples of when the least favourable function is symmetric. In Section 6.3.5 we give some particular examples of symmetric least favourable functions, as well as some asymmetric least favourable functions.

5.4 Other work on estimating linear functionals of a PET image

In this section we outline the work of Bickel and Ritov[5]. This is the only research known to us that addresses the problem of estimating linear functionals of a PET image, apart from our own work. We are very grateful to the authors of the paper for many helpful discussions and communications.

5.4.1 The work of Bickel and Ritov

One of the main points of [5] is that, in order to estimate a bounded linear functional, one does *not* need to go through the intermediate step of estimating the image. This result is, in a certain sense, contrary to what we found (see Sections 5.3.6 and 5.3.7), although the approach of [5] is not a minimax one. Bickel and Ritov[5] claim that a good image is oversmoothed for the estimation of particular properties of the image. We discuss this further in Section 5.4.3 below.

The treatment given in the paper is very technical, and we do not attempt to reproduce the details here. Rather, we try to outline the general approach. The authors assume that there are D detectors and that N_{ij} is the number of counts in the (i, j) th pair of detectors, where $1 \leq i < j \leq D$, and n is the expected number of counts. After a series of lemmas, which hold at least in the case when the image consists of a set of distinct uniform intensity discs and when the functional of interest satisfies certain smoothness assumptions, the authors arrive at what they refer to as the *fundamental expression* in which they establish the equality of

$$\mathbf{E}\left[\frac{1}{n}\sum_{i<j} N_{ij}\left(\frac{D}{2\pi}\right)^2 \int_{I_{ij}} h(\underline{\phi}, \overline{\phi})\, d\underline{\phi}\, d\overline{\phi}\right]$$

and

$$\Psi(f) - \frac{1}{3}\left(\frac{\pi}{D}\right)^2 \sum_{i<j} \int\!\!\int_{I_{i,j}} \left(\frac{\partial^2 h}{\partial \underline{\phi}^2} + \frac{\partial^2 h}{\partial \overline{\phi}^2}\right)(\underline{\phi}, \overline{\phi})\, d\underline{\phi}\, d\overline{\phi} + o(\frac{1}{D^2}),$$

where $\Psi(f)$ is the functional of interest and is defined in equation (5.62). The reader is referred to [5] for the precise definition of h: for the purposes of this discussion it suffices to say that h depends only upon the functional of interest.

The fundamental expression leads the authors to propose the estimator $\hat{\psi}$, which is defined as

$$\hat{\psi} = \frac{1}{n}\sum_{i<j} N_{ij}\left(\frac{D}{2\pi}\right)^2 \int_{I_{ij}} h(\underline{\phi}, \overline{\phi})\, d\underline{\phi}\, d\overline{\phi},$$

and to further propose a debiased $\hat{\psi}$, $\hat{\psi}_c$:

$$\hat{\psi}_c = \hat{\psi} + \frac{1}{12n}\sum_{i<j} N_{ij} \iint_{I_{ij}} \left(\frac{\partial^2 h}{\partial \underline{\phi}^2} + \frac{\partial^2 h}{\partial \overline{\phi}^2}\right)(\underline{\phi}, \overline{\phi})\, d\underline{\phi}\, d\overline{\phi}.$$

We note that these estimators are linear in the data.

The authors repeat the analysis outlined above under slightly stronger assumptions about the set of densities to produce $\hat{\hat{\psi}}$ and its debiased counterpart $\hat{\hat{\psi}}_c$, which are similar in form to $\hat{\psi}$ and $\hat{\psi}_c$. They state that they expect the debiasing to increase the variance or the estimator, and therefore study the behaviour of $\hat{\hat{\psi}}$ and $\hat{\hat{\psi}}_c$ by simulation.

5.4.2 Simulation study

The authors conduct a small simulation study in order to check the actual behaviour of the estimators that they have developed. First we give their definition of a linear functional $\Psi(f)$:

$$\Psi(f) = \iint f(x, y)\Psi(x, y)\, dx\, dy. \tag{5.62}$$

For the purposes of the simulation study the functional considered was the Gaussian

$$\Psi_G(x, y; x_p, y_p) = (2\pi\sigma^2)^{-1} \exp\left(-\frac{1}{2\sigma^2}\left[(x - x_p)^2 + (y - y_p)^2\right]\right), \tag{5.63}$$

where (x_p, y_p) is the *ccntre*. In [34] it is stated that one reason for selecting this Gaussian functional is that, although it is smoother than the indicator of a circle, it is quite close to the indicator. The phantom used was made up of a main circle of radius 1.0 and density 1.0, and 4 smaller circles of radius 0.1, positioned on the x and y axis, with centres at a distance 0.5 from centre of the main circle. The densities in these 4 smaller circles was 0.0, 2.0, 0.5 and 1.5.

The Gaussian functional given in equation (5.63) is applied at 9 points (x_p, y_p) over the unit circle, shown in Figure 2 of Bickel and Ritov[5], and 20 points equally spaced on the x axis, namely $(x_p, y_p) = 0.035\, i$, where $i = 1, \ldots, 20$. The results for the latter 20 points are presented by means of a graph, whereas the results for the former 9 are given in two tables. In both cases they consider 500 samples, each of 10^5 observations.

For the 20 points along the x axis the authors consider 256 detectors, and take $\sigma = 0.01$. The graph indicates that away from the centre of the unit circle both the original estimator and the bias-corrected estimator perform reasonably well. The bias-correction appears in all cases to reduce both variance and bias, and seems particularly effective at points away from the centre. However, near the centre of the unit circle (and indeed at the centre itself, which is one of the 9 points mentioned above) the values of both estimators were far off. Indeed, they were always negative. In [34] it is stated that the authors do not understand why this occurs.

In their two tables, the authors present the results for 8 out of the 9 points that they considered: the point $(0, 0)$ is excluded from these tables as the estimate obtained there was always so bad. For the first table there are 256 detectors and $\sigma = 0.008$, whereas for the second table there are 64 detectors and $\sigma = 0.08$. In the former case, there is very little difference between the two estimators from the point of view of bias, although this time $\hat{\hat{\psi}}_c$ seems to be better at points closer to the centre, whereas $\hat{\hat{\psi}}$ seems to be better at further points. Surprisingly, the standard deviation of the 'debiased' estimator is 70% lower than the standard deviation of $\hat{\hat{\psi}}$. In the latter case, the results are essentially the same. According to the authors, the difference between the two situations is that with 64 detectors we need a much lower sample size to make the bias and the standard deviation comparable.

5.4.3 Estimating the image density

The authors propose a method of estimating the overall density of the image. The general idea is to divide the image up into square pixels of sides ρ and to estimate the average intensity per pixel. As before this leads to an estimate $\hat{\hat{f}}$ and its debiased counterpart $\hat{\hat{f}}_c$.

In order to make a comparison with the work of Johnstone and Silverman[21], the authors consider the case when $\rho \to 0$ and $D = \infty$ (no discretization). They show that, under certain conditions on the smoothness of the density, if $\rho_n = O(n^{-1/2p+3})$, then the mean integrated square error is $O(n^{-2p/2p+3})$. This rate is faster than the $O(n^{-p/p+2})$ of [21]. However, the authors point out that there are differences between their approach and that of [21]: for example, it appears that the densities in [5] are assumed to belong to a smoothness class that is a subspace of the smoothness class used in [21].

Finally, the authors suppose that they have an estimate $\hat{f}_{\rho_n}$ of the entire image and ask whether $\Psi(\hat{f}_{\rho_n})$ is a good estimate of the functional $\Psi(f)$. (In [34] the definition of a good estimator is given. It seems that a *good estimator* is one that achieves the rate bound.) The answer to this question is given as follows:

- If $\rho = \rho_n$ is chosen to obtain the optimal rate $n^{-2p/2p+3}$, then the answer to the question is NO.
- If

  $$\rho^p \ll \frac{1}{\sqrt{n}}$$

  we can apply the functional to the reconstructed image without significantly increasing the bias. The image here is undersmoothed.

In [34] it is suggested that in order to obtain a good estimator of the picture as a whole, in the L_2 sense, say, more smoothing than would be needed to estimate a functional is required (see Section 5.4.1 also). Thus, estimating the functional from a reasonable estimate of the whole image may introduce unnecessary bias.

