Computational Methods for the Bayesian Analysis of Dynamic Models

Photis Stavropoulos

A Dissertation Submitted to the

University of Glasgow

for the degree of

Doctor of Philosophy

Department of Statistics

November 1998

Abstract

The topic of dynamic models has been extensively studied in statistics, both from classical and Bayesian perspectives. Some of the unknown components of a dynamic model evolve in time according to a probability model and give rise to observed data according to a second model. If one adopts a Bayesian point of view, as we do in this thesis, these models combined with any available prior information give rise to a sequence of posterior distributions for all the unknowns. They encompass all our knowledge about them but in practice the cases where they are analytically available are the exception to the rule. A wealth of methods for approximating them have been developed, originating mainly from the engineering community. They may however lead to unsatisfactory results and the only remaining resort will be to represent the posterior distributions by samples. As in all areas of Bayesian statistics Markov Chain Monte Carlo methods are the most widespread tool for this purpose.

The aim of this thesis is to present and study a group of techniques for sampling from intractable distributions; namely, the importance resampling methods. They are fast and easy to implement, and some of them have nice theoretical properties. Moreover, by their design they are well suited for application in dynamic model contexts.

The thesis is divided into two parts. The first one comprises Chapters 1 and 2. Chapter 1 offers an introduction to dynamic models and presents in detail the original resampling method, the weighted bootstrap. We examine how it can be applied in dynamic model problems and prove some of its theoretical properties. In Chapter 2 we present more recently suggested techniques that try to improve the characteristics of weighted bootstrap. We prove that one of them, the smooth bootstrap, has the same properties as weighted bootstrap. We also compare all the methods in several simulation experiments.

The second part consists of the three remaining chapters. Their unifying element is that they all deal with versions of the same problem: how to follow a moving particle in space when all the data we receive are noisy measurements of our squared distance from it. This is a geometrical description of problems that arise in industry. For example, the "moving particle" may be the changing optimal conditions for production of a commodity and the "squared distance" from the particle may be the cost we pay for not producing at these conditions. The problem can be expressed as a dynamic model with intractable posteriors and we want to see how resampling will perform in such a difficult situation. In Chapter 3 we suggest several solutions. They can all be adapted to any version of the problem. Resampling encounters some difficulties for which we manage to find an *ad hoc* solution. In Chapter 4 we deal with two more complicated versions of the problem. One of them causes resampling to break down. This leads us to Chapter 5, where we present and examine more recently proposed resampling algorithms. The conclusion is that there is still the need for further improvement of the resampling techniques.

To Vivi

Acknowledgements

- First of all I would like to thank my supervisor, Prof. Mike Titterington, for his invaluable guidance, his encouragement and his patience during the three years of my research. Without his help this thesis would not have been possible.

- I am indebted to the "Alexander S. Onassis" Public Benefit Foundation, Athens, Greece for funding my research and to the Engineering and Physical Sciences Research Council, UK for paying my fees.

- I would also like to thank all the members of academic and non-academic staff at the Department of Statistics of the University of Glasgow for providing me with a pleasant, friendly and functional working environment.

- A big thank you goes to all the past and current postgraduate students at the department for all the fun we have had together, not to mention the fruitful exchange of views and opinions.

- I owe a large debt of gratitude to my parents and brother for their love, moral support and care and for the financial reinforcements during all my studying years.

- Finally, a big thank you to Vivi for her love, support and patience.

Contents

List of Figures

List of Tables

Chapter 1

The Weighted Bootstrap in Bayesian Statistics

1.1 Introduction

The original motivation for this work was the analysis of series of images of a changing scene. Such problems are expressed in terms of dynamic models that are usually nonlinear and/or non-Gaussian. This means that the commonly used Kalman filter is not applicable. Moreover, the emerging conditional distributions (or posteriors if one adopts a Bayesian perspective) are very often analytically intractable.

The tool one has to resort to is Monte Carlo simulation. In order to study a distribution whose density is not available in closed form, a sample is obtained from it. Then, parameters of interest can be estimated by the corresponding parameters of the sample. Even the density function itself can be estimated by some form of kernel density estimation. See for example Silverman (1986),

West (1990) and Escobar and West (1995). However, it often happens that even sampling from the distribution of interest is impossible. In this case techniques that produce samples coming approximately from the target distribution are employed.

The most widely used group of such techniques are Markov Chain Monte Carlo methods (MCMC). They consist of finding an ergodic and irreducible Markov chain that has the distribution of interest as its invariant distribution. The idea is to simulate the chain for a long enough time so that it forgets its initial state and can be assumed to have converged to its invariant distribution. The values generated from that point onwards will form a sample of dependent observations from the target distribution. For more details on MCMC see Gilks *et al* (1996), Besag and Green (1993) and Smith and Roberts (1993).

However, MCMC methods have some characteristics which limit their applicability to dynamic model settings. We present these in the next section which deals with dynamic models in greater detail. In the rest of the chapter we present a method that could be used instead of MCMC for the analysis of such models.

1.2 Dynamic models

Throughout this thesis by the name of dynamic model we denote a discrete time dynamic model. A dynamic model is a mathematical description of the following situation. At instants t in time data y_t arise whose values depend, among other things, on unknown parameters x_t which we want to study. It is assumed that each y_t is generated according to the model

$$y_t = A(x_t, \epsilon_t), \tag{1.1}$$

where A is a function which can have any form and ϵ_t is a random variable. The parameters x_t do not remain constant in time, which explains the presence of the subscript t. It is assumed that their values evolve according to the model

$$x_{t+1} = B(x_t, \eta_t), \tag{1.2}$$

where B is another function of any form and η_t is a random variable. In dynamic model terminology the parameters x_t are thought to be describing the evolution of a system and x_t is called the **state** of the system at time t. Model (1.2) is called the **system evolution equation** and η_t the **system disturbance** at time t. Similarly, model (1.1) is called the **observation equation** and ϵ_t the **observation noise** at time t. The dimensionalities of y_t, x_t, ϵ_t and η_t can be arbitrary as long as functions A and B ensure that they conform with each other.

Since both (1.1) and (1.2) involve random variables they can be represented by the distribution of y_t given x_t and the distribution of x_{t+1} given x_t respectively. Of course, this implies that the forms of the distributions of ϵ_t and η_t are known, but normally this is true. Usually the distributions of ϵ_t and η_t are assumed not to change in time. We denote any density or probability mass function and any distribution by p. The distributions corresponding to (1.1) and (1.2) can be written as

$$p(y_t|x_t) \tag{1.3}$$

and

$$p(x_{t+1}|x_t). \tag{1.4}$$

To complete the description of the situation we assume that the state x_1 of the system at the first time instant also is a random variable with distribution

$$p(x_1). \tag{1.5}$$

Any prior information about the system is incorporated in (1.3),(1.4) and (1.5). (1.1),(1.2) and (1.5) or (1.3),(1.4) and (1.5) comprise a dynamic model.

To simplify things assume that the distributions of η_t, ϵ_t and x_1 are fully known, as are the functional forms of A and B. Moreover, assume that each ϵ_t is independent of x_s and η_s for any s and of ϵ_s for any $s \neq t$ and that each η_t is independent of x_s for any s. Then y_t is independent of anything else in the past given x_t and x_t is also independent of anything else in the past given x_{t-1}. Then interest focuses on inference about x_t. At time t the most complete and concise description of our knowledge about x_t is given by its conditional distribution given all the available data $\mathbf{y}_t = (y_1, \ldots, y_t)$. This distribution, $p(x_t|\mathbf{y}_t)$, in dynamic model jargon, is called the **filtering distribution**. Its evolution in time is described by two recursive formulae given by a straightforward application of Bayes' theorem and elementary probabilistic reasoning:

$$p(x_t|\mathbf{y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbf{y}_{t-1})}{\int p(y_t|x_t)p(x_t|\mathbf{y}_{t-1})dx_t}, \tag{1.6}$$

$$p(x_t|\mathbf{y}_{t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|\mathbf{y}_{t-1})dx_{t-1}. \quad (1.7)$$

Equation (1.6) is called the **update equation** and (1.7) is called the **propagation** or **prediction equation** or the **predictive distribution**. For a detailed presentation of the statistical theory of dynamic models see West and Harrison (1997).

In most cases it is impossible to obtain (1.6) and (1.7) in closed form. The most well behaved and consequently well studied case is when the functions A and B are linear and x_1, ϵ_t and η_t are Gaussian. This is the so-called **linear Gaussian model**. Then it is easily shown that (1.6) and (1.7) are also Gaussian and therefore can be described by just two parameters each, the mean vector and the covariance matrix. Probabilistic reasoning can lead to the value of these parameters at any time point. The most famous description of their evolution in time is the **Kalman filter** (Kalman (1960)). It is a system of two recursive equations that give the value of the mean vector μ_t and the covariance matrix Σ_t of (1.6) at time t given their values μ_{t-1} and Σ_{t-1} at time $t-1$ and the data y_t that has arrived at time t. We present these equations later on in Section 2.5.2, which deals with a linear Gaussian model.

From the discussion above it is clear that for a linear Gaussian dynamic model we do not need simulation. As soon as nonlinearities enter the model or some of the random variables are not Normally distributed the filtering distribution becomes intractable. Several methods have been proposed which try to analyse dynamic models in such cases. Some rely on transformations of the system and the observation equations. The most famous is the **extended Kalman filter** (Anderson and Moore (1979)). It assumes Normal noises and

linearizes with Taylor expansions any non-linear functions involved in the model. See also Fahrmeir (1992). For other transformations see Sorenson and Stubberud (1968), Masreliez (1975) and Julier and Uhlmann (1997). Instead of taking expansions of the functions involved Srinivasan (1970) takes Gram-Charlier expansions of the densities of interest. Alspach and Sorenson (1972) linearize (1.6) and (1.7) if they are not linear already and use **Gaussian sum approximation**, in other words approximation of all the densities of interest (including those for noise) by mixtures of Normal distributions. However good they are in particular applications all these approximations are based on potentially unrealistic assumptions and their general applicability is thus inhibited. In a different vein, Masreliez and Martin (1977) and Meinhold and Singpurwalla (1989) adapt the Kalman filter to cases of fat-tailed noise distributions which can better accommodate outlying observations.

Numerical techniques for evaluation of the integrals involved in (1.6) and (1.7) have also been put forward. Kitagawa (1987) and Kramer and Sorenson (1988) replace the integrals with sums by approximating the filtering distribution by piecewise constant functions. Such methods are called **grid methods** because the piecewise constant functions are evaluated only at a grid of points over the state space. They have the advantage that they do not rely on any simplifying assumptions about the model. Their problem however, is that the grid of points is static and therefore, it has to be very large (especially in multidimensional cases) if the piecewise constant function is to always approximate well the filtering distribution whose support may be shifting. Pole and West (1990) manage to get good results with very small grids by forcing the grids to shift dynamically in order to follow the support of the filtering distribution.

The intractability of (1.6) and (1.7) also meant that until the advent of MCMC it was very difficult, if not impossible, to sample from them. MCMC can give samples from them and it does not need any simplifying assumptions about the form of A and B or about the distribution of the random variables involved. Conclusions can be based on the samples and have been shown in practice to be more correct than those provided by the approximations mentioned above. However, MCMC also has its problems.

First of all, in order to apply MCMC we have to view $p(x_t|\mathbf{y}_t)$ as a marginal distribution of the full conditional distribution $p(\mathbf{x}_t|\mathbf{y}_t)$ of all the states up to time t given all the available data. This is because if, for example, the target density is $g(x)$ MCMC requires the knowledge of that part of g that depends on x. If $p(x_t|\mathbf{y}_t)$ is the target this means that at least the numerator of (1.6) must be known, but this involves $p(x_t|\mathbf{y}_{t-1})$ which is also intractable. The part of $p(\mathbf{x}_t|\mathbf{y}_t)$ that depends on $\mathbf{x}_t$, on the other hand, can be written as a product of terms of the form $p(y_t|x_t)$ and $p(x_t|x_{t-1})$ which are known since we know the distribution of ϵ_t and η_t. MCMC will thus give a sample from $p(\mathbf{x}_t|\mathbf{y}_t)$ from which we will keep only the x_t part and discard the rest. The same will have to be done at time $t+1$ for $p(\mathbf{x}_{t+1}|\mathbf{y}_{t+1})$ and so on. If the samples are very large and if also such samples are required for many time points this means a great waste of resources. The methods that we present in the rest of the chapter can sample from $p(x_t|\mathbf{y}_t)$ if desired and moreover in doing so they use the sample from $p(x_{t-1}|\mathbf{y}_{t-1})$ that has been generated at time $t-1$. They can also give samples from $p(x_t|\mathbf{y}_{t-1})$. They can essentially propagate and update these samples in the same way that their corresponding densities (1.6) and (1.7) evolve in time.

Another factor inhibiting the application of MCMC in dynamic model settings

is the issue of convergence of the Markov chain that is simulated. First, it can be very slow, which means that real-time analysis of a system may not be possible. Secondly, there is not yet any automatic method of assessing convergence which rules out an entirely automatic analysis of a system. The methods that we are going to present do not involve Markov chains. They are therefore a lot faster than MCMC and can be applied without any supervision.

To summarize, we are now going to present methods that work with any type of dynamic model without having to rely on unrealistic assumptions about it. They are also fast and reliable and can be applied in an automatic, real-time analysis set-up.

1.3 Importance sampling and the weighted bootstrap

Importance sampling is a well known simulation technique; see for example Ripley (1987) and, for a Bayesian viewpoint, Geweke (1989). It works as follows. Suppose that we want to estimate

$$E_h(k(X)) = \int k(x)h(x)dx$$

where $h(x) = \frac{f(x)}{\int f(x)dx}$ is a probability density function (pdf). The obvious estimator is

$$\bar{k} = \frac{1}{n}\sum_{i=1}^{n} k(x_i),$$

where $x_1, x_2, \ldots, x_n$ are independent random draws from h. This estimator

is unbiased and asymptotically Normally distributed given certain regularity conditions. If sampling from h is not possible but we can evaluate the value of h at any point, we can draw independent samples $x_1, x_2, \ldots, x_n$ from another pdf g, available for sampling, and obtain the estimator

$$\hat{k} = \frac{1}{n} \sum_{i=1}^{n} \frac{h(x_i)}{g(x_i)} k(x_i).$$

This estimator is also unbiased and asymptotically Normally distributed because we can write

$$E_h(k(X)) = \int k(x)h(x)dx = \int k(x)\frac{h(x)}{g(x)}g(x)dx = E_g\left(k(X)\frac{h(X)}{g(X)}\right).$$

Finally, if not h but only f can be evaluated, we can still estimate $E_h(k(X))$ based on the sample $x_1, x_2, \ldots, x_n$ from g, by

$$\tilde{k} = \frac{\sum_{i=1}^{n} \frac{f(x_i)}{g(x_i)} k(x_i)}{\sum_{i=1}^{n} \frac{f(x_i)}{g(x_i)}}.$$

This is because

$$E_h(k(X)) = \int k(x)\frac{h(x)}{g(x)}g(x)dx = \frac{\int k(x)\frac{f(x)}{g(x)}g(x)dx}{\int \frac{f(x)}{g(x)}g(x)dx} = \frac{E_g\left(k(X)\frac{f(X)}{g(X)}\right)}{E_g\left(\frac{f(X)}{g(X)}\right)}.$$

This estimator is asymptotically Normally distributed and asymptotically unbiased (Geweke (1989)).

An extension of importance sampling can be used in order to take a sample

from g and arrive at a sample from h. This is the method of **weighted bootstrap**.

1.3.1 Weighted Bootstrap

This method first appeared in Rubin (1987,1988) and was popularized by Smith and Gelfand (1992). It works as follows. Suppose that we want a random sample from a distribution h and that we cannot get it directly. Moreover, suppose that $h(x) = \frac{f(x)}{\int f(x)dx}$ and that we only know how to evaluate f. If we have another distribution g that is easy to sample from we obtain a sample $x_1, \ldots, x_n$ from it. Then to each point x_i we assign an importance weight w_i given by

$$w_i = \frac{f(x_i)}{g(x_i)} \tag{1.8}$$

and then normalise it to get

$$q_i = \frac{w_i}{\sum_{j=1}^{n} w_j}.$$

The w_i's are called **weights** and the q_i's are called **normalized weights**. We can then sample from among $x_1, x_2, \ldots, x_n$ with replacement, using the q_i's as probabilities of selection, and take a sample $y_1, \ldots, y_m$. In other words, for each $i = 1, \ldots, m$, $\Pr(y_i = x_j) = q_j$, for all j. The sample $y_1, \ldots, y_m$ can be considered as coming from h if n is large. The justification for this runs as follows[1].

[1]The proof is taken from Smith and Gelfand (1992).

Suppose that θ^* is drawn from among $x_1, x_2, \ldots, x_n$ with $\Pr(\theta^* = x_j) = q_j$. I is the indicator function. All limits are taken with $n \longrightarrow \infty$. Then,

$$\begin{aligned}
\Pr(\theta^* \le a) &= \sum_{i=1}^{n} q_i I(x_i \le a) \\
&= \frac{\frac{1}{n}\sum_{i=1}^{n} w_i I(x_i \le a)}{\frac{1}{n}\sum_{i=1}^{n} w_i} \\
&\xrightarrow{\text{a.s.}} \frac{E_g\left(\frac{f(X)}{g(X)} I(X \le a)\right)}{E_g\left(\frac{f(X)}{g(X)}\right)} \\
&= \frac{\int_{-\infty}^{\infty} \frac{f(x)}{g(x)} I(x \le a) g(x) dx}{\int_{-\infty}^{\infty} \frac{f(x)}{g(x)} g(x) dx} \\
&= \frac{\int_{-\infty}^{a} f(x) dx}{\int_{-\infty}^{\infty} f(x) dx} \\
&= \int_{-\infty}^{a} h(x) dx \\
&= \Pr_h(X \le a),
\end{aligned}$$

where "a.s." denotes almost sure convergence. In practice, since n is finite, this result will hold approximately with the approximation improving as n increases. The size m of the resulting sample should not be larger than n. In this thesis we usually take $m = n$ and sometimes $m < n$. If h is fully known, weighted bootstrap is still applied in the same way. The only difference is that we use h instead of f for the calculation of the weights in (1.8). The mathematical result for the resulting sample is valid again. Tanizaki and Mariano (1994) use a variation of importance sampling where the sample points are not drawn at random from the importance sampler but are chosen in a deterministic way.

The sample resulting from weighted bootstrap is used as any random sample from h. However, its members, although independent given the sample from the importance sampler, are not unconditionally independent. It is easy to see that

$$Cov(Y_i, Y_j) = V_g \left(\sum_{i=1}^{n} q_i X_i \right).$$

This is proven in appendix A as part of a larger proof. Because of its connections with importance sampling, weighted bootstrap is also known as **importance resampling**. In the rest of this thesis any density that plays the role of g is called the **importance sampler**.

From (1.8) it is clear that weighted bootstrap can be very useful in Bayesian settings. There the target is the posterior density of the parameters of interest which usually is not fully known. It usually happens that we know only that

$$\text{posterior} \propto \text{likelihood} \cdot \text{prior}.$$

Then, if the prior is easy to sample from we can use it as the importance sampler. If $x_1, x_2, \ldots, x_n$ is a sample from the prior and $\mathbf{y}$ are the available data, the weight associated with each sample point x_i is just its likelihood, $w_i = p(\mathbf{y}|x_i)$.

Nevertheless, the method has some disadvantages as well. First of all, it uses a discrete distribution, namely the one with support $\{x_1, \ldots, x_n\}$ and probability mass function $\{q_1, \ldots, q_n\}$ in order to approximate the target distribution which may be continuous. If moreover the target is a multivariate function the approximation can be very crude. The only remedy for this problem is to

increase n.

A second problem occurs if the support of g does not contain parts of the support of h. In other words, there may be values which are plausible under h but cannot be sampled from g. Such values are not going to be part of the sample resulting from weighted bootstrap either although it is supposed to come from h. For this reason the support of g should be the same as that of h and g should have fatter tails than h.

Another problem, possibly the worst, is **sample deterioration**. This will be made clearer with an example. Imagine a situation where data $y_1, y_2, \ldots$ arrive sequentially, generated by a probabilistic model $p(y_t|x)$ that depends on an unknown parameter x. From the model it is clear that $y_1, y_2, \ldots$ are mutually independent given x. We want to study the resulting posteriors $p(x|y_1), p(x|\mathbf{y}_2), \ldots, p(x|\mathbf{y}_t), \ldots$ where $\mathbf{y}_t = (y_1, \ldots, y_t)$. This accumulation of knowledge about a constant quantity based on sequentially arriving data is called **Bayesian learning**. Two successive posteriors are connected by the formula

$$p(x|\mathbf{y}_{t+1}) = \frac{p(y_{t+1}|x)p(x|\mathbf{y}_t)}{p(y_{t+1}|\mathbf{y}_t)} \propto p(y_{t+1}|x)p(x|\mathbf{y}_t). \tag{1.9}$$

The prior distribution $p(x)$ completes the chain. The denominators in (1.9) cannot usually be calculated and if $p(x)$ and $p(y_t|x)$ are not of any standard form the only way to study the posteriors will be to obtain samples from them. From (1.9) we can see that weighted bootstrap is very handy for this purpose. Each posterior can serve as the importance sampler for the subsequent one.

If we start with a sample $x(1), \ldots, x(n)$ from the prior $p(x)$ and assign to each

$x(i)$ weight $w(i) = p(y_1|x(i))$, application of weighted bootstrap will lead to a sample $x_1(1), \ldots, x_1(n)$ from $p(x|y_1)$. If we assign to each $x_1(i)$ weight $w_1(i) = p(y_2|x_1(i))$, application of weighted bootstrap will lead to a sample $x_2(1), \ldots, x_2(n)$ from $p(x|\mathbf{y}_2)$. Successive applications of weighted bootstrap in the same fashion will provide us with a sample $x_t(1), \ldots, x_t(n)$ from each posterior $p(x|\mathbf{y}_t)$. All these samples consist exclusively of points that were in the initial sample $x(1), \ldots, x(n)$. Moreover, during the transition from $x_t(1), \ldots, x_t(n)$ to $x_{t+1}(1), \ldots, x_{t+1}(n)$ some points contained in the first sample may not be picked at all during the resampling, especially if their weight is small, while points with large weight will probably be picked more than once. The points that are not picked are lost forever since $x_{t+1}(1), \ldots, x_{t+1}(n)$ will be the basis for all future resamplings. In other words each sample $x_t(1), \ldots, x_t(n)$ contains at most the same number of but most probably fewer distinct points than its preceding one. If t is very large we may end up with $x_t(1) = \ldots = x_t(n)$. Therefore, successive applications of weighted bootstrap cause the samples to deteriorate. A side-effect of deterioration is that the variance of the samples becomes unrealistically small and underestimates the variance of the distribution they are supposed to come from.

Sample deterioration does not only happen in Bayesian learning. Suppose again that g is the importance sampler and h is the target. If the main support of g is in an area of small support under h, then probably very few points among the sample from g will fall in the main support of h. These points will receive very large weights compared with the rest of the sample and will be favoured during the resampling. The resulting sample will mainly consist of many replicates of these few points.

Givens and Raftery (1996) propose a measure of sample deterioration. If the

sample from the importance sampler had size n and all the normalized weights were equal to $\frac{1}{n}$ we would expect the weighted bootstrap to give a sample of size m with $n(1-\exp(-m/n))$ distinct values. If we apply weighted bootstrap with the actual normalized weights and get a sample of size m with Q distinct values then the measure U of sample deterioration is

$$U = \frac{Q}{n\left(1 - \exp\left(-\frac{m}{n}\right)\right)}. \tag{1.10}$$

If among the weights there are a few large ones then one would expect the numerator of (1.10) to be a lot smaller than the denominator and U to be less than 1. If the weights are close to being equal to each other then we may even get U greater than 1.

Another way to measure sample deterioration is the **effective sample size**. This derives from importance sampling theory as follows. Suppose we want to estimate

$$E_h\left(k(X)\right) = \int k(x)h(x)dx.$$

If we have a sample $x_1, \ldots, x_n$ from $h(x) = \frac{f(x)}{\int f(x)dx}$ the estimator is

$$\bar{k} = \frac{\sum_{i=1}^{n} k(x_i)}{n},$$

while if the sample comes from an importance sampler g the best estimator is

$$\tilde{k} = \frac{\sum_{i=1}^{n} w_i k(x_i)}{\sum_{i=1}^{n} w_i},$$

with $w_i = \frac{f(x_i)}{g(x_i)}$. Comparing the variances of these two estimators, Kong *et al* (1994) state that, if k does not vary too quickly with x,

$$\frac{\text{var}_g(\tilde{k})}{\text{var}_h(\bar{k})} \approx 1 + \text{var}_g(w_i^*),$$

where $w_i^* = \frac{h(x_i)}{g(x_i)}$, i.e. it is the importance weight we would have if we knew h fully. Then the effective sample size (ESS) is defined as

$$\text{ESS} = \frac{n}{1 + \text{var}_g(w_i^*)}. \tag{1.11}$$

Imagine a sample of size ESS from h. Then the variance of the sample mean of $k(x)$ will be $\text{var}_h(k(X))/\text{ESS}$. But also $\text{var}_h(\bar{k}) = \text{var}_h(k(X))/n$. Then,

$$\frac{\text{var}_h(k(X))}{\text{ESS}} = \text{var}_g(\tilde{k}) \implies \frac{n\text{var}_h(\bar{k})}{\text{ESS}} = \text{var}_g(\tilde{k}) \implies \frac{n}{\text{ESS}} = \frac{\text{var}_g(\tilde{k})}{\text{var}_h(\bar{k})}$$
$$\implies \text{ESS} = \frac{n}{1 + \text{var}_g(w_i^*)}.$$

As we can see from the equation above, ESS is the size of a sample from h that would be required for that sample's mean of $k(x)$ to have the same variance as the mean $\tilde{k}$ of a sample of size n from g. If the variance of the weights is too large the effective sample size will be very small. If all the weights are

equal to each other, which will happen if $g = h$, then ESS $= n$. Also, large variance of the weights means that some of them will be a lot larger than the rest and therefore if we apply weighted bootstrap the resulting sample will be very poor. So, small ESS is a warning signal against applying weighted bootstrap.

In practice, instead of having w_i^* we have w_i. We can estimate w_i^* by

$$\hat{w}_i^* = \frac{nw_i}{\sum_{j=1}^n w_j} \approx \frac{\frac{f(x_i)}{g(x_i)}}{\int f(x)dx} = \frac{h(x_i)}{g(x_i)}.$$

Then the sample variance of $\hat{w}_i^*$ can be used to estimate ESS.

Carpenter *et al* (1997) suggest another estimator of ESS which however depends on the function k. Therefore, if many different functions are of interest the estimator of Kong *et al* (1994) is clearly advantageous.

Another indicator of potential problems from applying weighted bootstrap is the **entropy** of the normalized weights **relative to uniformity**

$$-\sum_{i=1}^{n} q_i \frac{\log q_i}{\log n}$$

which is proposed by West (1993). Small values of it indicate that there is a problem while values close to 1 show that the weights are close to being equal and we can go ahead with weighted bootstrap.

A remedy against all the problems mentioned above is to increase the ratio $\frac{n}{m}$ of sample sizes. This is quite inelegant since it relies just on "brute" force. Lee (1996) provides a theoretical argument giving the required n for any

combination of m, h and g. The author proves that the number of points that must be drawn from g until we get one belonging to the support of h follows a geometric distribution and then derives its expectation. If this expectation is α then n must be $\{m\alpha\}$ where $\{k\}$ denotes the smaller integer that is greater than or equal to k. Incidentally, Lee (1996) also provides a way of calculating both n and m. It is based on the maximum mean squared error that we are willing to allow between the true cumulative distribution function (cdf) and the cdf implied by the sample $y_1, \ldots, y_m$.

A second way of fighting sample deterioration is to ensure that the sample from the importance sampler does not contain points that are going to have a very small normalized weight. This is proposed in Gordon *et al* (1993) under the name of **prior editing**. In the Bayesian learning context, at each time t we want a good sample $x_t(1), \ldots, x_t(n)$ so that most of its members will have a good chance of also being part of $x_{t+1}(1), \ldots, x_{t+1}(n)$. This is achieved as follows. Suppose that we have already chosen $x_t(1), \ldots, x_t(k-1)$ by resampling from among $x_{t-1}(1), \ldots, x_{t-1}(n)$. At step k we pick x_t^* with $\Pr(x_t^* = x_{t-1}(i)) = \frac{p(y_t|x_{t-1}(i))}{\sum_{j=1}^{n} p(y_t|x_{t-1}(j))}$. If $p(y_{t+1}|x_t^*)$ is smaller than a threshold chosen by us we reject x_t^* and sample another point. When we find x_t^* with $p(y_{t+1}|x_t^*)$ larger than the threshold we set $x_t(k) = x_t^*$ and move on to get $x_t(k+1)$. Therefore, in order to implement this method we must wait for y_{t+1} before we can obtain the sample from $p(x|\mathbf{y}_t)$. Two disadvantages of the method are the subjective choice of the threshold and the fact that in order to acquire $x_t(1), \ldots, x_t(n)$ a random number of draws from among $x_{t-1}(1), \ldots, x_{t-1}(n)$ is required. Moreover, if very few points among $x_{t-1}(1), \ldots, x_{t-1}(n)$ have large $p(y_{t+1}|x_{t-1}(i))$ we will still suffer from sample deterioration.

To demonstrate the sample deterioration caused by weighted bootstrap in Bayesian learning problems we conduct a simple experiment. We are studying a $N(\mu, 1)$ population from which we sequentially obtain observations $y_1, y_2, \ldots$ Our prior knowledge about μ is expressed by the prior distribution $p(\mu)$ which we take to be $N(0, 1)$. After the arrival of each new observation y_t the posterior of μ becomes $p(\mu|\mathbf{y}_t) = p(\mu|y_1, \ldots, y_t)$. Because both the prior and the model generating the data are Normal the posterior is also Normal. See, for example, Gelman *et al* (1997, p.45). The mean and variance of $p(\mu|\mathbf{y}_t)$ are

$$\mu_t = \frac{\sum_{i=1}^{t} y_i}{1+t} \quad \text{and} \quad \sigma_t^2 = \frac{1}{1+t}.$$

However, we assume that we do not have the posteriors in closed form. In order to sample from them we use weighted bootstrap in the way we demonstrated above. We start with a sample of size 50 from the prior and end up with a sample of the same size from each posterior. Here we only consider 8 observations from $N(\mu, 1)$. We take $\mu = 0$ so that our prior for μ is as good a guess of the "unknown" value as could be.

On each posterior sample we performed a Kolmogorov-Smirnov goodness-of-fit test to see whether it indeed came from its corresponding distribution. For all 8 samples the p-values of the test statistic were smaller than 0.007. Figure 1.1 shows the histogram of the prior sample and the histograms of the posterior samples. Over each histogram is superimposed a plot of the distribution the sample is supposed to come from. We can see how quickly the samples deteriorate. The sample from $p(\mu|\mathbf{y}_8)$ contains only 8 distinct values. A sample size larger than 50 would have given better results but would have just delayed deterioration.

Before closing this section we present two methods designed in order to speed up the weighted bootstrap. They are both based on the following observation. If $x_1, \ldots, x_n$ is the sample from the importance sampler, $q_1, \ldots, q_n$ are the corresponding normalized weights and n is the required size of the sample from the target, each x_i is expected to be picked nq_i times during the resampling. Liu and Chen (1995) suggest doing no resampling but including each x_i $[nq_i]$ times in the resulting sample, where $[k]$ is the integer part of k. Because $[nq_i] \leq nq_i$ we will have $\sum_{i=1}^{n}[nq_i] \leq n$. If $\sum_{i=1}^{n}[nq_i] < n$ we fill the $n - \sum_{i=1}^{n}[nq_i]$ remaining places by resampling from among $x_1, \ldots, x_n$ with the probabilities of selection now being $q_i - \frac{[nq_i]}{n}$. Beadle and Djuric (1997) propose to pick during resampling each of $x_1, \ldots, x_n$ with probability $\frac{1}{n}$. If x_k is picked we include in the resulting sample $[nq_k]$ copies of it. The same process is repeated until the resulting sample reaches size n. If a point x_i with $[nq_i] < 1$ is picked, one copy of it is included in the resulting sample but then it cannot be included in it again. Finally, see Carpenter *et al* (1997) for a faster algorithm for implementing the ordinary weighted bootstrap.

It must be noted that wherever resampling is used **rejection sampling** (Ripley (1987)) could be used instead. If for example, the prior distribution $p(x)$ is easy to sample from and an upper bound $C(y)$ can be found for the likelihood $p(y|x)$ then we can generate points x^* from the prior and points u from a Uniform distribution on (0,1) and accept x^* as coming from the posterior if $u \leq p(y|x^*)/C(y)$. The advantage of this method over resampling is that the resulting sample comes from the target even for finite sample sizes. The disadvantage is that the prior sample size is a random variable and it is not known in advance. See also Ripley and Sutherland (1990), Smith and Gelfand (1992), Acklam (1996) and Hurzeler and Kunsch (1998).

1.3.2 Sequential importance sampling

We deviate briefly from our course in order to present the method of **sequential importance sampling**. Although it does not involve resampling, familiarity with it will facilitate the demonstration of the application of weighted bootstrap in dynamic model settings.

The method first appears in Kong *et al* (1994), under the name of **sequential imputation**. The objective of that work is the analysis of missing data problems. More specifically, data $y_1, \ldots, y_N$ arrive sequentially. They are not complete because each y_t is accompanied by a missing part x_t. If θ is a parameter of interest we would like to study its posterior distribution $p(\theta|\mathbf{x}_N, \mathbf{y}_N) = p(\theta|x_1, y_1, \ldots, x_N, y_N)$ but since only $\mathbf{y}_N$ is available we have to work with $p(\theta|\mathbf{y}_N)$. The former is called the complete data posterior and the latter the incomplete data posterior. Suppose that the complete data posterior is available in closed form while the incomplete data one is not. Then, because

$$p(\theta|\mathbf{y}_N) = \int p(\theta|\mathbf{x}_N, \mathbf{y}_N) p(\mathbf{x}_N|\mathbf{y}_N) d\mathbf{x}_N, \tag{1.12}$$

if we had a sample $\mathbf{x}_N(1), \ldots, \mathbf{x}_N(n)$ from $p(\mathbf{x}_N|\mathbf{y}_N)$ we could estimate (1.12) by

$$\hat{p}(\theta|\mathbf{y}_N) = \frac{1}{n} \sum_{i=1}^{n} p(\theta|\mathbf{x}_N(i), \mathbf{y}_N).$$

If $p(\mathbf{x}_N|\mathbf{y}_N)$ is not available for sampling we can use importance sampling. We will draw $\mathbf{x}_N(1), \ldots, \mathbf{x}_N(n)$ from an importance sampler g and use the

estimator

$$\tilde{p}(\theta|\mathbf{y}_N) = \frac{\sum_{i=1}^{n} w_i p(\theta|\mathbf{x}_N(i), \mathbf{y}_N)}{\sum_{i=1}^{n} w_i} \tag{1.13}$$

with $w_i \propto \frac{p(\mathbf{x}_N(i)|\mathbf{y}_N)}{g(\mathbf{x}_N(i))}$. Note that g can depend on the observed data $\mathbf{y}_N$ so that it is close to the target $p(\mathbf{x}_N|\mathbf{y}_N)$.

This generation of multiple copies of the missing data values from their conditional distribution given the observed data is called **multiple imputation**. If it is of interest to study any of the intermediate posteriors $p(\theta|\mathbf{y}_t)$,$1 \leq t < N$, we will have to employ sequential imputation and get a sample $\mathbf{x}_t(1), \ldots, \mathbf{x}_t(n)$ from each $p(\mathbf{x}_t|\mathbf{y}_t)$. Suppose that at time t such a sample is available from $g(\mathbf{x}_t)$ and that each point $\mathbf{x}_t(i)$ has the appropriate importance weight $w_t(i)$. Then, when y_{t+1} arrives we will need a sample $\mathbf{x}_{t+1}(1), \ldots, \mathbf{x}_{t+1}(n)$ from an importance sampler $g(\mathbf{x}_{t+1})$ with weights $w_{t+1}(i)$. Sequential imputation, or sequential importance sampling as it is also known, allows us to acquire this new sample without having to throw away the former. To do this we express $g(\mathbf{x}_{t+1})$ as $g(x_{t+1}|\mathbf{x}_t)g(\mathbf{x}_t)$. For each $\mathbf{x}_t(i)$ we generate $x_{t+1}(i)$ from $g(x_{t+1}|\mathbf{x}_t(i))$ and get $\mathbf{x}_{t+1}(i) = (x_{t+1}(i), \mathbf{x}_t(i))$. Then its weight is

$$\begin{aligned} w_{t+1}(i) &\propto \frac{p(\mathbf{x}_{t+1}(i)|\mathbf{y}_{t+1})}{g(\mathbf{x}_{t+1}(i))} \propto \frac{p(y_{t+1}|\mathbf{x}_{t+1}(i))p(\mathbf{x}_{t+1}(i)|\mathbf{y}_t)}{g(x_{t+1}(i)|\mathbf{x}_t(i))g(\mathbf{x}_t(i))} \\ &= \frac{p(\mathbf{x}_t(i)|\mathbf{y}_t)}{g(\mathbf{x}_t(i))} \cdot \frac{p(y_{t+1}|\mathbf{x}_{t+1}(i))p(x_{t+1}(i)|\mathbf{x}_t(i), \mathbf{y}_t)}{g(x_{t+1}(i)|\mathbf{x}_t(i))} \\ &\propto w_t(i) \cdot \frac{p(y_{t+1}|\mathbf{x}_{t+1}(i))p(x_{t+1}(i)|\mathbf{x}_t(i), \mathbf{y}_t)}{g(x_{t+1}(i)|\mathbf{x}_t(i))}. \end{aligned}$$

Sequential imputation can be applied in any dynamic model if the unknown

states x_t are viewed as missing data. Note that if $p(\mathbf{x}_t|\mathbf{y}_t)$ is approximated by the discrete distribution with probability masses $\left\{\frac{w_t(1)}{\sum_{i=1}^n w_t(i)}, \ldots, \frac{w_t(n)}{\sum_{i=1}^n w_t(i)}\right\}$ over the support $\{\mathbf{x}_t(1), \ldots, \mathbf{x}_t(n)\}$ then the filtering distribution $p(x_t|\mathbf{y}_t)$ can be approximated by the discrete distribution with the same masses as above over the support $\{x_t(1), \ldots, x_t(n)\}$. This view is taken for example by Liu and Chen (1995) for the estimation of hidden states in a signal processing problem. It is just a small step to go from sequential imputation to the repetitive use of weighted bootstrap. We describe how this is done in the next section.

1.3.3 Weighted bootstrap and dynamic models

Imagine a dynamic model like those described in Section 1.2. We have a system whose unknown state at time t is x_t and data y_t are generated by the probabilistic model $y_t = A(x_t, \epsilon_t)$ which can be described by the distribution of y_t given x_t, $p(y_t|x_t)$. Then, at time $t+1$ the state becomes x_{t+1} which is generated by the probabilistic model $x_{t+1} = B(x_t, \eta_t)$ that can also be described by a conditional distribution $p(x_{t+1}|x_t)$. Our knowledge about the initial state x_1 is expressed by the distribution $p(x_1)$. We have the same assumptions about ϵ_t and η_t which we made in Section 1.2. At each time point t we want a sample $x_t(1), \ldots, x_t(n)$ from $p(x_t|\mathbf{y}_t)$, the filtering distribution. Suppose that these distributions are not available in closed form and that we cannot sample directly from them. We will use weighted bootstrap.

Recall formula (1.6). It is clear from it that at time t the prior $p(x_t|\mathbf{y}_{t-1})$ could be a good importance sampler for $p(x_t|\mathbf{y}_t)$. For the time being we are going to consider only this choice. In later chapters we will present other alternatives. Suppose that somehow we have managed to get a sample $x_t^*(1), \ldots, x_t^*(n)$ from

$p(x_t|\mathbf{y}_{t-1})$. Then the weight assigned to each $x_t^*(i)$ will be $w_t(i) = p(y_t|x_t^*(i))$. Application of weighted bootstrap will give a sample $x_t(1), \ldots, x_t(n)$ which, if n is large, can be considered as coming from $p(x_t|\mathbf{y}_t)$. We call this an "update" step because we mimic with samples the effect that (1.6) has on the distributions they come from.

In order to proceed to the next time point we must obtain a sample $x_{t+1}^*(1)$, $\ldots, x_{t+1}^*(n)$ from the next prior $p(x_{t+1}|\mathbf{y}_t)$. Notice that we maintain the same sample size n throughout. We assume for the time being that the distribution of η_t is known and that we can sample from it. We then obtain a sample $\eta_t(1), \ldots, \eta_t(n)$ from it and for each $x_t(i)$ we get $x_{t+1}^*(i) = B(x_t(i), \eta_t(i))$. This is equivalent to sampling $x_{t+1}^*(i)$ from $p(x_{t+1}|x_t(i))$. Then $x_{t+1}^*(1), \ldots, x_{t+1}^*(n)$ is the sample we want. We call this a "propagation" step because here we mimic the effect that (1.7) has on the distributions the samples come from.

Having started with a sample $x_1^*(1), \ldots, x_1^*(n)$ from $p(x_1)$ (also assuming that $p(x_1)$ is available for sampling) we can perpetuate this alternation of update and propagation steps to obtain samples from all the distributions we are interested in. For applications of the method to contour tracking and to clinical patient monitoring see Isard and Blake (1996) and Berzuini *et al* (1997) respectively.

Notice that, unlike MCMC, weighted bootstrap makes very economical use of the available resources. The propagation step means that all the knowledge expressed via the sample from the current posterior at each time is used in the next time point instead of having to start from scratch. Moreover we do not simulate any Markov chain which must converge to an equilibrium distribution. Propagation has also another very significant effect. Each posterior sample may contain less than n distinct values since it is the result of weighted

bootstrap. The propagation step eliminates all replicates because each $x_t(i)$ is associated with its own individual $\eta_t(i)$. Of course, for this effect to take place η_t has to be a continuous variable but this is usually true for system disturbances.

However, sometimes an unlikely y_t may turn up leading to most of $x_t^*(i)$ having very small $p(y_t|x_t^*(i))$. Then a few values will dominate during resampling and the resulting sample will be very bad. Although the next propagation step will correct things, inference about $p(x_t|\mathbf{y}_t)$ will be unreliable. We can use any of the corrective methods described in Section 1.3.1. If we want the prior sample at time $t+1$ to be of size larger than n we can pass each $x_t(i)$ more than once through the system equation, each time with a different η_t point. We can also use prior editing by getting each x_t point that results from resampling at time t, passing it through the system equation with its own individual η_t point and rejecting it if the result, $\tilde{x}_{t+1}$ say, gives $p(y_{t+1}|\tilde{x}_{t+1})$ smaller than a chosen threshold.

However, we will present in the next chapter methods that do not need to resort to such measures and which still manage to avoid sample deterioration.

Weighted bootstrap can also provide samples from any other distribution of interest. For example at some time t we may be interested in $p(x_{t-k}|\mathbf{y}_t)$, for $0 < k < t$. This is called the **smoothing distribution**. We may also be interested in prediction, i.e. in probabilities of the form $p(x_{t+k}|\mathbf{y}_t)$, for $0 < k$. For an article detailing the use of weighted bootstrap for filtering, smoothing and predicting in dynamic models, see Kitagawa (1996).

1.4 Properties of some descriptive statistics of a weighted bootstrap sample

When a sample of independent observations $Y_1, \ldots, Y_n$ is taken from any population it is well known that the sample mean and variance are unbiased estimators of the respective population parameters while the variance of the sample mean is n times smaller than the population variance. If a sample is obtained by weighted bootstrap we would like its statistics to have the same properties. This seems to be the case only for the expectations of the sample statistics and only in the limit as $n \longrightarrow \infty$.

Suppose that $\mathbf{X} = (X_1, \ldots, X_n)$ is a sample from a distribution g while the target is another distribution $h \propto f$. Suppose that via weighted bootstrap we obtain $Y_1, \ldots, Y_n$. Define the sample statistics as

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \quad , \quad \hat{\sigma}_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2.$$

Their expectations can be calculated by a two-stage process. We first calculate expectations conditional on $\mathbf{X}$ and then calculate the expectations of the results with respect to g. Mathematical details are given in Appendix A.1 but the final results are

$$E_g(\bar{Y}) \approx E_h(X) - \frac{1}{n} Cov_h\left(X, \frac{h(X)}{g(X)}\right), \tag{1.14}$$

$$E_g(\hat{\sigma}^2_{\bar{Y}}) \approx V_h(X) - \frac{1}{n}Cov_h\left(X^2, \frac{h(X)}{g(X)}\right) + \frac{3}{n}E_h(X)Cov_h\left(X, \frac{h(X)}{g(X)}\right) - \frac{1}{n}Cov_h\left(X, X\frac{h(X)}{g(X)}\right). \quad (1.15)$$

Of course it is assumed that all the expectations involved exist and are finite. As we can see the non-leading terms in (1.14) and (1.15) go to 0 as $n \longrightarrow \infty$.

More credibility is lent to these results by a result mentioned in Geweke (1989), namely that under weak conditions

$$\frac{\sum_{i=1}^n \frac{f(x_i)}{g(x_i)}k(x_i)}{\sum_{i=1}^n \frac{f(x_i)}{g(x_i)}} \xrightarrow{\text{a.s.}} E_h\left[k(X)\right], \quad (1.16)$$

where k is any function of X such that its expectation exists and is finite. Notice that if $k(x) = x$ the ratio on the left-hand side of (1.16) is just $E[\bar{Y}|\mathbf{X}]$. (1.16) shows that one could easily generalize (1.14) and (1.15) for the sample mean and variance of any function $k(X)$, i.e. for

$$\frac{1}{n}\sum_{i=1}^n k(Y_i) \quad , \quad \frac{1}{n-1}\sum_{i=1}^n \left(k(Y_i) - \frac{1}{n}\sum_{i=1}^n k(Y_i)\right)^2.$$

As regards the variance of $\bar{Y}$, taking into account that the $Y_1, \ldots, Y_n$ are not independent, we arrive at the following result

$$V_g(\bar{Y}) \approx \frac{1}{n}V_h(X) + \left(\frac{3}{n^2} - \frac{1}{n}\right)E_h(X)Cov_h\left(X, \frac{h(X)}{g(X)}\right)$$

$$+ \quad \frac{n-1}{n^2} Cov_h\left(X, X\frac{h(X)}{g(X)}\right) - \frac{1}{n^2}\left[Cov_h\left(X, \frac{h(X)}{g(X)}\right)\right]^2$$
$$- \quad \frac{1}{n^2} Cov_h\left(X^2, \frac{h(X)}{g(X)}\right). \tag{1.17}$$

Here we see that there are non-leading terms going to zero at the same rate as the leading one. Therefore, the variance of the mean is not what we would like it to be. The simulations will show that it is actually larger.

If a sample consists of independent observations its mean is asymptotically Normal. In a weighted bootstrap sample the observations are not independent as we have seen. Berzuini *et al* (1997) however, prove a central limit theorem for the mean of a weighted bootstrap sample. In order to translate it into our case we first present it in the context in which it is proven by the authors. They refer to a dynamic model problem where at time t, the target of resampling is $p_t(\mathbf{x}_t)$, with $\mathbf{x}_t = (\mathbf{x}_{t-1}, x_t)$ and $\mathbf{x}_1 = x_1$. They implement resampling as follows. At $t = 1$ they obtain $x_1(1), \ldots, x_1(n_1)$ from $p_1(x_1)$. At time $t = k$ they draw $x_k(i)$ from a density $f_k(x_k|\mathbf{x}_{k-1}(i))$, for $i = 1, \ldots, n_{k-1}$. Then each point $(\mathbf{x}_{k-1}(i), x_k(i))$ gets weight

$$\frac{p_k(\mathbf{x}_{k-1}(i), x_k(i))}{p_{k-1}(\mathbf{x}_{k-1}(i)) f_k(x_k(i)|\mathbf{x}_{k-1}(i))}.$$

Resampling with probabilities proportional to these weights gives sample $\mathbf{x}_k(1), \ldots, \mathbf{x}_k(n_k)$, allegedly from $p_k(\mathbf{x}_k)$. Define the quantities

$$E_{f_{k,t+1}}(c|\mathbf{x}_t) = \begin{cases} \int c(\mathbf{x}_k) \prod_{l=t+1}^{k} f_l(x_l|\mathbf{x}_{l-1}) dx_l & \text{if} \quad t < k \\ c(\mathbf{x}_k) & \text{otherwise} \end{cases},$$

$$E_{p_t f_{t+1}}(c) = \int c(\mathbf{x}_{t+1}) f_{t+1}(x_{t+1}|\mathbf{x}_t) p_t(\mathbf{x}_t) d\mathbf{x}_{t+1}$$

and

$$w_{kt}(\mathbf{x}_k) = \begin{cases} \frac{p_k(\mathbf{x}_k)}{p_t(\mathbf{x}_t)\prod_{l=t+1}^{k} f_l(x_l|\mathbf{x}_{l-1})} & \text{if} \quad t < k \\ 1 & \text{otherwise} \end{cases}.$$

Then, if $\bar{c}_k = \sum_{i=1}^{n_k} c(\mathbf{x}_k(i))/n_k$, the authors prove that

$$\frac{\bar{c}_k - E_{p_k}(c)}{\sqrt{\sum_{l=1}^{k} \frac{1}{n_l} u_{k1l}}} \xrightarrow{\mathrm{D}} N(0,1), \quad \text{as} \quad n_1, \ldots, n_k \longrightarrow \infty$$

where $u_{k1l} = E_{p_l f_{l+1}}\left(E^2_{f_{k,l+2}}\left[(c - E_{p_k}(c))w_{kl}|\mathbf{x}_{l+1}\right]\right)$. For $k = 2$ we get

$$\frac{\bar{c}_2 - E_{p_2}(c)}{\sqrt{\frac{1}{n_1}u_{211} + \frac{1}{n_2}u_{212}}} \xrightarrow{\mathrm{D}} N(0,1), \quad \text{as} \quad n_1, n_2 \longrightarrow \infty \tag{1.18}$$

where

$$u_{211} = E_{p_1 f_2}\left[(c(\mathbf{x}_2) - E_{p_2}(c))^2 \left(\frac{p_2(\mathbf{x}_2)}{p_1(x_1)f_2(x_2|x_1)}\right)^2\right] \quad \text{and} \quad u_{212} = V_{p_2}(c).$$

In our case consider $h \equiv p_2$, $g \equiv p_1 f_2$ and $c(x) = x$. Then (1.18) says that

$$\frac{\bar{Y} - E_h(X)}{\sqrt{\frac{1}{n}(u_{211} + u_{212})}} \xrightarrow{\mathrm{D}} N(0,1), \quad \text{as} \quad n \longrightarrow \infty \tag{1.19}$$

with

$$u_{211} = E_g\left[(X - E_h(X))^2 \left(\frac{h(X)}{g(X)}\right)^2\right] \quad \text{and} \quad u_{212} = V_h(X).$$

Note that expanding $(u_{211} + u_{212})/n$ will give the terms that in (1.17) are divided by n. Equation (1.19) shows that even asymptotically the variance of the sample mean is larger than the variance of the mean of a random sample from the target. Of course (1.19) will hold for any function $c(x)$. To obtain all the results above it is assumed that the involved expectations exist and are finite.

1.4.1 Simulation study

In this section we present a simulation study conducted in order to examine the validity of formulae (1.14), (1.15) and (1.17) and of the asymptotic Normality of the sample mean.

Inference about a Normal population $N(\mu, \sigma^2)$ is based on a set of observations $y_1, \ldots, y_k$ obtained from it. If the mean μ of the population is unknown, while the variance σ^2 is known, we assign to μ a prior which usually is $N(\mu_0, \sigma_0^2)$, for specified μ_0, σ_0^2. The pdf of the observed data viewed as a function of μ,

$$l(\mu) \propto \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{k}(y_i - \mu)^2\right),$$

resembles a Normal density with mean $\bar{y} = \frac{\sum_{i=1}^{k} y_i}{k}$ and variance $\frac{\sigma^2}{k}$. The posterior distribution of μ is also Normal, $N(\mu_1, \sigma_1^2)$ with $\sigma_1^2 = \left(\frac{1}{\sigma_0^2} + \frac{k}{\sigma^2}\right)^{-1}$ and $\mu_1 = \sigma_1^2 \left(\frac{\mu_0}{\sigma_0^2} + \frac{k\bar{y}}{\sigma^2}\right)$. If on the other hand μ is known but σ^2 is unknown we assign to σ^2 a prior distribution which usually is $S_0\chi_\nu^{-2}$. This simply means that we assume that *a priori* $\frac{S_0}{\sigma^2} \sim \chi_\nu^2$ and that $E(\sigma^2) = \frac{S_0}{\nu-2}$ and $V(\sigma^2) = \frac{2S_0^2}{(\nu-2)^2(\nu-4)}$. If we are happy with a $S_0\chi_\nu^{-2}$ prior for σ^2 we can choose a particular one according to our beliefs about the mean and variance of σ^2. Taking $S = \sum_{i=1}^{k}(y_i - \mu)^2$ the pdf of the observed data viewed as a function of σ^2,

$$l(\sigma^2) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{k}{2}} \exp\left(-\frac{S}{2\sigma^2}\right),$$

resembles a $S\chi_{k-2}^{-2}$ distribution. The posterior of σ^2 can be proved to be $(S_0 + S)\chi_{\nu+k}^{-2}$.

For our experiment we took $\mu = 0$ and $\sigma^2 = 1$. We drew $k = 10$ observations from the population. The sample statistics were $\bar{y} = -0.2031$ and $S = 8.7545$.

Irrespective of which of the two parameters is the unknown one, the pdf of the data (or the likelihood) can be viewed as a member of the same distributional family as the prior. This is what is called **conjugacy**. We can then choose a prior that closely resembles the likelihood or a prior that is far from it. In our study we considered the following cases.

For μ unknown we took the following four priors.

$N(-0.15, 0.2)$,

$N(-0.15, 1.5)$,

$N(0.15, 1.5)$,

$N(0.15, 0.2)$.

For σ^2 unknown the likelihood resembles a distribution of the $S\chi_\nu^{-2}$ family with mean 1.4591 and variance 1.0645. We then considered the following four priors for σ^2.

$S_0 = 11.2, \nu = 9 \quad \rightarrow \quad$ mean=1.6 , var=1.024,

$S_0 = 4.8, \nu = 5 \quad \rightarrow \quad$ mean=1.6 , var=5.12,

$S_0 = 260, \nu = 54 \quad \rightarrow \quad$ mean=5 , var=1,

$S_0 = 60, \nu = 14 \quad \rightarrow \quad$ mean=5 , var=5.

The aim of the study was, considering each prior in turn, to take 1000 samples of size n from it and update each one of them by weighted bootstrap to transform them into samples from the corresponding posterior. We stored the mean and variance of each of the resulting samples. Then, if we want to see whether the weighted bootstrap sample mean is an unbiased estimator of the posterior mean we can average the 1000 simulated sample means. If we want to see whether the weighted bootstrap sample mean is asymptotically Normal we can draw a histogram of the 1000 sample means or perform a Kolmogorov-Smirnov goodness-of-fit test on them. The true posterior means and variances are available for comparison in all cases. We considered two different sample sizes, $n = 100$ and $n = 1000$. For all the simulations we used the same ten observations from $N(0, 1)$.

Tables 1.1 and 1.2 refer to the case of unknown mean μ. Table 1.1 presents the mean and variance of each posterior. We have also calculated for each sample size and prior the values of formulae (1.14),(1.15) and (1.17) and

present them in the rest of Table 1.1. We see that the expectations of the sample statistics are close to the corresponding posterior values. Only for the expectation of the variance and for $n = 100$ do we see a discrepancy from the corresponding posterior value. Note also that the variance of the sample mean is two to three times larger than one nth of the posterior variance. Table 1.2 presents the simulated expectations and variances of the sample statistics. All of them are close to their theoretical value. Only for the expectation of the sample variance with $n = 100$ is the simulated value closer to the posterior value rather than the theoretical one. Tables 1.3 and 1.4 refer to the case of unknown variance. We observe the same pattern there too, only this time all the expectations are close to the posterior values.

We performed Kolmogorov-Smirnov tests on the samples of sample means for all the cases considered. We always tested against the Normal distribution with mean and variance given by (1.14) and (1.17). Only for the case of unknown variance, S_0=60, ν=14 and n=1000 did we get a p-value smaller than 0.05, namely 0.006. Note that for this case when we tested against a Normal distribution with mean and variance equal to the simulated ones Normality was not rejected. Figure 1.2 shows the histograms of the 1000 simulated sample means and variances for each posterior in the case of unknown mean. We can see the values clustering around their theoretical expectations and that the histograms of the sample means do not refute the notion of Normality. The same happens in Figure 1.3 which refers to the case of unknown variance. Here, although the posterior is not Normal the sample means still seem to follow a Normal distribution.

1.5 Summary and discussion

In this chapter we have presented an alternative to MCMC for sampling from intractable distributions. This method, the weighted bootstrap, is not yet as widespread as MCMC and it probably never will be. This is because of two fundamental flaws that it has. The first is that it approximates the density it samples from by a discrete density no matter what the nature of the target density is. The second and worse problem is that the samples it produces usually contain more than one copy of some of their members. In some cases they may exclusively contain many copies of very few distinct points.

However we believe that when fast analysis of a dynamic model is needed weighted bootstrap is unchallenged. While MCMC has to start from scratch in order to sample from each filtering distribution, weighted bootstrap can give samples very quickly and, furthermore, can propagate them through time in a way that mimics the evolution of the filtering density. This propagation partly reverses the effects of sample deterioration as well. Moreover, MCMC is further slowed down by the need to monitor the convergence of the Markov chain being simulated. Weighted bootstrap on the other hand has simple to calculate diagnostics, like the effective sample size, that monitor its performance. Therefore, although MCMC will still be preferable for an off-line analysis, when speed is required weighted bootstrap is the choice.

We know that samples resulting from weighted bootstrap can asymptotically be considered as coming from the target density. We have proved in this chapter that the sample mean and variance are asymptotically unbiased estimators of the corresponding parameters of the target. However, it turns out that the variance of the sample mean is actually larger than that of the

mean of a random sample from the target. Moreover, although the weighted bootstrap sample does not consist of independent observations a central limit theorem can be established for the sample mean.

Although weighted bootstrap is intuitively simple, care is needed in its implementation. We must choose an importance sampler that covers the support of the target adequately. Otherwise, if all the sample points we get fall in an area assigned small probability by the target we may get a large ESS and a false impression that everything goes well. Before applying the weighted bootstrap some exploratory analysis of the target must be performed so that no important parts of it are missed by the importance sampler. Such analysis of course is the prerequisite of any method for sampling from intractable distributions.

In the next chapter we are going to present methods that modify the ordinary bootstrap so that it gets rid of some of its flaws while it maintains all its virtues.

Prior	$N(-0.15, 0.2)$	$N(-0.15, 1.5)$	$N(0.15, 1.5)$	$N(0.15, 0.2)$
Posterior	$\mu_1 = -0.1854$ $\sigma_1^2 = 0.0666$	$\mu_1 = -0.1998$ $\sigma_1^2 = 0.0938$	$\mu_1 = -0.181$ $\sigma_1^2 = 0.0938$	$\mu_1 = -0.0854$ $\sigma_1^2 = 0.0666$
n=100	$E(\bar{Y}) = -0.1853$ $V(\bar{Y}) = 0.0012$ $E(\sigma_Y^2) = 0.0635$	$E(\bar{Y}) = -0.1997$ $V(\bar{Y}) = 0.0023$ $E(\sigma_Y^2) = 0.0867$	$E(\bar{Y}) = -0.1807$ $V(\bar{Y}) = 0.0023$ $E(\sigma_Y^2) = 0.0873$	$E(\bar{Y}) = -0.0847$ $V(\bar{Y}) = 0.0013$ $E(\sigma_Y^2) = 0.0649$
n=1000	$E(\bar{Y}) = -0.1854$ $V(\bar{Y}) = 0.00012$ $E(\sigma_Y^2) = 0.0664$	$E(\bar{Y}) = -0.1998$ $V(\bar{Y}) = 0.00023$ $E(\sigma_Y^2) = 0.0931$	$E(\bar{Y}) = -0.181$ $V(\bar{Y}) = 0.00024$ $E(\sigma_Y^2) = 0.0931$	$E(\bar{Y}) = -0.0853$ $V(\bar{Y}) = 0.00013$ $E(\sigma_Y^2) = 0.0665$

Table 1.1: Posterior means and variances and expectations and variances of weighted bootstrap sample statistics in the case of inference for the mean of $N(\mu, 1)$.

Prior	$N(-0.15, 0.2)$	$N(-0.15, 1.5)$	$N(0.15, 1.5)$	$N(0.15, 0.2)$
n=100	$\hat{E}(\bar{Y}) = -0.1844$ $\hat{V}(\bar{Y}) = 0.0012$ $\hat{E}(\sigma_Y^2) = 0.0669$	$\hat{E}(\bar{Y}) = -0.1998$ $\hat{V}(\bar{Y}) = 0.0024$ $\hat{E}(\sigma_Y^2) = 0.0936$	$\hat{E}(\bar{Y}) = -0.183$ $\hat{V}(\bar{Y}) = 0.0024$ $\hat{E}(\sigma_Y^2) = 0.0935$	$\hat{E}(\bar{Y}) = -0.0847$ $\hat{V}(\bar{Y}) = 0.0013$ $\hat{E}(\sigma_Y^2) = 0.0664$
n=1000	$\hat{E}(\bar{Y}) = -0.1857$ $\hat{V}(\bar{Y}) = 0.00012$ $\hat{E}(\sigma_Y^2) = 0.0666$	$\hat{E}(\bar{Y}) = -0.2004$ $\hat{V}(\bar{Y}) = 0.00024$ $\hat{E}(\sigma_Y^2) = 0.0938$	$\hat{E}(\bar{Y}) = -0.181$ $\hat{V}(\bar{Y}) = 0.00025$ $\hat{E}(\sigma_Y^2) = 0.0939$	$\hat{E}(\bar{Y}) = -0.0847$ $\hat{V}(\bar{Y}) = 0.00013$ $\hat{E}(\sigma_Y^2) = 0.0664$

Table 1.2: Simulated expectations and variances of weighted bootstrap sample statistics in the case of inference for the mean of $N(\mu, 1)$.

Prior	$\mu_0 = 1.6, \sigma_0^2 = 1.024$ $S_0 = 11.2, \nu = 9$	$\mu_0 = 1.6, \sigma_0^2 = 5.12$ $S_0 = 4.8, \nu = 5$	$\mu_0 = 5, \sigma_0^2 = 1$ $S_0 = 260, \nu = 54$	$\mu_0 = 5, \sigma_0^2 = 5$ $S_0 = 60, \nu = 14$
Posterior	$\mu_1 = 1.1738$ $\sigma_1^2 = 0.1837$	$\mu_1 = 1.0427$ $\sigma_1^2 = 0.1977$	$\mu_1 = 4.3348$ $\sigma_1^2 = 0.6263$	$\mu_1 = 3.1252$ $\sigma_1^2 = 0.9767$
n=100	$E(\bar{Y}) = 1.1752$ $V(\bar{Y}) = 0.0032$ $E(\sigma_Y^2) = 0.1834$	$E(\bar{Y}) = 1.0437$ $V(\bar{Y}) = 0.0033$ $E(\sigma_Y^2) = 0.1978$	$E(\bar{Y}) = 4.3427$ $V(\bar{Y}) = 0.0169$ $E(\sigma_Y^2) = 0.615$	$E(\bar{Y}) = 3.1477$ $V(\bar{Y}) = 0.0373$ $E(\sigma_Y^2) = 0.951$
n=1000	$E(\bar{Y}) = 1.1739$ $V(\bar{Y}) = 0.00031$ $E(\sigma_Y^2) = 0.1837$	$E(\bar{Y}) = 1.0428$ $V(\bar{Y}) = 0.00033$ $E(\sigma_Y^2) = 0.1977$	$E(\bar{Y}) = 4.3355$ $V(\bar{Y}) = 0.0017$ $E(\sigma_Y^2) = 0.6252$	$E(\bar{Y}) = 3.1275$ $V(\bar{Y}) = 0.0038$ $E(\sigma_Y^2) = 0.9742$

Table 1.3: Posterior means and variances and expectations and variances of weighted bootstrap sample statistics in the case of inference for the variance of $N(0, \sigma^2)$.

Prior	$\mu_0 = 1.6, \sigma_0^2 = 1.024$ $S_0 = 11.2, \nu = 9$	$\mu_0 = 1.6, \sigma_0^2 = 5.12$ $S_0 = 4.8, \nu = 5$	$\mu_0 = 5, \sigma_0^2 = 1$ $S_0 = 260, \nu = 54$	$\mu_0 = 5, \sigma_0^2 = 5$ $S_0 = 60, \nu = 14$
n=100	$\hat{E}(\bar{Y}) = 1.1738$ $\hat{V}(\bar{Y}) = 0.0031$ $\hat{E}(\sigma_Y^2) = 0.1812$	$\hat{E}(\bar{Y}) = 1.0446$ $\hat{V}(\bar{Y}) = 0.0033$ $\hat{E}(\sigma_Y^2) = 0.1989$	$\hat{E}(\bar{Y}) = 4.346$ $\hat{V}(\bar{Y}) = 0.0169$ $\hat{E}(\sigma_Y^2) = 0.6107$	$\hat{E}(\bar{Y}) = 3.1502$ $\hat{V}(\bar{Y}) = 0.0377$ $\hat{E}(\sigma_Y^2) = 0.9413$
n=1000	$\hat{E}(\bar{Y}) = 1.175$ $\hat{V}(\bar{Y}) = 0.00031$ $\hat{E}(\sigma_Y^2) = 0.1847$	$\hat{E}(\bar{Y}) = 1.0428$ $\hat{V}(\bar{Y}) = 0.00033$ $\hat{E}(\sigma_Y^2) = 0.1971$	$\hat{E}(\bar{Y}) = 4.3351$ $\hat{V}(\bar{Y}) = 0.00174$ $\hat{E}(\sigma_Y^2) = 0.6253$	$\hat{E}(\bar{Y}) = 3.1336$ $\hat{V}(\bar{Y}) = 0.0041$ $\hat{E}(\sigma_Y^2) = 0.9733$

Table 1.4: Simulated expectations and variances of weighted bootstrap sample statistics in the case of inference for the variance of $N(0, \sigma^2)$.

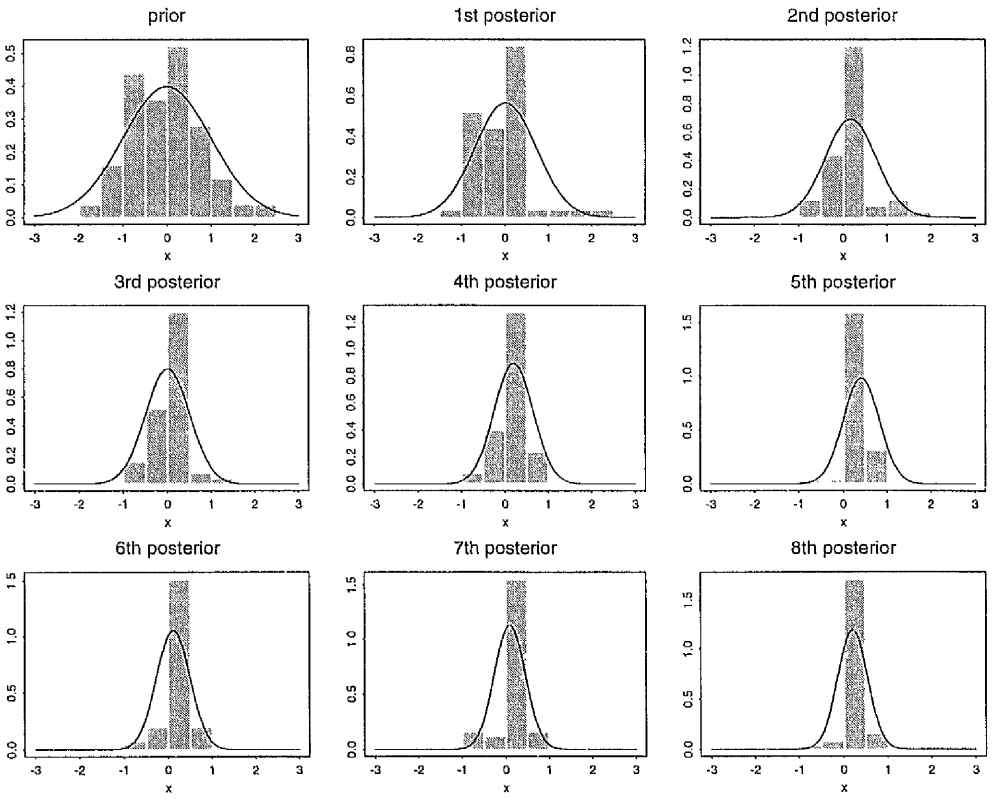


Figure 1.1: Prior and posterior samples for the mean of a $N(\mu, 1)$ population. The prior is $N(0, 1)$ and the posteriors are based on 8 data points taken sequentially from the $N(\mu, 1)$ distribution with $\mu = 0$. Bayesian learning has been used. The posterior samples were obtained via weighted bootstrap. The curves are the corresponding true posteriors.

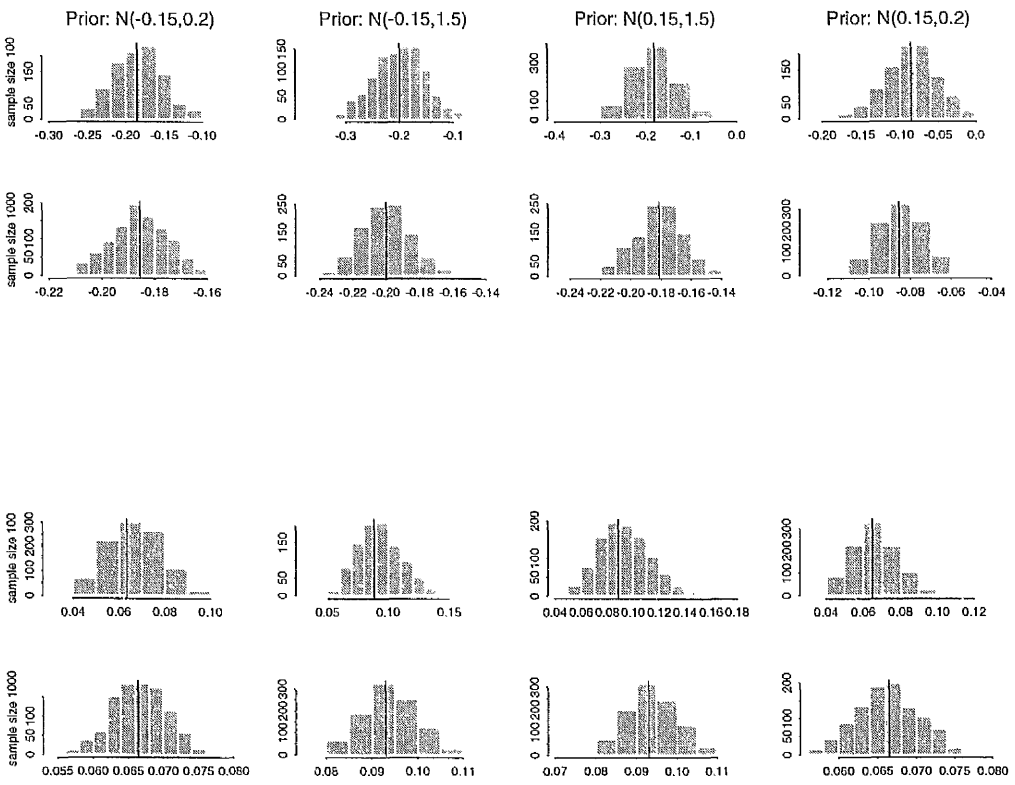


Figure 1.2: Means and variances of 1000 weighted bootstrap samples from the posterior of the mean of a $N(\mu, 1)$ population for four different choices of prior. The vertical lines denote the location of the expected value of the sample means and variances according to formulae (1.14) and (1.15). The top two rows show histograms of means and the bottom two show histograms of variances.

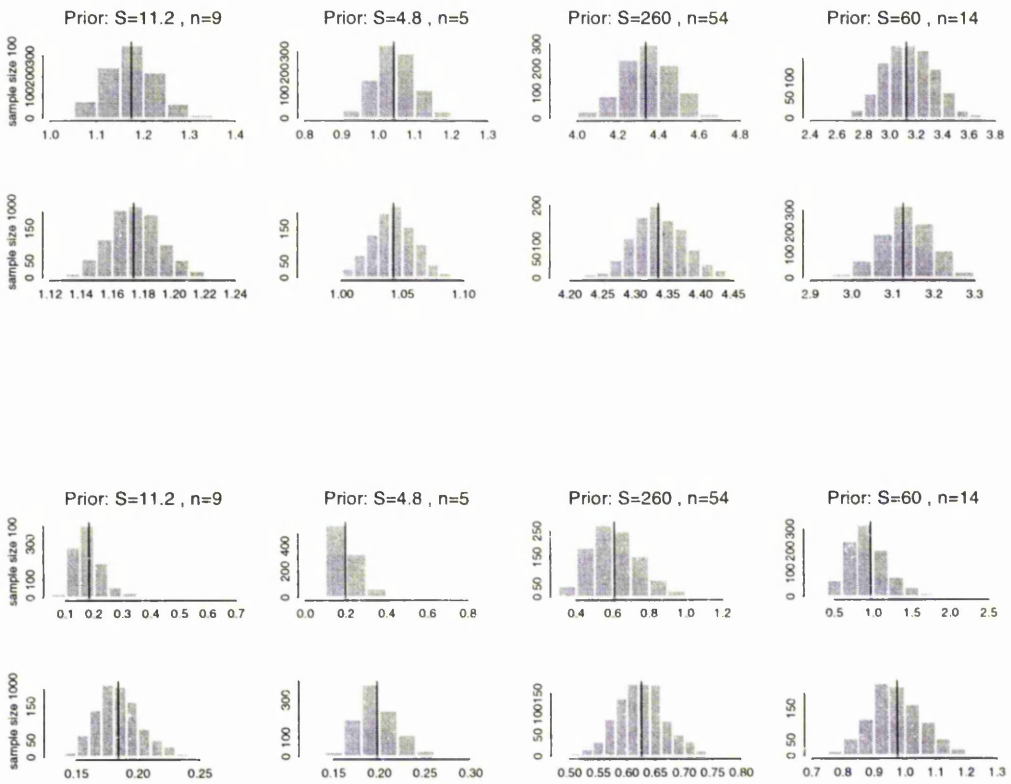


Figure 1.3: Means and variances of 1000 weighted bootstrap samples from the posterior of the variance of a $N(0, \sigma^2)$ population for four different choices of $S\chi_\nu^{-2}$ prior. The vertical lines denote the location of the expected value of the sample means and variances according to formulae (1.14) and (1.15). The top two rows show histograms of means and the bottom two show histograms of variances.

Chapter 2

Improving the weighted bootstrap

2.1 Introduction

In the previous chapter we presented the weighted bootstrap as an appealing alternative to MCMC for sampling from intractable distributions. The samples it generates have properties that resemble some of the properties of samples taken directly from the distributions of interest. It is also very straightforward to implement. It is especially advantageous over MCMC in situations where we need samples from many complicated distributions in a short time and with the least intervention from the user. An example of such a case is the unsupervised real-time analysis of a nonlinear and/or non-Gaussian dynamic system.

However, we also highlighted two disadvantages of the method. One of them is the deterioration experienced in the samples it produces which could be due

to many reasons: bad choice of an importance sampler, presence of unusual data or repetitive application of the method on the same pool of points. The other is that the distribution weighted bootstrap used to approximate the target is always discrete.

In this chapter we present some methods designed to overcome these problems maintaining at the same time, as far as possible, the good properties of weighted bootstrap. Whereas the latter samples $x_1, \ldots, x_n$ from the importance sampler and approximates the target distribution by a discrete one with masses $\{q_1, \ldots, q_n\}$ over the sample points, some of the corrective methods approximate the target by a continuous density constructed by $\{x_1, \ldots, x_n\}$ and $\{q_1, \ldots, q_n\}$. Others intervene after weighted bootstrap has been applied. In the next two sections we present methods from the former group. Then we present the latter group and finally we compare them all in several settings. The chapter closes with a general discussion of our work in the first two chapters.

2.2 Smooth weighted bootstrap

This is the most promising of all the corrective techniques. It is a modification of ordinary weighted bootstrap and was first presented in Gordon (1993). The setting is the same as before. We have a sample $x_1, \ldots, x_n$ from an importance sampler g and we want a sample from another distribution with density $h(x) = \frac{f(x)}{\int f(x)dx}$ when we have only $f(x)$ in closed form. Each point x_i is assigned its weight w_i and its normalized weight q_i.

Weighted bootstrap would sample $y_1, \ldots, y_m$ from among $x_1, \ldots, x_n$ with

$\Pr(y_i = x_j) = q_j$, for any i and j. Smooth weighted bootstrap, or smooth bootstrap as we will call it for brevity, samples $y_1, \ldots, y_m$ from a mixture of n symmetric distributions with means $x_1, \ldots, x_n$ and mixing weights $q_1, \ldots, q_n$. In other words, in order to get y_i we first sample a point from among $x_1, \ldots, x_n$ with probabilities of selection $q_1, \ldots, q_n$ and then draw y_i from the symmetric density with mean the sampled point, x_k say. We call these densities **smoothing kernels**. Denote by K the smoothing kernel and by $K(x; \mu)$ the value at point x of the smoothing kernel with mean μ. Smooth bootstrap approximates the target density by the continuous density

$$\hat{p}(x) = \sum_{i=1}^{n} q_i K(x; x_i). \tag{2.1}$$

As a result the samples it produces do not suffer from sample deterioration. Note also that like weighted bootstrap the samples produced by smooth bootstrap do not consist of independent observations.

We want smooth bootstrap to produce samples that "converge" to samples coming from the target density as n tends to infinity. Weighted bootstrap achieves this and it can be viewed as a case of smooth bootstrap with the smoothing kernels having zero variance. If we take the variance of the smoothing kernels to tend to zero as n goes to infinity, smooth bootstrap as a method will approach weighted bootstrap and the samples it produces should have the desired property.