5.5 Extensions by Silverman and others

Silverman[38] considers the following generalization of the problem we have discussed. Suppose that $\mathcal{H}$ is a Hilbert space of functions. Write $f * g$ for the inner product of two functions in $\mathcal{H}$, and write $||f||$ for the Hilbert norm $(f * f)^{1/2}$. Suppose that $Y_1, Y_2, \dots, Y_n$ are observations of bounded linear functionals $\eta_1(f), \eta_2(f), \dots \eta_n(f)$, such that $\mathrm{E}[Y_i] = \eta_i(f)$ and $\mathrm{Var}[Y_i] = I$. (Any other fixed covariance structure can be reduced to this case by applying an appropriate linear transformation to both the Y_i and the $\eta_i(f)$.) Write Y for the n-vector with elements Y_i and $\eta(f)$ for the n-vector of functionals $\eta_i(f)$. Assume that the functional T is a bounded linear functional with respect to the Hilbert space norm and consider linear estimators of $T(f)$ of the form $w^{\mathrm{T}}Y$ for weight vectors w. Define $\mathcal{F}$ to be the unit ball in $\mathcal{H}$:

$$\mathcal{F} = \{f \in \mathcal{H} : ||f|| \leq 1\}.$$

Silverman[38] explains that any class of the form $||f||^2 \leq C^2$ can be reduced to this form by rescaling the inner prduct by a factor of C^{-2}, and, moreover, general ellipsoidal conditions on

f are coped with by working within the function space equipped with the norm generated by the quadratic form defining the ellipsoid. Silverman now makes two definitions. The first is that the estimator $\hat{w}^{\mathrm{T}}Y$ will be said to be *linear minimax for estimating T* if and only if $\hat{w}$ is the minimiser over vectors w of

$$\sup_{f\in\mathcal{F}} \mathrm{E}\,\{T(f)-w^{\mathrm{T}}Y\}^2 = \sup_{f\in\mathcal{F}}\{T(f)-w^{\mathrm{T}}\eta(f)\}^2 + w^{\mathrm{T}}w. \tag{5.64}$$

The second definition is that $\hat{f}$ is a *penalized least squares estimator of f* if

$$\hat{f} \text{ minimises } \{Y-\eta(f)\}^{\mathrm{T}}\{Y-\eta(f)\} + ||f||^2 \text{ over } f \in \mathcal{H}, \tag{5.65}$$

where $||f||^2$ can be thought of as a measure of the smoothness of the function f. The main result of Silverman[38] is given in the following theorem:

Theorem 5 (Silverman[38], Theorem 1) *Assume that T is a bounded linear functional with respect to the norm* $||\cdot||$. *The linear minimax estimator of* $T(f)$ *will be* $T(\hat{f})$, *where* $\hat{f}$ *is the penalized least squares estimator of f as defined in (5.65).*

We briefly outline the proof used by Silverman[38]. First, let $\tilde{T}$ denote the element of $\mathcal{H}$ that is the representer of the bounded linear functional T: *i.e.* $T(f) = \tilde{T} * f$ for all f in $\mathcal{H}$. Similarly let $\tilde{\eta}_i \in \mathcal{H}$ be the representer of η_i, and let $\tilde{\eta}$ be the vector of such representers. Let H be the matrix with elements $\tilde{\eta}_i * \tilde{\eta}_j$ so that

$$H = \tilde{\eta} * \tilde{\eta}^{\mathrm{T}}.$$

First, Silverman[38] established that the penalized least squares estimator defined in equation (5.65) is given by

$$\hat{f} = Y^{\mathrm{T}}(I+H)^{-1}\tilde{\eta}.$$

Next he shows that

$$\hat{w} = (I+H)^{-1}\tilde{\eta} * \tilde{T}.$$

Finally, he notes that

$$\widehat{T(f)} = \hat{w}^{\mathrm{T}} Y = Y^{\mathrm{T}} \hat{w} = Y^{\mathrm{T}}(I + H)^{-1} \tilde{\eta} * \tilde{T} = \hat{f} * \tilde{T} = T(\hat{f}).$$

Silverman[38] goes on to extend the results under mild assumptions to seminorms, motivating this by saying that the appropriate measure of smoothness is often not a norm on the function space but a seminorm. Consider, for example, functions of a single variable on an interval I. A very common measure of the smoothness of a function G is $\int_I g''(t)^2\, dt$, and this is zero for any linear function. Hence, if we define $||\cdot||_s$ by

$$||g||_s^2 = \int_I g''(t)^2\, dt,$$

we have a seminorm. For this example $\eta_i(g) = g(t_i)$ for real points t_i and the assumptions referred to hold if there are at least two distinct t_is.

Other workers have independently produced similar results. Speckman[41] considers the problem in which $Y = \Upsilon f + \varepsilon$ is observed, where f belongs to a Hilbert space and Υ is a bounded linear transformation into n-dimensional Euclidean space, and ε is a mean-zero random vector. It is known that $||\Sigma f|| \leq \alpha$ for some bounded linear transformation Σ and constant α. Linear estimates of linear functionals of f are then found which minimax the mean square error. In particular Speckman[41] supposes that the Y_i are observed with $\mathrm{E}[Y_i] = f(x_i)$, where f is assumed to have absolutely continuous first derivatives and square integrable second derivatives on an interval I with

$$\int_I f''(x)^2\, dx \leq \alpha^2.$$

He considers the problem of estimating $f(x_0)$ for some fixed x_0 and establishes that the minimax estimate is a cubic smoothing spline, *i.e.* the function $\hat{f}$ that minimizes

$$\sum_{i=1}^{n} (Y_i - f(x_i))^2 + \frac{\sigma^2}{\alpha^2} \int_I f''(x)^2\, dx,$$

evaluated at x_0. Also $\hat{f}'(x_0)$ is the minimax estimate of $f'(x_0)$. Li[27] states that 'a minimax linear estimator for any bounded linear functional can be derived by ...operating the linear functional on the smoothing spline $\hat{f}$.' He refers to this as the method of regularization. His proof follows a variation of Speckman's approach and is similar to that presented by

Silverman[38], although not so immediate.

5.6 Conclusions

In this chapter we have considered the minimax estimation of linear functionals in the case of positron emission tomography, both in the idealised case when the ring of detectors is considered to be continuous, and in the more realistic case when the ring comprises a finite number N of detectors.

We have derived minimax estimators for linear functionals, and given the form of the minimax risk and the least favourable function in both the continuous and discrete cases. We have derived a sufficient condition for radial symmetry of the least favourable function. We have shown that if we write the estimate of the functional as $\widehat{T(f)}$, then $\widehat{T(f)} = T(\hat{f})$ where $\hat{f}$ does not depend upon the functional T, but minimizes a penalized least squares expression. In Chapter 6 we illustrate the theory of this chapter by presenting some numerical examples.

Chapter 6

Estimating Linear Functionals of a PET Image: Some Numerical Examples

6.1 Introduction

In this chapter we illustrate some of the theory we developed in Chapter 5. In Section 6.2 we introduce two particular linear functionals, and discuss some relevant properties and computational issues. In Section 6.3 we present the results of a numerical investigation carried out on these two functionals. We discuss the minimax risks that we obtained, and introduce the notion of efficiency. After drawing some conclusions that are connected with the construction of PET machines and the design of the experiment, we present a few examples of the least favourable function. Finally, in Section 6.4 we outline our conclusions.

6.2 Numerical examples and calculations

In this section we introduce two linear functionals. After briefly discussing a few of their properties, we consider the implementation of some of the theory developed in Chapter 5. In particular, we concern ourselves with the computation of the minimax risks and the least favourable function.

6.2.1 Two linear functionals

In this chapter we consider two types of linear functionals: the first type we denote as $T_{(x,s)}$, where

$$T_{(x,s)}(f) = \frac{\int_{D(x;s)} f(y)\, d\mu(y)}{\int_{D(x;s)} d\mu(y)}, \tag{6.1}$$

and $D(x; s)$ represents a region of the plane which is the intersection of the disc centre x, radius s, with the unit circle; the second type we denote as T_x, where

$$T_x(f) = f(x). \tag{6.2}$$

The functional $T_{(x,s)}(f)$ represents the 'average' of f over the above stated region, whereas $T_x(f)$ is the value of f at the point x. There is a relationship between these two functionals which we give in Proposition 7.

Proposition 7

$$\lim_{s \to 0} T_{(x,s)}(f) = T_x(f)$$

Proof. The proof is straightforward.

$$\begin{aligned}
T_{(x,s)}(f) - T_x(f) &= \frac{\int_{D(x;s)} f(y)\, d\mu(y)}{\int_{D(x;s)} d\mu(y)} - f(x) \\
&= \frac{\int_{D(x;s)} f(y)\, d\mu(y) - f(x) \int_{D(x;s)} d\mu(y)}{\int_{D(x;s)} d\mu(y)} \\
&= \frac{\int_{D(x;s)} (f(y) - f(x))\, d\mu(y)}{\int_{D(x;s)} d\mu(y)}.
\end{aligned}$$

Hence,

$$\begin{aligned}
|T_{(x,s)}(f) - T_x(f)| &= \left| \frac{\int_{D(x;s)} (f(y) - f(x))\, d\mu(y)}{\int_{D(x;s)} d\mu(y)} \right| \\
&\leq \sup_{y \in D(x;s)} |f(y) - f(x)| \\
&\to 0, \text{as } s \to 0
\end{aligned}$$

since f is assumed to be continuous. □

6.2.2 Norms

It is not difficult to show directly from the definition that, for the supremum norm $||\cdot||$ introduced in Section 5.3.1,

$$||T_{(x,s)}|| = \frac{1}{\int_{D(x;s)} d\mu(x)} = \frac{1}{\mu(D(x;s))}.$$

Accordingly, for $s > 0$, $T_{(x,s)}$ is a bounded linear functional. Therefore, we can apply Proposition 1 and con:lude that inequality (5.34), namely

$$\sum_{\nu} \frac{|\xi_\nu|^2}{a_\nu^2} < \infty,$$

holds. However, since $\lim_{s\to 0} \mu(D(x;s)) = 0$ and, by Proposition 7, $\lim_{s\to 0} T_{(x,s)}(f) = T_x(f)$, we have that T_x is unbounded with respect to $||\cdot||$. Nevertheless, inequality (5.34) still holds for $a > 1$, by Proposition 2.