We have found this to be true if a sufficient condition holds, namely that $E_g\left[\left(\frac{f(X)}{g(X)}\right)^2\right] < \infty$. We prove this in a special case. The target is a univariate continuous density, the smoothing kernel used is the Normal density and we have the same variance b_n^2 for all n smoothing kernels. Then (2.1) is written

as

$$p(x) = \sum_{i=1}^{n} q_i N(x; x_i, b_n^2), \tag{2.2}$$

where $N(x; \mu, \sigma^2)$ is the value at point x of the $N(\mu, \sigma^2)$ density. We denote the cumulative distribution function of the standard Normal distribution by Φ. All the limits mentioned are for $n \longrightarrow \infty$, and b_n is such that $b_n \longrightarrow 0$. Suppose that θ^* is a random variable with distribution (2.2). Then,

$$\begin{aligned} \Pr(\theta^* \leq a) &= \sum_{i=1}^{n} q_i \Pr(\theta^* \leq a | \theta^* \sim N(x_i, b_n^2)) \\ &= \frac{\frac{1}{n}\sum_{i=1}^{n} w_i \Phi\left(\frac{a-x_i}{b_n}\right)}{\frac{1}{n}\sum_{i=1}^{n} w_i} \end{aligned}$$

We derive the limit of numerator and denominator separately. For the denominator we have

$$\frac{1}{n}\sum_{i=1}^{n} w_i \longrightarrow E_g\left(\frac{f(X)}{g(X)}\right) = \int_{-\infty}^{\infty} f(x)dx.$$

For the numerator define

$$S_n = \frac{1}{n}\sum_{i=1}^{n} w_i \Phi\left(\frac{a-x_i}{b_n}\right)$$

and

$$\mu_n = E_g(S_n) = E_g\left[\frac{f(X)}{g(X)}\Phi\left(\frac{a-X}{b_n}\right)\right] = \int_{-\infty}^{\infty} f(x)\Phi\left(\frac{a-x}{b_n}\right)dx.$$

When n goes to infinity $\Phi\left(\frac{a-x}{b_n}\right) \longrightarrow I(x \leq a)$ where I is the indicator function. Therefore

$$\mu_n \longrightarrow \mu_\infty = \int_{-\infty}^{\infty} f(x)I(x \leq a)dx = \int_{-\infty}^{a} f(x)dx.$$

If $S_n \longrightarrow \mu_\infty$ then

$$\Pr(\theta^* \leq a) \longrightarrow \frac{\int_{-\infty}^{a} f(x)dx}{\int_{-\infty}^{\infty} f(x)dx} = \int_{-\infty}^{a} h(x)dx = \Pr_h(\theta^* \leq a). \tag{2.3}$$

We now prove that

$$S_n \longrightarrow \mu_\infty. \tag{2.4}$$

For any positive real number ϵ

$$\begin{aligned} \Pr(|S_n - \mu_\infty| > \epsilon) &\leq \frac{E_g[(S_n-\mu_\infty)^2]}{\epsilon^2} = \frac{Var_g(S_n) + (\mu_n - \mu_\infty)^2}{\epsilon^2} \\ &= \frac{\sigma_n^2}{n\epsilon^2} + \frac{(\mu_n-\mu_\infty)^2}{\epsilon^2}, \end{aligned} \tag{2.5}$$

where

$$\sigma_n^2 = Var_g\left[\frac{f(X)}{g(X)}\Phi\left(\frac{a-X}{b_n}\right)\right] = \int_{-\infty}^{\infty}\frac{f^2(x)}{g(x)}\Phi^2\left(\frac{a-x}{b_n}\right)dx - \mu_n^2.$$

However,

$$\sigma_n^2 \longrightarrow \sigma_\infty^2 = \int_{-\infty}^{a}\frac{f^2(x)}{g(x)}dx - \mu_\infty^2.$$

If $E_g\left[\left(\frac{f(X)}{g(X)}\right)^2\right] < \infty$ is finite so is $\int_{-\infty}^{a}\frac{f^2(x)}{g(x)}dx$ and therefore so is σ_∞^2. In this case $\frac{\sigma_n^2}{n} \longrightarrow 0$. Therefore from (2.5) we get

$$\Pr(|S_n - \mu_\infty| > \epsilon) \longrightarrow 0$$

and consequently (2.4) and (2.3) are true. This completes the proof.

For any symmetric kernel in the place of the standard Normal density the result should still be valid. The same could be said of the multivariate case. Intuitively though it seems plausible that convergence must be slower than that of the weighted bootstrap sample. To apply smooth bootstrap in a multivariate setting a variance matrix should be chosen for the smoothing kernels. Its entries should all go to zero as n goes to infinity. It is easier to define the variance matrix as $B_n = b_n^2 \cdot B$ with b_n going to zero and B some matrix with finite entries.

We have said that the variance of the smoothing kernels must go to zero as n goes to infinity but we have not said what that variance should be. The

technique of **kernel density estimation** comes to our aid in tackling this problem. Kernel density estimation, see Silverman (1986), tries to estimate the unknown probability density function (pdf) of a distribution based on a sample from that distribution. The estimator is another pdf formed as a mixture of distributions. It has as many components as the sample size, the components are equally weighted and their means are the sample points themselves. The distributions forming the mixture are called kernels, hence the name of the technique. Usually the same form of kernel with the same variance is used for all sample points. The estimator is a random function since it depends on a random sample. If h is the unknown pdf and $\hat{h}$ the estimator, a measure of the estimator's performance is the **mean integrated square error** (MISE),

$$\text{MISE} = E\left(\int (h(x) - \hat{h}(x))^2 dx\right).$$

The best estimator is the one that provides the minimum MISE. The value of MISE depends on the unknown h, the kernels used and their variance. The form of the kernel is not so important and usually the Normal density is used because of its nice mathematical properties. The optimal value of the variance for a given type of kernel is found by minimizing MISE. It depends on the unknown h and on the sample size, n say. Usually it is assumed that h comes from a certain family of distributions, Normal for example, and then the variance depends only on n.

For a univariate h and assuming that it is not far from Normal, Silverman (1986, p.48) suggests for Normal kernels variance

$$b_n^2 = \left(0.9An^{-\frac{1}{5}}\right)^2$$

where $A = \min\{\text{sample standard deviation, sample interquartile range}/1.34\}$. In multivariate settings also, Normal kernels are usually used. A suggestion for their variance matrix is Sb_n^2, where S is the sample variance matrix and

$$b_n^2 = \left(\frac{4}{(d+2)n}\right)^{\frac{2}{d+4}},$$

where d is the dimensionality of h.

Smooth bootstrap can be seen as a form of kernel density estimation. It approximates a target density by forming a mixture of kernels over a sample. The difference is that the sample has not been obtained from the target and for this reason the kernels in the mixture do not have equal mixing weights. However, the idea of minimizing the MISE between mixture and target can be used in order to find the variance of the smoothing kernels. Gordon (1993) proposes this approach. For example, imagine a case where we are interested in the posterior distribution of an unknown scalar parameter x given data $\mathbf{y}$ generated by the model $p(\mathbf{y}|x)$. To implement smooth bootstrap we have drawn a sample of size n from the prior distribution $p(x)$ and to each point x_i we have assigned weight q_i

$$q_i = \frac{p(\mathbf{y}|x_i)}{\sum_{i=1}^{n} p(\mathbf{y}|x_i)}.$$

As in ordinary kernel density estimation, in order to derive MISE and minimize it we have to assume that the target density comes from a certain family of distributions. If we assume that the posterior is Normal with variance $\sigma^2_{X|\mathbf{y}}$ and if we use Normal kernels, minimization of the MISE will give (Gordon (1993)), variance

$$b_n^2 = \left[\left(\frac{4}{3}\right)^{\frac{1}{5}} \sigma_{X|\mathbf{y}} \left(\frac{E(p^2(\mathbf{y}|x))}{p^2(\mathbf{y})} \right)^{\frac{1}{5}} n^{-\frac{1}{5}} \right]^2 .$$

The quantities $\sigma_{X|\mathbf{y}}$, $E(p^2(\mathbf{y}|x))$ and $p^2(\mathbf{y})$ are usually unknown and will have to be estimated from the sample $x_1, \ldots, x_n$ from the prior as follows

$$\begin{aligned} \hat{\sigma}_{X|\mathbf{y}} &= \sqrt{\sum_{i=1}^n q_i \left(x_i - \sum_{i=1}^n q_i x_i \right)^2}, \\ \hat{E}(p^2(\mathbf{y}|x)) &= \frac{1}{n} \sum_{i=1}^n p^2(\mathbf{y}|x_i), \\ \hat{p}^2(\mathbf{y}) &= \left(\frac{1}{n} \sum_{i=1}^n p(\mathbf{y}|x_i) \right)^2 \end{aligned}$$

where $q_1, \ldots, q_n$ are the normalized weights. Then the variance of the smoothing kernels becomes

$$b_n^2 = \left[\left(\frac{4}{3}\right)^{\frac{1}{5}} \sqrt{\sum_{i=1}^n q_i \left(x_i - \sum_{i=1}^n q_i x_i \right)^2} \left(\frac{\sum_{i=1}^n p^2(\mathbf{y}|x_i)}{\left(\sum_{i=1}^n p(\mathbf{y}|x_i)\right)^2} \right)^{\frac{1}{5}} \right]^2 .$$

The approach we adopt is simpler. Suppose that $x_1, \ldots, x_n$ is a sample from the importance sampler and $q_1, \ldots, q_n$ are the normalized weights. For uni-

variate targets the variance of the smoothing kernels is

$$b_n^2 = (0.9sn^{-\frac{1}{5}})^2 \tag{2.6}$$

with $s = \sqrt{\sum_{i=1}^n q_i \left(x_i - \sum_{i=1}^n q_i x_i\right)^2}$ being an importance sampling estimator of the target's variance. For multivariate targets, the variance matrix of the smoothing kernels is

$$B_n = Sb_n^2 \quad , \quad b_n^2 = \left(\frac{4}{(d+2)n}\right)^{\frac{2}{d+4}} \tag{2.7}$$

with $S = \sum_{i=1}^n q_i \left(x_i - \sum_{i=1}^n q_i x_i\right)\left(x_i - \sum_{i=1}^n q_i x_i\right)^T$ being an importance sampling estimator of the target's variance matrix.

West (1993) advocates the same approach and proposes a small amendment that renders smooth bootstrap more efficient. If in a multivariate setting we choose kernels with variance (2.7) then the variance of the mixture

$$\hat{p}(x) = \sum_{i=1}^n q_i K(x; x_i)$$

will be larger than the variance of the target which it approximates. For this reason West (1993) suggests taking a mixture

$$\tilde{p}(x) = \sum_{i=1}^n q_i K(x; m_i)$$

instead, where the smoothing kernels are not centred on the sample points x_i but on new means m_i given by

$$m_i = \alpha x_i + (1 - \alpha)\bar{x} \tag{2.8}$$

with $0 < \alpha < 1$ and $\bar{x} = \sum_{i=1}^{n} q_i x_i$. The kernels still have variance (2.7). This mixture has expected value $\bar{x}$ and variance matrix $S(b_n^2 + \alpha^2)$. Therefore choosing $\alpha = \sqrt{1 - b_n^2}$ will give it variance S, which is as good an estimate of the target's variance as we can have at that point. The same correction has to be done in univariate settings too but the notation is different. There, the estimator s^2 of the variance is incorporated in b_n^2. Considering (2.6) we see that we have to take $\alpha = \sqrt{1 - \frac{b_n^2}{s^2}}$. Notice that even with this correction the asymptotic result about the sample produced by smooth bootstrap is valid. This is because, as n goes to infinity, $\alpha \longrightarrow 1$.

If the sample from the importance sampler is very large and therefore the mixture that approximates the target has many components one could use the mixture reduction algorithm of Salmond (1990), as does Gordon (1997). This algorithm reduces the number of components in a mixture by merging significant components with neighbouring insignificant ones. The significance of the components is measured by their mixing weights. The algorithm maintains the mean and variance of the original mixture. The merging together of components goes on for as long as the mixture's structure has not been altered by more than a quantity specified by the user. The structure is measured by the relative contribution of the variance within the components and the variance between the components towards the constant overall variance. Reducing the number of components simplifies sampling from the mixture but it is ques-

tionable whether this simplification counterbalances the computational effort of reduction.

In the light of all the above, the way we will be implementing smooth bootstrap is as follows. We will be using Normal kernels with a global variance given by (2.6) or (2.7) and with means given by (2.8) with α suitably calculated. We will not use mixture reduction.

2.2.1 Properties of smooth bootstrap sample statistics

We examine here whether the mean and variance of a smooth bootstrap sample are asymptotically unbiased and whether the variance of the sample mean is n times smaller than the variance of the target. Moreover, we are trying to see whether a central limit theorem holds for the sample mean.

In fact we have proved (see the Appendix for the mathematical details) that, if we use the same importance sampler, the expectation of the sample mean and variance and the variance of the sample mean are the same irrespective of whether we use weighted bootstrap or smooth bootstrap in order to obtain the final sample. Smooth bootstrap has to be used with correction (2.8) with a suitable α for this result to be valid. Since smooth bootstrap is weighted bootstrap with Normal noise added to it we expect Normality of the sample mean to be true here too.

To verify this we repeat the simulation experiment of section 1.4.1, using smooth bootstrap this time. We have as target the same "unknown" $N(0,1)$ distribution from which we have obtained the same ten data points as in 1.4.1. We treat either the mean or the variance as unknown and in each case we consider four different priors. For each combination of unknown and prior we

take 1000 samples from the prior and update them to arrive at 1000 samples of size n from the corresponding posterior at each time. Two different sample sizes, $n = 100$ and $n = 1000$, are considered.

Table 2.1 is an exact copy of Table 1.1, giving for the case of unknown mean the posterior means and variances and the expected values and variances of the sample statistics. The latter were derived by evaluation of (1.14),(1.15) and (1.17). Table 2.2 presents the simulated expectations and variances of the sample statistics. Its comparison with Table 2.1 leads to the same conclusions as in the case of weighted bootstrap. The sample mean and variance seem unbiased and more so for $n = 1000$. The variance of the sample mean is once more two to three times larger than one nth of the posterior variance. Comparing Tables 2.2 and 1.2 also confirms that the sample statistics have the same expectation and variance no matter whether weighted bootstrap or smooth bootstrap has produced the sample. Tables 2.3 and 2.4 refer to the case of unknown variance and also lead to the same conclusions.

The same Kolmogorov-Smirnov tests as for weighted bootstrap were performed here too and Normality of the sample means was not rejected for any case. Figure 2.1 presents histograms of the means and variances of the 1000 simulated samples for the case of unknown mean and for each combination of sample size and prior. We see from the histograms of the means that their distribution looks Normal. Figure 2.2 refers to the case of unknown variance and from it too Normality of the sample means does not look unreasonable.

Therefore, after the agreement of simulations with theory we can say that smooth bootstrap sample statistics have the same properties as the corresponding statistics of weighted bootstrap samples.

2.2.2 Conclusions

As we have demonstrated, smooth bootstrap has all the advantages of weighted bootstrap: it provides samples that asymptotically come from the target distribution, possibly at a lower rate of convergence, it is fast and easy to implement, the sample statistics have properties that resemble those of a random sample and the mean of the samples it produces seems to be asymptotically Normal. Moreover, it is devoid of most of weighted bootstrap's disadvantages; it approximates the target with a continuous distribution and the samples it produces do not suffer from the deterioration that so much afflicts weighted bootstrap samples. In general, smooth bootstrap is a very effective replacement for weighted bootstrap and should be used in the latter's place. The only point of caution, as in weighted bootstrap, is the choice of a good importance sampler. We will see in the following section and in a later chapter ideas about how to tackle this problem.

2.3 Other methods related to the smooth bootstrap

In this section we present three other methods related to the smooth bootstrap. The first combines it with weighted bootstrap while the second uses many iterations of it in order to arrive at a good importance sampler. The third "marries" MCMC with the ability to propagate samples through time.

2.3.1 Partially smooth bootstrap

This method is a hybrid of weighted bootstrap and smooth bootstrap. It works as follows. Let $x_1, \ldots, x_n$ be a sample from an importance sampler g. We apply weighted bootstrap with appropriate weights and get a sample of size m which for large n can be treated as coming from a target density h. In this sample each x_i will appear n_i times, with $0 \leq n_i \leq m$ and $\sum_{i=1}^{n} n_i = m$. For any x_i with $n_i \geq 2$ we drop from the final sample all its replicates but one and replace them by points drawn from a symmetric kernel placed over x_i. The parameters of the kernel are those the kernel would have if we were using smooth bootstrap. Therefore, it may not be centred over x_i but over m_i given by (2.8). This is the way we use the method later in the chapter.

Note that even given $x_1, \ldots, x_n$ the members of the resulting sample are not independent random variables. Therefore we cannot apply the law of large numbers in order to establish whether the sample tends to coming from the target distribution. However, as n tends to infinity, the variance of the smoothing kernels tends to zero and the method approaches weighted bootstrap. Therefore, it seems plausible that this method also produces samples that asymptotically come from the target density.

2.3.2 Iterated smooth bootstrap

As its name suggests, this method consists of iterating the smooth bootstrap. The idea will be clarified with the help of mathematics. The exposition is based on West (1993).

The aim is the same as before. We have a sample $x_1^{(0)}, \ldots, x_n^{(0)}$ from a distri-

bution g and want a sample from another distribution h. Having the usual weights, which we now denote by $w_i^{(0)}$ and $q_i^{(0)}$, we apply smooth bootstrap and obtain a sample $x_1^{(1)}, \ldots, x_n^{(1)}$ which, we hope, is close to having come from h. However, we know the exact distribution g_1 from which this sample has been obtained. It is a mixture of symmetric kernels with appropriate variance $B^{(0)}$ and means $x_i^{(0)}$ or $m_i^{(0)}$ given by (2.8),

$$g_1(x) \equiv \sum_{i=1}^{n} q_i^{(0)} K(x; m_i^{(0)}, B^{(0)}),$$

where $K(x; m, B)$ is the value at x of the kernel with mean m and variance matrix B. This is expected to be closer to h than g was and therefore, to be a better importance sampler. We can then calculate new weights for the $x_1^{(1)}, \ldots, x_n^{(1)}$ sample,

$$w_i^{(1)} = \frac{h(x_i^{(1)})}{g_1(x_i^{(1)})}$$

and $q_i^{(1)}$'s respectively. Then smooth bootstrap can be applied to $x_1^{(1)}, \ldots, x_n^{(1)}$ to get a sample $x_1^{(2)}, \ldots, x_n^{(2)}$ which should be even closer to h. We can either stop there or consider that the new sample comes from a density g_2 and do smooth bootstrap again.

This iteration of smooth bootstrap steps is performed so that a possibly unsatisfactory initial importance sampler can be improved upon. When the necessary improvement has been achieved or when no more improvement is possible we retain the final sample as the sample from the target. We can at each step monitor the effective sample size or the entropy of the weights

relative to uniformity. Supposing that we keep a constant sample size n throughout, a value of ESS very close to n or of entropy close to 1 indicates that not much greater improvement is possible and that we should stop; for discussion see West (1993).

When the method is applied in dynamic model settings there is a potential problem. Suppose that we have started the iterated smooth bootstrap with a sample $x_t^{(0)}(1), \ldots, x_t^{(0)}(n)$ from $p(x_t|\mathbf{y}_{t-1})$ and that the target is $p(x_t|\mathbf{y}_t) = p(x_t|y_t, \mathbf{y}_{t-1})$. After the kth iteration of smooth bootstrap we have $x_t^{(k)}(1), \ldots, x_t^{(k)}(n)$, coming from a density $g^{(k)}(x_t)$. Then their weights for the $(k+1)$st iteration will be

$$w_t^{(k+1)}(i) \propto \frac{p(x_t^{(k)}(i)|\mathbf{y}_t)}{g^{(k)}(x_t^{(k)}(i))} \propto \frac{p(y_t|x_t^{(k)}(i))p(x_t^{(k)}(i)|\mathbf{y}_{t-1})}{g^{(k)}(x_t^{(k)}(i))}.$$

However, usually in dynamic models $p(x_t|\mathbf{y}_{t-1})$ is not available in closed form. Therefore it may not be possible to calculate the weight either. A possible solution is to approximate $p(x_t|\mathbf{y}_{t-1})$ by a kernel density estimate based on $x_t^{(0)}(1), \ldots, x_t^{(0)}(n)$.

Iterated smooth bootstrap can obviously be very time consuming because of the need to evaluate many times mixtures of densities possibly consisting of thousands of components. West (1993) suggests an algorithm for reducing substancially the number of components of a mixture without altering its structure. Alternatively the method of Salmond (1990) can be used. For a different way of speeding up the iterated smooth bootstrap see Oehlert (1998).

Givens and Raftery (1996) propose an alternative approach that is thought to give even better importance samplers for multivariate targets with irregularly

shaped supports. Their method is iterated smooth bootstrap, or **adaptive importance sampling** as they call it, but they do not use the same variance matrix for all kernels. The kernel over point x_i has a variance matrix that reflects the variation of the sample points close to x_i. This variance matrix is equal to a multiple of the variance matrix of the $[\lambda_i n]$ near neighbours of x_i, with $\frac{2}{n} \leq \lambda_i \leq 1$. The multiplying factor goes to zero as n increases and can be different for each x_i. When calculating the local variance matrix of a group of points we can either weigh them equally or divide their normalized weights with their total normalized weight. Moreover, the kernels have means x_i and not m_i. The mixture formed with such smoothing kernels follows more effectively the shape of the support of the target as indicated by the alignment of the sample points in space. For the very first resampling we must use a global variance matrix because the alignment of the points of the initial sample follows the shape of the support of the importance sampler.

Givens and Raftery (1996) report simulation results which show that when the target is very structured and the importance sampler is very diffuse the performance of iterated smooth bootstrap is extremely variable. Sometimes it is better and sometimes it is worse than weighted bootstrap. On the other hand their method performs better than weighted bootstrap when n is small. Moreover it can give with a lot smaller n, estimators of the same variance as weighted bootstrap. However, in one of their examples with a 10-dimensional target and $n = 20000$ their method and iterated smooth bootstrap do worse than weighted bootstrap. They believe that when n is large the latter is the best method because it gives similar results with less computational effort. They have not tried smooth bootstrap.

2.3.3 Bayesian Metropolis filter

As we have mentioned, one of the disadvantages of MCMC is that for sampling from $p(x_t|\mathbf{y}_t)$, in a dynamic model context, we have to sample from $p(\mathbf{x}_t|\mathbf{y}_t)$ and then keep only the x_t part of the sample. This happens because $p(x_t|\mathbf{y}_{t-1})$ is not available in closed form. Gordon and Whitby (1995) present a method which they call the Bayesian Metropolis filter. It is a usual Metropolis-Hastings algorithm with target

$$\hat{p}(x_t|\mathbf{y}_t) \propto p(y_t|x_t)\hat{p}(x_t|\mathbf{y}_{t-1}),$$

where $\hat{p}(x_t|\mathbf{y}_{t-1})$ is a kernel density estimate of $p(x_t|\mathbf{y}_{t-1})$ based on a sample from it. To speed things up the authors use Salmond's algorithm to reduce the number of components in $\hat{p}(x_t|\mathbf{y}_{t-1})$. The sample from $p(x_t|\mathbf{y}_{t-1})$ is obtained by passing through the system evolution equation the sample from $p(x_{t-1}|\mathbf{y}_{t-1})$ that the algorithm has given in the previous time point.

The advantage of the method is that it combines the theoretical properties of MCMC with the propagation of samples through time. However, we still need to monitor for the convergence of the Markov chain at each time point. Therefore, the method can still be time consuming and is unwieldy for automatic real time analysis of dynamic models. For this reason we are not going to consider it in the simulations section.

2.4 Augmentation methods

The methods that we have presented so far correct the deficiencies of weighted bootstrap by changing the distribution from which it obtains its samples. We will now examine two methods that apply weighted bootstrap as usual but then modify the sample it produces.

For both of them the setting is the same. We want to obtain a sample from $h(x) = f(x)/\int f(x)dx$ when only $f(x)$ is known. We get a sample $x_1, \ldots, x_n$ from another density $g(x)$ and assign to each point x_i a normalized weight $(f(x_i)/g(x_i)) \,/\, \left(\sum_{j=1}^{n} f(x_j)/g(x_j)\right)$. Weighted bootstrap gives a sample $y_1, \ldots, y_m$ that can be considered as coming from h if n is very large. However, this sample is unrealistic and unsatisfactory, especially if h is continuous, because it contains more than one copy of at least some of its members. The methods we are going to present are called **augmentation** methods because they "augment" the weighted bootstrap sample by bringing the number of distinct values in it back to m.

2.4.1 The method of Sutherland and Titterington

This method, which from now on we denote by "S-T", first appeared in a slightly simpler form than that presented here in Sutherland and Titterington (1994). It relies on a very straightforward idea.

Since the sample $y_1, \ldots, y_m$ is treated as coming from h, a kernel density estimate (kde) based on it will be an approximation to h. This kde will, as we have mentioned before, be a mixture of m equally weighted components. The ith component will be a symmetric kernel with mean y_i, for $i = 1, \ldots, m$.

In other words, if B_m is the variance matrix of the kernels the kde will be

$$\hat{h}(x) = \frac{1}{m}\sum_{i=1}^{m} K(x; y_i, B_m). \tag{2.9}$$

We can then sample m points $z_1, \ldots, z_m$ from it. Since (2.9) is a continuous density, the new sample will almost certainly consist of m distinct points. We can base all our inference about h on $z_1, \ldots, z_m$. To sample m points from (2.9) we only have to repeat m times the following. At step i

- sample θ with $\Pr(\theta = y_j) = \frac{1}{m}$, for all $j = 1, \ldots, m$.
- draw z_i from $K(x; \theta, B_m)$.

Our usual choice of kernel is the Normal density. The variance is chosen according to kernel density estimation theory. In univariate cases we take

$$b_m^2 = \left(0.9Am^{-\frac{1}{5}}\right)^2 \tag{2.10}$$

with $A = \min\{\text{sample standard deviation, sample interquartile range}/1.34\}$. In multivariate cases, if d is the dimensionality of the target we have

$$B_m = B \cdot b_m^2 \qquad , \qquad b_m^2 = \left(\frac{4}{(d+2)m}\right)^{\frac{2}{d+4}},$$

where B is the sample variance matrix of $y_1, \ldots, y_m$. In the original version, in Sutherland and Titterington (1994), a simpler form of variance matrix for the Normal kernels is used. The variance matrix, in the d-dimensional case, is diagonal with its kth diagonal element being calculated by applying (2.10)

to the kth component of $y_1, \ldots, y_m$, for $k = 1, \ldots, d$. It is obvious that this choice fails to take into account the presence of any correlations in the sample.

Although our choice of variance is theoretically optimal when h is not far from being Normal, the simulations show that it works well for many types of target distribution.

2.4.2 The method of Gordon, Salmond and Smith

This method, which from now on we denote by "G et al", appeared in Gordon *et al* (1993). This time, from the sample $y_1, \ldots, y_m$ we consider each point y_i once, replacing it with a point z_i drawn from the Normal distribution $N(y_i, B'_m)$. B'_m is a diagonal variance matrix. Its kth element on the diagonal, σ_k^2, is given by

$$\sigma_k^2 = \left(TE_k m^{-\frac{1}{d}}\right)^2,$$

where d is the dimensionality of the target density, T is a tuning constant chosen by the user and E_k is the range of the sample in dimension k, for $k = 1, \ldots, d$.

Criticism of this method focuses mainly on the choice of variance. First, in multivariate settings the variance matrix does not take into account any correlations that may be present in the sample. Secondly, while according to what kernel density estimation theory dictates for the optimal estimator, the variance should decay very slowly, being proportional to $m^{-2/(d+4)}$, here it is proportional to $m^{-2/d}$. Finally, in specifying the variance we have to

subjectively choose a value for the tuning parameter T. However, as the simulations are going to show, the method performs equally well as the other methods.

2.4.3 A short note

Acklam (1996) proposes yet another way of augmenting a posterior sample. There, the augmentation by sampling from a kernel density estimate of the posterior is combined with shrinking of the original posterior sample towards its mean. We will not pursue this method any further.

2.4.4 Discussion

We presented in this section methods that improve the results of weighted bootstrap by modifying the samples it produces after it has produced them. These methods are no harder to implement than the smoothing methods presented earlier in this chapter.

We have seen that, especially in Bayesian learning contexts, successive applications of weighted bootstrap produce samples with ever decreasing variance. A part of this decrease is natural since information is being gathered about the unknown parameters. The rest of it is attributable to sample deterioration. Augmentation convolves the samples with the kernels and therefore increases their variance again. However, if in a Bayesian learning problem augmentation was employed after each weighted bootstrap application it could eliminate all the decrease in variance. This would lead to samples that do not reflect in their variance the learning about the unknown parameters. For

this reason Sutherland and Titterington (1994) suggest using it sparingly in Bayesian learning situations.

Finally, there is a technicality about augmentation which luckily does not affect its practical performance. Smoothing methods base the variance of their kernels on the size n of the sample from the importance sampler. As $n \longrightarrow \infty$ the methods automatically approach weighted bootstrap which in that case gives samples directly from the target. In augmentation the variance of the kernels depends on the size m of the sample resulting from weighted bootstrap. Therefore they can theoretically be applied even when $n \longrightarrow \infty$, in which case they will "damage" a sample that has come from the target density. In practice, n is always finite and, as we will see in the simulations below, there is little difference between the results of smoothing and augmentation methods.

2.5 Comparisons of the methods

Here we gather together all the methods presented in this and the previous chapter and compare them in several settings. We try to see how well they estimate the posterior distribution and the values of the "unknown" parameters.

2.5.1 A simple univariate case

For this first experiment we have obtained 100 observations from $N(0,1)$ but we are going to treat $\mu = 0$ as unknown. We treat the data sequentially. Therefore, we have a case of Bayesian learning with 100 posteriors for μ. We

assign to μ a prior $N(0.1, 1)$, a fairly good guess of its true value. If we sample n points from the prior the aim will be to update this sample in order to get samples of size n from each of the 100 posteriors. Of course these posteriors are Normal and known analytically, facilitating the assessment of the performance of the methods. The iterated smooth bootstrap is not considered here since the problem is simple and the entropies of the weights were found to be higher than 0.99.

The simulation procedure is as follows. For each of the methods, 30 times we start with a sample of $n = 10000$ points from the prior and we update it using the observations from $N(0, 1)$ so that for each method and each of the 100 posteriors we have 30 sample sets, i.e. 30 estimates of the posterior's mean, variance, quantiles or any other quantity of interest.

Figure 2.3 presents the true 95% highest density intervals (HDI) of the 100 posteriors and against them the corresponding intervals obtained by the five methods. These were constructed by taking the medians of the 0.025 and 0.975 percentiles of the 30 sample sets corresponding to each posterior and method. Twice the variances of the same percentiles were added to and subtracted from their means and this gave rough confidence bands for the interval endpoints. Figure 2.4 presents the logarithm of the true posterior variance and the logarithms of the 0.025 and 0.975 percentiles of the 30 sample variances corresponding to each posterior and method.

From Figure 2.3 we see that all the methods track the true intervals extremely well, with S-T and partially smooth bootstrap giving estimates slightly wider than the truth. For S-T in particular and for some time instants the confidence bands for the HDI endpoints do not contain the endpoints of the true HDI's. Figure 2.4 leads to the same conclusions. S-T overestimates the posterior

variance and so does partially smooth bootstrap, something that was seen in the previous figure. Notice however that, although this is a Bayesian learning problem, having employed augmentation after each application of weighted bootstrap has not eliminated the characteristic phenomenon, in such problems, of decreasing posterior variance. Only in the last samples provided by S-T is there perhaps a hint of the sample variance stabilizing. The ordinary weighted bootstrap is doing so well here because the case is very simple, with good overlap between prior and posterior at each step.

2.5.2 A simple dynamic model

Although all the methods presented in these two chapters were developed for non-linear and non-Gaussian dynamic models, we examine them here in the case of a bivariate, linear and Gaussian dynamic model. This is done so that we have again, this time provided by the Kalman filter equations, the true posteriors of the parameters of interest as a standard which the methods are trying to attain.

The system evolution equation is $x_t = Gx_{t-1} + \eta_t$, while an observation y_t becomes available at time t given by the model $y_t = Hx_t + \epsilon_t$. The quantities η_t and ϵ_t are random system and observation noise. They are mutually independent and independent of their past values and any value of x_t. Their distribution is Normal with means $\mu_\eta = (1, -1)^T$ and $\mu_\epsilon = (-1, 1)^T$ and variance matrices

$$\Sigma_\eta = \Sigma_\epsilon = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

G and H are deterministic matrices,

$$G = \begin{pmatrix} 2 & 2 \\ 1 & 3 \end{pmatrix} \quad , \quad H = \begin{pmatrix} 1.1 & 0.8 \\ 0.6 & 1.3 \end{pmatrix}.$$

The state x_0 of the system at time 0 is unknown and follows a Normal distribution with mean $\mu_0 = (5, 3)^T$ and variance matrix

$$\Sigma_0 = \begin{pmatrix} 2 & 1 \\ 1 & 1.5 \end{pmatrix}.$$

All the above are assumed known. A starting value for x_0 was chosen from this distribution and then the system was simulated for six time steps, giving rise to observations $y_1, \ldots, y_6$. We are interested in the six resulting posterior distributions $p(x_t|\mathbf{y}_t), t = 1, \ldots, 6$. Since the model is linear and Gaussian we know that all these posteriors will be Gaussian too. A straightforward application of Bayes Theorem gives the following recursive formulae for the posterior mean μ_t and variance matrix Σ_t at time t.

$$\Sigma_t = \left[(G\Sigma_{t-1}G^T + \Sigma_\eta)^{-1} + H^T\Sigma_\epsilon^{-1}H\right]^{-1}$$

$$\mu_t = (G\mu_{t-1} + \mu_\eta) + \Sigma_t H^T\Sigma_\epsilon^{-1}(y_t - \mu_\epsilon - H(G\mu_{t-1} + \mu_\eta)).$$

These are none other than the Kalman filter equations mentioned earlier in Section 1.2.

Subsequently we treated all these posteriors as unavailable. We consider the same five methods as in the previous example. Iterated smooth bootstrap is excluded again, for reasons of speed of implementation. Each method was initialized on a sample of 20000 points from the distribution of x_0 and then provided a sample of the same size from each of the six posteriors.

Figures 2.5 and 2.6 show true and sample 95% HDI's for the two components of x_t separately, for all six posteriors and for each sampling method. The sample intervals are displayed as segments, their endpoints being their corresponding sample's 2.5% and 97.5% quantiles. We cannot distinguish any clear winner among the methods since they all produce intervals close to the true ones.

Since the target distributions are bivariate these two figures do not provide us with the full picture because they convey no information about the correlation of the two components. Table 2.5 gives the true correlation coefficient ρ for each posterior as well as the corresponding Fisher ζ-score,

$$\zeta = \frac{1}{2} \ln \left(\frac{1+\rho}{1-\rho} \right).$$

In Table 2.6 we present the sample correlation coefficients r and corresponding z-scores for each method and posterior. Using the approximate formula $z \sim N(\zeta, \frac{1}{n-3})$, where for us $n = 20000$, we calculated 95% confidence intervals for ζ, abandoning for a while our Bayesian perspective. We say that a method has "tracked" a ζ-score if its true value is included in the confidence interval for it given by the method. We can see that the methods are not doing well in this respect. The best, partially smooth bootstrap, tracks four out of the six ζ-scores.

2.5.3 A simple imaging problem - 1

Here we present a very simple image analysis problem which we formulate in a Bayesian learning context. A rectangle lies on a 128×128 pixel plane and it starts rotating around the centre of the plane with a constant angular velocity of ω rads per time unit. If at time $t-1$ a point of the rectangle is at coordinates $x^{(t-1)}, y^{(t-1)}$ then at time t it will be at

$$x^{(t)} = (x^{(t-1)} - 64)\cos(\omega) - (y^{(t-1)} - 64)\sin(\omega) + 64, \quad (2.11)$$

$$y^{(t)} = (x^{(t-1)} - 64)\sin(\omega) + (y^{(t-1)} - 64)\cos(\omega) + 64. \quad (2.12)$$

All the points inside the rectangle have brightness 1 and those outside it have brightness 0. To find all the points that lie inside the rectangle it suffices to know the coordinates $(x_1^{(t)}, y_1^{(t)}), (x_2^{(t)}, y_2^{(t)})$ of the midpoints of its two short sides and its width. For a presentation of how, given these coordinates and the width, we can find which pixels belong to the rectangle see appendix B. We can find the coordinates of the midpoints at time t by repetitive application of (2.11) and (2.12), if we know their initial coordinates $(x_1^{(0)}, y_1^{(0)}), (x_2^{(0)}, y_2^{(0)})$ and ω.

At time t we cannot observe the true image but only a degraded version of it. We denote by $c_i^{(t)}$ the true brightness of pixel i at time t and by $d_i^{(t)}$ its brightness in the degraded version of the image. Each pixel is degraded independently of all the other pixels by Gaussian noise $N(0, 0.36)$. In other words, if $c_t = (c_1^{(t)}, \ldots, c_N^{(t)})$ is the true image and $d_t = (d_1^{(t)}, \ldots, d_N^{(t)})$ is the degraded one with $N = 128^2$, we have

$$p(d_t|c_t) \propto \exp\left[-\frac{1}{2 \cdot 0.36}\sum_{i=1}^{N}(d_i^{(t)} - c_i^{(t)})^2\right]. \tag{2.13}$$

The assumption of Normal noise for a black and white image is of course unrealistic since all the noisy measurements should be grey, i.e. between 0 and 1. However the image can be a description for some other type of system where noisy measurements can take any real value.

In this problem we assume that the only unknowns are $x_1^{(0)}, y_1^{(0)}, x_2^{(0)}, y_2^{(0)}, \omega$ and the **halfwidth** h. We assume that they are independent *a priori* with uniform distributions, on [20,100] for the coordinates, on [5,15] for the halfwidth and on $[0, 2\pi)$ for the angular velocity. Note that all the parameters are continuous variables. Only when coordinates are transformed into pixel coordinates do truncations take place. The brightnesses $d_i^{(t)}$ are also continuous.

If we start with a sample from the prior we can update it by weighted bootstrap or any other of the methods described thus far and obtain samples from each of the arising posteriors. After we have obtained a sample from $p(x_1^{(0)}, y_1^{(0)}, x_2^{(0)}, y_2^{(0)}, \omega, h|\mathbf{d}_{t-1})$, where $\mathbf{d}_{t-1} = (d_0, \ldots, d_{t-1})$, each point $x_1^{(0)}(i), y_1^{(0)}(i), x_2^{(0)}(i), y_2^{(0)}(i), \omega(i), h(i)$ needs a weight $w_t(i)$ in the light of the new data d_t. By applying (2.11) and (2.12) t times successively, we get coordinates $x_1^{(t)}(i), y_1^{(t)}(i), x_2^{(t)}(i), y_2^{(t)}(i)$ which in combination with $h(i)$ produce an image $c_t(i)$. The weight then will be $w_t(i) \propto p(d_t|c_t(i))$, which is given by (2.13).

The prior can be very diffuse compared with the first posterior, $p(x_1^{(0)}, y_1^{(0)}, x_2^{(0)}, y_2^{(0)}, \omega, h|d_0)$, and for this reason we use the Metropolis-Hastings algorithm to sample from the latter. The first degraded image d_0 does not convey any information about ω since rotation has not started yet. Therefore, the

algorithm will give samples from $p(x_1^{(0)}, y_1^{(0)}, x_2^{(0)}, y_2^{(0)}, h|d_0)$ and after sampling has finished we will attach to each sample point a point taken from the prior of ω. The starting values are chosen at random from the prior. The proposal density considers the unknowns as independent. At each step of the algorithm one of the five unknowns is chosen at random with equal probabilities of selection for all of them and a new value is proposed for it. The proposal densities are Normal for all the unknowns. Each one of them has the current value of the corresponding unknown as mean and variance 25 for the coordinates and 1 for the halfwidth. If a proposed value lies outside [20,100] for the coordinates or [5,15] for the halfwidth we propose a new one.

For this particular application we simulated d_0 and degraded images for the first nine time points after the beginning of rotation. The true parameter values were (40,40) for the first midpoint, (40,80) for the second, 10 for the halfwidth and 0.1 for the angular velocity. All the samples had size 20000. The Metropolis-Hastings algorithm was ran for 1000000 iterations. The results of the first 900000 were discarded as "burn-in" and from the remaining sample points we stored every fifth so as to avoid the presence of serial correlations in the samples.

Thus we started the simulations for each of the methods compared, which are the same as in the previous experiment. All the methods analysed the same set of ten degraded images but each one started with its own Metropolis-Hastings sample. Iterated smooth bootstrap was extremely slow and we do not present results for it. The weighted bootstrap sample after $t = 1$ already contained many copies of two distinct values and therefore, although those values are close to the true ones, we do not present results for it either.

Figure 2.7 gives posterior sample boxplots for each parameter separately, for

the time instants after rotation begins. Because of their intractability we do not know the true posteriors and against the boxplots we can only plot the true parameter values. In most cases they are not contained in the boxplots but are so close to them that the discreteness of the pixel grid eliminates the difference. We do not give the reconstructions of the images for reasons of economy of space, but the results are very good, similar to those of the next example. No method really stands out.

2.5.4 A simple imaging problem - 2

The same problem as above is now viewed from a dynamic modelling point of view. We assume that the rectangle does not rotate continuously but changes location only every time unit. The angular velocity $\omega = 0.1$ is now known. The unknowns at time t are the coordinates $(x_1^{(t)}, y_1^{(t)}), (x_2^{(t)}, y_2^{(t)})$ of the midpoints of the two short sides of the rectangle and the halfwidth h. Let $U_t = (x_1^{(t)}, y_1^{(t)}, x_2^{(t)}, y_2^{(t)}, h)^T$. We assume that U_t and U_{t-1} are related via the formula

$$U_t - C = A(U_{t-1} - C) + V_t, \tag{2.14}$$

where $C = (64, 64, 64, 64, 0)^T$ and $V_t = (v_1^{(t)}, v_2^{(t)}, v_3^{(t)}, v_4^{(t)}, 0)^T$, with $v_i^{(t)}$'s being independent $N(0, 1)$ random variables. The matrix A is

$$A = \begin{pmatrix} \cos 0.1 & -\sin 0.1 & 0 & 0 & 0 \\ \sin 0.1 & \cos 0.1 & 0 & 0 & 0 \\ 0 & 0 & \cos 0.1 & -\sin 0.1 & 0 \\ 0 & 0 & \sin 0.1 & \cos 0.1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Therefore, (2.14) is (2.11) and (2.12) in matrix notation and, moreover, system noise is introduced. The observation equation is non-linear and consequently the Kalman filter cannot be applied. Therefore, the posteriors are not analytically available. Here we cannot use Bayesian learning because with each new frame we have to sample from the posterior of a new set of parameters (only the halfwidth remains constant although unknown). Only four time points are considered after the rectangle starts moving.