6.2.3 Zernike polynomials

We have seen that the set of functions $\{\phi_\nu\}$ form an orthonormal system on H, where ϕ_ν is defined in equation (5.14) in terms of Zernike polynomials. These polynomials are discussed in detail in [6], which also presents a table showing some explicitly. In this study it is necessary to be able to calculate the value of any given Zernike polynomials, $Z_n^m(\rho)$, where $m \geq 0$, $n \geq m$ and $n - m$ is even, at any given ρ, $0 \leq \rho \leq 1$, both quickly and accurately. There are two cases to consider.

Case 1: $m = 0$. If we write $n = 2j$, we have

$$Z_{2j}^0(\rho) = Z_j(2\rho^2 - 1),$$

where Z_j is the Legendre polynomial of order j (see [25]). There are many poor ways to evaluate Legendre polynomials numerically. It seems that any method involving the computation of factorials is unsatisfactory. However, these polynomials can be computed easily and accurately by means of the following relationships:

$$\begin{aligned} Z_0(t) &= 1 \\ Z_1(t) &= t, \end{aligned}$$

and the recurrence relationship

$$(2j+1)\,t\,Z_j(t) = (j+1)\,Z_{j+1}(t) + j\,Z_{j-1}(t),\ j = 1, 2, \dots .$$

Integrals of Zernike polynomials can also be computed recursively in several different ways. We give one example. It is a well known result that

$$\frac{d}{dt}\left[(1-t^2)\,Z_j'(t)\right] + j\,(j-1)\,Z_j(t) = 0$$

where the prime $'$ denotes differentiation with respect to t. Hence

$$\int_a^b Z_j(t)\,dt = -\frac{1}{j\,(j-1)}[(1-t^2)\,Z_j'(t)]\,\Big|_a^b .$$

Fortunately,

$$(1-t^2)\,Z_j'(t) = j\,Z_{j-1}(t) - j\,t\,Z_j(t),$$

where Z_j and Z_{j-1} can be computed in the way described above.

Case 2: $m \neq 0$. This case is much more difficult. First, we derive recurrence relationships between certain Zernike polynomials from recurrence relationships between polynomials $Q_{n,k}(t)$, where

$$Q_{n,k}(t) = \frac{1}{k!\,t^n}\frac{d^k}{dt^k}\{t^{n+k}\,(t-1)^k\},$$

as defined in equation (A.2) of [28]. The recurrence relationships that we use for the Q polynomials are equations (A.12) and (A.13) of [28]:

$$Q_{0,k+1} = 2t\,Q_{1,k} - Q_{0,k} \tag{6.3}$$

$$Q_{n+1,k+1} = Q_{n,k+1} + \frac{k+1}{n+k+2}[Q_{n,k+1} - Q_{n+1,k}]. \tag{6.4}$$

It is easy to show that

$$Z_{m+2s}^m(\rho) = t^{\frac{m}{2}}\,Q_{m,s}(t)\,\Big|_{t=\rho^2} .$$

Hence, we obtain from equation (6.3)

$$Z^0_{2(k+1)}(\rho) = 2\rho Z^1_{1+2k}(\rho) - Z^0_{2k}(\rho),$$

or

$$Z^0_n(\rho) = 2\rho Z^1_{n-1}(\rho) - Z^0_{n-2}(\rho) \ n \text{ even},$$

and from equation (6.4)

$$Z^{n+1}_{n+1+2(k+1)}(\rho) = \rho Z^n_{n+2(k+1)}(\rho) + \frac{k+1}{n+k+2}[\rho Z^n_{n+2(k+1)}(\rho) - Z^{n+1}_{n+1+2k}(\rho)],$$

or

$$Z^m_n(\rho) = \rho Z^{m-1}_{n-1}(\rho) + \frac{n-m}{n+m}[\rho Z^{m-1}_{n-1}(\rho) - Z^m_{n-2}(\rho)].$$

Finally, observing that $Z^m_n(\rho) = \rho^m$, when $m = n$, we can compute the Zernike polynomials according to Table 6.1. To compute $Z^m_n(\rho)$ we adopt the following algorithm:

- Check that $m + n$ is even;
- Set $m + n = 2l$;
- Compute, in the order of Table 6.1, $Z^0_0(\rho), \dots, Z^0_{2(l-1)}, Z^l_l(\rho)$;
- Compute, in the order of Table 6.1, up to $Z^m_n(\rho)$, if necessary.

6.2.4 The integral of ϕ_v over a disc

For the purposes of evaluating the minimax risk when estimating $T_{(x,s)}$, we must be able to calculate $\int_{D(x;s)} \phi_v = \int_{D(x;s)} \phi_{(j,k)}$, where ϕ_v is defined in equation (5.14). We consider only points x on the $\theta = 0$ axis. The reason for doing this is discussed in Section 6.3.1.

Let x be the point on the $\theta = 0$ axis a distance R from the origin (in Cartesian coordinates x is the point $(R, 0)$). Let (x_1, y_1) be any point inside the disc. Let l be the line joining the point $(R, 0)$ to the point (x_1, y_1). Set ρ to be the length of l and let ϕ be the angle between l and the x axis. Thus, (ρ, ϕ) are polar coordinates for the point (x_1, y_1) with respect to the the centre of the disc and the x axis. The definitions of ρ and ϕ can be seen in Figure 6.1.

m	n
0	**0**
1	**1**
0	2
2	**2**
1	3
0	4
3	**3**
2	4
1	5
0	6
$\vdots$	$\vdots$
i	i
$i-1$	$i+1$
$\vdots$	$\vdots$
1	$2i-1$
0	$2i$
$\vdots$	$\vdots$

Table 6.1: *The order in which the Zernike polynomials were computed*

Now we find the point (x_1, y_1) in the original coordinates (*i.e.* with respect to the centre of brain space B and the $\theta = 0$ axis) by some elementary trigonometry:

$$\begin{aligned} x_1 &= R + \rho\cos\phi \\ y_1 &= \rho\sin\phi. \end{aligned} \tag{6.5}$$

Let us now turn our attention to the integral $\int_{D(x;s)} \phi_{(j,k)}$. Since $\phi_{(j,k)}(r,\theta) = f(r)g(\theta)$, where $f(r) = (j+k+1)^{1/2} Z_{j+k}^{|j-k|}(r)$ and $g(\theta) = e^{i(j-k)\theta}$, then this integral can be written as

$$\int_{D(x;s)} f\left(\sqrt{x_1^2 + y_1^2}\right) g\left(\tan^{-1}(y_1/x_1)\right)\, dx_1\, dy_1,$$

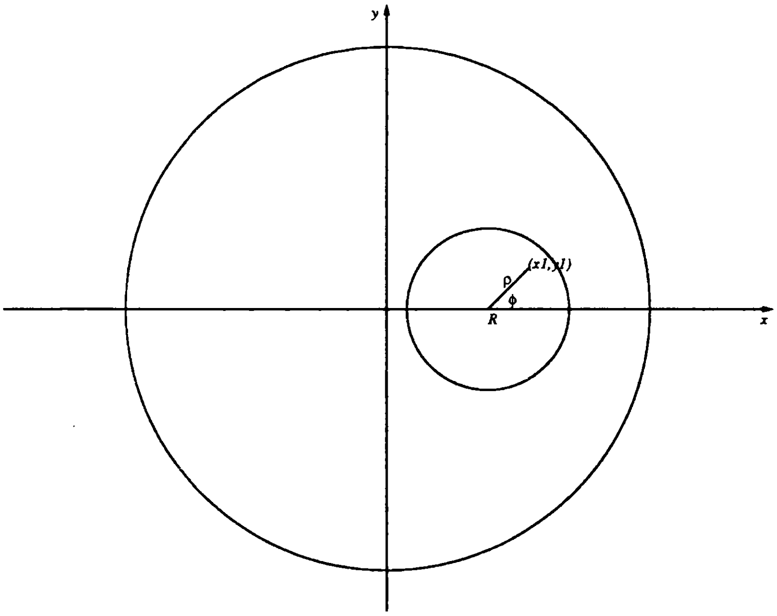


Figure 6.1: *Diagram of the trigonometry of the set-up*

where the point (x_1, y_1) lies within the disc $D(x; s)$. We shall refer to this integral as I. Using the relationships (6.5), we now transform the variables of integration form (x_1, y_1) to (ρ, ϕ) to get that

$$I = \int_0^{2\pi}\int_0^a f\left(\sqrt{R^2 + 2\rho R\cos\phi + \rho^2}\right) g\left(\tan^{-1}(\frac{\rho\sin\phi}{R + \rho\cos\phi})\right) \rho\, d\rho\, d\phi,$$

a quantity that in general we compute by numerical integration.

However, in the case when the disc is centred at the origin, we have $R = 0$ and obtain the following equation

$$\begin{aligned} I &= \int_0^{2\pi}\int_0^a f(\rho)g(\phi)\,\rho\, d\rho\, d\phi \\ &= \int_0^a f(\rho)\,\rho\, d\rho \int_0^{2\pi} g(\phi)\, d\phi. \end{aligned}$$

Now

$$\int_0^{2\pi} g(\phi)\, d\phi = \int_0^{2\pi} e^{i\,(j-k)\phi}\, d\phi,$$

which is zero, unless $j = k$, in which case

$$\begin{aligned}\int_0^a f(\rho)\rho\, d\rho &= (2j+1)^{1/2}\int_0^a Z_{2j}^0(\rho)\rho\, d\rho \\ &= (2j+1)^{1/2}\int_0^a Z_j(2\rho^2-1)\rho\, d\rho \\ &= \frac{(2j+1)^{1/2}}{4}\int_{-1}^{2a^2-1} Z_j(u)\, du.\end{aligned}$$

The evaluation of this last integral can be done recursively, as described above, and this can be used to check our numerical integration routines.

We emphasize that in the case when $R = 0$, $\xi_{(j,k)} = 0$, unless $j = k$. Thus, by Propositions 6 and 5, the least favourable function will be radially symmetric in both the continuous and discrete case.

Finally in this section we state and prove a useful proposition that is employed in Section 6.2.6.

Proposition 8 *The integral*

$$\int_{D(x;s)} f\left(\sqrt{x_1^2+y_1^2}\right) g\left(\tan^{-1}(y_1/x_1)\right)\, dx_1\, dy_1,$$

where $f(r) = (j+k+1)^{1/2} Z_{j+k}^{|j-k|}(r)$ *and* $g(\theta) = e^{i(j-k)\theta}$ *is real.*

Proof. The integral is clearly real when $j = k$. Let $j \neq k$ and change the order of integration:

$$\int_{D(x;s)} f\left(\sqrt{x_1^2+y_1^2}\right) g\left(\tan^{-1}(y_1/x_1)\right)\, dy_1\, dx_1.$$

For fixed x_1 in the appropriate range the inner integral can be written as

$$\int_{-y_1(x_1)}^{y_1(x_1)} f\left(\sqrt{x_1^2+y_1^2}\right) g\left(\tan^{-1}(y_1/x_1)\right)\, dy_1,$$

since the x axis goes through the centre of the disc. Consider the imaginary part

$$\int_{-y_1(x_1)}^{y_1(x_1)} f\left(\sqrt{x_1^2+y_1^2}\right) \sin\left((j-k)\tan^{-1}(y_1/x_1)\right)\, dy_1. \tag{6.6}$$

A simple argument shows that the function $f(\sqrt{x_1^2+y_1^2})$ is an even function of y_1. On the other hand, if we now note that $\tan^{-1}(-z) = -\tan^{-1}(z)$, then we see that $\sin((j-k)\tan^{-1}(y_1/x_1))$ is an odd function of y_1. Thus the integrand of the integral given in expression (6.6) is an odd function of y_1 and therefore that integral is 0. Since x_1 was an arbitrary element of the appropriate range, we obtain the required result. □

The proposition tells us that $T_{(x,a)}(\phi_\nu) = \xi_\nu$ is real for all ν.

6.2.5 The evaluation of ϕ_ν at a point

In this section we consider the evaluation of ϕ_ν at a point which lies along the x axis (see Section 6.3.1). Accordingly, we need to be able to evaluate $\phi_\nu(r, \theta)$, when $\theta = 0$. This quantity will be equal to $(j+k+1)^{1/2} Z_{j+k}^{|j-k|}(r)$, and is clearly real. Moreover, if the point of interest is the origin, then $r = 0$. Now

$$Z_{j+k}^{|j-k|}(0) = \begin{cases} (-1)^j & \text{if } j = k \\ 0 & \text{otherwise.} \end{cases}$$

Hence,

$$\phi_{(j,k)} = \begin{cases} (-1)^j (2j+1)^{1/2} & \text{if } j = k \\ 0 & \text{otherwise} \end{cases}$$

at the origin. Again we emphasize that in the case when $r = 0$, $\xi_{(j,k)} = 0$, unless $j = k$, and thus, by Propositions 6 and 5, the least favourable function will be radially symmetric in both the continuous and discrete case.

We can evaluate ϕ_ν at any other point using the recurrence relations discussed above.

6.2.6 The computation of the least favourable function

To find the least favourable function we need to compute

$$\sum_\nu (f_{LF})_\nu \, \phi_\nu(r, \theta)$$

over an appropriate region of r and θ within the unit circle defined by B. In general ϕ_ν is complex. However, we work with *real* densities f. Accordingly, as we saw in Section 5.2.5, we may identify this complex basis with an equivalent real orthonormal basis in a standard fashion. To do this we write $f = \sum_v f_v \phi_v = \sum_v \tilde{f}_v \tilde{\phi}_v$, where

$$\tilde{\phi}_{(j,k)} = \begin{cases} \sqrt{2}\,\mathrm{Re}\left(\phi_{(j,k)}\right) & \text{if } j > k \\ \phi_{(j,j)} & \text{if } j = k \\ \sqrt{2}\,\mathrm{Im}\left(\phi_{(j,k)}\right) & \text{if } j < k, \end{cases}$$

or

$$\tilde{\phi}_{(j,k)}(r,\theta) = \begin{cases} \sqrt{2}\,(j+k+1)^{1/2}\, Z^{|j-k|}_{j+k}(r)\, \cos((j-k)\theta) & \text{if } j > k \\ (2j+1)^{1/2}\, Z^{0}_{2j}(r) \qquad\qquad Z^{|j-k|}_{i+k}(r) & \text{if } j = k \\ \sqrt{2}\,(j+k+1)^{1/2}\, Z^{|j-k|}_{j+k}(r)\, \sin((j-k)\theta) & \text{if } j < k, \end{cases}$$

and

$$\tilde{f}_{(j,k)} = \begin{cases} \sqrt{2}\,\mathrm{Re}(f_{(j,k)}) & \text{if } j > k \\ f_{(j,j)} & \text{if } j = k \\ -\sqrt{2}\,\mathrm{Im}(f_{(j,k)}) & \text{if } j < k. \end{cases}$$

We have seen in Proposition 8 and in Section 6.2.5 that in the two cases considered ξ_v is real for all v. Equation (5.41) in the continuous case and equations (5.46) and (5.48) in the discrete case immediately give us that $(f_{LF})_v$ is real for all v. Accordingly, the calculation of the least favourable functional is simplified to the computation of

$$2\sum_{j>k} \mathrm{Re}(f_{(j,k)})\,\mathrm{Re}(\phi_{(j,k)}) + \sum_{j=k} f_{(j,k)}\,\phi_{(j,k)}$$

where $\mathrm{Re}(\phi_{(j,k)}(r,\theta)) = (j+k+1)^{1/2} Z^{|j-k|}_{j+k}(r,\theta)\cos((j-k)\,\theta)$. Moreover, the least favourable function can be seen to be symmetric in θ. This means that in the cases that we consider the least favourable function is symmetric about the $\theta = 0$ axis. This is not surprising, given the general set up.

6.3 Results obtained

In this section we present some of our findings. First, we briefly outline the investigation. After a discussion of the minimax risks themselves, we introduce the notion of efficiency as a meaningful way of interpreting the minimax risks. We make some comments about the implication of our findings for the design of experiments and machines, before computing some least favourable functions. This last part provides an illustration of one use of the contouring package CONICON (see [36]).

6.3.1 Outline of the investigation

In this section we investigate the estimation of two different linear functionals of the density f. These were first introduced in Section 6.2.1. The first, defined in equation (6.1), is denoted by $T_{(x,s)}$, where

$$T_{(x,s)}(f) = \frac{\int_{D(x;s)} f(y)d\mu(y)}{\int_{D(x;s)} d\mu(y)},$$

and is the 'average' of f over the disc $D(x; s)$. The second, defined in equation (6.2), is denoted by T_x, where

$$T_x(f) = f(x),$$

and is the value of the density f at the point x. In Proposition 7 we showed that if we take the limit of the first functional as the radius $s \to \infty$, we obtain the second functional.

In this section we consider three different xs. In polar coordinates (r, θ) relative to the centre of brain space B and the x axis, these are $(0.0, \theta)$, $(0.4, \theta)$ and $(0.7, \theta)$. We take $\theta = 0.0$, *i.e.* x is constrained to lie along the x axis, in Cartesian coordinates. In the continuous case we can do this completely without loss of generality as the choice of x axis is arbitrary. In the discrete case, however, the situation is a little different. We suppose that the unit circle is divided into an even number N of detectors of equal size, the intervals having polar angular coordinates $(2\pi d/N, 2\pi(d+1)/N)$ for $d = 0, 1, \ldots, N-1$. The problem of estimating the integral of f over a disc centre (r, θ), or the value of f at this point is equivalent to the same problem, but with θ replaced by θ_1, where $\theta_1 \equiv \theta \pmod{2\pi/N}$ and $0.0 \le \theta_1 \le 2\pi/N$. Thus, without loss of generality, we may take our original $\theta \in (0, 2\pi/N)$. Moreover, in this sector the problem is symmetric in θ about $\theta = \pi/N$. Accordingly, a more thorough investigation would consider various θs in the range 0.0 to π/N, and this could be an area for further work. It is not, however, expected that there will be much variation in the results for θ in this range, and so in this work we consider only $\theta = 0.0$.