The priors are again the same as before apart from the fact that ω is not among the unknowns any more. For frame 0 we use the same Metropolis-Hastings algorithm as in the previous experiment. Then, weighted bootstrap and the other methods are applied. The sample size is again 20000 for all samples. The samples from each posterior are passed through the system evolution equation to give samples from the next prior.

For reasons of comparison we also use the low-level technique of Hainsworth and Mardia (1992). This is done in order to demonstrate the superiority of using information about the types of object present in an image over pixel-level analysis, when such information is available. This method, to be denoted by "H-M" from now on, is a low-level technique for the estimation of binary images. It does not make any assumptions about what objects may be present

in them. It views each image as an unknown arrangement of black (background) and white (objects) pixels. It tries to reconstruct the true image given the observed data. The reconstruction is not only affected by the data but also by the model that generates them and by the prior distribution over the space of all possible images. A usual choice of prior is the one that gives smaller probabilities to images that contain large numbers of pairs of neighbouring pixels where one is black and one is white. In binary images there are pairs of neighbouring pixels where this can happen legitimately; these are the pairs through which passes the border of an object. To allow the prior to accommodate such pairs an edge variable is introduced for each pair of neighbouring pixels. It has two possible values, "present" or "absent". If an edge is present then the pixels can have different colours without being penalized for that. The aim of the technique is to reconstruct the true colours of the pixels and the true values of the edges at each time point. Because we have a sequence of images, the prior for the image at time t depends on the reconstructions at previous time points. Considering these as the truth for the previous time points, the prior gives smaller probabilities to images that have many pixels changing colour or many edges changing value from time $t-1$ to time t. With model (2.13) generating the data it is not difficult to estimate the mode of the posterior distribution of true pixel colours and edge values by using techniques like **simulated annealing** (Geman and Geman (1984)) or **iterated conditional modes** (Besag (1986)). We use simulated annealing in this application with the following "cooling" scheme suggested by the authors. We start with "temperature" $T_1 = 2$ and lower it according to

$$T_r = T_1 \frac{\ln 2}{\ln r} \quad , \quad r = 1, \ldots, 5.$$

At each temperature 200 complete updates of the image and edge values are performed. Each pixel colour and edge value is separately updated by sampling from its conditional distribution given the other pixel colours and edge values and the data.

Figure 2.8 presents posterior sample boxplots for all parameters separately, for the time instants after rotation begins. The true parameter values are also shown. We do not show results for the weighted bootstrap because, as in the previous experiment, it produced very degenerate data. There are no boxplots for H-M either since it does not produce samples for the parameters. The method that is better is G et al, catching most of the true values in the boxplots or being nearer to them than any other method. Again though, the discreteness of the pixel grid eliminates all differences in the reconstructions, as may be seen in Figure 2.9. All the reconstructions are practically identical. The maximum number of mis-coloured pixels in any of them was 22. As was expected the reconstructions are much better than those of H-M because the latter by its nature cannot give sharp edges to the rectangle. This method gives larger numbers of mis-coloured pixels.

2.5.5 Another imaging problem

Another image analysis problem is considered here. A cuboid is hovering in 3D space above the plane of the image. Light is shone on the plane in a direction perpendicular to the cuboid so that its shadow is cast on it. The colour of the image (background) is black (brightness 0) and that of the shadow (foreground) is white (brightness 1). The cuboid starts rotating in discrete time around an axis passing through its centre and vertical to the

direction of light. Before and during rotation it is always at such positions that the images we would see on the plane would be of a rectangle (the cuboid's shadow) contracting and expanding but having constant width and with its centre coinciding all the time with the centre of the image (pixel (64,64)). Figure 2.10 will clarify this exposition. The width is assumed known. The images are degraded by the same noise as in the previous experiments. The unknowns in this problem are the two other dimensions of the cuboid, l and b, the angle θ it was forming with the plane before rotation and the angular velocity ω. At each time t the true length of the shadow x_t is

$$x_t = l|\cos(\theta + \omega t)| + b|\sin(\theta + \omega t)|. \tag{2.15}$$

The width of the rectangle is 40 pixels. We observe a degraded image before rotation and nine more after rotation begins, at equidistant time points. This is again a Bayesian learning problem with each new image giving rise to a new posterior distribution for l, b, θ and ω. Here we use resampling methods even for frame 0 in order to see whether the diffuse prior will cause any problems to resampling. The methods are the same as in the previous experiments. ω is again excluded in frame 0 and is just sampled from its prior. The prior distribution for l is taken to be uniform on (20,100), for b uniform on (10,40), for θ uniform on $(-\frac{\pi}{2}, \frac{\pi}{2})$ and for ω uniform on $(0, 2\pi)$. The posteriors once more are not analytically available. The true parameter values, used to form the images, were $l = 65, b = 23, \theta$=0 and ω=0.5. The samples have size 20000. The weighted bootstrap sample again degenerated very quickly and after the third frame contained 20000 replications of the same value. We do not show results for it.

Figure 2.11 gives posterior sample boxplots and the true parameter values. Here all the methods are doing very well, a lot better than in the rectangle problem. All the samples home in on the true values. In most cases there is a sharp contrast between what happens before and after frame 3. This is because each frame basically provides an estimate of the current shadow length x_t. Equations (2.15) from different time points form a system in four unknowns. Until frame 3 we have fewer equations than unknowns and therefore less certainty about the parameters' true values.

2.5.6 An alternative point of view

We close this section by returning to simpler cases again. This time the performance of the methods is examined from a different perspective. If any method produces a sample $x_1, \ldots, x_n$, allegedly from the target density h, we judge its performance by the MISE of the kernel density estimator (kde) $\hat{h}$ based on the sample, in other words by

$$E\left(\int (h(x) - \hat{h}(x))^2 dx\right). \tag{2.16}$$

The expectation is taken with respect to all the sources of randomness that play part in the generation of the sample. It is difficult to calculate this expectation or even the integral in many cases. We consider two simple cases where h is Normal, $N(\mu, \sigma^2)$ and where the kde is

$$\hat{h}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{2\pi b^2}} \exp\left(-\frac{1}{2b^2}(x - x_i)^2\right)$$

which is the usual kde with Normal kernels. The variance b^2 which leads to the kde with minimum MISE, since the target is Normal is

$$b^2 = \left(1.06\hat{\sigma}n^{-\frac{1}{5}}\right)^2 \tag{2.17}$$

where $\hat{\sigma}$ is the sample standard deviation (Silverman (1986, p. 45)). Then the integral, which is the L_2 distance between h and $\hat{h}$, is

$$\begin{aligned}\int (h(x) - \hat{h}(x))^2 dx &= \frac{1}{n\sqrt{4\pi b^2}}\left[1 + \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq i}\exp\left(-\frac{(x_i - x_j)^2}{4b^2}\right)\right] \\ &- \frac{2}{n\sqrt{2\pi(b^2+\sigma^2)}}\sum_{i=1}^{n}\exp\left(\frac{(x_i-\mu)^2}{2(b^2+\sigma^2)}\right) + \frac{1}{\sqrt{2\pi\sigma^2}}.\end{aligned}$$

The expectation of this quantity will depend on the method that generated the sample. We estimate it for each method separately by generating many samples of size n and averaging the L_2 distances of the kernel density estimates they produce. Each kde is formed with the optimal variance given by (2.17). Here we compare all the methods, iterated smooth bootstrap included. The latter is applied with three iterations of smooth bootstrap.

In the first setting we once again use the $N(0,1)$ prior for the "unknown" mean of a $N(0,1)$ population. We have 30 observations from this population which give rise to 30 posteriors if considered sequentially. All these posteriors are Normal. For each method we obtain 200 samples of size $n = 50$ from the prior and then update them. Thus for each posterior and method we have 200 samples and therefore 200 density estimates and 200 L_2 distances. We average these 200 distances and the results are shown in Figure 2.12.

The improvement created by all the smoothing and augmentation methods is evident. As weighted bootstrap is applied again and again the estimates it gives drift further and further away from the posteriors. We see that the average L_2 distances for the other methods also increase, but very slowly. If we want to pick a winner among the methods this will be the partially smooth bootstrap. It is surprising that the iterated smooth bootstrap does worse than the smooth bootstrap and the augmentation methods.

A different picture emerges when we consider a dynamic model variation of the previous example. Now we have 30 observations $y_t \sim N(x_t, 1)$ with $x_t \sim N(x_{t-1}, 0.1)$ and $x_1 \sim N(0, 1)$. The x_t's are the only unknowns. The posterior distributions the observations give rise to are Normal again since this is a linear Gaussian model. Again for each method we take 200 samples of size 50 from $p(x_1)$ and update them. The average L_2 distances are shown in Figure 2.13. Here all the methods give estimates that are very close to the true posteriors. All the average L_2 distances are less than 0.2. Nevertheless, the iterated smooth bootstrap is now the worst of the methods, while the partially smooth bootstrap is again the best, followed closely by the smooth bootstrap and G et al. Weighted bootstrap is now doing well thanks to the effect of the propagation equation on the samples.

2.6 Discussion

We now pull together and discuss our findings in the first two chapters. Our purpose was to study alternative techniques for obtaining random samples from intractable distributions. MCMC is the established group of methods that serve this purpose. However there are situations where samples from

many distributions are needed in a short time. One such situation is the real-time analysis of dynamic models. MCMC can be very slow because the Markov chain it simulates must converge to the target distribution. Moreover convergence must be monitored and this precludes an automatic analysis. The methods that we focused on are based on resampling. They all essentially acquire an initial sample from the importance sampler, a distribution easy to sample from, and based on this sample they construct an approximation to the target. They then obtain their final sample from this approximation.

The simplest method is weighted bootstrap. It approximates the target by a discrete distribution over the initial sample. If this distribution appropriately assigns probabilities to the sample's members it is proved that sampling from it gives samples that tend to come from the target as the initial sample's size tends to infinity. We proved that the mean and variance of the final sample are asymptotically unbiased and it has also been proved that the sample mean is asymptotically Normal. However, because the approximation is a discrete distribution, the samples that the method produces sometimes contain many replicates of very few values, which is unrealistic if the target is continuous.

The other methods try to address this deficiency. Smooth bootstrap forms a mixture of continuous distributions, with means given by the initial sample's members, and samples from it. We proved that, if the mixing weights have appropriately been assigned to the components of the mixture, the samples it gives also tend to come from the target as the initial sample's size tends to infinity. Moreover we proved that its sample statistics have the same properties as those of a weighted bootstrap sample. Partially smooth bootstrap is a hybrid between smooth and weighted bootstrap. Iterated smooth bootstrap consists of iterating the smooth bootstrap many times by using each sample

as the basis of a new better approximation to the target. Its performance can be unstable and a large computational effort is required. Augmentation methods perform weighted bootstrap but then construct a kernel density estimate of the target based on the weighted bootstrap sample and draw a new sample from it.

The methods have performance diagnostics which are easy to compute and to monitor automatically. They are very handy for the analysis of dynamic models because they are fast (with the exception of iterated smooth bootstrap) and can utilise samples from posterior distributions of previous time points in obtaining the samples from the current one. We compared them in many simulation experiments and there was no clear winner among them. However, because of its probabilistic soundness and ease of applicability, we will be using smooth bootstrap.

A problem with all resampling methods is the choice of a good importance sampler. This has to be as close to the target as possible and its support must include that of the target. Usually the latter is not fully known and sometimes its support may have an awkward shape. There has been a lot of recent research on this issue. In one of the following chapters we will examine some recent proposals.

Prior	$N(-0.15, 0.2)$	$N(-0.15, 1.5)$	$N(0.15, 1.5)$	$N(0.15, 0.2)$
Posterior	$\mu_1 = -0.1854$ $\sigma_1^2 = 0.0666$	$\mu_1 = -0.1998$ $\sigma_1^2 = 0.0938$	$\mu_1 = -0.181$ $\sigma_1^2 = 0.0938$	$\mu_1 = -0.0854$ $\sigma_1^2 = 0.0666$
n=100	$E(\bar{Y}) = -0.1853$ $V(\bar{Y}) = 0.0012$ $E(\sigma_Y^2) = 0.0635$	$E(\bar{Y}) = -0.1997$ $V(\bar{Y}) = 0.0023$ $E(\sigma_Y^2) = 0.0867$	$E(\bar{Y}) = -0.1807$ $V(\bar{Y}) = 0.0023$ $E(\sigma_Y^2) = 0.0873$	$E(\bar{Y}) = -0.0847$ $V(\bar{Y}) = 0.0013$ $E(\sigma_Y^2) = 0.0649$
n=1000	$E(\bar{Y}) = -0.1854$ $V(\bar{Y}) = 0.00012$ $E(\sigma_Y^2) = 0.0664$	$E(\bar{Y}) = -0.1998$ $V(\bar{Y}) = 0.00023$ $E(\sigma_Y^2) = 0.0931$	$E(\bar{Y}) = -0.181$ $V(\bar{Y}) = 0.00024$ $E(\sigma_Y^2) = 0.0931$	$E(\bar{Y}) = -0.0853$ $V(\bar{Y}) = 0.00013$ $E(\sigma_Y^2) = 0.0665$

Table 2.1: Posterior means and variances and expectations and variances of smooth bootstrap sample statistics in the case of inference for the mean of $N(\mu, 1)$.

Prior	$N(-0.15, 0.2)$	$N(-0.15, 1.5)$	$N(0.15, 1.5)$	$N(0.15, 0.2)$
n=100	$E(\bar{Y}) = -0.1872$ $V(\bar{Y}) = 0.0012$ $E(\sigma_Y^2) = 0.0667$	$E(\bar{Y}) = -0.2$ $V(\bar{Y}) = 0.0022$ $E(\sigma_Y^2) = 0.0946$	$E(\bar{Y}) = -0.1823$ $V(\bar{Y}) = 0.0027$ $E(\sigma_Y^2) = 0.094$	$E(\bar{Y}) = -0.0852$ $V(\bar{Y}) = 0.0014$ $E(\sigma_Y^2) = 0.0661$
n=1000	$E(\bar{Y}) = -0.185$ $V(\bar{Y}) = 0.00012$ $E(\sigma_Y^2) = 0.0668$	$E(\bar{Y}) = -0.1998$ $V(\bar{Y}) = 0.00025$ $E(\sigma_Y^2) = 0.0937$	$E(\bar{Y}) = -0.1811$ $V(\bar{Y}) = 0.00024$ $E(\sigma_Y^2) = 0.094$	$E(\bar{Y}) = -0.0852$ $V(\bar{Y}) = 0.00013$ $E(\sigma_Y^2) = 0.0665$

Table 2.2: Simulated expectations and variances of smooth bootstrap sample statistics in the case of inference for the mean of $N(\mu, 1)$.

Prior	$\mu_0 = 1.6, \sigma_0^2 = 1.024$ $S_0 = 11.2, \nu = 9$	$\mu_0 = 1.6, \sigma_0^2 = 5.12$ $S_0 = 4.8, \nu = 5$	$\mu_0 = 5, \sigma_0^2 = 1$ $S_0 = 260, \nu = 54$	$\mu_0 = 5, \sigma_0^2 = 5$ $S_0 = 60, \nu = 14$
Posterior	$\mu_1 = 1.1738$ $\sigma_1^2 = 0.1837$	$\mu_1 = 1.0427$ $\sigma_1^2 = 0.1977$	$\mu_1 = 4.3348$ $\sigma_1^2 = 0.6263$	$\mu_1 = 3.1252$ $\sigma_1^2 = 0.9767$
n=100	$E(\bar{Y}) = 1.1752$ $V(\bar{Y}) = 0.0032$ $E(\sigma_Y^2) = 0.1834$	$E(\bar{Y}) = 1.0437$ $V(\bar{Y}) = 0.0033$ $E(\sigma_Y^2) = 0.1978$	$E(\bar{Y}) = 4.3427$ $V(\bar{Y}) = 0.0169$ $E(\sigma_Y^2) = 0.615$	$E(\bar{Y}) = 3.1477$ $V(\bar{Y}) = 0.0373$ $E(\sigma_Y^2) = 0.951$
n=1000	$E(\bar{Y}) = 1.1739$ $V(\bar{Y}) = 0.00031$ $E(\sigma_Y^2) = 0.1837$	$E(\bar{Y}) = 1.0428$ $V(\bar{Y}) = 0.00033$ $E(\sigma_Y^2) = 0.1977$	$E(\bar{Y}) = 4.3355$ $V(\bar{Y}) = 0.0017$ $E(\sigma_Y^2) = 0.6252$	$E(\bar{Y}) = 3.1275$ $V(\bar{Y}) = 0.0038$ $E(\sigma_Y^2) = 0.9742$

Table 2.3: Posterior means and variances and expectations and variances of smooth bootstrap sample statistics in the case of inference for the variance of $N(0, \sigma^2)$.

Prior	$\mu_0 = 1.6, \sigma_0^2 = 1.024$ $S_0 = 11.2, \nu = 9$	$\mu_0 = 1.6, \sigma_0^2 = 5.12$ $S_0 = 4.8, \nu = 5$	$\mu_0 = 5, \sigma_0^2 = 1$ $S_0 = 260, \nu = 54$	$\mu_0 = 5, \sigma_0^2 = 5$ $S_0 = 60, \nu = 14$
n=100	$E(\bar{Y}) = 1.1745$ $V(\bar{Y}) = 0.003$ $E(\sigma_Y^2) = 0.183$	$E(\bar{Y}) = 1.0471$ $V(\bar{Y}) = 0.0034$ $E(\sigma_Y^2) = 0.2013$	$E(Y) = 4.3474$ $V(\bar{Y}) = 0.0167$ $E(\sigma_Y^2) = 0.609$	$E(Y) = 3.151$ $V(\bar{Y}) = 0.0367$ $E(\sigma_Y^2) = 0.9676$
n=1000	$E(\bar{Y}) = 1.1739$ $V(\bar{Y}) = 0.00031$ $E(\sigma_Y^2) = 0.1832$	$E(\bar{Y}) = 1.0422$ $V(\bar{Y}) = 0.00033$ $E(\sigma_Y^2) = 0.1974$	$E(\bar{Y}) = 4.3362$ $V(\bar{Y}) = 0.0018$ $E(\sigma_Y^2) = 0.6273$	$E(\bar{Y}) = 3.1275$ $V(\bar{Y}) = 0.0035$ $E(\sigma_Y^2) = 0.9735$

Table 2.4: Simulated expectations and variances of smooth bootstrap sample statistics in the case of inference for the variance of $N(0, \sigma^2)$.

Posterior	Corr. Coeff. (ζ-score)	
1	-0.296	(-0.305)
2	-0.313	(-0.324)
3	-0.31	(-0.321)
4	-0.308	(-0.318)
5	-0.308	(-0.318)
6	-0.307	(-0.317)

Table 2.5: True posterior correlation coefficients and ζ-scores for Example 2.5.2.

	Corr. Coeff. (z-score) (95% confidence intervals for ζ)				
Posterior	W.B.	Sm. Boot.	Par. Sm. B.	S - T.	G et al
1	-0.302 (-0.312) (-0.326,-0.298)	-0.278 (-0.286) (-0.3,-0.272)	-0.301 (-0.311) (-0.324,-0.296)	-0.331 (-0.344) (-0.358,-0.33)	-0.287 (-0.295) (-0.309,-0.281)
2	-0.299 (-0.308) (-0.322,-0.295)	-0.275 (-0.282) (-0.296,-0.268)	-0.321 (-0.333) (-0.346,-0.319)	-0.333 (-0.346) (-0.36,-0.332)	-0.309 (-0.319) (-0.333,-0.306)
3	-0.302 (-0.312) (-0.326,-0.298)	-0.298 (-0.307) (-0.321,-0.293)	-0.311 (-0.322) (-0.336,-0.308)	-0.324 (-0.336) (-0.35,-0.322)	-0.297 (-0.306) (-0.32,-0.292)
4	-0.28 (-0.288) (-0.302,-0.274)	-0.244 (-0.249) (-0.263,-0.235)	-0.342 (-0.356) (-0.37,-0.342)	-0.334 (-0.347) (-0.361,-0.333)	-0.323 (-0.335) (-0.349,-0.321)
5	-0.294 (-0.303) (-0.317,-0.289)	-0.289 (-0.297) (-0.311,-0.283)	-0.321 (-0.333) (-0.346,-0.319)	-0.306 (-0.316) (-0.333,-0.302)	-0.335 (-0.348) (-0.362,-0.335)
6	-0.3 (-0.31) (-0.323,-0.296)	-0.302 (-0.312) (-0.326,-0.298)	-0.304 (-0.314) (-0.328,-0.3)	-0.329 (-0.342) (-0.36,-0.328)	-0.322 (-0.334) (-0.348,-0.32)

Table 2.6: Posterior sample correlation coefficients, z-scores and 95% confidence intervals for the ζ-scores, in Example 2.5.2.

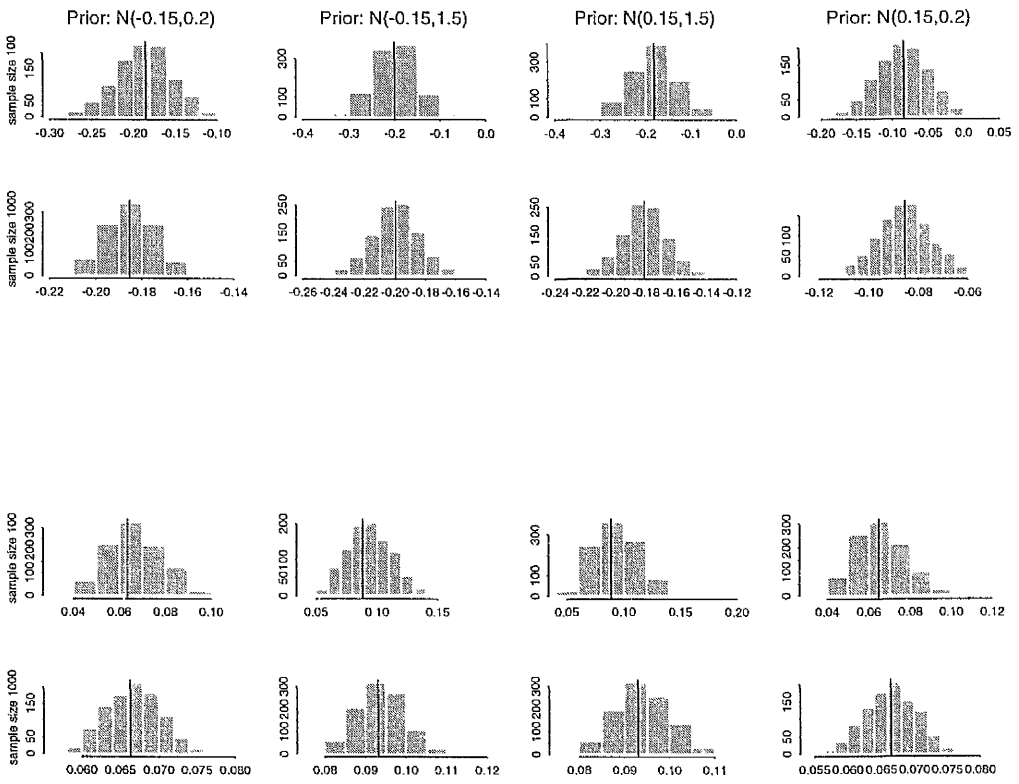


Figure 2.1: Means and variances of 1000 smooth bootstrap samples from the posterior of the mean of a $N(\mu, 1)$ population for four different choices of prior. The vertical lines denote the location of the expected value of the sample means and variances according to formulae (1.14) and (1.15). The top two rows show histograms of means and the bottom two show histograms of variances.

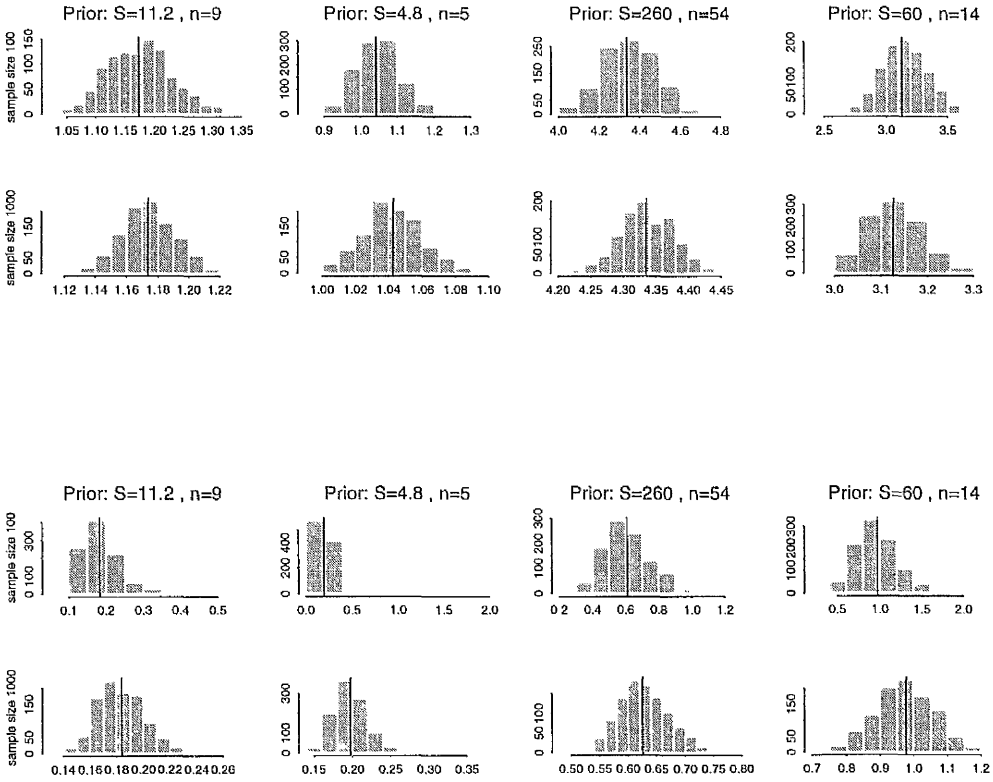


Figure 2.2: Means and variances of 1000 smooth bootstrap samples from the posterior of the variance of a $N(0, \sigma^2)$ population for four different choices of $S\chi_\nu^{-2}$ prior. The vertical lines denote the location of the expected value of the sample means and variances according to formulae (1.14) and (1.15). The top two rows show histograms of means and the bottom two show histograms of variances.

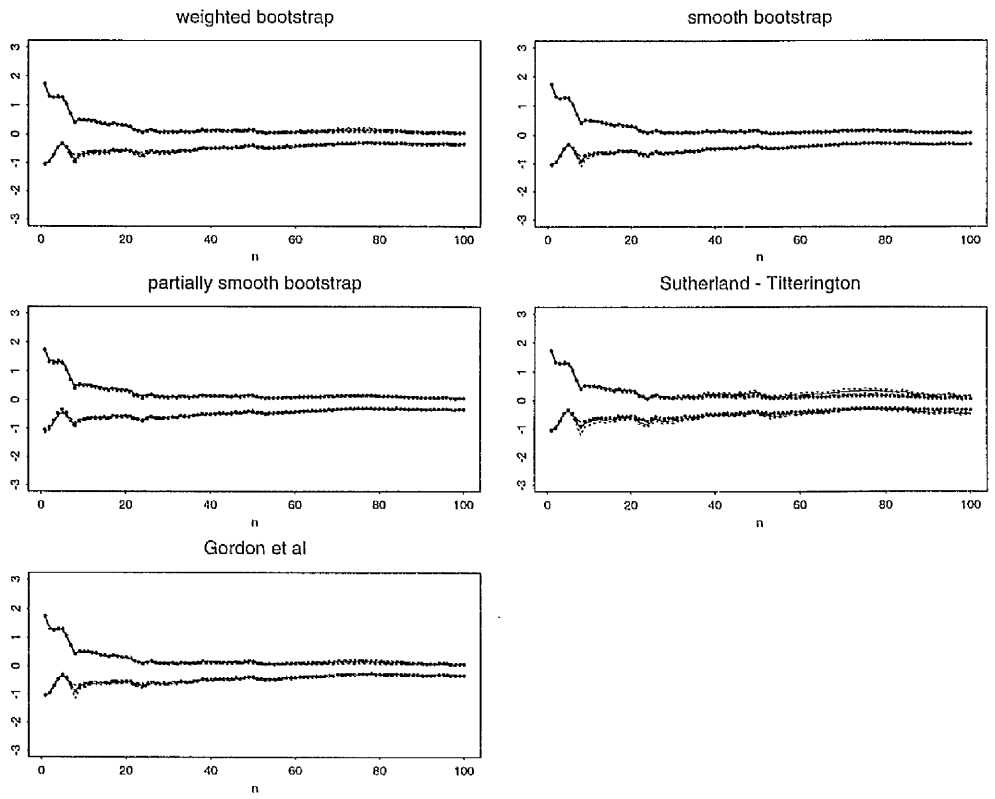


Figure 2.3: True and estimated 95% HDI's of the 100 posteriors arising in Example 2.5.1. Points: true interval endpoints. Solid lines: estimated intervals. Broken lines: Confidence bands for interval endpoints.

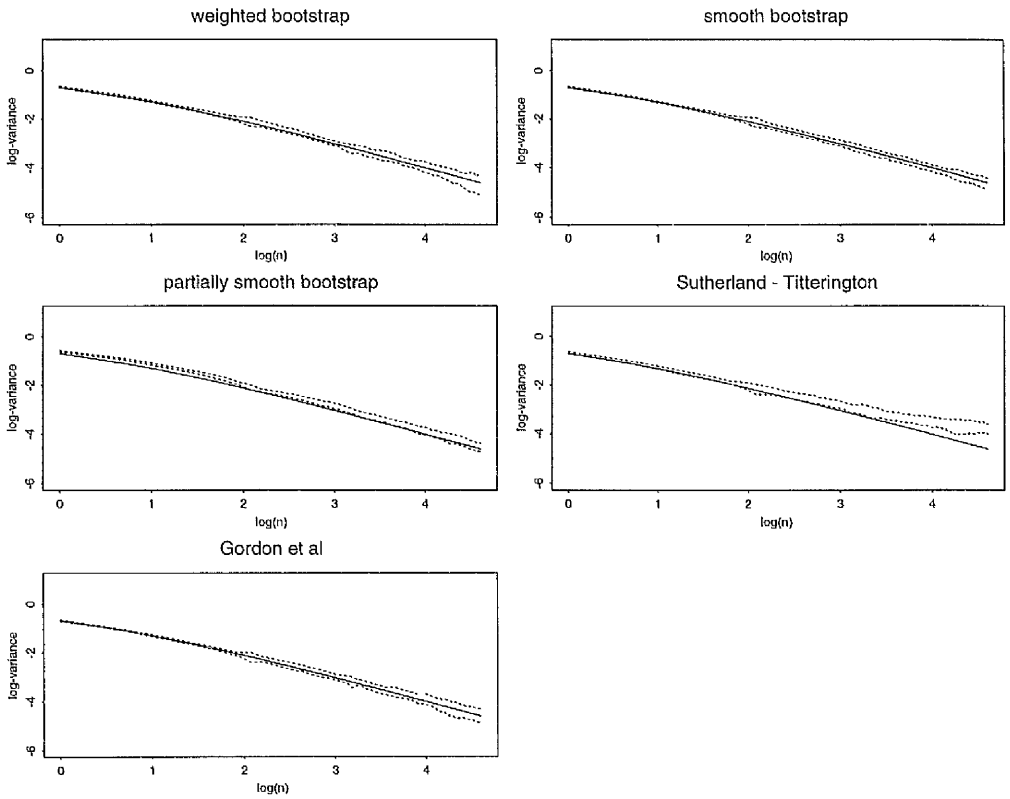


Figure 2.4: True posterior variances and 95% interval estimates of them (in logarithmic scale) for Example 2.5.1. Solid line: true variance. Broken lines: interval estimates.

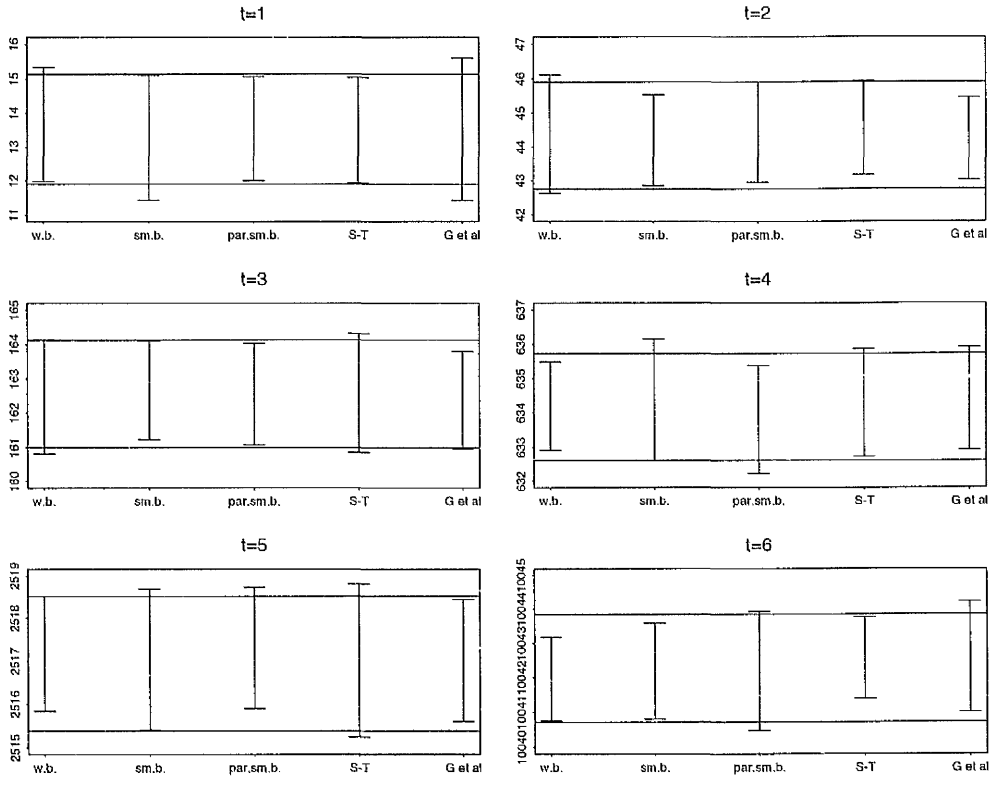


Figure 2.5: True 95% HDI's and sample 95% HDI's of the posteriors of the first component of x_t for $t = 1\ldots,6$, in Example 2.5.2. Lines: True 95% HDI's. Segments: Sample 95% HDI's.

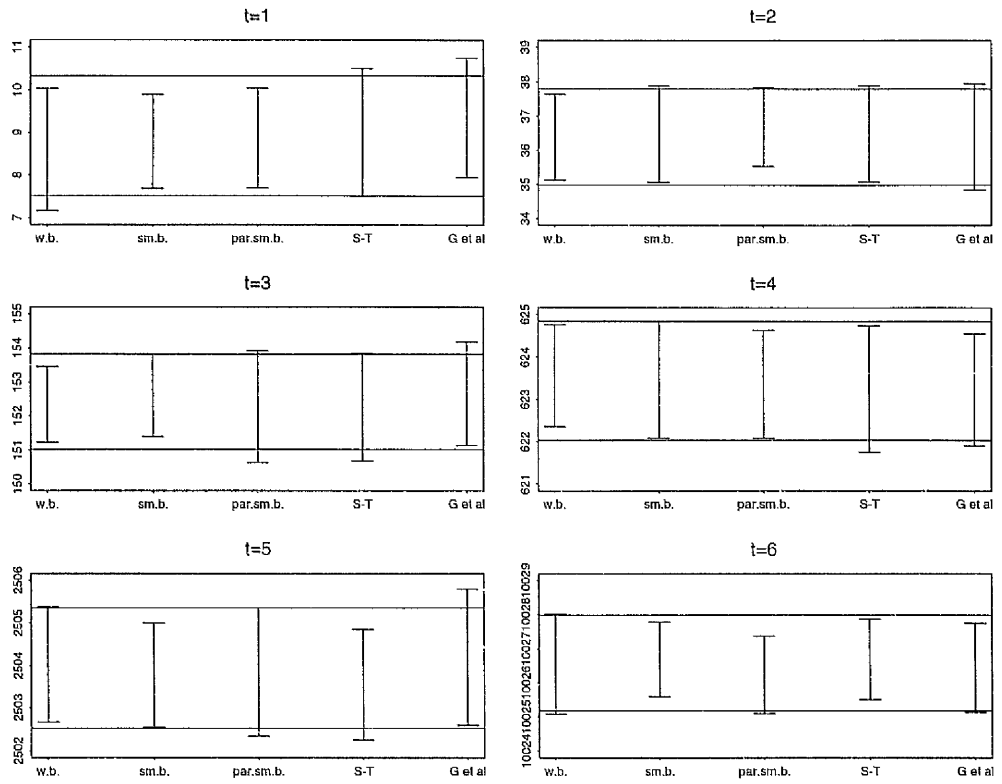


Figure 2.6: True 95% HDI's and sample 95% HDI's of the posteriors of the second component of x_t for $t = 1 \ldots, 6$, in Example 2.5.2. Lines: True 95% HDI's. Segments: Sample 95% HDI's.

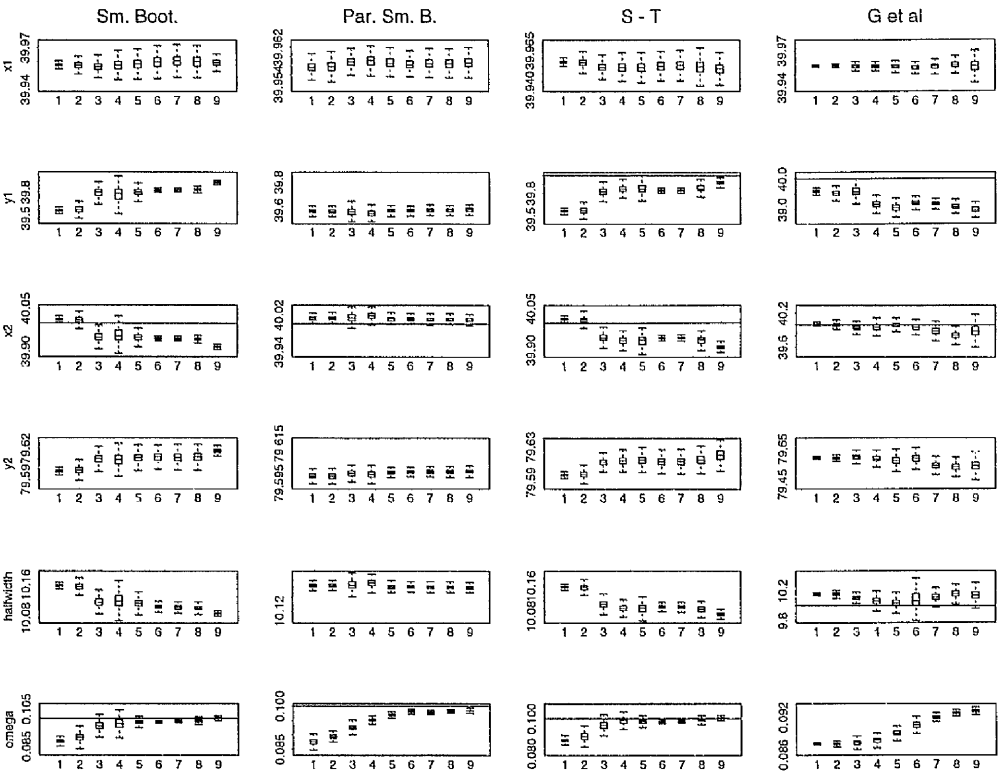


Figure 2.7: True parameter values and posterior sample boxplots for Example 2.5.3. Lines: true parameter values (when they could be accommodated in the graphs!).