Now that we have selected the three different vectors x to use for both types of functional described above, all that remains is for us to state the different values of the radius s that we

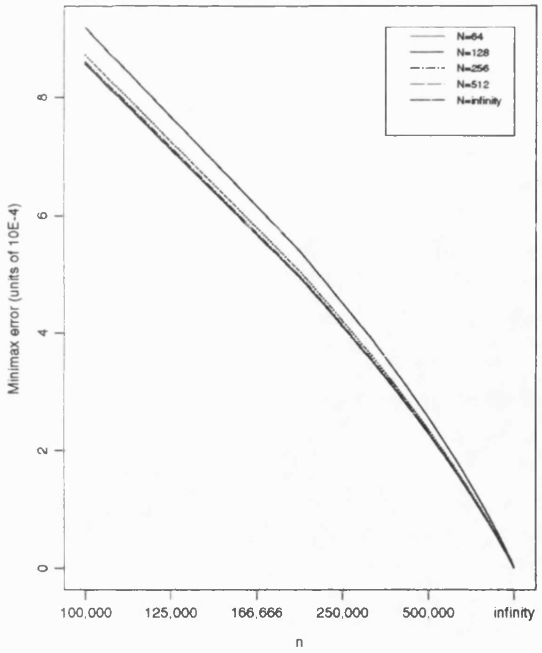


Figure 6.2: *The minimax risk for various numbers of detectors N for n* $> 100,000$

take in equation (6.1). In this work we consider $s = 0.1$ and $s = 0.3$. Overall this gives us 9 cases to consider.

6.3.2 Minimax risks

We computed the minimax risks as a function of the expected number of emissions n for various values of the number of detectors N. The values that we consider in this section are $N = 64$, 128, 256, 512 and $N = \infty$, the last corresponding to the continuous case as was mentioned in Section 5.3.8. Throughout this section we set $a = 1.0$ and $C = \sqrt{2}$, the upper bound on C, as calculated from equation (5.21), required to ensure that all density functions in the class $\mathcal{F}$ are nonnegative.

We present the results of the calculations of the minimax risks themselves in only one case: that of estimating the functional $T_{(0,0.1)}(f)$. Figure 6.2 shows the minimax risks for each of the above values of N, for n ranging from 100,000 to ∞. We can display the minimax risks for $n = \infty$ by plotting on the x axis $1 - 100,000/n$, which means that $n = \infty$ corresponds to $x = 1$. Because of this, the interval just before $x = 1$ covers an extremely large range of values of n.

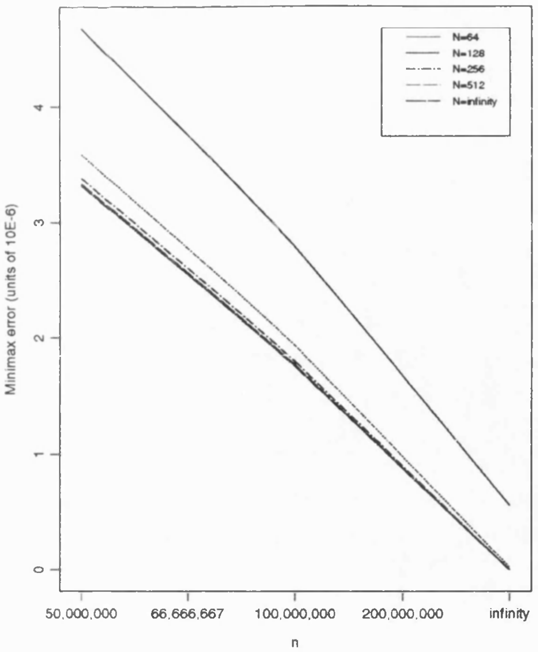


Figure 6.3: *The minimax risk for various numbers of detectors N for n* > 50, 000, 000

It can clearly be seen that these minimax risks are decreasing functions of n, as was proved in Sections 5.3.5 and 5.3.7. Figure 6.3 shows an enlargement of the x interval [0.998, 1.0]. Here the minimax risks seem to be straight lines (*i.e.* the minimax risks seems to decreases at a rate proportional to $1/n$) as was suggested by the asymptotic expansion (5.59) given in Section 5.3.10 (although the meaning of that expansion is questionable).

We now make three remarks about these graphs. The first is a direct consequence of the result found in Section 5.3.9, where we showed that, for a fixed n, if we doubled the number of detectors N, the minimax risk decreased. This can be seen clearly for both graphs as the $N = 64$ curve lies uniformly above the $N = 128$ curve, and so on.

The second remark concerns the difference between the curves for finite N, and the curve for the continuous case ($N = \infty$). What is clear from both graphs, but especially from the second, is that the minimax risks for $N = 256, 512$ and the continuous case ($N = \infty$) are almost identical, whereas there is a noticeable (although not large) distance between the $N = 64$ and $N = 128$ curves, and the curve for the continuous case. This observation, which has potentially important implications for the design of these PET machines, will be quantified and discussed further in Section 6.3.3 below.

The third remark concerns the behaviour of the curves as $n \to \infty$. As described in Sections 5.3.8 and 5.3.10, the limit of the minimax risk as $n \to \infty$ is a positive quantity when the number of detectors N is finite and zero when N is infinite. This result can be clearly seen from an examination of the second graph at $x = 1$ ($n = \infty$). (The values of the minimax risk given here tie up with the coefficients of the constant terms given in Table 5.3 in Section 5.3.10.)

The discussion in this section has concentrated on the minimax risks. However, these quantities are not very meaningful in themselves. In Section 6.3.3 we present a possible way of quantifying the information contained in these curves, and the results of doing this for all of the 9 cases outlined in Section 6.3.1. We also discuss further the implication of these results for machine design.

6.3.3 Efficiency

In this section we present a meaningful way of interpreting the minimax risks by means of a quantity that we shall refer to as efficiency. We now outline the basic idea; the quantities that we shall use are illustrated in Figure 6.4. First we assume that with a fixed number N of detectors and a fixed expected number n of emissions we achieve a minimax risk of $r(n, N)$. We then compute $n^* = n^*(n, N)$, the expected number of emissions necessary to achieve the same or better (*i.e.* lower) value $r(n, N)$ of the minimax risk in the continuous case. Clearly, $n^* \leq n$ as the class of estimators when N is finite is a subset of the class of estimators in the continuous case (compare the argument of Section 5.3.9). We now define the *efficiency* $\mathrm{eff}(n, N)$ (when there are N detectors *relative to* the continuous case) as the ratio of n to n^*:

$$\mathrm{eff}(n, N) = \frac{n^*}{n}. \tag{6.7}$$

Immediately we have that $0 \leq \mathrm{eff}(n, N) \leq 1$, with $\mathrm{eff}(n, \infty) = 1$ for all values of n. Also, since the results of Section 5.3.9 give us that $n^*(n, N) \leq n^*(n, 2N)$, we can see that $\mathrm{eff}(n, N) \leq \mathrm{eff}(n, 2N)$, $\forall n, N$. Moreover, we present the following proposition about the behaviour of $\mathrm{eff}(n, N)$ as $n \to \infty$.

Proposition 9 *For finite N,*

$$\lim_{n \to \infty} \mathrm{eff}(n, N) = 0$$

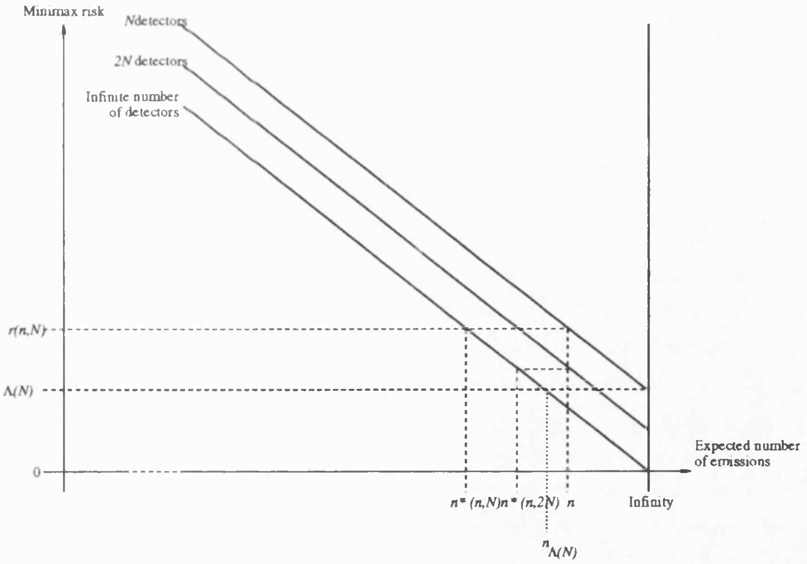


Figure 6.4: *Diagram to explain that* $\lim_{n\to\infty} \text{eff}\,(n, N) = 0$

Proof. This proof is based on Figure 6.4. Let the number of detectors N be fixed, but finite. We established in Section 5.3.8 that the limit of the minimax risk as $n \longrightarrow \infty$ is strictly positive when N is finite. Moreover, because the minimax risk is strictly decreasing in n (see Section 5.3.7 for this result in the case when N is finite (and Section 5.3.5 for the infinite case)), this limit is approached from above. In this proof we shall refer to the limit when there are N detectors as $\Lambda\,(N)$. Since $r\,(n, \infty) \leq r\,(n, N)$, $\forall n$ and since the limit of the minimax risk as $n \longrightarrow \infty$ is exactly zero in the continuous case, there exists a finite $n_{\Lambda\,(N)}$ such that $\forall n > n_{\Lambda\,(N)}$, $r\,(n, \infty) \leq \Lambda\,(N)$. Now let $n > n_{\Lambda\,(N)}$ be arbitrary. Since $r\,(n, N) > \Lambda\,(N)$, the number of observations required in the continuous case to achieve a risk of $r\,(n, N)$ or lower, namely $n^* = n^*(n, N)$, is clearly such that $n^* < n_{\Lambda\,(N)}$. Thus,

$$\text{eff}\,(n, N) = \frac{n^*}{n} \leq \frac{n_{\Lambda\,(N)}}{n},$$

for all $n > n_{\Lambda\,(N)}$. The right hand side of this inequality tends to zero from above as $n \longrightarrow \infty$. Hence, the result is established. □