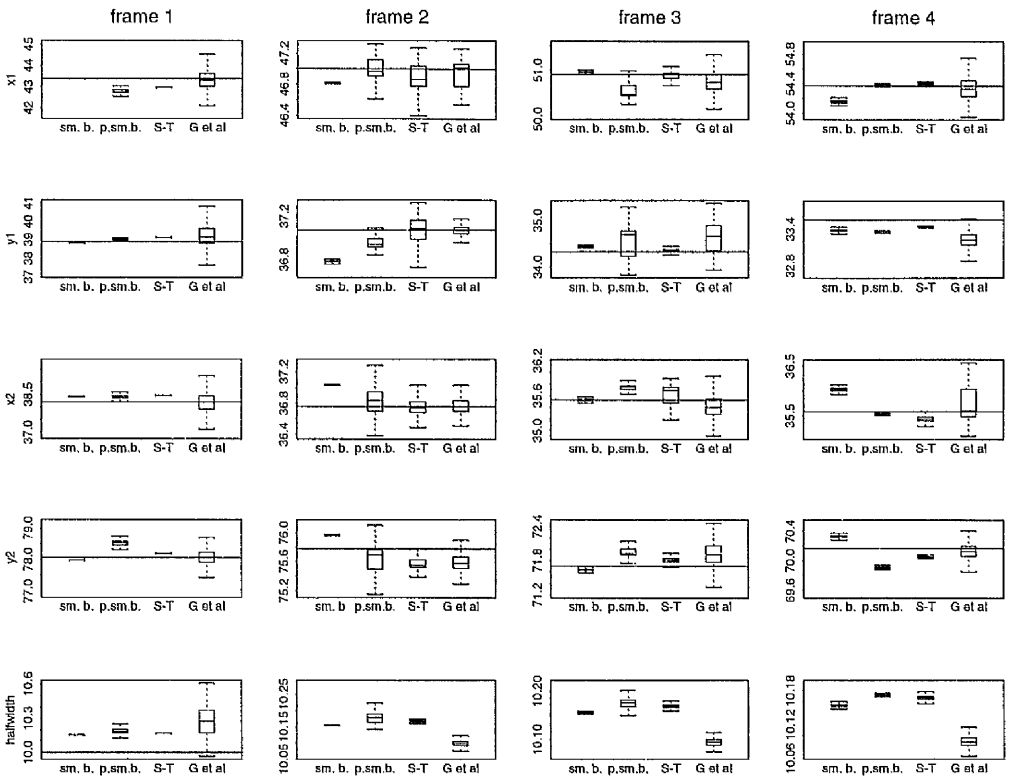


Figure 2.8: True parameter values and posterior sample boxplots for Example 2.5.4. Lines: true parameter values (when they could be accomodated in the graphs!).

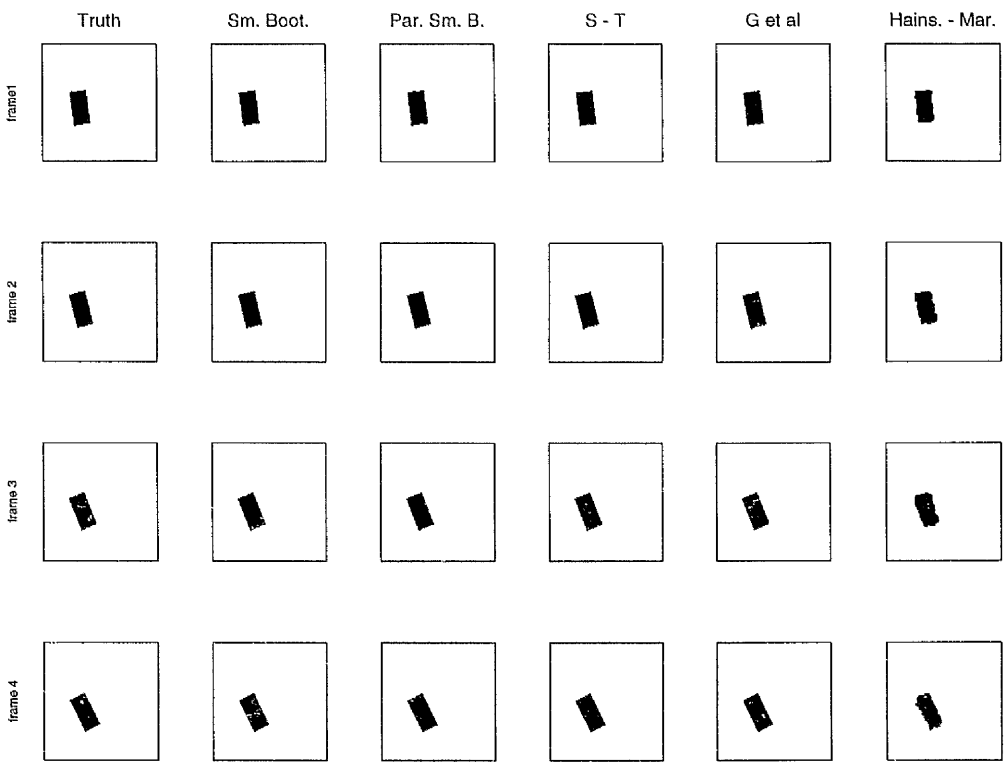


Figure 2.9: True images and reconstructions for Example 2.5.4.

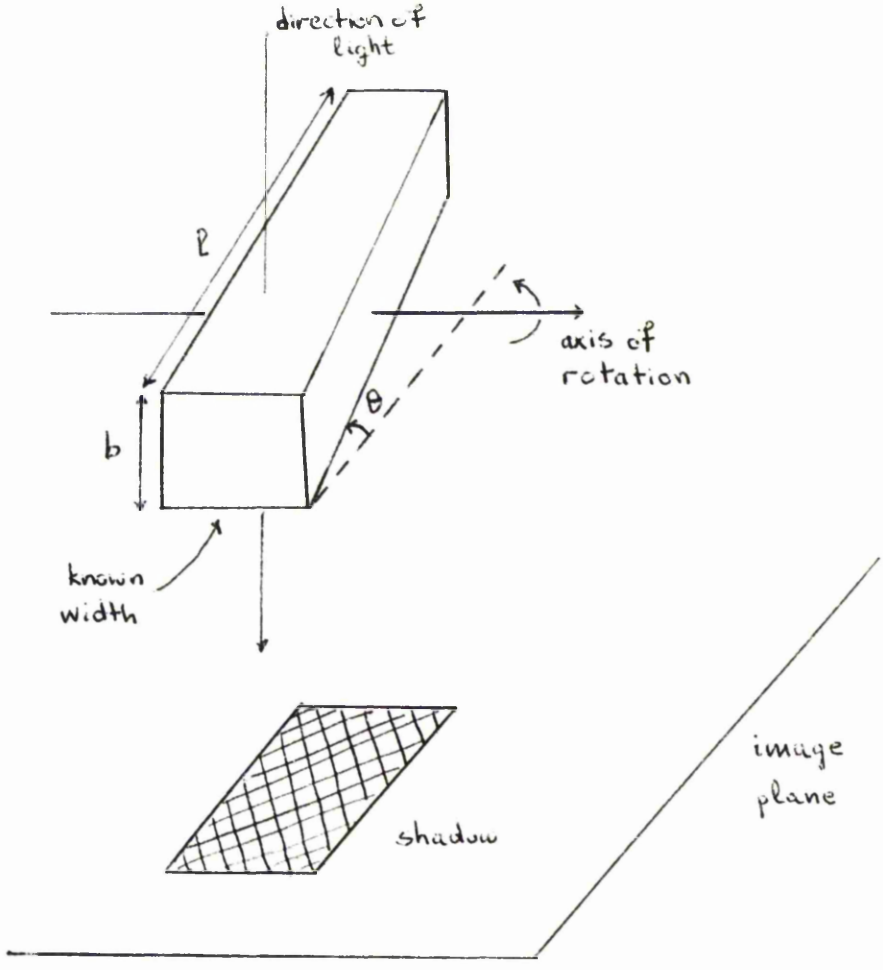


Figure 2.10: The cuboid at a starting position (forming angle θ with the image plane), its shadow and the directions of light and rotation, from example 2.5.5.

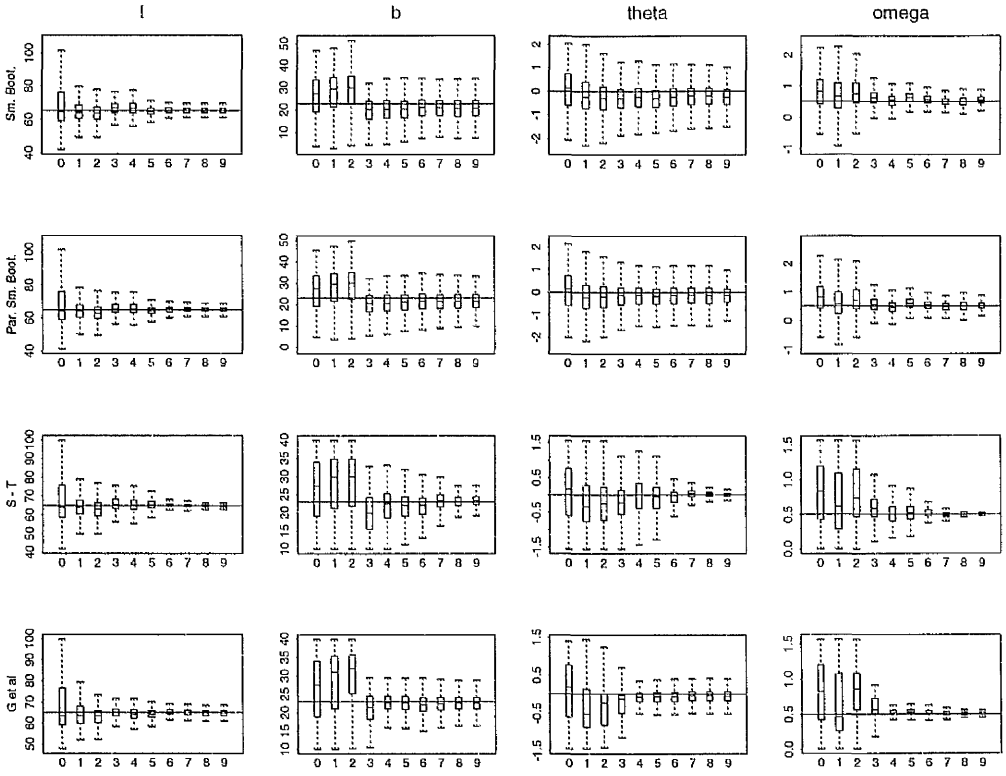


Figure 2.11: True parameter values and posterior sample boxplots for Example 2.5.5. Lines: true parameter values.

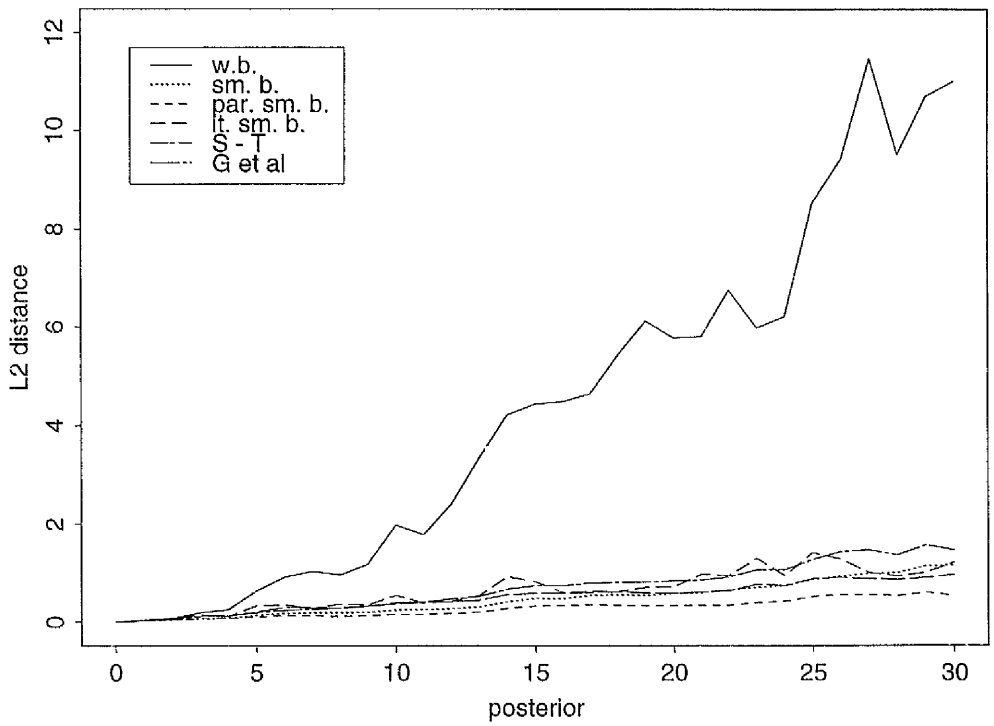


Figure 2.12: Average L_2 distances of density estimates from the true posteriors, in the Bayesian learning setting of Example 2.5.6.

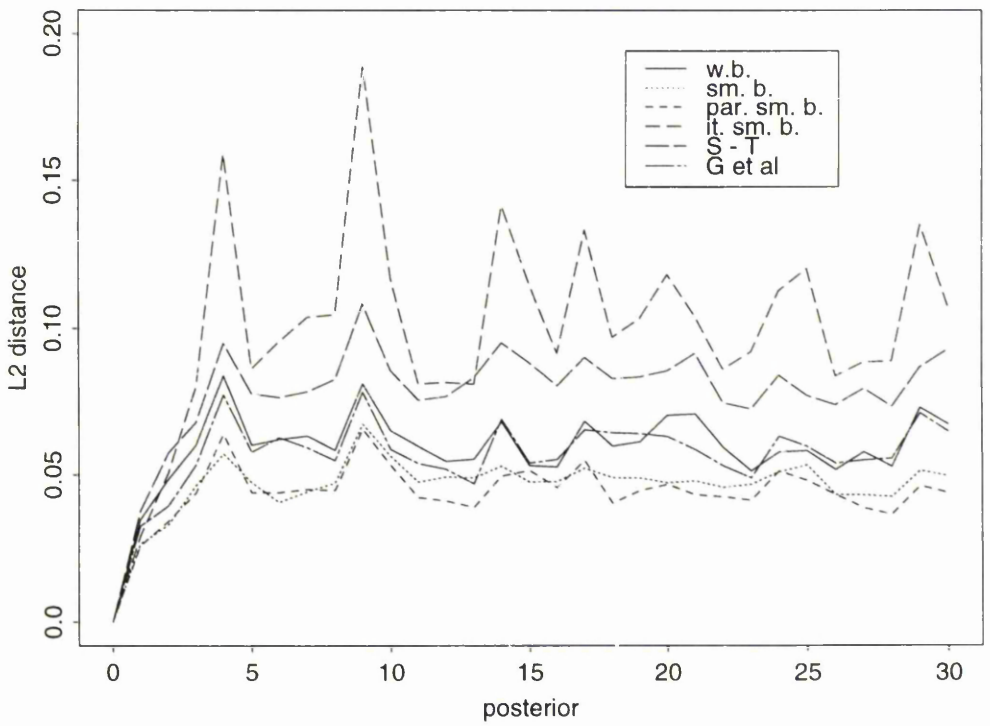


Figure 2.13: Average L_2 distances of density estimates from the true posteriors, in the dynamic models setting of Example 2.5.6.

Chapter 3

A control problem

3.1 Introduction

In this and the subsequent chapters we deal with a problem arising in the area of stochastic control. We tackle it using smooth bootstrap. The motivation for this work is twofold: first, the problem is interesting in its own right and second, it causes the smooth bootstrap in its usual form to break down. This fact leads to the consideration of possible improved resampling algorithms. As an introduction we give a brief description of control in layman's terms.

Control is used in numerous mechanical and biological systems. Some examples will make this clearer. When a virus attacks an organism, the immune system releases specialized cells in order to destroy the intruder. Sensors in air-conditioning devices "read" a room's temperature and regulate accordingly the amount of cool air introduced into it. An automatic pilot maintains the desired direction of the aircraft or makes adjustments to it when necessary. In general, we can say that we have a system whose state can be specified in

some way. When this state differs from a desired one by a significant amount control is employed to eliminate the difference.

In a mechanical system we observe an output which can either be the system's state itself or some other quantity related to it. If the output leads us to believe that the state is not the one we want we can change it by supplying some input to the system. This input is called control. Therefore **control** is the name for both the process that tries to change the system's state and for the actual input that causes this change. The most interesting cases are those where the state changes dynamically in time even without intervention by us. Then the relevant output must be monitored continuously and the appropriate action must be taken every time it is necessary. Sometimes more than one input can have the same effect on the state and then we seek the optimal one for each situation. This is what the mathematical theory of control deals with.

A usual situation with which we deal here is when the dynamics of the state's evolution in time can be described by a dynamic model that incorporates random disturbances. Moreover, the output is a quantity related to the state and contaminated by random noise. When in a control problem random quantities are involved we talk of **stochastic control**.

This chapter is organised as follows. In the remainder of this section we present the problem we are studying. Then we give a mathematical description of the general control theory. We show why this cannot be applied to our problem and we present an alternative approach. We adapt this and some other methods to our problem and present their practical implementation via resampling. A simulations section compares the methods under several variants of the problem. After the simulations we provide a theoretical analysis

of a simpler version of the problem trying to explain the performance of the method that has turned out to be the best. Finally, we close the chapter with a short discussion.

3.1.1 The problem we are dealing with

The problem presented in this chapter belongs to the category of **extremum adaptation** problems; see Titterington (1973). These arise in cases where we have a response curve or surface which changes location in time and provides us with noisy observations which are used in order to track, in an optimal way, an extreme point of this curve.

In the particular case at hand, the response curve is a parabola. Its minimum touches the real line and the curve moves along the real line in a random manner. The movement takes place in discrete time. Suppose that at time t the minimum is at x_t. Since we do not know where it is, if we guess that it is at u_t, what we will observe is the value of the parabola at u_t, $\phi(u_t, x_t)$ say, with some random noise ϵ_t added to it,

$$y_t = \phi(u_t, x_t) + \epsilon_t.$$

We can assume that ϵ_t are independent, identically distributed (i.i.d.) $N(0, \sigma_\epsilon^2)$ random variables. We also assume that they are independent of x_s for any s. The form of the parabola is

$$\phi(u_t, x_t) = \frac{1}{2}(x_t - u_t)^2. \tag{3.1}$$

Furthermore, we assume that x_1, the minimum of the curve at the first time point, comes from a $N(0, \sigma_1^2)$ distribution and that the parabola moves in such a way that the movement of x_t is a random walk:

$$x_t = x_{t-1} + \eta_t,$$

where η_t are i.i.d. $N(0, \sigma_\eta^2)$ random variables. Moreover, η_t is independent from ϵ_s and x_s for all s and t.

Equation (3.1) can be seen as a cost function. It gives the cost we expect to pay for not knowing the value of x_t and guessing it is u_t. In industrial applications the cost could be production cost, u_t and x_t could be production conditions, and trying to track x_t would amount to trying to manufacture the product at the optimal conditions each time. If y_t could be observed without any noise there would again be ambiguity concerning the location of the minimum. At each instant t there would be two candidates for its location, namely the solutions of the equation

$$y_t = \frac{1}{2}(x_t - u_t)^2,$$

which are $u_t - \sqrt{2y_t}$ and $u_t + \sqrt{2y_t}$. The presence of noise complicates things even further.

3.2 General control theory

Here we present the mainstream control theory following mainly Aoki (1967). We only consider the discrete time case. Suppose that the state of a system at time $t+1$ is given by the equation

$$x_{t+1} = F_t(x_t, u_t, \eta_t, \alpha_t),$$

while what can be observed at time t is the output

$$y_t = G_t(x_t, \epsilon_t, \beta_t).$$

u_t is the control to be chosen at time t. Note that it affects the state of the system at time $t+1$. In the problem we are studying u_t affects the output at time t but not the next state of the system. The α's and β's are unknown parameters which may vary with time. The η's and ϵ's are random noise. For an exposition of general control theory no assumption of independence is necessary about them. Their distributions may also involve unknown quantities. We denote by D_t all the information available at time t prior to choosing u_t. It comprises $\mathbf{y}_t = (y_1, \ldots, y_t), \mathbf{u}_{t-1} = (u_0, \ldots, u_{t-1})$, the prior information about x_0 and the posterior distributions of all the unknown quantities given $\mathbf{y}_t$ and $\mathbf{u}_{t-1}$. We consider **closed-loop** controls which means that u_t is a function of D_t only.

In choosing the controls $u_0, u_1, \ldots$ a **performance index** has to be decided upon so that the chosen controls are optimal according to it. The index will

normally be indicated by the system itself. It may also depend on whether the controls must be chosen for the next N time units or for an infinite period. The first case is easier and we start with it. Assume that currently we are at time 0. A usual form of the performance index is

$$J = E\left[\sum_{t=1}^{N} W_t(x_t, u_{t-1})\right] = \sum_{t=1}^{N} E\left[W_t(x_t, u_{t-1})\right] = \sum_{t=1}^{N} R_t.$$

$W_t(x_t, u_{t-1})$ is a generic notation for a **cost function** that can have any form as long as it takes non-negative values. Its arguments are x_t and u_{t-1} because the latter affects the former. The sequence of optimal controls is the one that minimizes J. It can be found either by **backward recursion** or by **induction**, both methods leading to the same result.

In backward recursion we start by assuming that $u_0, u_1, \ldots, u_{N-2}$ have already been chosen and that $\mathbf{y}_{N-1}$ is available. All that remains is to find the optimal control u^*_{N-1} for time $N-1$. According to the performance index this will be the minimizer of $E\left[W_N(x_N, u_{N-1})\right]$. But $E[W_N] = E\left(E[W_N|D_{N-1}]\right)$ and therefore $E[W_N]$ is minimized only if $E[W_N|D_{N-1}]$ is minimized for every D_{N-1}. The derivation of $E[W_N|D_{N-1}]$ involves the posterior distributions of all the unknown quantities but is at least in principle feasible, and so is its minimization. We denote $E\left[W_N(x_N, u^*_{N-1})|D_{N-1}\right]$ by γ^*_N. Note that γ^*_N depends, among other things, on $u_0, u_1, \ldots, u_{N-2}$.

Then we proceed to find u^*_{N-2}. This will be the minimizer of

$$E\left[W_{N-1}(x_{N-1}, u_{N-2}) + W_N(x_N, u^*_{N-1})|D_{N-2}\right]$$

$$= E\left[W_{N-1}(x_{N-1}, u_{N-2})|D_{N-2}\right] + E\left(E\left[W_N(x_N, u^*_{N-1})|D_{N-1}\right]|D_{N-2}\right)$$
$$= E\left[W_{N-1}(x_{N-1}, u_{N-2}) + \gamma^*_N|D_{N-2}\right].$$

The minimization of this quantity is also feasible in principle. Moving further back in the same fashion we can derive $u^*_0, u^*_1, \ldots, u^*_{N-1}$.

In induction the reverse procedure is followed. At first u^*_0 is derived for $N = 1$ as the minimizer of $E\left[W_1(x_1, u_0)|D_0\right]$. Then u^*_0 is derived for $N = 2$ as the minimizer of $E\left[W_1(x_1, u_0) + W_2(x_2, u^*_1)|D_0\right]$ where u^*_1 is given by the same formula as u^*_0 for $N = 1$ but all the quantities that there depended on D_0 here depend on D_1. Going on in the same way $u^*_0, u^*_1, \ldots, u^*_{N-1}$ are derived. Of course irrespective of the order in which we derive their mathematical formulae, the values of the controls are calculated and applied in the proper order.

If we wish to consider the infinite horizon case (i.e. infinite N) we must find the limit of u^*_0 as N goes to infinity. All controls will be given by the same formula, changing only the quantities referring to the unknown parameters so that they reflect their changing posterior distributions as new data are being collected. Of course in the infinite horizon case the performance index may tend to infinity so that it cannot be minimized. Then we may consider a new index

$$J = E\left[\sum_{t=1}^{N} \frac{W_t(x_t, u_{t-1})}{N}\right].$$

If the index represents total loss then this last form can be seen as the average loss per unit time. This new J may also tend to infinity. However, if we can

find a sequence of control points that keeps it finite the system is called **controllable**.

For more information on control theory see Aoki(1967), Whittle (1969), Wishart (1969) and Bar-Shalom (1981) among others. For an example of the infinite horizon case see Drenick and Shaw (1964).

3.2.1 Difficulties with the problem at hand

The standard control theory we presented in the previous section breaks down when applied to the problem we study. Suppose that M optimal controls $u_1^*, \ldots, u_M^*$ have already been chosen and that we also want the N next ones. The performance index indicated by the system is slightly different than the one used in the previous exposition because in our problem we want each u_t to be close to x_t. Therefore we take

$$J = E\left[\sum_{t=1}^{N}(x_{M+t} - u_{M+t})^2 | D_M\right].$$

Note that here D_M is the same as in the previous section apart from the fact that it also includes u_M. We try to derive the optimal controls by induction.

For $N = 1$ we have to minimize

$$J = E\left[(x_{M+1} - u_{M+1})^2 | D_M\right] = E(x_{M+1}^2 | D_M) - 2u_{M+1}E(x_{M+1}|D_M) + u_{M+1}^2,$$

which leads to

$$\frac{\partial J}{\partial u^*_{M+1}} = 0 \Rightarrow 2u^*_{M+1} - 2E(x_{M+1}|D_M) = 0 \Rightarrow u^*_{M+1} = E(x_{M+1}|D_M).$$

Therefore, u^*_{M+1}, when the horizon is $M+1$, is the mean of the predictive distribution of x_{M+1} given D_M. That it minimizes J can be seen from the fact that J is a quadratic function of u_{M+1}.

For $N = 2$ problems appear. We have to minimize

$$J = E\left[(x_{M+1} - u_{M+1})^2 + (x_{M+2} - u^*_{M+2})^2|D_M\right],$$

where u^*_{M+2} corresponds to u^*_{M+1} above with the time index increased by one, i.e. $u^*_{M+2} = E(x_{M+2}|D_{M+1})$. Thus,

$$\begin{aligned} J &= E\left[(x_{M+1} - u_{M+1})^2 + (x_{M+2} - E(x_{M+2}|D_{M+1}))^2|D_M\right] \\ &= E\left((x_{M+1} - u_{M+1})^2 + E\left[(x_{M+2} - E(x_{M+2}|D_{M+1}))^2|D_{M+1}\right]|D_M\right) \\ &= E\left[(x_{M+1} - u_{M+1})^2 + Var(x_{M+2}|D_{M+1})|D_M\right]. \end{aligned}$$

But $x_{M+2} = x_{M+1} + \eta_{M+2}$ and therefore,

$$J = E\left[(x_{M+1} - u_{M+1})^2 + Var(x_{M+1}|D_{M+1}) + \sigma^2_\eta|D_M\right].$$

Unfortunately, the posterior variance of x_{M+1} given D_{M+1} cannot be obtained

in closed form because of the non-linearity of y_{M+1} and therefore the procedure breaks down. This means that at least in the particular case at hand a different approach must be taken.

3.3 An alternative approach

Titterington (1973) analyses the same problem but in a continuous time setting. The non-linearity of the observation equation prevents him from using the standard control theory. We hereby summarize his approach.

Assuming that x_0 *a priori* has a unimodal symmetric distribution with mode 0 we can set $u_0 = 0$. If we keep $u_t = 0$ the posterior distribution of x_t, for any t, will also be symmetric around 0. If x_t moves far from 0 its distribution will moreover become bimodal. The author argues that when x_t is very close to 0 the noise dominates the observations and we should keep $u_t = 0$. We should choose a value K such that by the time $|x_t|$ gets close to it the observations are dominated by their deterministic part. When this happens we set u_t equal to K or $-K$ at random. Of course we cannot observe x_t so we change u_t when the modes of the posterior distribution of x_t reach K and $-K$. When we have made the change a **test period** ensues during which we use the observations gathered to check whether the correct mode was chosen as u_t. If it turns out that the wrong mode had been picked, we take the other one as u_t. Then a new **passage period** begins with u_t constant and the variable of interest now being $|x_t - u_t|$. The whole process is an alternation of passage and test periods. The length T of the test period depends on the power of the test. The author chooses a very large power so that the possibility that after the test u_t is not in the correct place can effectively be dismissed. If moreover

T is not very large x_t will not move a lot during the test and all passage periods can be considered probabilistically identical. The same applies to all test periods.

The author's performance index represents expected loss and for a period of length T_0 is defined as

$$J = \int_0^{T_0} E\left[\phi(x_t, u_t)\right] dt.$$

Because the horizon is infinite the author uses the expected rate of loss

$$\gamma(K) = \frac{\sum_{passages} E(\text{loss during passage}) + \sum_{tests} E(\text{loss during test})}{\sum_{passages} E(\text{passage time to } \pm K) + \sum_{tests} \text{test time}}$$

which because of the above arguments about passage and test periods becomes

$$\gamma(K) = \frac{E(\text{loss during one passage}) + E(\text{loss during one test})}{E(\text{first passage time to } \pm K) + T},$$

The optimal value of K can be found by minimizing $\gamma(K)$.

A practical consideration is that of assessing that $|x_t|$ has reached K. The posterior distribution of x_t has to be used in this respect but it is not of a nice form. The author uses the technique of Gaussian sums approximation (Alspach and Sorenson (1972)), mentioned in an earlier chapter. More specifically the author approximates the posterior distribution $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ by a mixture, with equal weights, of two Normal densities with the same variance v_t and with means $-m_t$ and m_t and provides formulae for the updating

of m_t and v_t in time given the observations. Therefore, we monitor m_t and when it reaches K, u_t is changed. Simulations made by the author show that m_t approximates $|x_t|$ well when this is in the proximity of K, which is what matters.

Titterington's idea is also appropriate in our discrete time setting. His method however cannot be directly adapted to it for several reasons. For example, when the author calculates $\gamma(K)$ he uses a theorem that gives the expected loss during one passage period and the duration of a passage period. The equivalent of this theorem cannot be easily derived in the discrete time case. Moreover, in calculating T the author assumes that the aggregate observation of the test period is Normal, which does not hold in our case. In the next section we will implement Titterington's idea using a sample-based approach.

3.4 Sample-based methods

In this section we propose some methods for choosing u_t, derived directly from the discrete time setting. They are modified so that they can be applied when only samples are available from the arising distributions of interest.

First of all we recall and introduce some notation to be used in all the presentation that follows. $\mathbf{y}_t = (y_1, \ldots, y_t)$ are the data collected up to time t and $\mathbf{u}_t = (u_1, \ldots, u_t)$ are the corresponding control points. $\mathbf{x}_t = (x_1, \ldots, x_t)$ denotes the sequence of true curve minima up to time t.

3.4.1 An adaptation of Titterington's method

The logic behind this method is the same as in the continuous time setting. At each time t we want the distance between u_t and x_t to be at most K, where K is a value chosen by us. All information about x_t is encompassed in its posterior distribution $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. After observing y_t we want to choose u_{t+1}. We argue that the best choice of u_{t+1} is the mode of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. Although it seems even better to choose as u_{t+1} the mode of the prior distribution of x_{t+1}, $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$, our simulations have proved otherwise. We will try to explain why this is so in our simulations section. From a theoretical point of view the best choice for u_{t+1} is the mean of $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$. However, careful consideration reveals that if this was our choice we would have $u_t = E(x_1)$ for all t. This is clearly a very unsatisfactory solution to the problem. When x_t is far from u_t, $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ will be symmetric around u_t and bimodal. If the distance of the modes of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ from u_t is larger than K we set u_{t+1} equal to one of the modes chosen at random. Then, $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$ is not symmetric around u_{t+1} and therefore, neither is $p(x_{t+1}|\mathbf{y}_{t+1}, \mathbf{u}_{t+1})$. Moreover even if $p(x_{t+1}|\mathbf{y}_{t+1}, \mathbf{u}_{t+1})$ is bimodal only one of its two modes will be global. An altogether different case is when x_t is close to u_t where we may get a unimodal $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ with u_t as its mode.

In the light of all the above the argument about choosing u_{t+1} is as follows. If the distance between the global mode of the posterior distribution $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ and u_t is greater than K set u_{t+1} equal to the mode, otherwise retain $u_{t+1} = u_t$. If $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ is bimodal and both modes are global choose one of them at random as u_{t+1}. If at some point t x_t happens to get very far from u_t the global mode of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ will probably also be far from u_t. If its distance from u_t is larger than K we will set u_{t+1} equal to it. Therefore even if we

begin to lose track of x_t this is automatically corrected. This eliminates the need for a test period.

As in the continuous time setting, the distributions $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ are analytically intractable because of the non-linearity of the observation equation. We suggest representing them by samples taken from them. We will deal with how to obtain the samples later on.

We have developed a heuristic for finding the global mode of a sample from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. As we will see in the simulations section, if the sample is bimodal and we split it in the middle into two subsamples, each of them presents an almost symmetric histogram. Applying the formula appropriate for slightly non-symmetric unimodal distributions

$$\text{mode} = 3 \cdot \text{median} - 2 \cdot \text{mean}, \tag{3.2}$$

(Kendall and Stuart (1963), sec. 2.11), we can find the mode of each subsample and therefore the two modes of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. However, $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ could also be unimodal with u_t as its mode as we have said. In that case (3.2) will give two pseudomodes. From a sample we cannot distinguish automatically between unimodality and bimodality. Therefore, we do the following at each time t.

- Obtain the sample from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$.
- Calculate the average of its smallest and largest member.
- Using that as a split point, create two subsamples.
- Find the pseudomode in each subsample.

- Compare the two pseudomodes and u_t to get the global mode.
- If the distance between the global mode and u_t is greater than K set u_{t+1} equal to the global mode, otherwise set $u_{t+1} = u_t$.

We take as global mode the point that has the most sample values around it in a zone of width equal to half the distance between the two pseudomodes.

The discreteness of the situation does not allow us to use theoretical arguments like Titterington (1973) does in order to derive an optimal value for K. This will have to be chosen on the basis of simulations of the system.

3.4.2 A probabilistic criterion

Here and in the next section we look at the problem of choosing u_{t+1} from a different perspective. We wish u_{t+1} to be close to x_{t+1}. How close can it be? We can never answer this since x_{t+1} is a random quantity. We can only try to maximize the probability that its distance from u_{t+1} will be smaller than a quantity V. This probability is stated with respect to the predictive distribution of x_{t+1} given the information up to time t. In other words we want u_{t+1} to maximize

$$\Pr\left(|X_{t+1} - u_{t+1}| \leq V | \mathbf{y}_t, \mathbf{u}_t\right).$$

If we define an indicator variable

$$I(x_{t+1}, u_{t+1}) = \begin{cases} 1 & \text{if} \quad |x_{t+1} - u_{t+1}| \leq V \\ 0 & \text{otherwise} \end{cases}$$

then

$$\Pr\left(|X_{t+1} - u_{t+1}| \leq V | \mathbf{y}_t, \mathbf{u}_t\right) = E\left[I(X_{t+1}, u_{t+1}) | \mathbf{y}_t, \mathbf{u}_t\right].$$

If we have a sample $x^*_{t+1}(1), \ldots, x^*_{t+1}(n)$ from $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$ then the probability above can be estimated by

$$\hat{\Pr}\left(|X_{t+1} - u_{t+1}| \leq V | \mathbf{y}_t, \mathbf{u}_t\right) = \frac{1}{n} \sum_{i=1}^{n} I(x^*_{t+1}(i), u_{t+1}) \tag{3.3}$$

Alternatively, we can write

$$\begin{aligned}
&\Pr\left(|X_{t+1} - u_{t+1}| \leq V | \mathbf{y}_t, \mathbf{u}_t\right) \\
&= \int_{-\infty}^{\infty} \Pr\left(|X_{t+1} - u_{t+1}| \leq V | x_t\right) p(x_t | \mathbf{y}_t, \mathbf{u}_t) dx_t \\
&= \int_{-\infty}^{\infty} p(x_t | \mathbf{y}_t, \mathbf{u}_t) \int_{u_{t+1}-V}^{u_{t+1}+V} p(x_{t+1}|x_t) dx_{t+1} dx_t \\
&= \int_{-\infty}^{\infty} p(x_t | \mathbf{y}_t, \mathbf{u}_t) \left[\Phi\left(\frac{u_{t+1} + V - x_t}{\sigma_\eta}\right) - \Phi\left(\frac{u_{t+1} - V - x_t}{\sigma_\eta}\right)\right] dx_t \\
&= E\left[\Phi\left(\frac{u_{t+1} + V - X_t}{\sigma_\eta}\right) - \Phi\left(\frac{u_{t+1} - V - X_t}{\sigma_\eta}\right) | \mathbf{y}_t, \mathbf{u}_t\right]
\end{aligned}$$

where Φ denotes the cumulative distribution function of the standard Normal

distribution. This probability can also be estimated if we have a sample $x_t(1), \ldots, x_t(n)$ from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ by

$$\begin{aligned}&\hat{\Pr}\left(|X_{t+1} - u_{t+1}| \leq V|\mathbf{y}_t, \mathbf{u}_t\right) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\Phi\left(\frac{u_{t+1}+V-x_t(i)}{\sigma_\eta}\right) - \Phi\left(\frac{u_{t+1}-V-x_t(i)}{\sigma_\eta}\right)\right]. \quad (3.4)\end{aligned}$$

3.4.3 Working directly with the cost

Instead of requiring a small distance between u_{t+1} and x_{t+1} with high probability we may want to have small cost y_{t+1} with high probability. After all, the cost is what we observe and we will know at each time point whether our target has been met or not. We want a u_{t+1} that maximizes

$$\Pr(Y_{t+1} \leq b|\mathbf{y}_t, \mathbf{u}_t, u_{t+1})$$

for a quantity b. We can write

$$\begin{aligned}\Pr(Y_{t+1} \leq b|\mathbf{y}_t, \mathbf{u}_t, u_{t+1}) &= \int_{-\infty}^{b} p(y_{t+1}|\mathbf{y}_t, \mathbf{u}_t, u_{t+1})dy_{t+1} \\ &= \int_{-\infty}^{b}\int_{-\infty}^{\infty} p(y_{t+1}, x_{t+1}|\mathbf{y}_t, \mathbf{u}_t, u_{t+1})dx_{t+1}dy_{t+1} \\ &= \int_{-\infty}^{b}\int_{-\infty}^{\infty} p(y_{t+1}|x_{t+1}, u_{t+1})p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)dx_{t+1}dy_{t+1}\end{aligned}$$

$$
\begin{aligned}
&= \int_{-\infty}^{\infty} p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t) \int_{-\infty}^{b} p(y_{t+1}|x_{t+1}, u_{t+1}) dy_{t+1} dx_{t+1} \\
&= \int_{-\infty}^{\infty} p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t) \Phi\left(\frac{b - \frac{1}{2}(x_{t+1} - u_{t+1})^2}{\sigma_\epsilon}\right) dx_{t+1} \\
&= E\left[\Phi\left(\frac{b - \frac{1}{2}(X_{t+1} - u_{t+1})^2}{\sigma_\epsilon}\right) |\mathbf{y}_t, \mathbf{u}_t\right].
\end{aligned}
$$

Therefore, if $x^*_{t+1}(1), \ldots, x^*_{t+1}(n)$ is a sample from $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$ we can estimate the probability by

$$
\hat{\Pr}(Y_{t+1} \leq b|\mathbf{y}_t, \mathbf{u}_t, u_{t+1}) = \frac{1}{n} \sum_{i=1}^{n} \Phi\left(\frac{b - \frac{1}{2}(x^*_{t+1}(i) - u_{t+1})^2}{\sigma_\epsilon}\right). \tag{3.5}
$$

(3.3),(3.4) and (3.5) are functions of u_{t+1} and can be maximized with the aid of numerical techniques. Instead of changing u_{t+1} each time we could change it only if the relevant probability falls below a limit specified by us. This limit should be set close to 1 so that bad u_{t+1} are not retained. V or b can have any value as long as it is not very large.

3.4.4 Unknown variances

In the discussion so far, all the variances involved (of the prior of x_1, of the noise and of the system disturbances) have been considered known. This may not be the case in practice, but, fortunately, our methods do not change considerably in such a situation. We consider here two different possible settings.

- All variances, $\sigma_1^2, \sigma_\eta^2, \sigma_\epsilon^2$ are unknown.

- σ_1^2 is known, $\sigma_\eta^2 = \beta\sigma_\epsilon^2$ with β known and σ_ϵ^2 unknown.

In both situations it is easier for Bayesian analysis to work with the precisions, which are the reciprocals of the variances. The posteriors are now joint distributions for x_t and the unknown precisions. For Titterington's method no modification is needed. When the posterior sample has been obtained, we only deal with the x_t part of it. For the probabilistic methods slight changes are needed in the calculation of the probabilities involved. In the case of totally unknown variances they are derived as follows.

$$
\begin{aligned}
&\Pr\left(|X_{t+1} - u_{t+1}| \le V | \mathbf{y}_t, \mathbf{u}_t\right) \\
&= \int_{u_{t+1}-V}^{u_{t+1}+V} p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t) dx_{t+1} \\
&= \int_{u_{t+1}-V}^{u_{t+1}+V} \int_{-\infty}^{\infty} \int_{0}^{\infty} p(x_{t+1}, x_t, \phi_\eta | \mathbf{y}_t, \mathbf{u}_t) d\phi_\eta dx_t dx_{t+1} \\
&= \int_{-\infty}^{\infty} \int_{0}^{\infty} p(x_t, \phi_\eta | \mathbf{y}_t, \mathbf{u}_t) \int_{u_{t+1}-V}^{u_{t+1}+V} p(x_{t+1}|x_t, \phi_\eta) dx_{t+1} d\phi_\eta dx_t \\
&= E\left[\Phi\left((u_{t+1} + V - X_t)\phi_\eta^{1/2}\right) - \Phi\left((u_{t+1} - V - X_t)\phi_\eta^{1/2}\right) | \mathbf{y}_t, \mathbf{u}_t\right].
\end{aligned}
$$

Therefore, if $(x_t(1), \phi_\eta(1)), \ldots, (x_t(n), \phi_\eta(n))$ is a sample from $p(x_t, \phi_\eta|\mathbf{y}_t, \mathbf{u}_t)$ we have the estimator

$$
\begin{aligned}
\hat{\Pr}\left(|X_{t+1} - u_{t+1}| \le V | \mathbf{y}_t, \mathbf{u}_t\right) &= \frac{1}{n}\sum_{i=1}^{n}\left[\Phi\left((u_{t+1} + V - x_t(i))\phi_\eta^{1/2}(i)\right)\right. \\
&- \left.\Phi\left((u_{t+1} - V - x_t(i))\phi_\eta^{1/2}(i)\right)\right]. \qquad (3.6)
\end{aligned}
$$

If one prefers to use (3.3), this is directly applicable by using only the x_{t+1} part of a sample from $p(x_{t+1}, \phi_\eta, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t)$.

In the method involving Y_{t+1} we have

$$\begin{aligned}
\Pr(Y_{t+1} \leq b | \mathbf{y}_t, \mathbf{u}_t, u_{t+1}) &= \int_{-\infty}^{b} p(y_{t+1} | \mathbf{y}_t, \mathbf{u}_t, u_{t+1}) dy_{t+1} \\
&= \int_{-\infty}^{b} \int_{-\infty}^{\infty} \int_{0}^{\infty} p(y_{t+1}, x_{t+1}, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t, u_{t+1}) d\phi_\epsilon dx_{t+1} dy_{t+1} \\
&= \int_{-\infty}^{\infty} \int_{0}^{\infty} p(x_{t+1}, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t) \int_{-\infty}^{b} p(y_{t+1} | x_{t+1}, \phi_\epsilon, u_{t+1}) dy_{t+1} d\phi_\epsilon dx_{t+1} \\
&= E\left[\Phi\left((b - \frac{1}{2}(X_{t+1} - u_{t+1})^2) \phi_\epsilon^{1/2} \right) | \mathbf{y}_t, \mathbf{u}_t \right].
\end{aligned}$$

Now, if $(x_{t+1}^*(1), \phi_\epsilon^*(1)), \ldots, (x_{t+1}^*(n), \phi_\epsilon^*(n))$ is a sample from $p(x_{t+1}, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t)$ we have the estimator

$$\hat{\Pr}(Y_{t+1} \leq b | \mathbf{y}_t, \mathbf{u}_t, u_{t+1}) = \frac{1}{n} \sum_{i=1}^{n} \left[\Phi\left((b - \frac{1}{2}(x_{t+1}^*(i) - u_{t+1})^2) \phi_\epsilon^*(i)^{1/2} \right) \right]. \tag{3.7}$$

Of course, all the above means that at each instant we have samples for all $x, \phi_\epsilon, \phi_\eta$ and we use whichever part of them is relevant to each estimator.

When only σ_ϵ^2 is unknown we have $\sigma_\eta^2 = \beta \sigma_\epsilon^2$ which means $\phi_\epsilon = \beta \phi_\eta$ and therefore, while (3.7) remains unchanged, (3.6) becomes

$$\hat{\Pr}\left(|X_{t+1} - u_{t+1}| \leq V | \mathbf{y}_t, \mathbf{u}_t\right) = \frac{1}{n}\sum_{i=1}^{n}\left[\Phi\left(\frac{(u_{t+1} + V - x_t(i))\phi_\epsilon^{1/2}(i)}{\sqrt{\beta}}\right) - \Phi\left(\frac{(u_{t+1} - V - x_t(i))\phi_\epsilon^{1/2}(i)}{\sqrt{\beta}}\right)\right] \tag{3.8}$$

where $(x_t(1), \phi_\epsilon(1)), \ldots, (x_t(n), \phi_\epsilon(n))$ is a sample from $p(x_t, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t)$.

3.5 Resampling implementation

It is obvious from the previous discussion that the application of any of the methods in any situation in practice will require the generation of samples from a possibly very large number of distributions. The dynamic nature of the problem makes the resampling techniques the most suitable for this purpose, as has been made clear in the previous chapters. Because we may have to follow x_t for a long time we will use smooth bootstrap so that the samples do not degenerate.

Depending on our degree of knowledge about the variances, different samples and therefore different resampling weights are required. In all cases though, if we start with a sample from the prior we can easily update it to get samples from all the subsequent posterior and prior distributions. The resampling weights will always be equal to the likelihood of the prior sample points.

3.5.1 Known variances

The prior for x_1 is $N(0, \sigma_1^2)$ and therefore easy to sample from. Suppose that at time t a sample $x_t^*(1), \ldots, x_t^*(n)$ from $p(x_t|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ is available. After y_t is observed at u_t, the weight of sample point $x_t^*(i)$ is equal to

$$w_t(i) = \exp\left[-\frac{1}{2\sigma_\epsilon^2}\left(y_t - \frac{1}{2}(x_t^*(i) - u_t)^2\right)^2\right].$$

After resampling, we end up with a sample $x_t(1), \ldots, x_t(n)$ from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. Then, because

$$p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t) = \int_{-\infty}^{\infty} p(x_{t+1}|x_t)p(x_t|\mathbf{y}_t, \mathbf{u}_t)dx_t$$

we can obtain a sample $x_{t+1}^*(1), \ldots, x_{t+1}^*(n)$ from it by setting

$$x_{t+1}^*(i) = x_t(i) + \eta_{t+1,i}$$

for each $i = 1, 2, \ldots, n$, where $\eta_{t+1,1}, \ldots, \eta_{t+1,n}$ are points sampled from $N(0, \sigma_\eta^2)$. This constitutes a full transition from one prior to the next.

3.5.2 Unknown noise and disturbance variances with known ratio β

The unknowns here are x and ϕ_ϵ, the precision of the noise. For simplicity we denote the latter by ϕ. At each time instant we need samples from both x_t and

ϕ. Since the variance of x_1 is known we can consider x_1 and ϕ independent *a priori*. Following common Bayesian practice we take the prior of ϕ to be Gamma. Its parameterization will be explained in the simulations section. It is easy to obtain a sample from $p(x_1, \phi)$. We get points from $N(0, \sigma_1^2)$ and to each one of them we attach a point from the Gamma prior of ϕ.

Suppose that at time t a sample $(x_t^*(1), \phi^*(1)), \ldots, (x_t^*(n), \phi^*(n))$ from $p(x_t, \phi|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ is available. When y_t is observed at u_t, the weight of point $(x_t^*(i), \phi^*(i))$ is equal to

$$w_t(i) = \phi^*(i)^{1/2} \exp\left[-\frac{\phi^*(i)}{2}\left(y_t - \frac{1}{2}(x_t^*(i) - u_t)^2\right)^2\right].$$

Resampling gives us the sample $(x_t(1), \phi(1)), \ldots, (x_t(n), \phi(n))$ from $p(x_t, \phi|\mathbf{y}_t, \mathbf{u}_t)$. Then, because

$$p(x_{t+1}, \phi|\mathbf{y}_t, \mathbf{u}_t) = \int_{-\infty}^{\infty} p(x_{t+1}|x_t, \phi) p(x_t, \phi|\mathbf{y}_t, \mathbf{u}_t) dx_t,$$

we can get a sample $(x_{t+1}^*(1), \phi^*(1)), \ldots, (x_{t+1}^*(n), \phi^*(n))$ from it by setting

$$\begin{aligned} x_{t+1}^*(i) &= x_t(i) + \eta_{t+1,i} \\ \phi^*(i) &= \phi(i) \end{aligned}$$

for each $i = 1, 2, \ldots, n$, where $\eta_{t+1,i}$ is a point from $N(0, \frac{\beta}{\phi(i)})$. This is easily explained by the fact that $p(x_{t+1}|x_t, \phi)$ is $N(0, \frac{\beta}{\phi})$ since $\beta = \frac{\phi_\epsilon}{\phi_\eta}$.

We should note here that the very first prior distribution, $p(x_1, \phi)$ can be very diffuse, especially in the direction of ϕ, if very little is known about ϕ. A sample of size n from it may contain very few points or no point at all close to the true values of x_1 and ϕ. Then the sample from $p(x_1, \phi|y_1, u_1)$ will not be truly representative of the posterior. All the subsequent analysis will be based on very unstable foundations. For this reason we select the size of the first prior sample to be a lot larger than n. Alternatively one could use MCMC for obtaining this first sample. We set u_1 equal to the prior mean of x_1, i.e. $u_1 = 0$.

3.5.3 Totally unknown variances

The unknowns here are $x, \phi_1, \phi_\eta, \phi_\epsilon$, although ϕ_1 is not necessary in the main bulk of the calculations. We leave the sampling from the first prior aside for the moment because of some peculiarities which have to be taken into account.

Suppose that, at time t, $(x_t^*(1), \phi_\eta^*(1), \phi_\epsilon^*(1)), \ldots, (x_t^*(n), \phi_\eta^*(n), \phi_\epsilon^*(n))$ from $p(x_t, \phi_\eta, \phi_\epsilon|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ are available. After y_t is observed at u_t, the weight of point $(x_t^*(i), \phi_\eta^*(i), \phi_\epsilon^*(i))$ is equal to

$$w_t(i) = \phi_\epsilon^*(i)^{1/2} \exp\left[-\frac{\phi_\epsilon^*(i)}{2}\left(y_t - \frac{1}{2}(x_t^*(i) - u_t)^2\right)^2\right].$$

Note that ϕ_η makes no contribution at all to the weight. The resampling gives $(x_t(1), \phi_\eta(1), \phi_\epsilon(1)), \ldots, (x_t(n), \phi_\eta(n), \phi_\epsilon(n))$ from $p(x_t, \phi_\eta, \phi_\epsilon|\mathbf{y}_t, \mathbf{u}_t)$. Following similar thinking to before, we get a sample $(x_{t+1}^*(1), \phi_\eta^*(1), \phi_\epsilon^*(1)), \ldots,$ $(x_{t+1}^*(n), \phi_\eta^*(n), \phi_\epsilon^*(n))$ from $p(x_{t+1}, \phi_\eta, \phi_\epsilon|\mathbf{y}_t, \mathbf{u}_t)$ by setting

$$\begin{aligned} x^*_{t+1}(i) &= x_t(i) + \eta_{t+1,i} \\ \phi^*_\eta(i) &= \phi_\eta(i) \\ \phi^*_\epsilon(i) &= \phi_\epsilon(i), \end{aligned}$$

where $\eta_{t+1,i}$ is a point from $N(0, \frac{1}{\phi_\eta(i)})$ for each $i = 1, 2, \ldots, n$.

As we see, while ϕ_η is essential in propagating posterior samples, any value of it could in theory go through the resampling step. We have to ensure that the initial pool of points of ϕ_η is good. This makes the sampling from the very first prior and the first resampling very crucial. If we use only the first observation y_1 any point ϕ_η from the prior can pass through the resampling. For this reason we have to consider $x_1, x_2, \phi_1, \phi_\eta, \phi_\epsilon$ as a block. Then, using observations y_1 and y_2 we will get a sample from $p(x_1, x_2, \phi_1, \phi_\eta, \phi_\epsilon|\mathbf{y}_2, \mathbf{u}_2)$. The prior of this block of unknowns is

$$p(x_1, x_2, \phi_1, \phi_\eta, \phi_\epsilon) = p(\phi_1)p(\phi_\eta)p(\phi_\epsilon)p(x_1|\phi_1)p(x_2|x_1, \phi_\eta).$$

The priors for the precisions are again Gamma, $p(x_1|\phi_1)$ is the density (p.d.f.) of $N(0, \frac{1}{\phi_1})$ and $p(x_2|x_1, \phi_\eta)$ is the p.d.f. of $N(x_1, \frac{1}{\phi_\eta})$. The parameterization of the Gamma priors is explained in the simulations section. To get $(x^*_1(i), x^*_2(i), \phi^*_1(i), \phi^*_\eta(i), \phi^*_\epsilon(i))$ from the prior, we independently sample $\phi^*_1(i), \phi^*_\eta(i)$ and $\phi^*_\epsilon(i)$ from their corresponding priors, then $x^*_1(i)$ from $N(0, \frac{1}{\phi^*_1(i)})$ and finally $x^*_2(i)$ from $N(x^*_1(i), \frac{1}{\phi^*_\eta(i)})$. The weight corresponding to it is

$$w(i) = \phi_\epsilon^*(i) \exp\left\{-\frac{\phi_\epsilon^*(i)}{2}\left[\left(y_1 - \frac{1}{2}(x_1^*(i) - u_1)^2\right)^2 + \left(y_2 - \frac{1}{2}(x_2^*(i) - u_2)^2\right)^2\right]\right\}.$$

The resampling will favour "good" pairs of x_1 and x_2 and therefore "good" points for ϕ_η too. We do not care about "good" points for ϕ_1 since straight after the resampling we forget about it. The size of the first prior sample is again a lot larger than n. We set $u_1 = u_2 = 0$.

3.5.4 A small amendment to the resampling algorithms

In simulations we have observed that smooth bootstrap sometimes breaks down when applied to the problem at hand. More specifically, sometimes all the resampling weight goes to a single point of the prior sample and the subsequent samples "lose" x_t. In this section we explain why this happens and we suggest a remedy. We refer separately to the cases of known and unknown variances.

Known variances. The weight assigned to a prior sample point x, when we have observed y at u, viewed as a function of x is

$$w(x) = \exp\left[-\frac{1}{2\sigma_\epsilon^2}\left(y - \frac{1}{2}(x-u)^2\right)^2\right].$$

Since $w(x)$ is proportional to the likelihood of x, it is maximized at the maximum likelihood estimates of x

$$x = \begin{cases} u + \sqrt{2y} \quad \text{and} \quad u - \sqrt{2y} & \text{if} \quad y > 0 \\ u & \text{if} \quad y \leq 0 \end{cases}$$

which we shall call MLE's from now on. Incidentally, the value of the MLE for $y \leq 0$ can be seen as another justification for not changing u when the noise dominates the deterministic part of the observation. A negative y can be the result of either the true x being very close to u or of very large noise, and in either case it is better to retain the same u until more data have been gathered.

Suppose that we have, at time t, a sample from $p(x_t|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$. Being a sample it does not cover the whole support of the prior it represents. If the true x_t comes from the tails of the prior it will lead to a positive y_t and the two MLE's may be outside the sample's range. If this happens and σ_ϵ is very small, the weight function will be very peaked around the MLE's and almost 0 away from them. Then the sample point closest to one of the MLE's will get all the weight and will be exclusively favoured during resampling. If this point happens to be far from the true x_t the resulting sample will lose x_t. The same will hold for all the subsequent samples unless by chance another extreme x_t occurs in the future which "lands" in the corresponding prior sample's range.

To prevent this from happening we suggest the following solution. When $y_t > 0$ and both MLE's are outside the prior sample's range we scrap this sample and we do not perform resampling. Instead we take as the posterior $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ a mixture, with equal mixing weights, of two Normals: $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$, where

$$\begin{aligned}\mu_1 &= u_t + \sqrt{2y_t} \\ \mu_2 &= u_t - \sqrt{2y_t} \\ \sigma &= \frac{\sqrt{2y_t + 4\sigma_\epsilon}}{2}.\end{aligned}$$

The value of σ is derived heuristically based on the following argument. It holds with probability higher than 0.975 that

$$\frac{1}{2}(x_t - u_t)^2 \leq y_t + 2\sigma_\epsilon \Longrightarrow u_t - \sqrt{2y_t + 4\sigma_\epsilon} \leq x_t \leq u_t + \sqrt{2y_t + 4\sigma_\epsilon}.$$

Taking $\sigma = \frac{\sqrt{2y_t+4\sigma_\epsilon}}{2}$ ensures that the mixture will cover with probability higher than 0.975 the range

$$\begin{aligned}&\left(\mu_2 - \sqrt{2y_t + 4\sigma_\epsilon}, \mu_1 + \sqrt{2y_t + 4\sigma_\epsilon}\right) \\ &= \left(u_t - \sqrt{2y_t} - \sqrt{2y_t + 4\sigma_\epsilon}, u_t + \sqrt{2y_t} + \sqrt{2y_t + 4\sigma_\epsilon}\right)\end{aligned}$$

which is wider than $\left(u_t - \sqrt{2y_t + 4\sigma_\epsilon}, u_t + \sqrt{2y_t + 4\sigma_\epsilon}\right)$. Therefore, when we sample n points from the mixture the chance that the true x_t will not be in the range of the sample is negligible.

From this mixture we take a sample of size n and then we return to the usual procedure until such **intervention** is needed again. In our case the extreme observations are possible although unlikely under our model. In practice however, we cannot be sure that a model explains the behaviour of

the system under study. Therefore, outlying observations throw doubt on the model and intervention becomes necessary; see West and Harrison (1989, sec. 1.2.3).

With this amendment, our samples have never "lost" x_t in all our simulations so far.

Unknown variances. Irrespective of whether we know the ratio of noise and disturbance variances or not, the weight is a function of x and the noise precision, ϕ say,

$$w(x, \phi) = \phi^{1/2} \exp\left[-\frac{\phi}{2}\left(y - \frac{1}{2}(x-u)^2\right)^2\right].$$

The function is again proportional to the likelihood. If $y < 0$ the function has a maximum at $(u, 1/y^2)$. The case that can cause trouble is $y > 0$. For $x = u + \sqrt{2y}$ and $x = u - \sqrt{2y}$ (the MLE's) the value of w increases monotonically as ϕ increases and therefore w has no global maximum. However, for any ϕ, $w(x, \phi) < w(u + \sqrt{2y}, \phi) = w(u - \sqrt{2y}, \phi)$. Therefore, w forms two ridges in the direction of the MLE's which become steeper and taller as ϕ increases. In a sample from the prior, points with x not very close to the MLE's get very small weights. As a result of this the same problems as in the known variances case may appear. The solution we propose is the following.

When $y_t > 0$ and both MLE's are outside the range of the prior sample's x_t part, from the sample we keep the precision parts and, instead of resampling, we get x_t points from a mixture of two Normal distributions with the MLE's as their means and standard deviation

$$\sigma = \frac{\sqrt{2y_t + 4\bar{\sigma}_\epsilon}}{2},$$

where $\bar{\sigma}_\epsilon$ is an estimate of σ_ϵ derived from the noise precision sample points as

$$\bar{\sigma}_\epsilon = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{\phi_\epsilon^*(i)}}.$$

Again, with this amendment the samples have never "lost" x_t in any of our simulations so far.

3.6 Simulations

In this section we examine how the techniques mentioned so far in this chapter perform in practice. We begin with two very simple experiments for demonstrative purposes. With the first one we show what kind of pseudo-modes we get with the heuristic of section 3.4.1. We consider a case where the prior and system standard deviations are equal to 1 while the noise standard deviation is 0.1. This allows quite quick passage from unimodality to bimodality for $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. We keep $u_t = 0$ for all t and consider 25 time points. Figure 3.1 shows histograms of the samples taken from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$, for $t = 1, \ldots, 25$, and the two pseudo-modes for each one of them. The samples have size 1000 and were obtained with smooth weighted bootstrap. We can see that the global mode is picked by one of the two pseudo-modes unless the sample is unimodal without a very prominent mode.

With the second experiment we want to show that the probabilistic methods actually pick out the modes of the distributions to which they refer. This is expected because, in bell-shaped densities with not very narrow peaks, observations fall with the highest probability close to the modes. In Figure 3.2 we show the samples from $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$, for $t = 1, \ldots, 25$, with the maximizers of (3.4) and (3.5). In most cases the two values fall very close to each other. When they do not it is because the distribution is bimodal and each maximizer picks out a different mode. In all samples the overall mode is located by at least one of the two methods. In the maximization we used the non-linear minimization routine **e04bbf** of the NAG library for FORTRAN 77. The values of V and b were 1.5 and 0.15 respectively.

Although these results correspond to the case of known variances the methods pick out modes equally well in cases of unknown variances.

In the remainder of this section we present and discuss the results of an experiment designed to give us more insight into the problem of choosing K for Titterington's method and also in order to compare the performance of the methods in several situations. The criterion used to assess the performance is the mean rate of loss

$$E(\gamma_N) = \frac{E\left(\sum_{t=1}^{N}(x_t - u_t)^2\right)}{N}.$$

We consider five different combinations of values for the variances:

- $\sigma_1^2 = \sigma_\eta^2 = 1, \sigma_\epsilon^2 = 0.01$
- $\sigma_1^2 = \sigma_\eta^2 = \sigma_\epsilon^2 = 1$

- $\sigma_1^2 = \sigma_\eta^2 = 1, \sigma_\epsilon^2 = 4$
- $\sigma_1^2 = \sigma_\eta^2 = 4, \sigma_\epsilon^2 = 1$
- $\sigma_1^2 = \sigma_\eta^2 = \sigma_\epsilon^2 = 4.$

For each combination we also consider all three different situations concerning the degree of our knowledge about the variances. Therefore, in all, we compare the methods under fifteen settings.

For each setting we simulate the mean rate of loss using fifty simulated chains $x_1, \ldots, x_{100}$ of realizations of x. Each method is applied in turn, producing its own fifty chains of optimal control points $u_1, \ldots, u_{100}$. If x_{ti} and u_{ti} signify the true minimum and the control for time t and chain i, the mean rate of loss is estimated by

$$\hat{\gamma}_N = \frac{\sum_{t=1}^{N} \sum_{i=1}^{50} (x_{ti} - u_{ti})^2}{50 \cdot N} \quad , \quad N = 1, \ldots, 100.$$

In each setting we consider the two probabilistic methods referred to as probabilistic criteria 1 and 2, and for criterion 1 we use both estimators (3.4) and (3.3) based on the indicator variable. For probabilistic criterion 1 we set $V = 1.5$ which is quite small for the values of σ_η involved. For criterion 2 we set $b = 1.5$ unless $\sigma_\epsilon = 0.1$ where we set $b = 0.15$. Again these values are quite small for the corresponding values of σ_ϵ. In all cases if the probability involved is greater than 0.9 we set $u_{t+1} = u_t$. If it falls below 0.9 we choose as u_{t+1} the maximizer of the probability; 0.9 is large for the chosen values of V and b so that rarely do we have $u_{t+1} = u_t$. We also consider Titterington's method with three different values for K. The smaller of the three is 0 which

means that we change u_t each time. The larger is always one that we believe is too large for the variance values of the particular setting. In other words, for that value we change u_t less frequently. The particular values of K for each case are shown in the respective graphs.

The prior distribution for x_1 when the variance σ_1^2 is known is just $N(0, \sigma_1^2)$. The precisions have a Gamma(a, b) distribution assigned to them as prior. Its p.d.f. at point z is $p(z) \propto z^{a-1} \exp(-bz)$. Its mean is a/b and its variance is a/b^2. We choose a and b so that the variance is equal to 10 and the mean is equal to 1, unless the true value of the precision is 100 in which case the mean is also equal to 100. This leads to quite diffuse priors. The first control point, u_1, is always chosen equal to 0, the mean of x_1. The samples drawn from the distributions of interest have size 1000. When there are unknown variances the sample from the very first prior has size 10000.

Each one of the five Figures (3.3)-(3.7) refers to a single combination of variance values. The figures show the evolution of $\hat{\gamma}_N$ for each method as N goes to 100. A look at them reveals a few general characteristics. We see that each expected rate of loss quickly reaches a limit. In other words the system is in all cases controllable. It seems that in our problem the limit of γ_N does not depend on the degree of our knowledge about the variances. It does depend though on the system variances and, to a lesser extent, on the noise variance too.

In particular, increasing the system variances from 1 to 4 almost quadruples the limit as well. This happens if the noise variance is equal to 1. When it is equal to 4 the increase in the limit is smaller. Therefore, the same change in the system variances has a different effect according to the noise variance. Changes in the noise variance when the system variances remain constant

have less noticeable effects on the limit. When the system variances are equal to 1, increasing the noise variance from 0.01 to 1 and then to 4 causes small increases in the limit. When the system variances are equal to 4 increasing the noise variance from 1 to 4 does not have any effect.

In all settings but one, Titterington's method with $K = 0$ turns out to be the best. The exception is the case of known variances $\sigma_1^2 = \sigma_\eta^2 = 4, \sigma_\epsilon^2 = 1$, where $K = 1$ performs better than $K = 0$. Trying even larger ratios of noise to disturbance variance has not given any concrete evidence that a value of K other than 0 would be optimal. Intuitively, it does not seem wholly unreasonable that the best strategy is to change u_t after each new observation. A tentative explanation runs as follows.

In the continuous-time setting cost for not getting observations y_t at the minimum of the response curve is being incurred continuously. This is the reason why, when we change u, moving to K for example, we want very quickly to resolve whether we have made the right move or whether we should have gone to $-K$. Larger K leads to faster decisions and hence smaller costs. When we observe the system in discrete time we incur cost only every time we get a new observation. Therefore if, for example, after receiving y_{t-1} we decide to move u_t to m away from where x_t turns out to be the only cost we are going to have will be $(x_t - m)^2/2$ and y_t will lead us to the correct direction, because such is the nature of our method. In other words, there is no need for test periods and hence no need to have K other than 0. This shows that continuous and discrete time settings of the problem are not directly analogous.

In any case, since the theoretical results of Titterington (1973) cannot be adapted to the discrete time case our only tool for choosing K, if we are not

persuaded that $K = 0$ is the best, is simulation. We can simulate the system off-line, trying several values of K and picking out the one that performs best. There is no problem when the variances involved are known. If they are not, we can observe the system on-line for a while with $K = 0$. This will quickly give us an idea about the variances in the form of posterior distributions. Then we can again simulate the system off-line using combinations of quantiles of those distributions. However, the larger the number of unknowns the more difficult this approach becomes because we will have to simulate the system under more combinations of quantiles.

As we said earlier the mode of $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$ seems a more reasonable choice for u_{t+1} rather than the mode of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. However, in all the cases we have studied the expected rates of loss turn out to be higher when the former choice of u_{t+1} is used. We believe that the reason lies in the relationship between the posterior distribution of x_t and the prior distribution of x_{t+1}. The latter is the convolution of the former with a zero-mean Normal distribution. This means that if the posterior is symmetric and unimodal the two distributions will have the same modes. If however, the posterior is bimodal then the modes of the prior will lie between the modes of the posterior. Our choice of u_{t+1} will be conservative, not opting clearly for any of the posterior modes which are the more likely locations of the mean of x_{t+1}. This will be reflected in the posterior distribution of x_{t+1} which will be more diffuse than the one we would get if we had set u_{t+1} equal to the dominant mode of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$.

Each of the two policies for choosing u_t - we will call them "prior" and "posterior" policy - produces its own series of observations and its own succession of posterior distributions of x_t. Those given by the "posterior" policy are more accurate and this more than counterbalances the suboptimal choice of

u_t. This is demonstrated in Figure 3.8. We have applied both policies in order to track the same 10 values $x_1, \ldots, x_{10}$ of x generated by a system with $\sigma_1 = \sigma_\eta = 1$ and $\sigma_\epsilon = 0.1$. Both policies have started with the same sample of size 1000 from the prior distribution of x_1. The graphs show the 10 posterior samples of the same size produced by each of the policies and we see that those of the "posterior" policy are more accurate for most of the time. This fact explains why the probabilistic methods perform worse than our adaptation of Titterington's. They pick as u_{t+1} the mode of $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$.

3.7 Theoretical analysis

In this section we study mathematically the behaviour of the rate of loss and its expectation. We focus on Titterington's method with $K = 0$. The situation we have studied so far is complicated since it involves random variation of the minimum, non-linearity of the observation equation and random observation noise. For this reason we examine two analogous but simpler cases which, we hope, offer sufficient insight into the problem. In both of them we have a randomly varying minimum x_t and we want to choose u_t as "close" to it as possible. The movement of x_t is described by the same random walk as before.

First, suppose that x_t can be observed directly and without any error but that we still have to provide a good guess u_t for it. Having observed $x_1, \ldots, x_{t-1}$ the best guess is $u_t = x_{t-1}$. For $t = 1$ we choose $u_1 = E(x_1)$. Then the rate of loss is

$$\gamma_N = \frac{\sum_{t=1}^{N}(x_t - u_t)^2}{N} = \frac{(x_1 - E(x_1))^2}{N} + \frac{\sum_{t=2}^{N}\eta_t^2}{N}.$$

Then,

$$E(\gamma_N) = \frac{\sigma_1^2}{N} + \frac{N-1}{N}\sigma_\eta^2 \longrightarrow \sigma_\eta^2 \quad \text{as} \quad N \longrightarrow \infty$$

and therefore the system is controllable.

We now turn to a more complicated case. At each time t we observe not x_t but $y_t = \frac{1}{2}(x_t - u_t)^2$ without any noise. Therefore, after observing y_{t-1} we know that x_{t-1} can have been equal to either $u_{t-1} + \sqrt{2y_{t-1}}$ or $u_{t-1} - \sqrt{2y_{t-1}}$. This is closer to the actual problem we are dealing with. Again, we have to pick u_t optimally and, as we have seen, the optimal value would be x_{t-1}. Based on the current and the previous observations we will be able to express probabilistically our beliefs about which of the two points is more likely to coincide with x_{t-1}. This point will be chosen as u_t while the other will be denoted by u_t'. The same happens at each time point apart from the first one, where we have $u_1 = E(x_1)$. Summarizing, at time t we have two candidates for u_t. The one chosen, u_t, satisfies $\Pr(x_{t-1} = u_t|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}) = a_t$ while the other, u_t', satisfies $\Pr(x_{t-1} = u_t'|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}) = b_t$. Obviously, $b_t = 1 - a_t$ and $b_t \leq \frac{1}{2}$. Now we show how these probabilities can be derived.

Hereafter, $\phi(a|b, c)$ denotes the value that the p.d.f of a Normal distribution with mean b and standard deviation c takes at point a. At time $t-1$, before observing y_{t-1} and in the light of all the currently available data we can say

that the prior pdf of x_{t-1} is a mixture of two Normal densities,

$$p(x_{t-1}|\mathbf{y}_{t-2}, \mathbf{u}_{t-1}) = a_{t-1}\phi(x_{t-1}|u_{t-1}, \sigma_\eta) + b_{t-1}\phi(x_{t-1}|u'_{t-1}, \sigma_\eta).$$

This applies to x_1 too, where $u_1 = u'_1 = E(x_1)$ and $b_1 = \frac{1}{2}$. When y_{t-1} is observed the posterior density of x_{t-1} becomes

$$p(x_{t-1}|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}) \propto p(y_{t-1}|x_{t-1}, u_{t-1})p(x_{t-1}|\mathbf{y}_{t-2}, \mathbf{u}_{t-1}).$$

In fact, the posterior is discrete with its probability mass divided between $m = u_{t-1} + \sqrt{2y_{t-1}}$ and $m' = u_{t-1} - \sqrt{2y_{t-1}}$ since for all other points the likelihood is zero. The posterior is

$$\Pr(x_{t-1} = m|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}) = \frac{a_{t-1}\phi(m|u_{t-1}, \sigma_\eta) + b_{t-1}\phi(m|u'_{t-1}, \sigma_\eta)}{d},$$

where

$$d = a_{t-1}\phi(m|u_{t-1}, \sigma_\eta)+b_{t-1}\phi(m|u'_{t-1}, \sigma_\eta)+a_{t-1}\phi(m'|u_{t-1}, \sigma_\eta)+b_{t-1}\phi(m'|u'_{t-1}, \sigma_\eta).$$

Note that m and m' have the same likelihood and therefore it cancels out in the derivation of the posterior probabilities.

It is easy to see that the highest posterior probability is given to that point which is on the same side of u_{t-1} as is u'_{t-1}. Therefore, this point is u_t while the other is u'_t. At time 1, however, we get $b_2 = \frac{1}{2}$ because $u_1 = u'_1$ and we

choose one of the two candidates at random.

We now proceed to derive the expectation of γ_N.

$$E\left[(x_t - u_t)^2\right] = E\left\{E\left[(x_t - u_t)^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right]\right\}$$

and

$$\begin{aligned} & E\left[(x_t - u_t)^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right] \\ &= a_t E\left[(x_t - u_t)^2 | x_{t-1} = u_t\right] + b_t E\left[(x_t - u_t)^2 | x_{t-1} = u'_t\right] \\ &= a_t \sigma_\eta^2 + b_t E\left[(x_t - u'_t)^2 | x_{t-1} = u'_t\right] + b_t (u'_t - u_t)^2 \\ &= \sigma_\eta^2 + b_t (u'_t - u_t)^2 = \sigma_\eta^2 + 4 b_t (2 y_{t-1}) = \sigma_\eta^2 + 4 b_t (x_{t-1} - u_{t-1})^2. \end{aligned}$$

Therefore,

$$E\left[(x_t - u_t)^2\right] = \sigma_\eta^2 + 4E\left[b_t (x_{t-1} - u_{t-1})^2\right]. \tag{3.9}$$

b_t cannot be taken out of the expectation because its value depends on the deviation $|x_{t-1} - u_{t-1}|$. This formula has been verified in simulations.

We deviate briefly from our discussion in order to show why it is better to choose as u_t the point among m and m' with which x_{t-1} is more likely to coincide. The alternative would be to choose at random between the two points. Without loss of generality assume that $\Pr(x_{t-1} = m | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}) = a_t$ and $\Pr(x_{t-1} = m' | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}) = b_t$. Then we would choose m for u_t with

probability a_t. In this case

$$\begin{aligned} & E\left[(x_t - u_t)^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right] \\ &= a_t E\left[(x_t - m)^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right] + b_t E\left[(x_t - m')^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right]. \end{aligned}$$

Following the same line of thought as above, we have

$$\begin{aligned} & E\left[(x_t - m)^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right] \\ &= a_t \sigma_\eta^2 + b_t E\left[(x_t - m)^2 | x_{t-1} = m'\right] = \sigma_\eta^2 + 4b_t (x_{t-1} - u_{t-1})^2 \end{aligned}$$

and similarly

$$E\left[(x_t - m')^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right] = \sigma_\eta^2 + 4a_t (x_{t-1} - u_{t-1})^2.$$

Then,

$$\begin{aligned} & E\left[(x_t - u_t)^2 | \mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right] \\ &= \sigma_\eta^2 + 8a_t b_t E\left[(x_{t-1} - u_{t-1})^2\right] \geq \sigma_\eta^2 + 4b_t E\left[(x_{t-1} - u_{t-1})^2\right] \end{aligned}$$

since $a_t \geq \frac{1}{2}$.

We now return to where we left off. The expected rate of loss is

$$E(\gamma_N) = \frac{\sum_{t=1}^{N} E\left[(x_t - u_t)^2\right]}{N}$$
$$= \frac{E\left[(x_1 - u_1)^2\right] + E\left[(x_2 - u_2)^2\right] + \sum_{t=3}^{N} E\left[(x_t - u_t)^2\right]}{N}.$$

However, $E\left[(x_1 - u_1)^2\right] = \sigma_1^2$ and $E\left[(x_2 - u_2)^2\right] = \sigma_\eta^2 + 2\sigma_1^2$ because $b_2 = \frac{1}{2}$ always. Then,

$$E(\gamma_N) = \frac{3\sigma_1^2 + (N-1)\sigma_\eta^2}{N} + \frac{4\sum_{t=3}^{N} E\left[b_t(x_{t-1} - u_{t-1})^2\right]}{N}.$$

We now write

$$E\left[b_t(x_{t-1} - u_{t-1})^2\right] = \frac{\beta_t}{4} E\left[(x_{t-1} - u_{t-1})^2\right], \forall t \geq 3. \tag{3.10}$$

As we will see in the simulations section below, after some point M in time which arrives very soon, β_t becomes $\beta < 1$ and β does not depend either on time or on σ_1 or σ_η. Up to M, β_t is a function of $\delta = \frac{\sigma_1}{\sigma_\eta}$ only. Its functional form depends on time t. We have calculated the formulae of β_t for $t = 3, 4, 5$ and we give them further below. The stabilization of β_t to β seems correct intuitively although we have not been able to prove it. We can explain it as follows. The method by its construction will very quickly find a true x_t. Thereafter, all the deviations of the form $|x_{t-1} - u_{t-1}|$ will be affected by σ_η only and will follow the same distribution. The value b_t depends on all past deviations $|x_{t-1} - u_{t-1}|$ but more on the most recent ones. Simulations show that very quickly its distribution becomes independent of t. But β_t is a measure of the distribution of b_t and therefore becomes constant. It is not

influenced by σ_η either because σ_η affects both sides of (3.10) in the same way. At the beginning, but only then, σ_1 also plays a role and then its ratio with σ_η affects β_t.