As we shall see below, in practical situations high efficiencies can often be obtained, even though eff $(n, N) \longrightarrow 0$ as $n \longrightarrow \infty$. We also note that $n_{\Lambda\,(N)}$ is an increasing function of N as $\Lambda\,(N)$ decreases as N increases (and in fact $n_{\Lambda\,(N)} \longrightarrow \infty$ as $N \longrightarrow \infty$).

In Figure 6.5 we present plots of the efficiency for the 9 cases described in Section 6.3.1. In all 9 cases we consider values of n from 5 million to 50 million, and N = 64, 128 and 256, as well as the continuous case. We take these values to illustrate the method and we do not necessarily intend to suggest that they are more representative of reality than any other values. For this reason we present a qualitative description of the graphs. In each graph there are four lines: the top line corresponds to the continuous case, the next line corresponds to the $N = 256$ case, the third line corresponds to the $N = 128$ case and the bottom line corresponds to the $N = 64$ case. The three columns of the graph represent the three points under consideration, namely $x = (0.0, 0.0)$, $(0.4, 0.0)$ and $(0.7, 0.0)$. The top row gives the efficiencies for estimating the functional T_x, the second row gives the efficiencies for estimating the functional $T_{(x,0.1)}$, and the third row gives the efficiencies for estimating the functional $T_{(x,0.3)}$. From the graphs we can see easily the effect of discretization. In the first row the efficiency when $N = 256$ is near to 1.0 and seems to be fairly constant over the range of n considered. Moreover, with this functional the position of the point at which we are trying to estimate the density does not seem to have much effect on the efficiency. Thus, it seems that when $N = 256$, there is not much to choose between the discrete case and the continuous case. When $N = 128$ the efficiency is again almost constant at over 0.9, although it does seem that as the point moves away from the centre of the circle, the efficiency decreases slightly. This last feature is more clearly visible in the $N = 64$ case. We note also that in the $N = 64$ case, the efficiency noticeably decreases as n increases. In general, similar comments can be made for the middle row, which concerns the functional $T_{(x,0.1)}$, except that here the efficiency seems a little higher when compared to the top row. Finally, the third row, which concerns the functional $T_{(x,0.3)}$ displays a marked decline in efficiency as the disc moves from the centre of brain space, B. Moreover, the effect of discretization is clearly visible in the case when $x = (0.7, 0.0)$.

6.3.4 Implications for the design of experiments and machines

In this section we assume that it is of interest to discover the average intensity in a certain region of brain space. (In the above work we considered only circular regions, but the theory developed covers regions of any shape, and the extension of the FORTRAN programs to more general regions, possibly defined by the user with the mouse on the screen, is a topic for further work.) From the graphs of the minimax risk the experimenter can assess the effect of, for example, doubling the expected number of emissions n. In addition, the effect of doubling the

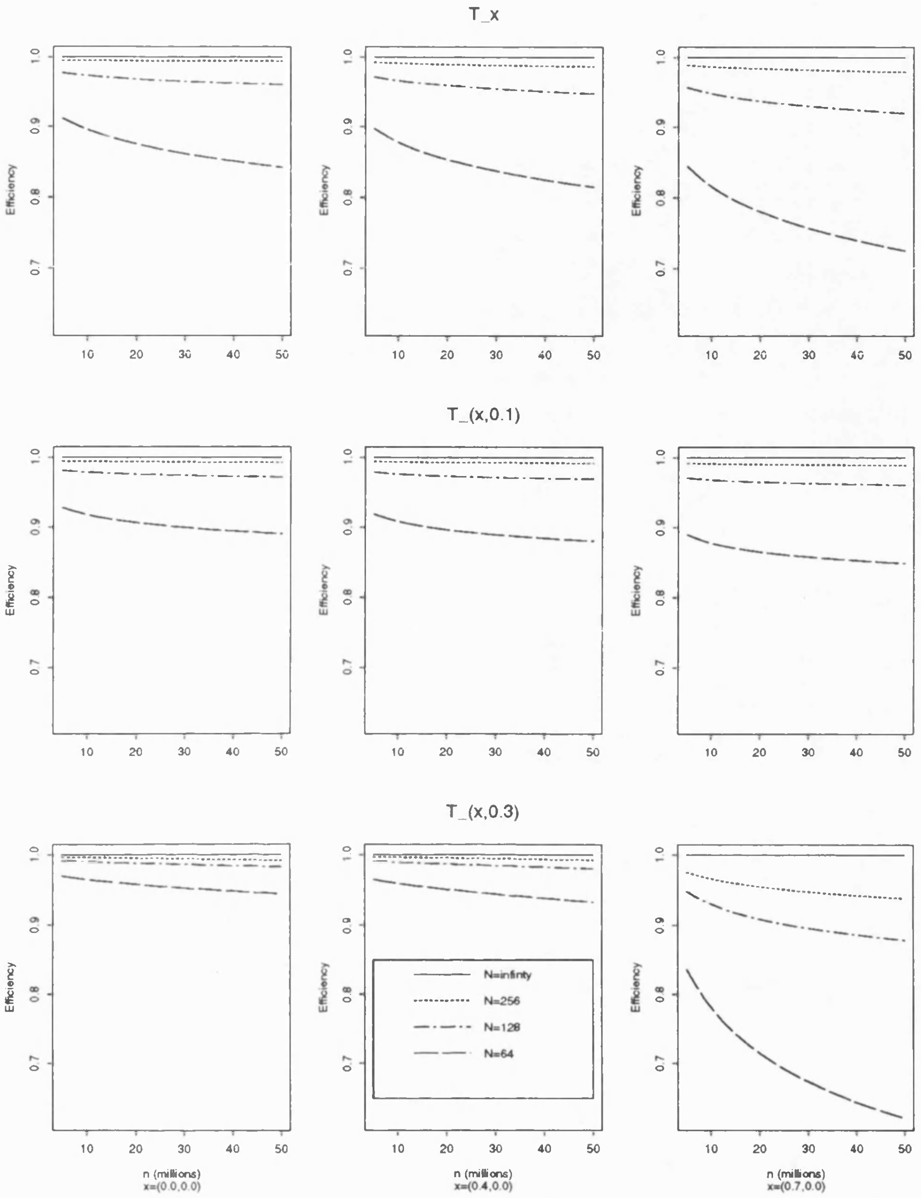


Figure 6.5: *The efficiency of estimating the functionals* T_x, $T_{(x,0.1)}$ *and* $T_{(x,0.3)}$ *at* $x = (0.0, 0.0)$, $(0.4, 0.0)$ *and* $(0.7, 0.0)$

number of detectors N can also be assessed. In the example given in Figure 6.2 and Figure 6.3 for instance, doubling the expected number of emissions may noticeably reduce the minimax risk, whereas doubling the number of detectors beyond a certain point may achieve very little. Such statements clearly have implications for the design of experiments and machines.

6.3.5 Least favourable functions

In this section we present some examples of the least favourable functions. The forms of the least favourable function are given in equation (5.41) in the continuous case and equation (5.48) in the discrete case. There, and in this section, we disregard the first term of the expansion, namely 1. Accordingly, the integral of the functions we show in this section, with respect to the measure μ over the unit circle, is zero rather than one. The ± sign tells us that an equivalent least favourable function can be found by reflection in the appropriate axis. All the examples in this section are presented in such a way that the value of the function at the point of interest, x, is positive.

In Section 5.3.11 we considered the symmetry of the least favourable function and in Proposition 6 we gave a sufficient condition for the least favourable function to be radially symmetric. In Section 6.2.4 we showed that the least favourable function for estimating $T_{(x,s)}$ is symmetric if $x = 0$. Similarly, in Section 6.2.5 we showed that the least favourable function for estimating T_x is symmetric again if $x = 0$. In this discussion we consider examples of both symmetric and asymmetric least favourable functions. In all the examples that we give the least favourable function is symmetric about the x axis, as we established in Section 6.2.6.