In the light of all this the expected rate of loss for $N > M$ becomes

$$\begin{aligned} E(\gamma_N) &= \frac{3\sigma_1^2 + (N-1)\sigma_\eta^2 + \sum_{t=3}^{M} \beta_t E\left[(x_{t-1} - u_{t-1})^2\right]}{N} \\ &+ \frac{\beta \sum_{t=M+1}^{N} E\left[(x_{t-1} - u_{t-1})^2\right]}{N} \\ &= \frac{(3-\beta)\sigma_1^2 + (N-1)\sigma_\eta^2 + \sum_{t=3}^{M} (\beta_t - \beta) E\left[(x_{t-1} - u_{t-1})^2\right]}{N} \\ &+ \frac{\beta \sum_{t=1}^{N-1} E\left[(x_t - u_t)^2\right]}{N} \\ &= \frac{(3-\beta)\sigma_1^2 + \sum_{t=3}^{M} (\beta_t - \beta) E\left[(x_{t-1} - u_{t-1})^2\right]}{N} \\ &+ \frac{(N-1)\sigma_\eta^2}{N} + \frac{\beta(N-1)E(\gamma_{N-1})}{N}. \end{aligned} \tag{3.11}$$

Let $c = (3-\beta)\sigma_1^2 + \sum_{t=3}^{M} (\beta_t - \beta) E\left[(x_{t-1} - u_{t-1})^2\right], x_n = E(\gamma_n)$ and $\alpha = \sigma_\eta^2$. Then (3.11) becomes

$$x_n = \frac{c}{n} + \frac{n-1}{n}\alpha + \beta \frac{n-1}{n} x_{n-1}, \forall n > M.$$

To investigate convergence formally, define a new sequence $\{y_n\}$ as $y_n = x_{n+M}$. Then,

$$y_n = \frac{c}{n+M} + \frac{n+M-1}{n+M}\alpha + \beta \frac{n+M-1}{n+M} y_{n-1}. \tag{3.12}$$

The first term of the sequence, y_1 is finite. A non recursive formula for y_n can be found by applying (3.12) recursively. It is not difficult to see that

$$y_n = \frac{c\left(\sum_{i=0}^{n-2}\beta^i\right)}{M+n} + \frac{\alpha\sum_{i=0}^{n-2}\beta^i(M+n-i-1)}{M+n} + \beta^{n-1}\frac{M+1}{M+n}y_1.$$

We now examine the limit of each term as $n \longrightarrow \infty$ separately.

$$\beta^{n-1}\frac{M+1}{M+n}y_1 \longrightarrow 0 \quad \text{since} \quad \beta < 1.$$

$$\frac{c\left(\sum_{i=0}^{n-2}\beta^i\right)}{M+n} = \frac{c}{M+n}\frac{1-\beta^{n-1}}{1-\beta} \longrightarrow 0.$$

The limit of $\sum_{i=0}^{n-2}\beta^i(M+n-i-1)$ is harder to find but we observe that

$$\sum_{i=0}^{n-2}\beta^i\frac{(M+n-i-1)}{M+n} < \sum_{i=0}^{n-2}\beta^i \quad \text{since} \quad \frac{(M+n-i-1)}{M+n} < 1.$$

Moreover,

$$\begin{aligned}
&\sum_{i=0}^{n-2}\beta^i - \sum_{i=0}^{n-2}\beta^i\frac{(M+n-i-1)}{M+n} \\
&= \sum_{i=0}^{n-2}\beta^i\frac{1+i}{M+n} = \frac{1}{M+n}\left(\frac{\beta-\beta^{n-1}}{(1-\beta)^2} - \frac{1-(n-1)\beta^{n-1}}{1-\beta}\right) \\
&= \frac{1}{M+n}\frac{\beta-\beta^{n-1}}{(1-\beta)^2} - \frac{1-(n-1)\beta^{n-1}}{(M+n)(1-\beta)} \longrightarrow 0.
\end{aligned}$$

In other words, $\sum_{i=0}^{n-2} \beta^i \frac{(M+n-i-1)}{M+n}$ is positive, bounded from above by and gets closer and closer to a sequence whose limit is $\frac{1}{1-\beta}$. Therefore,

$$\frac{a\sum_{i=0}^{n-2} \beta^i (M+n-i-1)}{M+n} \longrightarrow \frac{\alpha}{1-\beta}.$$

So $\lim y_n = \frac{\alpha}{1-\beta}$. Therefore,

$$\lim E(\gamma_N) = \frac{\sigma_\eta^2}{1-\beta}. \tag{3.13}$$

We see that the limit is directly proportional to the system variance, as we also observed in the more complicated case we have been studying.

Extension. Suppose now that the system equation has the more general form $x_t = \kappa x_{t-1} + \eta_t$, where κ is a known non-zero coefficient. Then the above ideas apply here as well. In other words when we observe y_{t-1} we get two candidates for the location of x_{t-1} with posterior probabilities a_t and b_t. The one with probability a_t is multiplied by κ and becomes u_t, while the other, multiplied by κ, becomes u_t'. It is then easy to see that

$$E\left[(x_t - u_t)^2\right] = \sigma_\eta^2 + 4\kappa^2 E\left[b_t (x_{t-1} - u_{t-1})^2\right].$$

If we try to prove convergence of the expected rate of loss we will arrive at the fact that as N goes to infinity

$$\lim E(\gamma_N) = \frac{\sigma_\eta^2}{1-\kappa^2\beta} \tag{3.14}$$

provided that $\kappa^2\beta < 1$. As we will see in the simulations below the system remains controllable and $\kappa^2\beta < 1$. We have not been able to prove this either but intuitively one could expect it. As κ becomes larger the separation of u_t from u'_t also becomes larger. This makes picking out the correct control point easier and should lead to a decrease in β that makes up for the increase in κ.

A simulation study. In order to check the theoretical results, we conducted the following simulation. At each time point t, we estimated $E(\gamma_t)$, $E\left[(x_t - u_t)^2\right]$, $E\left[b_t(x_{t-1} - u_{t-1})^2\right]$ and β_t by taking 10000 simulations of the system for 1000 time points and letting

$$\begin{aligned}
\hat{E}\left[(x_t - u_t)^2\right] &= \frac{\sum_{i=1}^{10000}\left[(x_{ti} - u_{ti})^2\right]}{10000}, \\
\hat{E}\left[b_t(x_{t-1} - u_{t-1})^2\right] &= \frac{\sum_{i=1}^{10000}\left[b_{ti}(x_{t-1,i} - u_{t-1,i})^2\right]}{10000}, \\
\hat{E}(\gamma_t) &= \frac{\sum_{j=1}^{t}\hat{E}\left[(x_j - u_j)^2\right]}{t}, \\
\hat{\beta}_t &= \frac{4\hat{E}\left[b_t(x_{t-1} - u_{t-1})^2\right]}{\hat{E}\left[(x_{t-1} - u_{t-1})^2\right]}
\end{aligned}$$

where x_{ti} is the true state, u_{ti} is the control and b_{ti} is b_t, all for chain i at time t. This was done for four different combinations of σ_1 and σ_η, namely

- $\sigma_1 = 1, \sigma_\eta = 1$
- $\sigma_1 = 1, \sigma_\eta = 2$
- $\sigma_1 = 2, \sigma_\eta = 1$
- $\sigma_1 = 2, \sigma_\eta = 2$.

In Figure 3.9 we present the progress of $\hat{E}(\gamma_N)$, $\hat{E}\left[(x_t - u_t)^2\right]$ and $\hat{\beta}_t$. The first column of graphs shows that the convergence of the expected rate of loss is extremely fast. We also see from the second column that $E\left[(x_t - u_t)^2\right]$ also reaches a constant level, and finally that the value of β_t given by the graphs in the third column quickly reaches ($M \simeq 5$) a constant level which is the same irrespective of δ. β is somewhere close to $\frac{4}{7}$. The limit of the expected rate of loss is in all four cases close to the theoretical one given by (3.13). Moreover, it is in agreement with the respective limits observed in the simulations of the previous section.

In Figure 3.10 we examine the evolution of the distribution of b_t in time. We plot the histograms of the 10000 simulated values of $b_3, b_{10}, b_{50}, b_{100}$ and b_{1000} for all combinations of σ_1 and σ_η. As expected we see that the distributions of b_{10}, b_{50}, b_{100} and b_{1000} are very similar everywhere while differing from that of b_3, which moreover changes according to δ. We can therefore say that the simulation results verify our theoretical derivations and assumptions.

We also conducted the same experiment for several combinations of the variances and for several values of κ other than 1. A problem that affects the simulation of the system with the aid of a computer is that if $\kappa > 1$ the absolute value of x_t grows exponentially with time. More specifically, $|x_t|$ is of order $\kappa^t|x_1|$. This causes memory overflows and rounding errors that lead the computer to report zero deviations $|x_t - u_t|$ for large t. For this reason we simulated the system for only 100 time points and for values of κ close to 1. The pattern observed was the same for all combinations of σ_1 and σ_η and therefore we only present results for $\sigma_1 = \sigma_\eta = 1$. The values of κ considered were 0.5, 1.1 and 1.3. Figure 3.11 shows the progress of $\hat{E}(\gamma_N)$, $\hat{E}\left[(x_t - u_t)^2\right]$ and $\kappa^2\hat{\beta}_t$. The first column demonstrates that the system is controllable and

that the rate of loss reaches its limit quickly. The second column shows that $E[(x_t - u_t)^2]$ also stabilizes very fast. Finally we see in the third column that $\kappa^2\beta_t$ quickly attains a level $\kappa^2\beta$ below 1. Moreover, in each case the limit of the rate of loss is that postulated by formula (3.14). Note however that as κ increases so does the limit of the expected rate of loss and $\kappa^2\beta$ approaches 1. This means that, although the system remains controllable as κ increases, the limit of the expected rate of loss approaches infinity.

The formulae of β_3, β_4 and β_5. Here we give the formulae of β_3, β_4 and β_5 and we briefly sketch how they were calculated.

For each t

$$\beta_t = \frac{4E\left[b_t(x_{t-1} - u_{t-1})^2\right]}{E\left[(x_{t-1} - u_{t-1})^2\right]}.$$

If $E\left[(x_{t-1} - u_{t-1})^2\right]$ is known, calculating $E\left[b_t(x_{t-1} - u_{t-1})^2\right]$ will lead to β_t. The random quantities involved in $b_t(x_{t-1} - u_{t-1})^2$ are all the deviations $|x_l - u_l|$ for $l = 1, 2, \ldots, t-1$. The distribution of each of them, in decreasing order of l, given all its predecessors is not difficult to find. Multiplying by each distribution and integrating in turn will lead eventually to the desired expectation. In appendix C we give the derivation of β_3 in more detail. The formulae for β_3, β_4 and β_5 are

$$\beta_3 = \frac{\pi + 2(4\delta^2 + 1)\tan^{-1}\left(\frac{1}{2\delta}\right) - 4\delta}{\pi(2\delta^2 + 1)},$$

$$\beta_4 = \frac{m}{n},$$

where,

$$\begin{aligned} m &= 4\left[\frac{3\pi-4}{8} + \frac{5\tan^{-1}\left(\frac{1}{2}\right)}{4} + (4\delta^2+1)\tan^{-1}\left(\frac{1}{2\delta}\right)\right. \\ &- \frac{2\delta}{4\delta^2+1} - \frac{1}{2\sqrt{4\delta^2+1}} + \frac{25\delta}{(16\delta^2+5)\sqrt{5}} \\ &- \frac{10\delta^2}{(16\delta^2+5)\sqrt{4\delta^2+1}} + \frac{5}{4}\tan^{-1}\left(\frac{1}{2}\right)\sqrt{\frac{5\tan^{-1}\left(\frac{1}{2}\right)}{5\tan^{-1}\left(\frac{1}{2}\right)+8\delta^2}} \\ &- \frac{16\delta^2+5}{4}\tan^{-1}\left(\frac{\sqrt{5}}{4\delta}\right) + 4\delta^2\tan^{-1}\left(\frac{1}{2}\right)\left(\frac{5\tan^{-1}\left(\frac{1}{2}\right)}{5\tan^{-1}\left(\frac{1}{2}\right)+8\delta^2}\right)^{3/2} \\ &+ \left.\frac{80\delta^3}{(16\delta^2+5)\sqrt{5}} - \frac{8\delta^2}{4\delta^2+1}\right] \end{aligned}$$

and

$$n = 2\pi + 2(4\delta^2+1)\tan^{-1}\left(\frac{1}{2}\right) - 4\delta,$$

while

$$\beta_5 = \frac{q}{p},$$

where,

$$
\begin{aligned}
q &= 4\left[\frac{7\pi-20}{16}+\frac{1014}{504\sqrt{5}}+\frac{25}{8}\tan^{-1}(\tfrac{1}{2})-\frac{21}{8}\tan^{-1}(\tfrac{\sqrt{5}}{4})\right.\\
&+ \frac{5}{8}\tan^{-1}(\tfrac{1}{2})\sqrt{\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}}+2\tan^{-1}(\tfrac{1}{2})\left(\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}\right)^{3/2}\\
&- \frac{1}{2}+\frac{1}{2\pi}\tan^{-1}(\frac{1}{2\delta})+\frac{5}{4}\tan^{-1}(\frac{1}{2})-\frac{5}{4\pi}\tan^{-1}(\frac{1}{2})\tan^{-1}(\frac{1}{2\delta})\\
&- \frac{21\pi}{8}+\frac{21}{8}\tan^{-1}\left(\frac{\sqrt{21}}{8\delta}\right)+\frac{5\pi}{2}-\frac{5}{2}\tan^{-1}(\frac{\sqrt{5}}{4\delta})\\
&+ \frac{105}{42\sqrt{5(4\delta^2+1)}}-\frac{1}{\sqrt{4\delta^2+1}}-\frac{41}{42\sqrt{16\delta^2+5}}+\frac{100\delta}{(16\delta^2+5)\sqrt{5}}\\
&+ \frac{5}{2}\tan^{-1}(\frac{1}{2})\sqrt{\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8\delta^2}}\\
&- \frac{21}{8}\tan^{-1}(\frac{\sqrt{5}}{4})\sqrt{\frac{21\tan^{-1}(\frac{\sqrt{5}}{4})}{21\tan^{-1}(\frac{\sqrt{5}}{4})+16\delta^2\sqrt{5}}}\\
&- \frac{34994\delta}{21(64\delta^2+21)\sqrt{84}}+\frac{41}{42\sqrt{16\delta^2+5}}\\
&- \frac{41}{42\pi\sqrt{16\delta^2+5}}\tan^{-1}\left(\frac{\sqrt{16\delta^2+5}}{2\delta}\right)\\
&- \frac{100\delta}{(16\delta^2+5)\sqrt{20}}-\frac{20\delta^2}{(16\delta^2+5)\sqrt{4\delta^2+1}}+\frac{441\delta}{(64\delta^2+21)\sqrt{21}}\\
&+ \frac{42\delta^2\sqrt{10}}{(64\delta^2+21)\sqrt{2(4\delta^2+1)}}-\frac{525\delta}{21(64\delta^2+21)\sqrt{21}}\\
&- \frac{100\delta^2}{21(64\delta^2+21)\sqrt{16\delta^2+5}}\\
&+ \frac{5}{4\pi}\tan^{-1}(\tfrac{1}{2})\sqrt{\frac{5\tan^{-1}(\frac{1}{2})}{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}}\tan^{-1}\left(\frac{\sqrt{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}}{2\delta\sqrt{5\tan^{-1}(\frac{1}{2})}}\right)\\
&+ \frac{4}{\pi}\tan^{-1}(\tfrac{1}{2})\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}\sqrt{\frac{5\tan^{-1}(\frac{1}{2})}{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}}\\
&\quad \left.\tan^{-1}\left(\frac{\sqrt{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}}{2\delta\sqrt{5\tan^{-1}(\frac{1}{2})}}\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&- \frac{16\delta}{2\pi(4\delta^2+1)}\tan^{-1}(\frac{1}{2})\left(\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}\right)^2 \\
&+ 16\delta^2\tan^{-1}(\frac{1}{2})\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}\left(\frac{5\tan^{-1}(\frac{1}{2})}{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}\right)^{3/2} \\
&- 16\delta^2\tan^{-1}(\frac{1}{2})\left(\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}\right)^{5/2}\left(\frac{5\tan^{-1}(\frac{1}{2})}{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}\right)^{3/2} \\
&+ \frac{16\delta^2}{\pi}\tan^{-1}(\frac{1}{2})\left(\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}\right)^{5/2}\left(\frac{5\tan^{-1}(\frac{1}{2})}{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}\right)^{3/2} \\
&\quad \tan^{-1}\left(\frac{\sqrt{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}}{2\delta\sqrt{5\tan^{-1}(\frac{1}{2})}}\right) \\
&- \frac{32\delta^3}{\pi(4\delta^2+1)}\tan^{-1}(\frac{1}{2})\frac{5\tan^{-1}(\frac{1}{2})}{32\delta^2+5\tan^{-1}(\frac{1}{2})+8}\left(\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8}\right)^3 \\
&- \frac{250\delta\left(\tan^{-1}(\frac{\sqrt{21}}{2})\right)^2}{\pi(105\tan^{-1}(\frac{\sqrt{21}}{2})+320\tan^{-1}(\frac{\sqrt{21}}{2})\delta^2+8\delta^2\sqrt{21})\sqrt{21}} \\
&- 8\delta^2\tan^{-1}(\frac{\sqrt{5}}{4\delta})\,] + \frac{160\delta^3}{(16\delta^2+5)\sqrt{5}} + 8\delta^2\tan^{-1}(\frac{\sqrt{21}}{8\delta}) - \frac{64\delta^3\sqrt{21}}{64\delta^2+21} \\
&+ 40\delta^2\frac{\left(\tan^{-1}(\frac{1}{2})\right)^2}{5\tan^{-1}(\frac{1}{2})+8\delta^2}\sqrt{\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8\delta^2}} \\
&- 8\delta^2\tan^{-1}(\frac{\sqrt{5}}{4})\left(\frac{21\tan^{-1}(\frac{\sqrt{5}}{4})}{21\tan^{-1}(\frac{\sqrt{5}}{4})+16\delta^2\sqrt{5}}\right)^{3/2}\Bigg]
\end{aligned}
$$

and

$$
\begin{aligned}
p &= \pi + \frac{3\pi-4}{2} + 5\tan^{-1}(\frac{1}{2}) + 4(4\delta^2+1)\tan^{-1}(\frac{1}{2\delta}) - \frac{8\delta}{4\delta^2+1} - \frac{2}{\sqrt{4\delta^2+1}} \\
&+ \frac{100\delta}{(16\delta^2+5)\sqrt{5}} - \frac{40\delta^2}{(16\delta^2+5)\sqrt{4\delta^2+1}} + 5\tan^{-1}(\frac{1}{2})\sqrt{\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8\delta^2}} \\
&- (16\delta^2+5)\tan^{-1}(\frac{\sqrt{5}}{4\delta}) + 16\delta^2\tan^{-1}(\frac{1}{2})\left(\frac{5\tan^{-1}(\frac{1}{2})}{5\tan^{-1}(\frac{1}{2})+8\delta^2}\right)^{3/2}
\end{aligned}
$$

$$+ \frac{320\delta^3}{(16\delta^2+5)\sqrt{5}} - \frac{32\delta^3}{4\delta^2+1}.$$

We can see that β_3, β_4 and β_5 depend only on δ and not on the variances and this agrees with the simulations. Unfortunately the formulae are too complicated to allow us to derive any convergence results about β_t.

3.8 Discussion

In this chapter we studied at some depth and proposed solutions for a control problem that finds applications in industry. The system involved can be described by a dynamic model. All the suggested solutions require the availability of the relevant prior and posterior distributions, but non-linearities in the system make it impossible to have them in closed form. Resampling techniques, however, and more specifically smooth bootstrap allow us to obtain random samples from all the distributions of interest in very short time.

The application of smooth bootstrap revealed a problem that we believe is inherent in all sampling techniques. In essence the problem is due to the fact that a sample from any continuous distribution, in whatever way it is obtained, is bound to under-represent the tails of the distribution. However, the problem becomes more serious in resampling because samples form the foundation for the acquisition of future samples. In the present application we overcame the problem by discarding the results of resampling when it was based on such a bad sample and replacing them by a sample formed in a way that guaranteed it would be a good basis for future resamplings. In other applications, however, it may not be easy to detect a bad sample.

The distributions that arise in the control problem can be either unimodal or bimodal. We developed an automatic heuristic that finds their global mode when all we have is a sample from them. We do not claim it to be an all-purpose tool, however, since it is designed with a particular type of distribution in mind.

As far as the control problem is concerned all the proposed methods manage to keep the system under control. After an initial transient period they maintain the expected rate of loss at a constant finite level. This level is not affected by our degree of knowledge about the variances of the random noises involved. The best method is our adaptation of Titterington's method with $K = 0$. The fact that in a continuous time setting (Titterington (1973)) the optimal solution required $K > 0$ shows that in some problems discrete and continuous time settings do not have similar solutions. A similar situation is reported in Drenick and Shaw (1964). A theoretical analysis of a simpler version of the problem proved that the system is controllable, verified the limits of the expected rates of loss and explained some patterns observed in the simulations.

As a conclusion we can say that we have found a method that is easy to implement and which deals efficiently with the control problem at hand. We are now going to apply it in more complicated variants of it where it could encounter trouble.

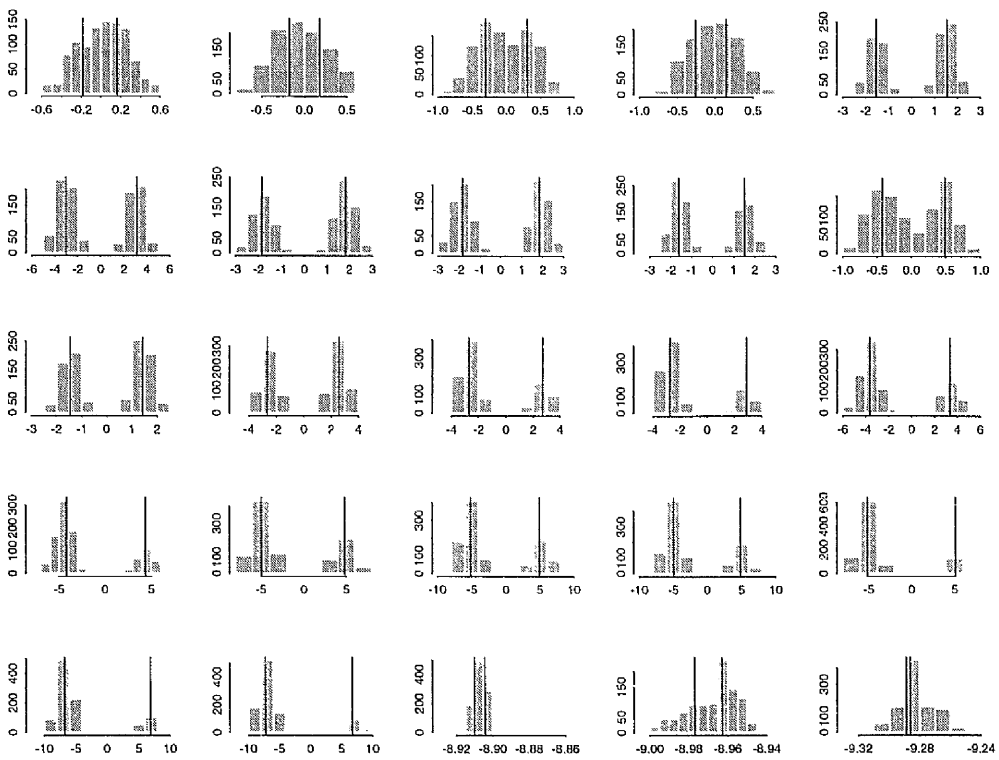


Figure 3.1: Simulation results from section 3.6. Histograms of samples from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$, $t = 1, \ldots, 25$, $\sigma_1 = \sigma_\eta = 1, \sigma_\epsilon = 0.1$. Vertical lines denote pseudo-modes found with the adaptation of Titterington's method.

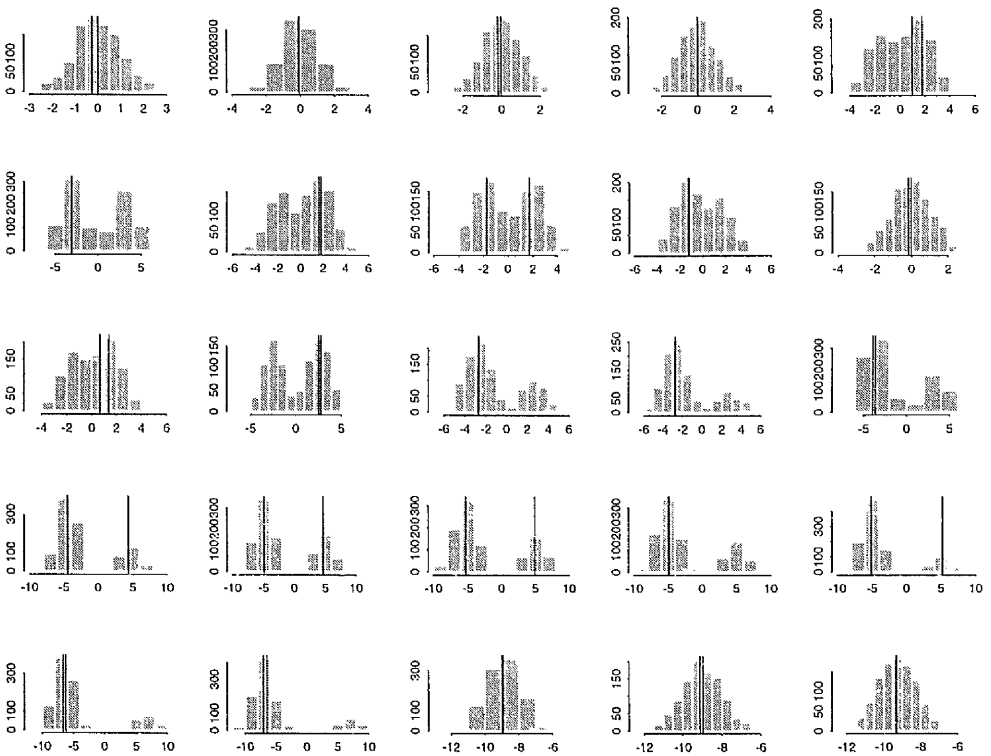
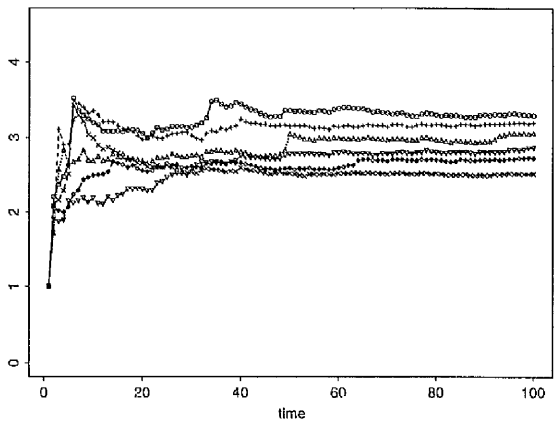
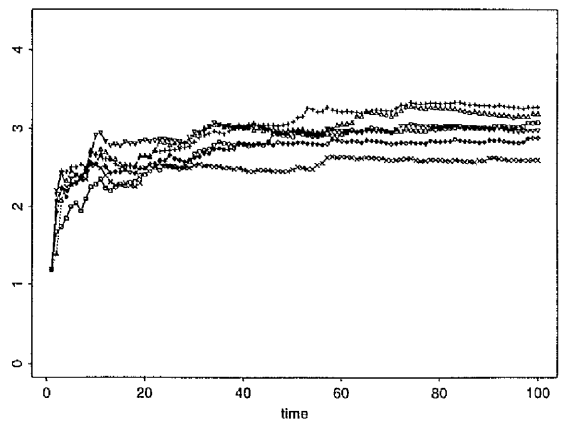


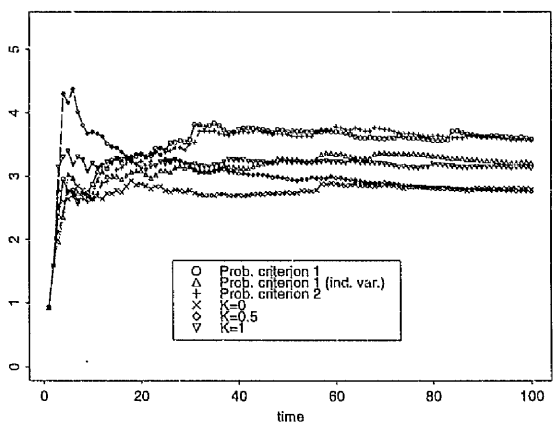
Figure 3.2: Simulation results from section 3.6. Histograms of samples from $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$, $t = 1, \ldots, 25$, $\sigma_1 = \sigma_\eta = 1, \sigma_\epsilon = 0.1$. Vertical lines denote the maximizers of (3.4) and (3.5).



(a)

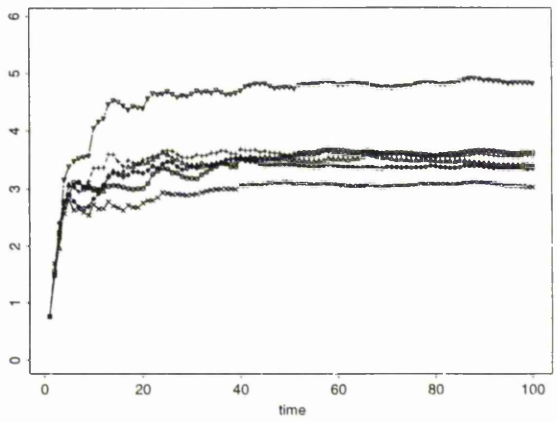


(b)

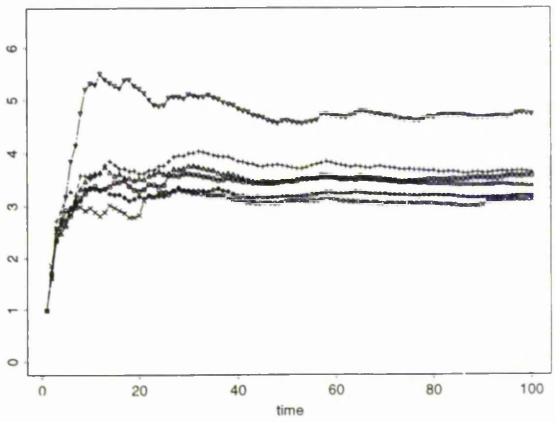


(c)

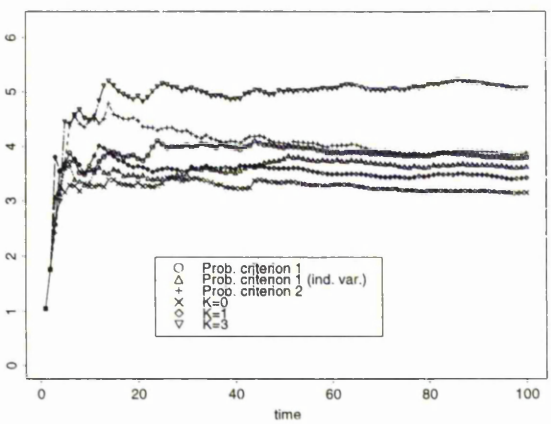
Figure 3.3: Simulation results from section 3.6. Estimated expected rates of loss. Top left image: known variances $\sigma_1^2 = 1, \sigma_\eta^2 = 1, \sigma_\epsilon^2 = 0.01$. Top right image: Only σ_1^2 and $\frac{\sigma_\eta^2}{\sigma_\epsilon^2}$ known. Bottom image: entirely unknown variances.



(a)

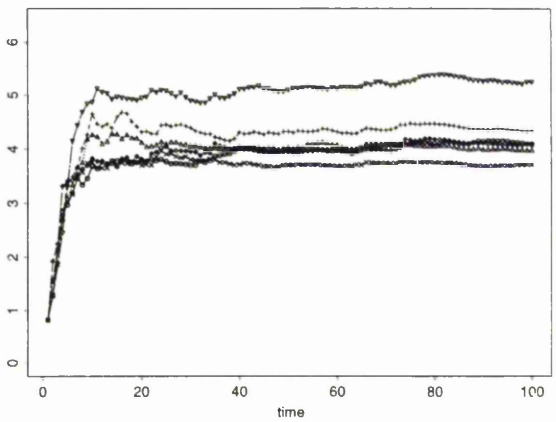


(b)

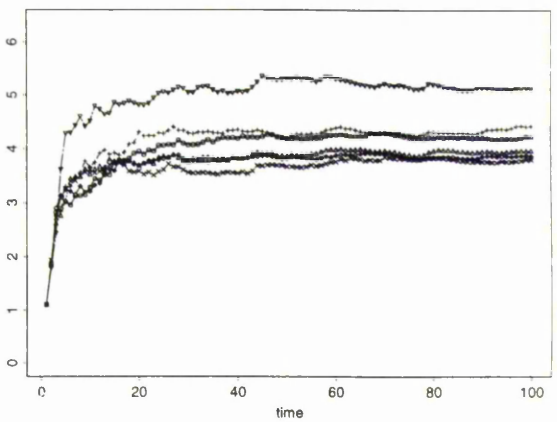


(c)

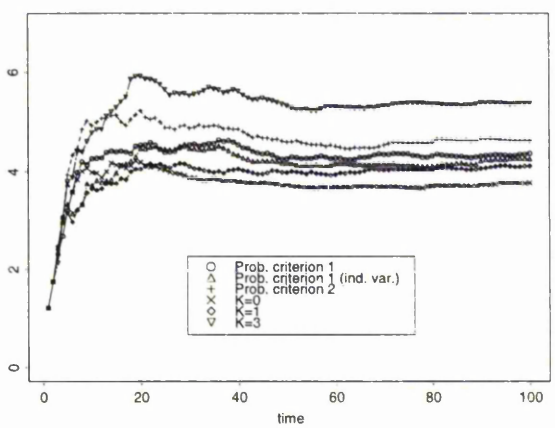
Figure 3.4: Simulation results from section 3.6. Estimated expected rates of loss. Top left image: known variances $\sigma_1^2 = 1, \sigma_\eta^2 = 1, \sigma_\epsilon^2 = 1$. Top right image: Only σ_1^2 and $\frac{\sigma_\eta^2}{\sigma_\epsilon^2}$ known. Bottom image: entirely unknown variances.



(a)

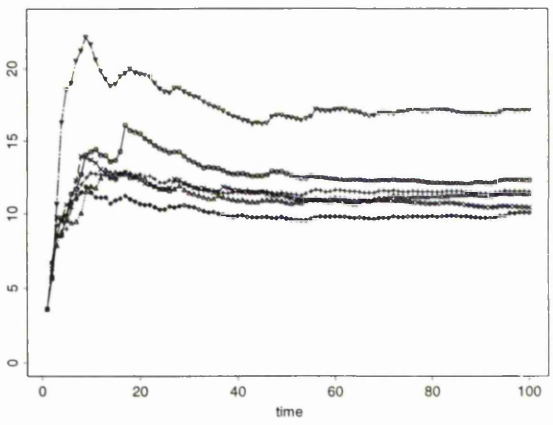


(b)

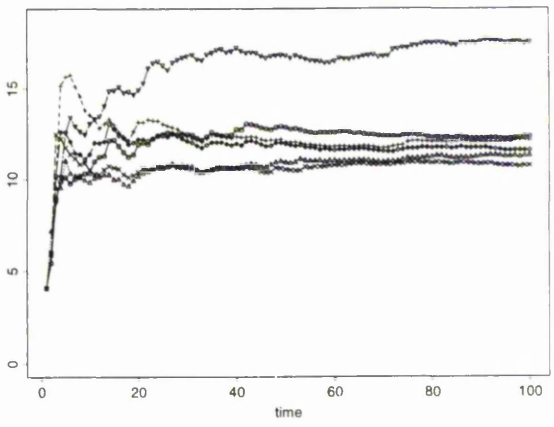


(c)

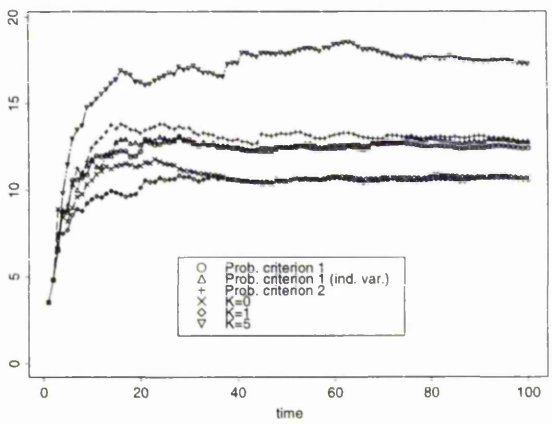
Figure 3.5: Simulation results from section 3.6. Estimated expected rates of loss. Top left image: known variances $\sigma_1^2 = 1, \sigma_\eta^2 = 1, \sigma_\epsilon^2 = 4$. Top right image: Only σ_1^2 and $\frac{\sigma_\eta^2}{\sigma_\epsilon^2}$ known. Bottom image: entirely unknown variances.



(a)

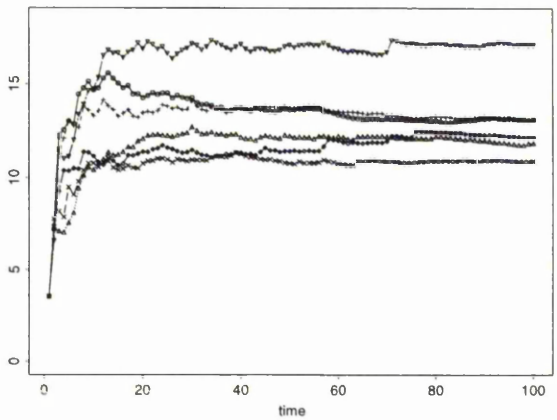


(b)

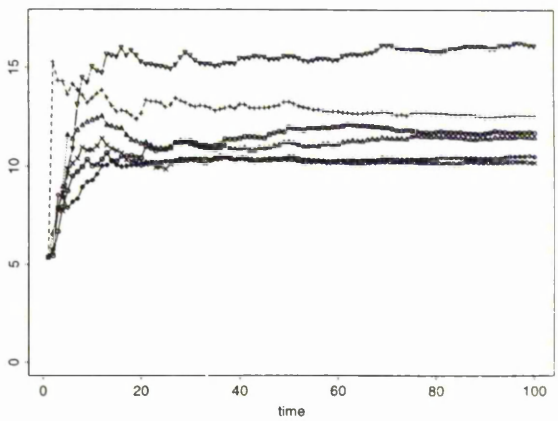


(c)

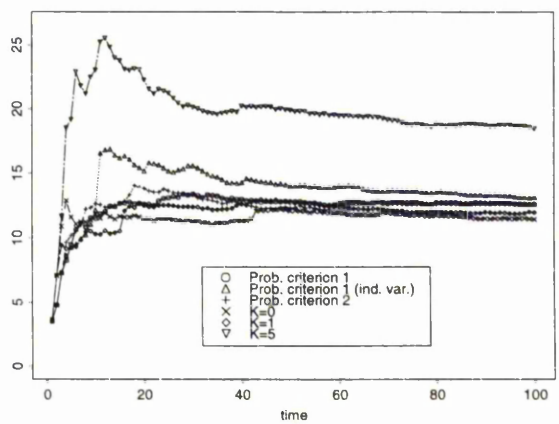
Figure 3.6: Simulation results from section 3.6. Estimated expected rates of loss. Top left image: known variances $\sigma_1^2 = 4, \sigma_\eta^2 = 4, \sigma_\epsilon^2 = 1$. Top right image: Only σ_1^2 and $\frac{\sigma_\eta^2}{\sigma_\epsilon^2}$ known. Bottom image: entirely unknown variances.



(a)



(b)



(c)

Figure 3.7: Simulation results from section 3.6. Estimated expected rates of loss. Top left image: known variances $\sigma_1^2 = 4, \sigma_\eta^2 = 4, \sigma_\epsilon^2 = 4$. Top right image: Only σ_1^2 and $\frac{\sigma_\eta^2}{\sigma_\epsilon^2}$ known. Bottom image: entirely unknown variances.

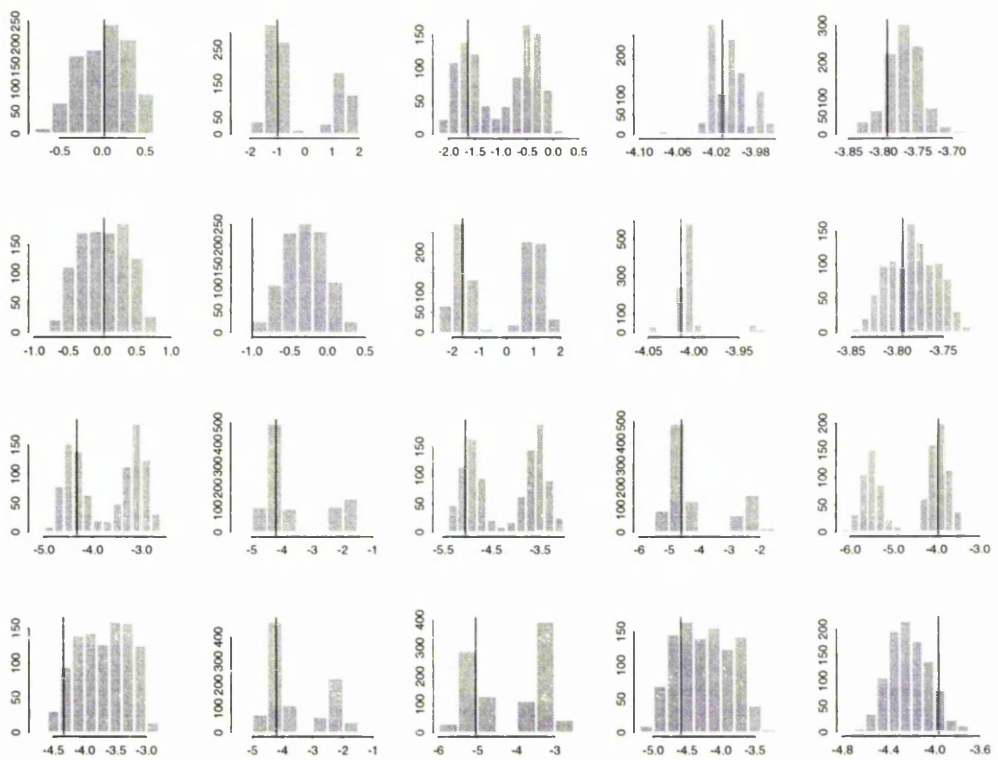


Figure 3.8: Simulation results from section 3.6. Histograms of samples from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$, $t = 1, \ldots, 10$, $\sigma_1 = \sigma_\eta = 1, \sigma_\epsilon = 0.1$. First and third rows: Samples created with "posterior" policy. Second and fourth rows: Samples created with "prior" policy. The vertical lines denote the corresponding true value of x_t.