First, we give examples and a qualitative description of symmetric least favourable functions. In all the cases that we shall present the (expected) number of emissions n is 10,000,000. We set $a = 1.5$ and $C = 2.0$. Again we attach no special significance to these examples. Their purpose is to illustrate the theory we have developed above. The first graph of Figure 6.6 shows the least favourable function for estimating both $T_{(0,0.1)}$ and T_0 in the continuous case ($N = \infty$), the middle graph presents the same functions in the case when there are 128 detectors ($N = 128$), and the bottom graph deals with the case when there are 32 detectors ($N = 32$). Because of the radial symmetry the functions are shown for only a region of the positive x axis (they remain almost indistinguishable from zero after about $x = 0.4$). The function itself is the surface of revolution obtained by rotating these curves about the vertical axis. There is clearly some difference between the graphs: the range of the curves is greater

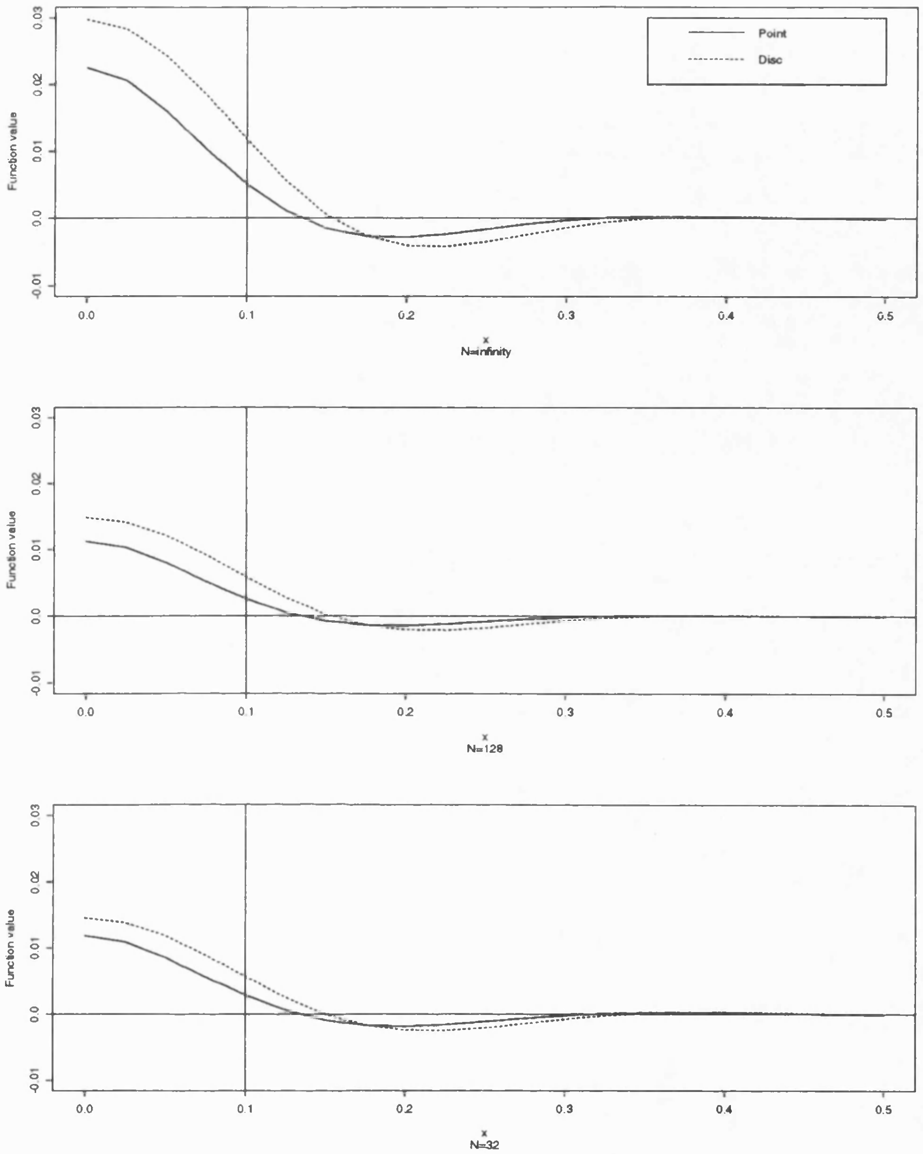


Figure 6.6: *Radially symmetric least favourable functions for the functionals* $T_{(0,0.1)}$ *and* T_0*, when there are 32 and 128 detectors, and in the continuous case*

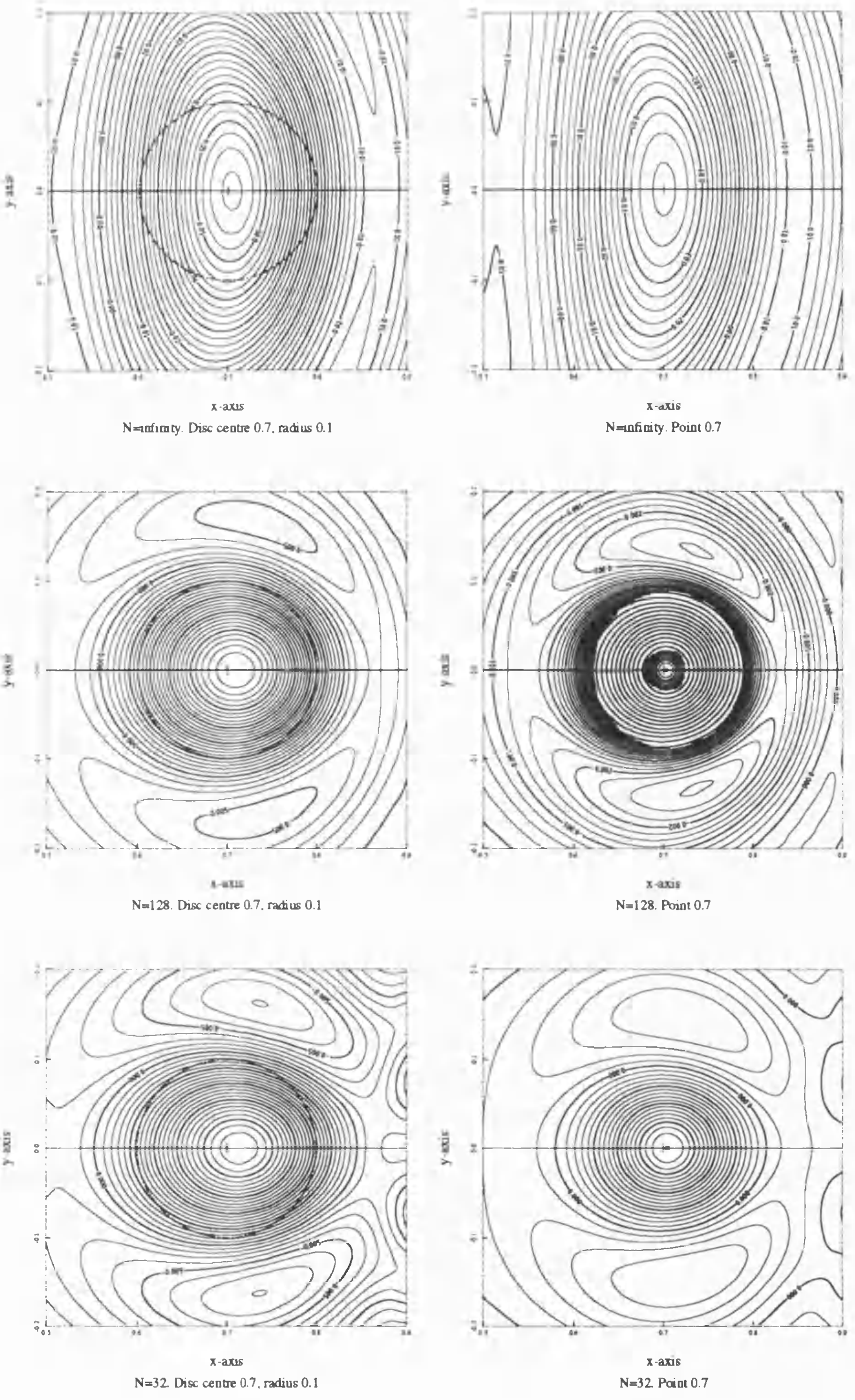
in the top ($N = \infty$) graph than in the bottom ($N = 32$) graph. Moreover, in the top graph, the curves seem a little more separated. They seem to take their most extreme values near the point $x = 0$ and the curve that corresponds to $T_{(0,0.1)}(f)$ seems to change rapidly over the boundary of that disc, which is indicated by the vertical line. In Table 6.2 we give the value of these least favourable functions and others when $N = 64$ and $N = 256$ at the origin. We

N	$T_{(0.0,0.1)}$	$T_{0.0}$
32	0.01456	0.01185
64	0.01466	0.01139
128	0.01485	0.01129
256	0.01487	0.01127
∞	0.02976	0.02252

Table 6.2: *The value of the radially symmetric least favourable functions at the origin*

make two comments. First the value of the least favourable function for $T_{(0.0,0.1)}$ at the origin is greater than the value of the least favourable function for $T_{0.0}$ for all the values of N considered. Secondly, for finite N, this value for $T_{(0.0,0.1)}$ increases with N, whereas for $T_{0.0}$ it decreases with N.

Secondly, we give examples and a qualitative description of asymmetric least favourable functions away from the centre of the unit circle. We consider $T_{(x,0.1)}(f)$ and $T_x(f)$, where $x = 0.7$. Again we take $n = 10,000,000$, $a = 1.5$ and $C = 2.0$. In Figure 6.7 we present contour plots of that part of the asymmetric least favourable functions that lies in a square centre $(0.7, 0)$, of sides 0.4. We use the excellent CONICON programs of Sibson[36]. In particular we use an interface to these programs written by Dr Glenn Stone. The axes and the labels are produces by means of a POSTSCRIPT program written by the author. The reader is advised that the numbers printed on the contour lines follow the convention of having their top in the direction of increasing height. We indicate on the plots the disc and its centre, or the point, as appropriate. The top row is concerned with the continuous case ($N = \infty$), the middle row with the case when there are 128 detectors ($N = 128$), and the bottom row with the case when there are 32 detectors ($N = 32$). The least favourable functions for the continuous case are relatively simple, whereas for the $N = 32$ and $N = 128$ cases they are very complicated. For



N=infinity. Disc centre 0.7, radius 0.1

N=infinity. Point 0.7

N=128. Disc centre 0.7, radius 0.1

N=128. Point 0.7

N=32. Disc centre 0.7, radius 0.1

N=32. Point 0.7

Figure 6.7: *Least favourable functions for the functionals $T_{((0.0,0.7),0.1)}$ and $T_{(0.0,0.7)}$, when there are 32 and 128 detectors, and in the continuous case*

comparison, Table 6.3 gives the value (taken to be positive) of the least favourable function at the point (0.7, 0.0) in Cartesian coordinates. We observe the same behaviour here as we noted for the results given in Table 6.2.

N	$T_{((0.7,0.0),0.1)}$	$T_{(0.7,0.0)}(f)$
32	0.0197	0.0147
64	0.0217	0.0135
128	0.0223	0.0133
256	0.0224	0.0132
∞	0.0546	0.0361

Table 6.3: *The value of the least favourable function at the point* (0.7, 0.0)

6.4 Conclusions

In this chapter we have presented some numerical examples to illustrate the theory developed in Chapter 5. We introduced two linear functionals, discussed some relevant computational aspects and investigated the behaviour of the minimax risk as a function of the expected number of emissions n, for various values of the number of detectors N. We provide an interpretation of these minimax risks by means of the notion of efficiency. The examples that we have considered have provided us with some insight into the construction of the PET machine (in terms of doubling the number of detectors N) and the design of the experiment (in terms of increasing the expected number of emissions n). Finally, we presented some examples of the least favourable functions.

Bibliography

[1] Apostol, T. M. (1982). *Mathematical Analysis*, 2nd ed. Addison-Wesley, Reading, Massachusetts.

[2] Besag, J. E. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Statist. Soc. B*, **36**, 192–236.

[3] Besag, J. E. (1986). On the statistical analysis of dirty pictures (with discussion). *J. R. Statist. Soc. B*, **48**, 259–302.

[4] Besag, J. E. (1989). Towards Bayesian image analysis. *J. Appl. Stat.*, **16**, 395–407.

[5] Bickel, P. J. and Ritov, Y. (1991). Estimating linear functionals of a PET image. University of California–Berkeley (unpublished).

[6] Born, M. and Wolf, E. (1975). *Principles of Optics*, 5th ed. Pergamon, New York.

[7] Cross, G. R. and Jain, A. K. (1983). Markov random field texture models. *IEEE Trans. Pattn Anal. Mach. Intell.*, **PAMI-5**, 25–39.

[8] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Statist. Soc. B*, **39**, 1–38.

[9] Donoho, D. L., Liu, R. C. and MacGibbon, B. (1990). Minimax risk over hyperrectangles, and applications. *Ann. Statist.*, **18**, 1416–1437.

[10] Ford, L. R. and Fulkerson, D. R. (1962). *Flows in Networks.* Princeton University Press, Princeton.

[11] Frigessi, A. and Piccioni, M. (1988). Parameter estimation for 2-dimensional Ising fields corrupted by noise. *Spatial Statistics and Imaging, IMS Lecture notes*. Maine, USA.

[12] Geman, D. and Geman, S. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattn Anal. Mach. Intell.*, **PAMI-6**, 721–741.

[13] Geman, S. and McClure, D. E. (1987). Statistical methods for tomographic image reconstruction. *Bull. Int. Statist. Inst.*, **52 (Bk. 4)**, 5–21.

[14] Geman, D. and Reynolds, G. (1990). Constrained restoration and the recovery of discontinuities. University of Massachusetts at Amherst (unpublished).

[15] Green, P. J. (1990). Bayesian reconstructions from emission tomography data using a modified EM algorithm. *IEEE Trans. Med. Imgng.*, **MI-9**, 84–93.

[16] Green, P. J. (1990). On use of the EM algorithm for penalized likelihood estimation. *J. R. Statist. Soc. B*, **52**, 443–452.

[17] Greig, D. M., Porteous, B. T. and Seheult, A. H. (1989). Exact maximum *a posteriori* estimation for binary images. *J. R. Statist. Soc. B*, **51**, 271–279.

[18] Hajek, B. (1988). Cooling schedules for optimal annealing. *Math. Oper. Res.*, **13**, 311–329.

[19] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

[20] Jennison, C. and Silverman, B. W. (1991). How to charge for boundaries in a pixel image. In Mardia, K. V. (ed.), *Essays in Statistics: in Honour of G. S. Watson*. Wiley, New York.

[21] Johnstone, I. M. and Silverman, B. W. (1990). Speed of estimation in positron emission tomography and related inverse problems. *Ann. Statist.*, **18**, 251–280.

[22] Johnstone, I. M. and Silverman, B. W. (1991). Discretization effects in statistical inverse problems. *Journal of Complexity*, **7**, 1–34.

[23] Jones, M. C. and Silverman, B. W. (1989). An orthogonal series density estimation approach to reconstructing positron emission tomography images. *J. Appl. Stat.*, **16**, 177–191.

[24] Jubb, M. D. (1989). Image reconstruction. Ph. D. Thesis, University of Bath.

[25] Kreyszig, E. (1978). *Introductory Functional Analysis with Applications.* Wiley, New York.

[26] Laarhoven, P. J. M. and Aarts, E. H. L. (1987). *Simulated Annealing: Theory and Applications.* D. Reidel Publishing Company, Dordrecht, Holland.

[27] Li K.-C. (1982). Minimaxity of the method of regularization on stochastic processes. *Ann. Statist.*, **10**, 937–942.

[28] Marr, R. B. (1974). On the reconstruction of a function on a circular domain from a sampling of its line integrals. *J. Math. Anal. Appl.*, **45**, 357–374.

[29] Marroquin, J. L. (1984). Surface reconstructions preserving discontinuities. *AI Memo 792.* MIT Artificial Intelligence Laboratory, Cambridge, MA, USA.

[30] Murray, D. W., Kashko, A. and Buxton, H. (1986). A parallel approach to the picture restoration algorithm of Geman and Geman. *Image Vision Comput.*, **4**, 133–142.

[31] Nychka, D. (1990). Some properties of adding a smoothing step to the EM algorithm. *Statist. Probab. Lett.*, **9**, 187–193.

[32] Ripley, B. D. (1987). *Stochastic Simulation.* Wiley, New York.

[33] Ripley, B. D. (1988). *Statistical Inference for Spatial Processes.* Cambridge University Press, Cambridge.

[34] Ritov, Y. (1991). *Personal Communication.*

[35] Seheult, A. H. (1989). *Personal Communication.*

[36] Sibson, R. (1987). CONICON3 *Handbook.* School of Mathematical Sciences, University of Bath.

[37] Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

[38] Silverman, B. W. (1991). Minimax estimation of linear functionals. University of Bath (unpublished).

[39] Silverman, B. W., Jennison, C., Stander, J. and Brown, T. C. (1990). The specification of edge penalties for regular and irregular pixel images. *IEEE Trans. Pattn Anal. Mach. Intell.*, **PAMI-12**, 1017–1024.

[40] Silverman, B. W., Jones, M. C., Wilson, J. D. and Nychka, D. W. (1990). A smoothed EM approach to indirect estimation problems, with particular reference to stereology and emission tomography (with discussion). *J. R. Statist. Soc. B*, **52**, 271–324.

[41] Speckman, P. (1979). Minimax estimates of linear functionals in a Hilbert space. University of Oregon (unpublished).

[42] Stander, J., Farrington, D. P., Hill, G. and Altham, P. M. E. (1989). Markov chain analysis and specialization in criminal careers. *British Journal of Criminology*, **29**, 317–335.

[43] Thompson, A. M., Brown, J. C., Kay, J. W. and Titterington, D. M. (1991). A study of methods of choosing the smoothing parameter in image restoration by regularization. *IEEE Trans. Pattn Anal. Mach. Intell.*, **PAMI-13**, 326–339.

[44] Vardi, Y., Shepp, L. A. and Kaufman, L. (1985). A statistical model for positron emission tomography (with comments). *J. Amer. Statist. Assoc.*, **80**, 8–37.

[45] Wichmann, B. A. and Hill, J. D. (1982). Algorithm AS183. An efficient and portable pseudo-random number generator. *Appl. Statist.*, **31**, 188–190.

[46] Wright, W. A. (1989). A Markov random field approach to data fusion and colour segmentation. *Image Vision Comput.*, **7**, 144–150.