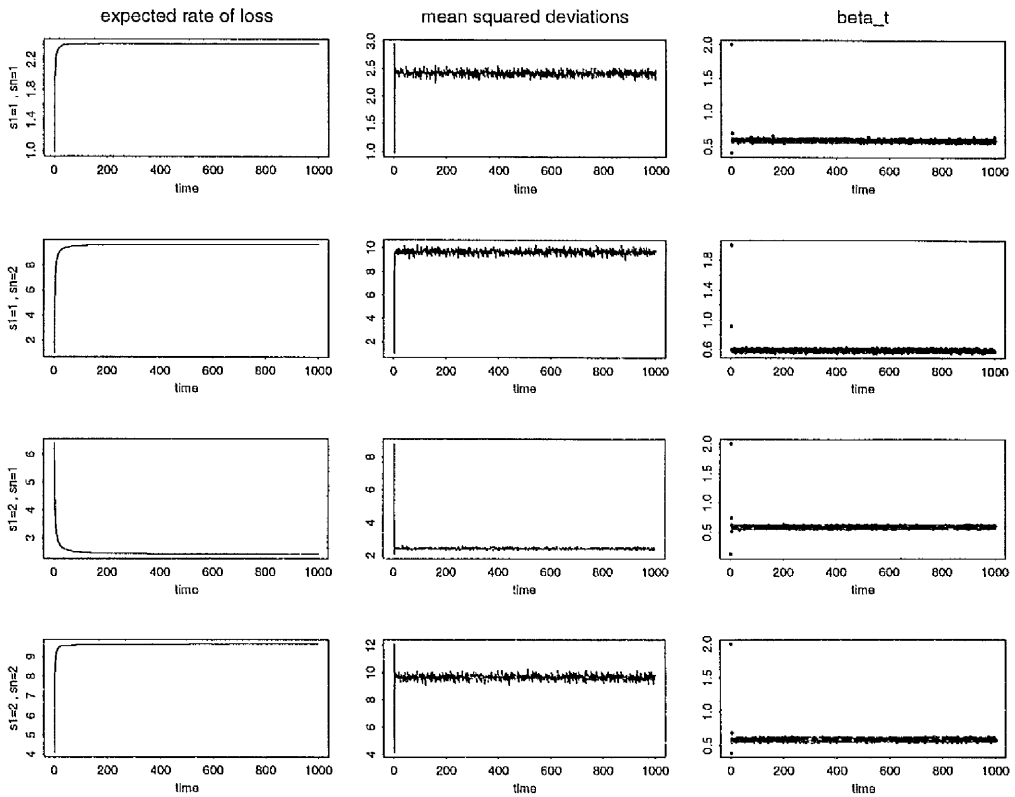


Figure 3.9: Simulation results from the first experiment in section 3.7. First column: graphs of the estimated expected rate of loss, $\hat{E}(\gamma_N)$. Second column: graphs of $\hat{E}[(x_t - u_t)^2]$. Third column: graphs of $\hat{\beta}_t$. All graphs for $t = 1, \ldots, 1000$ and for four combinations of σ_1 and σ_η.

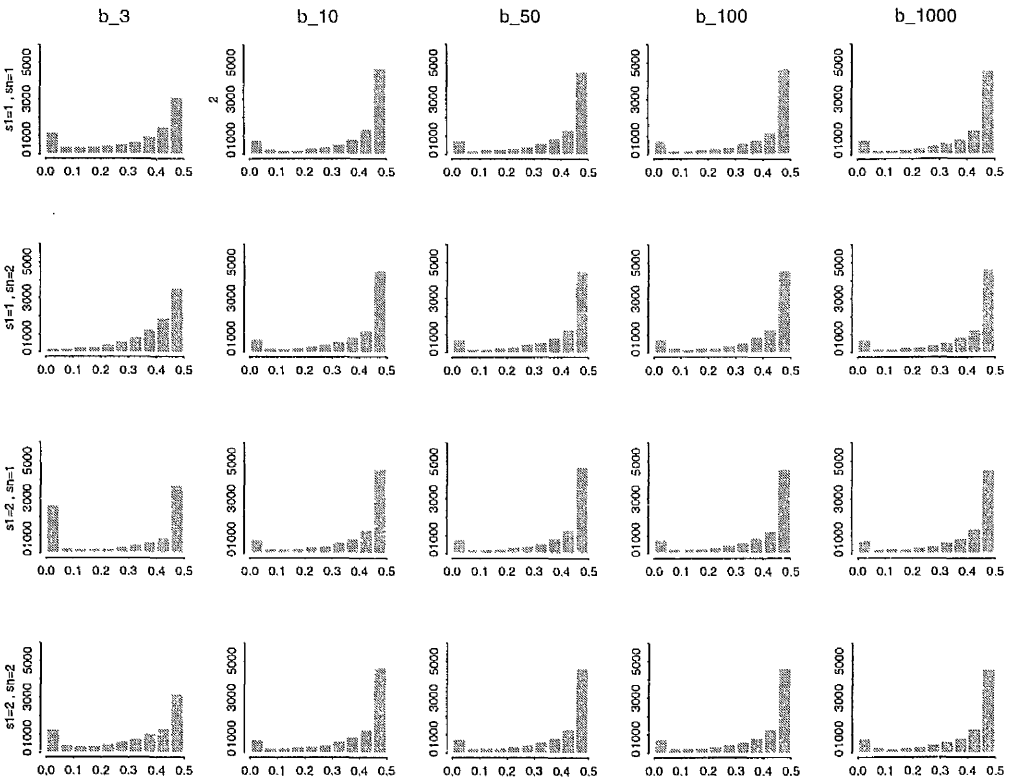


Figure 3.10: Simulation results from the first experiment in section 3.7. Histograms of the simulated distributions of $b_3, b_{10}, b_{50}, b_{100}$ and b_{1000} for four combinations of σ_1 and σ_η.

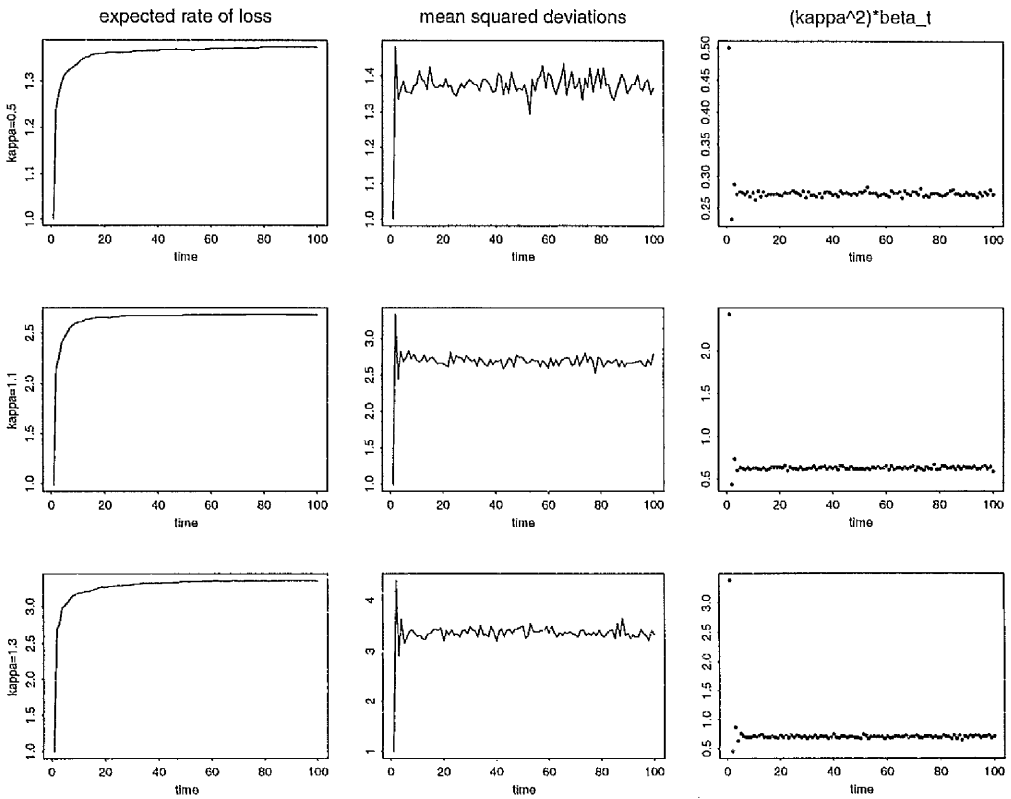


Figure 3.11: Simulation results from the second experiment in section 3.7. First column: graphs of the estimated expected rate of loss, $\hat{E}(\gamma_N)$. Second column: graphs of $\hat{E}[(x_t - u_t)^2]$. Third column: graphs of $\kappa^2\hat{\beta}_t$. All graphs for $t = 1, \ldots, 100$. All for $\sigma_1 = \sigma_\eta = 1$.

Chapter 4

Extensions of the control problem

4.1 Introduction

In the present chapter we study two new variants of the extremum adaptation problem. The first incorporates unknown coefficients in the system evolution and the observation equations. The second refers to the tracking of an extreme point of a multivariate function. We propose a solution for each case and we examine whether this solution can be implemented with the aid of resampling methods. We find that in principle this is feasible in both cases but in practice when the equations contain unknown coefficients resampling can cause problems.

The chapter is organised as follows. The two following sections deal with the first variant of the extremum adaptation problem. In the first of them

we present the problem, propose solutions and show how they can be implemented with the help of resampling. In the second we demonstrate the implementational problems faced by resampling. The next section deals with extremum adaptation in multidimensional spaces. A simpler version of the problem is analysed theoretically and the conclusion is that the problem is solvable. Then we analyse the problem in its full form with the help of resampling. In the final section we summarize the findings of the chapter and present some conclusions.

4.2 Unknown coefficients in the system

In this section the problem analysed in the last chapter is treated in its most general form. More specifically, the movement of x_t is now described by the model

$$x_t = \beta x_{t-1} + \eta_t \quad , \quad \eta_t \sim N(0, \phi_\eta^{-1}) \tag{4.1}$$

while the observation equation now takes the form

$$y_t = \alpha(x_t - u_t)^2 + \epsilon_t \quad , \quad \epsilon_t \sim N(0, \phi_\epsilon^{-1}). \tag{4.2}$$

The prior distribution of x_1 is Normal once more,

$$x_1 \sim N(0, \phi_1^{-1}).$$

The random variables $\{\epsilon_t\}$ and $\{\eta_t\}$ are mutually independent and independent across time. They are also independent from $\{x_t\}$. Apart from the value of x_t, the values of $\alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon$ are also unknown. Our aim still is to have, at each time t, u_t as close to x_t as possible.

We tackle this problem in exactly the same way as in the last chapter. We maintain our Bayesian approach and we assign to $\alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon$ a prior distribution. At time $t-1$, after observing y_{t-1}, our knowledge about all the unknowns is mathematically described by the posterior distribution $p(x_{t-1}, \alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$. At time t interest shifts from x_{t-1} to x_t and we derive the prior distribution

$$\begin{aligned} & p(x_t, \alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}) \\ &= \int p(x_t|x_{t-1}, \beta, \phi_\eta) p(x_{t-1}, \alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}) dx_{t-1}. \end{aligned}$$

Finally we observe y_t at our chosen u_t and arrive at the posterior distribution

$$p(x_t, \alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon|\mathbf{y}_t, \mathbf{u}_t) \propto p(y_t|x_t, \alpha, \phi_\epsilon, u_t) p(x_t, \alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}).$$

Although ϕ_1 is among the unknowns, in practice we are not interested in it for $t \geq 2$ and we integrate it out of the prior and posterior distributions. Our choice for u_t is the same as in the last chapter. In other words, u_t is the mode of $p(x_{t-1}|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ which is a marginal distribution of the posterior of all the unknowns at time $t-1$.

The distributions involved are intractable and we replace them with samples

obtained from them. Because of the dynamic nature of the problem the most promising way of obtaining the samples is through resampling. Its implementation involves trivial calculations. When $(x_t^*(1), \alpha_1^*, \beta_1^*, \phi_\eta^*(1), \phi_\epsilon^*(1)), \ldots, (x_t^*(n), \alpha_n^*, \beta_n^*, \phi_\eta^*(n), \phi_\epsilon^*(n))$ is a sample from $p(x_t, \alpha, \beta, \phi_\eta, \phi_\epsilon | \mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ the resampling weight of each point $(x_t^*(i), \alpha_i^*, \beta_i^*, \phi_\eta^*(i), \phi_\epsilon^*(i))$ is

$$w_i = \phi_\epsilon^*(i)^{1/2} \exp\left[-\frac{\phi_\epsilon^*(i)}{2}\left(y_t - \alpha_i^*(x_t^*(i) - u_t)^2\right)^2\right].$$

Resampling gives a sample $(x_t(1), \alpha_1, \beta_1, \phi_\eta(1), \phi_\epsilon(1)), \ldots, (x_t(n), \alpha_n, \beta_n, \phi_\eta(n), \phi_\epsilon(n))$ from $p(x_t, \alpha, \beta, \phi_\eta, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t)$ which then gives a sample from the next prior $p(x_{t+1}, \alpha, \beta, \phi_\eta, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t)$ by replacing each point $(x_t(i), \alpha_i, \beta_i, \phi_\eta(i), \phi_\epsilon(i))$ with $(x_{t+1}^*(i), \alpha_i^*, \beta_i^*, \phi_\eta^*(i), \phi_\epsilon^*(i))$ where

$$\begin{aligned}
&x_{t+1}^*(i) \quad \text{is a draw from} \quad N(\beta_i x_t(i), \phi_\eta(i)^{-1}),\\
&\alpha_i^* = \alpha_i,\\
&\beta_i^* = \beta_i,\\
&\phi_\eta^*(i) = \phi_\eta(i),\\
&\phi_\epsilon^*(i) = \phi_\epsilon(i).
\end{aligned}$$

We again analyse the first two time points together in order to get a good sample for β and ϕ_η which do not appear in the weight function. The starting sample $(x_1^*(1), x_2^*(1), \alpha_1^*, \beta_1^*, \phi_1^*(1), \phi_\eta^*(1), \phi_\epsilon^*(1)), \ldots, (x_1^*(m), x_2^*(m), \alpha_m^*, \beta_m^*, \phi_1^*(m), \phi_\eta^*(m), \phi_\epsilon^*(m))$ has size m a lot larger than n and is obtained by drawing $(\alpha_i^*, \beta_i^*, \phi_1^*(i), \phi_\eta^*(i), \phi_\epsilon^*(i))$ from the prior of $\alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon$, then $x_1^*(i)$ from $N(0, 1/\phi_1^*(i))$ and finally $x_2^*(i)$ from $N(\beta_i^* x_1^*(i), 1/\phi_\eta^*(i))$, for $i = 1, \ldots, m$. We take $u_1 = u_2 = 0$ and when we observe y_1 and y_2 the resampling weight of

each point $(x_1^*(i), x_2^*(i), \alpha_i^*, \beta_i^*, \phi_1^*(i), \phi_\eta^*(i), \phi_\epsilon^*(i))$ is

$$w_i = \phi_\epsilon^*(i) \exp\left[-\frac{\phi_\epsilon^*(i)}{2}\left(\left(y_1 - \alpha_i^* x_1^*(i)^2\right)^2 + \left(y_2 - \alpha_i^* x_2^*(i)^2\right)^2\right)\right].$$

Resampling gives $(x_1(1), x_2(1), \alpha_1, \beta_1, \phi_1(1), \phi_\eta(1), \phi_\epsilon(1)), \ldots, (x_1(n), x_2(n), \alpha_n, \beta_n, \phi_1(n), \phi_\eta(n), \phi_\epsilon(n))$ from $p(x_1, x_2, \alpha, \beta, \phi_1, \phi_\eta, \phi_\epsilon|\mathbf{y}_2, \mathbf{u}_2)$. From this sample we drop the x_1 part since we are not interested in x_1 anymore and the ϕ_1 part which amounts to integrating ϕ_1 out of the posteriors.

Apart from taking the mode of $p(x_{t-1}|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ as u_t we can also use with straightforward modifications any of the two probabilistic methods presented in the previous chapter. If we want to maximize $\Pr\left(|X_t - u_t| \leq V|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right)$ for some V, we can estimate this probability by either

$$\hat{\Pr}\left(|X_t - u_t| \leq V|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right) = \frac{1}{n}\sum_{i=1}^{n} I(x_t^*(i), u_t)$$

or

$$\hat{\Pr}\left(|X_t - u_t| \leq V|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}\right) = \frac{1}{n}\sum_{i=1}^{n}\left[\Phi\left((u_t + V - \beta_i x_{t-1}(i))\phi_\eta^{1/2}(i)\right) - \Phi\left((u_t - V - \beta_i x_{t-1}(i))\phi_\eta^{1/2}(i)\right)\right],$$

where $x_t^*(1), \ldots, x_t^*(n)$ is a sample from $p(x_t|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ and $(x_{t-1}(1), \beta_1, \phi_\eta(1)), \ldots, (x_{t-1}(n), \beta_n, \phi_\eta(n))$ is a sample from $p(x_{t-1}, \beta, \phi_\eta|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$. $I(x_t, u_t)$ denotes an indicator function taking value 1 if $|x_t - u_t| \leq V$ and 0 otherwise,

while Φ denotes the cdf of the standard Normal distribution. If on the other hand we want u_t to maximize $\Pr(y_t \leq b|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}, u_t)$ for some b, we can estimate this probability by

$$\hat{\Pr}(y_t \leq b|\mathbf{y}_{t-1}, \mathbf{u}_{t-1}, u_t) = \frac{1}{n}\sum_{i=1}^{n} \Phi\left[\left(b - \alpha_i^*(x_t^*(i) - u_t)^2\right)\phi_\epsilon^*(i)^{1/2}\right],$$

where $(x_t^*(1), \alpha_1^*, \phi_\epsilon^*(1)), \ldots, (x_t^*(n), \alpha_n^*, \phi_\epsilon^*(n))$ is a sample from $p(x_t, \alpha, \phi_\epsilon|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$. All the samples from marginal distributions are obtained by using the necessary parts of samples from the corresponding distributions of all the unknown quantities.

4.3 Implementational difficulties of the methods

We have seen that the control problem of the previous section can be solved by an adaptation of the methods used in the last chapter for a related problem. The distributions involved are intractable and must therefore be represented by samples obtained from them. The most efficient way to do that seems to be resampling. This approach ought from a theoretical point of view to perform well. In practice however, unexpected problems crop up and they are caused precisely by resampling. In this section we explain what goes wrong and why.

We begin with the exposition for the case of known α. At time t the weight assigned to a point $(x_t^*(i), \beta_i^*, \phi_\eta^*(i), \phi_\epsilon^*(i))$ from the prior distribution of x_t, β, ϕ_η, ϕ_ϵ is

$$w_i = \phi_\epsilon^*(i)^{1/2} \exp\left[-\frac{\phi_\epsilon^*(i)}{2}\left(y_t - \alpha(x_t^*(i) - u_t)^2\right)^2\right].$$

In Figure 4.1 we present the graph of the generic function w with formula $w(\phi, \kappa) = \phi^{1/2} \exp\left[-\frac{\phi}{2}(y-\kappa)^2\right]$ for a particular value of y and for positive κ. We see that the larger values of w correspond to points where κ is close to y. For values of κ further away from y small values of ϕ give largish values to w.

For our problem this means that if y_t is positive the larger weights will go to the prior sample points that give $\alpha(x_t^*(i) - u_t)^2$ close to y_t. These are the points with $x_t^*(i)$ close to one of the maximum likelihood estimates (MLE's) of x_t, $u_t \pm \sqrt{y_t/\alpha}$. Each $x_t^*(i)$ is a random draw from a $N(\beta_i x_{t-1}(i), \phi_\eta(i)^{-1})$ distribution, where $(x_{t-1}(i), \beta_i, \phi_\eta(i))$ is the corresponding sample point from the previous posterior distribution. It is much more likely to get a point $x_t^*(i)$ close to one of the MLE's from a $N(\beta_i x_{t-1}(i), \phi_\eta(i)^{-1})$ distribution with $\beta_i x_{t-1}(i)$ close to one of the MLE's rather than from one with $\beta_i x_{t-1}(i)$ far from them even if the latter has a lot smaller $\phi_\eta(i)$. If x_t happens to come from the tails of $N(\beta x_{t-1}, \phi_\eta^{-1})$ and if y_t is close to $\alpha(x_t - u_t)^2$ then sample points with $x_{t-1}(i), \beta_i$ and $\phi_\eta(i)$ close to the true values of these unknowns will most probably give points $x_t^*(i)$ far from the MLE's while points with bad $x_{t-1}(i)$ and β_i but for which $\beta_i x_{t-1}(i)$ is close to one of the MLE's will give $x_t^*(i)$ which will receive the larger weights. Therefore, the posterior sample will be very poor in terms of β values. Unsatisfactory values of β will generate unsatisfactory values of x_{t+1} in the next prior sample. If there are no good values of β left then all the prior sample of x_{t+1} will be poor. This will lead to u losing track of x in the future. Fortunately this situation is slowly reversed.

First, when both MLE's at time t are outside the range of the prior sample of x_t we draw the posterior sample of it from a mixture of two equally weighted Normal distributions with means the MLE's and variance σ^2, where

$$\sigma = \frac{\sqrt{(y_t + 2\bar{\sigma}_\epsilon)/\alpha}}{2} \quad \text{with} \quad \bar{\sigma}_\epsilon = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\sqrt{\phi_\epsilon^*(i)}}.$$

This is what we called intervention in the last chapter. Notice that if $\alpha = 1/2$ we have the same σ as in the last chapter. Intervention brings the posterior sample of x_t close to the true value of x_t again. The posterior sample of the other unknowns is the same as their prior sample.

Secondly, when x_t is not very far from βx_{t-1} even if all the values of β in the prior sample are bad there is a reasonable chance that some of the large weights will go to values closer to the true value of β. If we use smooth bootstrap as our resampling method (which is what we do) the posterior sample of β will contain more values closer to the truth. If this keeps on happening it will eventually lead to a good sample for β again. However, a long time may elapse before this good sample is obtained during which we are paying a high cost. Moreover, nothing guarantees that the samples will not deteriorate again as a result of another x_t far from βx_{t-1}.

If the prior and posterior distributions where available analytically such problems would not occur. This is so because if a point $(x_t, \beta, \phi_\eta, \phi_\epsilon)$ has a bad β value even if it has a high likelihood its small prior probability will counterbalance this and its posterior probability will not be large. When the prior distribution is represented by a sample, however, even one bad point with very large weight is enough to produce a poor posterior sample.

When α is also unknown matters can get even worse. The weight assigned to a point $(x_t^*(i), \alpha_i^*, \beta_i^*, \phi_\eta^*(i), \phi_\epsilon^*(i))$ will now be

$$w_i = \phi_\epsilon^*(i)^{1/2} \exp\left[-\frac{\phi_\epsilon^*(i)}{2}\left(y_t - \alpha_i^*(x_t^*(i) - u_t)^2\right)^2\right].$$

All the things that we mentioned above can still go wrong and moreover, sample points with unsatisfactory values of α_i^* and $x_t^*(i)$ but with $\alpha_i^*(x_t^*(i) - u_t)^2$ close to y_t get large weights. On top of that, there are now infinite MLE's at each time t and therefore intervention cannot be used.

In general we can say that there is a confounding between the effects of β and those of the disturbance η_t in the system evolution equation and between the effects of α and those of the noise ϵ_t in the observation equation. Although this does not affect the theoretical analysis of the problem, it creates difficulties for a sample-based approach if at least one of α and β is not known. A remedy could be to increase the size of the samples that represent the distributions of interest. This would improve the chance of getting good sample points from resampling even when x_t is far from βx_{t-1}. These good sample points would dominate future resamplings if future values of x were close to βx_{t-1}.

We now demonstrate a case where problems appear. We have simulated a chain x_t for $t = 1, \ldots, 100$ and we apply Titterington's method with $K = 0$ in order to produce the u points. The unknowns are $\beta, \phi_1, \phi_\eta, \phi_\epsilon$ and their true values are $\beta = 1, \phi_1 = 0.25, \phi_\eta = 0.25, \phi_\epsilon = 1$. The value of α is $1/2$ and it is known. The prior distribution of each of the precisions is Gamma(0.1,0.1) while the prior of β is Uniform on the interval $(0, 3)$. In the spirit of the previous chapter β, ϕ_1, ϕ_η and ϕ_ϵ are considered independent *a priori* and the

first two time points are analysed together. All the samples that represent prior and posterior distributions have size 1000 apart from the sample from $p(x_1, x_2, \beta, \phi_1, \phi_\eta, \phi_\epsilon)$ which has size 10000.

Figure 4.3 presents the values of x_t and u_t for $t = 1, \ldots, 100$. We see that somewhere after time 50 u_t loses x_t to recover it only occasionally from then onwards. Serious problems first appear at time 50. The true value of x_{50} is 31.77 which is more than 1.5 standard deviations from 28.61, the true value of x_{49}. The MLE's are 5.5 and 31.64. In Figure 4.4 we display the posterior samples of x_{49} and β given $\mathbf{y}_{49}$. It is clear that the combinations of x_{49} and β points that will give x_{50} prior points close to the MLE's are those with $x_{49} \approx 28$ and $\beta \geq 1$. This is verified by the histogram of the prior sample of x_{50} and by the plot of the weights this sample's members receive. In the rest of Fig. 4.4 we see the posterior samples of x_{50} and β. Almost all β values are larger than 1. This causes future samples of x to move rightwards and therefore, to drift further away from the true values of x. Only when intervention is applied a few time points later does u_t recover x_t.

A different sort of problem occurs at time 66. By then, a string of true x_t values close to each other has led to a quite good sample of β. Figure 4.5 shows the posterior samples of x_{65} and β given $\mathbf{y}_{65}$. We see that the main mode of the x sample is far from the true value of x_{65}. This leads to u_{66} being far from x_{66} which produces a large y_{66}. From among all the points belonging to the prior sample of x_{66} only a few receive all the weight and they all are far from x_{66} and with large β. From then on only frequent interventions manage to keep u_t close to x_t. In Figure 4.6 we present boxplots of the posterior samples of β,ϕ_η and ϕ_ϵ. Before the samples collapsed at time 66 the samples of ϕ_η and ϕ_ϵ were very spread out and we do not present them because this

would obscure all detail in the boxplots of the later samples. We see how bad the posterior samples are even after 100 items of data have been processed.

The last two sections have demonstrated a problem with resampling methods. When the importance sampler is the prior distribution the resampling weights are exclusively influenced by the likelihood of the sample points. A single bad sample point with high likelihood can result in a very poor posterior sample if the sample size is not big enough. A different importance sampler can be helpful but then the weights may not be easy to compute. We will examine some alternative importance samplers in the next chapter but for the time being we turn our attention to another variant of the control problem.

4.4 Multivariate control

In this section we deal with the multivariate equivalent of the control problem of the last chapter. In other words we now have a response surface that changes location in d-dimensional space in discrete time. We are again trying to follow an extreme point of the surface as closely as possible. We use the same notation as before and we denote the location of the extremum at time t by x_t. If we guess that it is at u_t we obtain a noisy observation of the surface's height over u_t

$$y_t = \phi(u_t, x_t) + \epsilon_t.$$

In our case the surface is a paraboloid,

$$\phi(u_t, x_t) = \frac{1}{2}||x_t - u_t||^2,$$

where $||x_t - u_t||^2$ is the squared distance between x_t and u_t,

$$||x_t - u_t|| = \sqrt{\sum_{i=1}^{d}(x_{ti} - u_{ti})^2}.$$

The extremum x_t is the paraboloid's minimum. The observation is univariate and we assume that the ϵ_t are i.i.d. $N(0, \sigma_\epsilon^2)$ random variables.

The movement of the surface is such that the movement of x_t is described by a random walk in d dimensions:

$$x_t = x_{t-1} + \eta_t,$$

where the η_t are i.i.d. $N_d(\mathbf{0}, \Sigma_\eta)$ random variables, i.e. they follow the d-variate Normal distribution with mean vector $\mathbf{0}$ and variance-covariance matrix Σ_η. We also assume that $\{x_t\}$, $\{\epsilon_t\}$ and $\{\eta_t\}$ are mutually independent. For x_1 we assume that it follows a $N_d(\mathbf{0}, \Sigma_1)$ distribution *a priori* and for this reason we always choose $u_1 = \mathbf{0}$.

We can see that even if there is no observation noise there will still be ambiguity about where x_t lies since all the points on the d-dimensional hypersphere with centre u_t and radius $\sqrt{2y_t}$ can give observation y_t. It is clear that the scope for choice increases as d becomes larger. The presence of noise increases the number of candidates even more. Probabilistic reasoning can lead at each

time t to a posterior distribution of x_t given all the data $\mathbf{y}_t$ gathered up to that time. This posterior will in turn give rise by a simple integration to the prior distribution of x_{t+1}. We argue that the mode of the posterior of x_t is the best choice for u_{t+1}. We first provide a theoretical analysis of the problem to show that the system is controllable.

4.4.1 Theoretical analysis

As in the univariate case we will analyse the problem theoretically in the simpler case of no noise in the observations. Our aim is to examine whether the system is controllable, in other words whether the expected rate of loss

$$\gamma_N = \frac{E\left(\sum_{t=1}^{N} ||x_t - u_t||^2\right)}{N}$$

has a finite limit. We will only consider the case of $d = 2$ and briefly mention $d = 3$ because we believe that the results generalize to higher dimensions.

Our first aim is to find a relationship between $E[||x_{t+1} - u_{t+1}||^2]$ and $E[||x_t - u_t||^2]$. The ideas rely on 2-dimensional geometry and Figure 4.2 can be helpful in understanding them. When we have u_t and we observe $y_t = \frac{1}{2}||x_t - u_t||^2$, we know that x_t lies on a circle $\Delta_{t+1} = \left\{x \in \mathbf{R}^2 : ||x - u_t|| = \sqrt{2y_t} = r_{t+1}\right\}$, i.e. the circle of centre u_t and radius $\sqrt{2y_t}$. We assign a subscript $t+1$ to the circle because it is used for choosing u_{t+1}. We do not know which point on Δ_{t+1} is x_t but, based on all our observations $\mathbf{y}_t$ and control points $\mathbf{u}_t$, as well as on the prior for x_1, we formulate our beliefs as a posterior distribution with density function $b_{t+1}(x)$ defined on Δ_{t+1}, such that

$$\int_{x \in \Delta_{t+1}} b_{t+1}(x)dx = 1$$

and

$$\int_{x \in AB} b_{t+1}(x)dx = Pr(x_t \in AB | \mathbf{y}_t, \mathbf{u}_t),$$

where AB is any segment or collection of segments of Δ_{t+1}.

The prior density for x_{t+1} will then be a mixture of Normal densities

$$\begin{aligned} p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t) &= \int_{x \in \Delta_{t+1}} b_{t+1}(x) p(x_{t+1}|x_t = x)dx \\ &\propto \int_{x \in \Delta_{t+1}} b_{t+1}(x) \phi_2 \left(\Sigma_\eta^{-1/2} ||x_{t+1} - x|| \right) dx, \end{aligned}$$

where $\phi_d(a)$ is the value of the density function of the standard d-variate Normal distribution at point a.

Now suppose that $x_t = x^*$ and we have chosen u_{t+1} as our control point. It is not difficult to verify that

$$\begin{aligned} & E[||x_{t+1} - u_{t+1}||^2 | x_t = x^*, \mathbf{y}_t, \mathbf{u}_t] \\ &= ||x^* - u_{t+1}||^2 + E[||x_{t+1} - x^*||^2 | x_t = x^*]. \end{aligned}$$

If $X \sim N_d(\mu, \Sigma)$ then

$$E[||X-\mu||^2] = \sum_{i=1}^{d} E[(X_i - \mu_i)^2] = \sum_{i=1}^{d} \sigma_{ii}^2$$

where σ_{ii}^2 is the i-th element on the diagonal of Σ. Here therefore we have

$$\begin{aligned} & E[||x_{t+1} - u_{t+1}||^2 | \mathbf{y}_t, \mathbf{u}_t] \\ &= \int_{x \in \Delta_{t+1}} b_{t+1}(x) E[||x_{t+1} - u_{t+1}||^2 | x_t = x, \mathbf{y}_t, \mathbf{u}_t] dx \\ &= E[||x_{t+1} - x||^2 | x_t = x] + \int_{x \in \Delta_{t+1}} b_{t+1}(x) ||x - u_{t+1}||^2 dx \\ &= \sigma_{\eta 1}^2 + \sigma_{\eta 2}^2 + \int_{x \in \Delta_{t+1}} b_{t+1}(x) ||x - u_{t+1}||^2 dx, \end{aligned}$$

where $\sigma_{\eta,i}^2$ is the i-th element on the diagonal of Σ_η.

For a particular $x \in \Delta_{t+1}$ denote by $\psi(x)$ the angle between the radii connecting u_{t+1} with u_t and x with u_t. Denote by α the length of the perpendicular from u_t on to the chord joining u_{t+1} and x. Then

$$\begin{aligned} ||x - u_{t+1}||^2 &= \left(2\sqrt{r_{t+1}^2 - \alpha^2}\right)^2 \\ &= 4r_{t+1}^2 \left(1 - \cos^2(\frac{\psi(x)}{2})\right) = 4||x_t - u_t||^2 \left(1 - \cos^2(\frac{\psi(x)}{2})\right). \end{aligned}$$

Therefore,

$$E[||x_{t+1} - u_{t+1}||^2|\mathbf{y}_t, \mathbf{u}_t]$$
$$= \sigma_{\eta 1}^2 + \sigma_{\eta 2}^2 + 4||x_t - u_t||^2 \int_{x \in \Delta_{t+1}} b_{t+1}(x) \left(1 - \cos^2(\frac{\psi(x)}{2})\right) dx.$$

The integral is always between 0 and 1 since b_{t+1} is a density and $0 \leq \cos^2(\psi) \leq 1$ for any angle ψ. If we denote it by B_{t+1} we get

$$E[||x_{t+1} - u_{t+1}||^2|\mathbf{y}_t, \mathbf{u}_t] = \sigma_{\eta 1}^2 + \sigma_{\eta 2}^2 + 4B_{t+1}||x_t - u_t||^2$$

and hence,

$$E[||x_{t+1} - u_{t+1}||^2] = \sigma_{\eta 1}^2 + \sigma_{\eta 2}^2 + 4E[B_{t+1}||x_t - u_t||^2]. \tag{4.3}$$

Furthermore, we can write

$$E[B_{t+1}||x_t - u_t||^2] = \frac{\beta_{t+1}}{4} E[||x_t - u_i||^2].$$

Thie results are directly analogous to those for the univariate case, see (3.9) and (3.10). If, after finite time M, β_{t+1} stabilizes to β and if $\beta < 1$ then the proof of the univariate case can be used to show that the expected rate of loss $E(\gamma_N)$ is finite and that

$$\lim E(\gamma_N) = \frac{\sigma_{\eta 1}^2 + \sigma_{\eta 2}^2}{1 - \beta}.$$

Therefore the system is again controllable.

One last note in order to complete the argument concerns the choice of u_{t+1} and then the derivation of the posterior distribution for x_{t+1} after we have observed y_{t+1}. It is clear that u_{t+1} is the maximizer of $b_{t+1}(x)$. After y_{t+1} is observed a new circle Δ_{t+2} with centre u_{t+1} and radius $\sqrt{2y_{t+1}}$ emerges. We have already derived the prior distribution of x_{t+1}. Its posterior, whose density will be denoted by b_{t+2}, will be the same density but confined to the points of Δ_{t+2}. In other words,

$$b_{t+2}(x) = \begin{cases} \dfrac{\int_{z\in\Delta_{t+1}} b_{t+1}(z)\phi_2\left(\Sigma_\eta^{-1/2}||x-z||\right)dz}{\int_{y\in\Delta_{t+2}}\int_{z\in\Delta_{t+1}} b_{t+1}(z)\phi_2\left(\Sigma_\eta^{-1/2}||y-z||\right)dzdy} & \text{if} \quad x\in\Delta_{t+2} \\ 0 & \text{otherwise} \end{cases} . \qquad (4.4)$$

The very first posterior we need is b_2. It is easy to see that if Σ_1 is diagonal b_2 will be a uniform distribution over Δ_2.

In three dimensions Δ_{t+1} is a sphere of centre u_t and radius r_{t+1}. Then doing the same calculations as above we get that

$$E[||x_{t+1}-u_{t+1}||^2|\mathbf{y}_t,\mathbf{u}_t] = \sigma_{\eta 1}^2+\sigma_{\eta 2}^2+\sigma_{\eta 3}^2+\int_{x\in\Delta_{t+1}} b_{t+1}(x)||x-u_{t+1}||^2dx.$$

Now imagine a point $x\in\Delta_{t+1}$ and the two radii connecting u_{t+1} with u_t and x with u_t. They define a plane P whose intersection with Δ_{t+1} is a circle with the same centre and radius as Δ_{t+1}. If $\psi(x)$ is the angle between these radii on P we again have

$$E[||x_{t+1} - u_{t+1}||^2] = \sigma_{\eta 1}^2 + \sigma_{\eta 2}^2 + \sigma_{\eta 3}^2 + 4E[B_{t+1}||x_t - u_t||^2],$$

where

$$B_{t+1} = \int_{x \in \Delta_{t+1}} b_{t+1}(x) \left(1 - \cos^2(\frac{\psi(x)}{2})\right) dx.$$

Then, reasoning similar to that of the bivariate case leads to the result that the expected rate of loss is finite and that

$$\lim E(\gamma_N) = \frac{\sigma_{\eta 1}^2 + \sigma_{\eta 2}^2 + \sigma_{\eta 3}^2}{1 - \beta}.$$

In the same way we can generalize the results for spaces of any dimension $d > 3$.

An important feature of the formulae we have derived is that the presence of correlations between the components of the disturbance does not complicate things. The mathematical results still hold. Only the posteriors b_{t+1} (and therefore β) change.

As in the univariate case, if the system evolution equation has the form $x_t = \kappa x_{t-1} + \eta_t$, where κ is a known non-zero scalar quantity, the results will be similar with only the appropriate changes to incorporate κ. We do not pursue this further.

4.4.2 Checking the theory

In this section we present a simulation study that we conducted in order to examine whether (4.3) holds and whether the value of β_{t+1} stabilizes to a level below 1, i.e. whether the system is indeed controllable. We only simulated the 2-dimensional case because we took discrete approximations of the necessary continuous densities. It was felt that in higher dimensions we would either obtain very crude approximations or we would have to run the simulations for too long.

We now describe the set-up of the experiment. The calculation of posterior densities involves integration over the circumference of a circle and this is analytically impossible. This leads us to discretize the densities in the following way. We represent each circle Δ_t by a group of 1000 points $z_{t,1}, \ldots, z_{t,1000}$ spread equidistantly on its circumference. The corresponding pdf b_t is replaced by the discrete probability function $\hat{b}_t$ which is such that

$$\sum_{i=1}^{1000} \hat{b}_t(z_{t,i}) = 1.$$

Therefore, (4.4) becomes

$$\hat{b}_{t+2}(z_{t+2,k}) = \frac{\sum_{i=1}^{1000} \hat{b}_{t+1}(z_{t+1,i}) \phi_2 \left(\Sigma_\eta^{-1/2} || z_{t+2,k} - z_{t+1,i} || \right)}{\sum_{j=1}^{1000} \sum_{i=1}^{1000} \hat{b}_{t+1}(z_{t+1,i}) \phi_2 \left(\Sigma_\eta^{-1/2} || z_{t+2,j} - z_{t+1,i} || \right)} \quad , \quad k = 1, \ldots, 1000. \tag{4.5}$$

We only consider a system evolution equation of the form $x_{t+1} = x_t + \eta_t$, which means that we choose as u_{t+1} the maximizer of b_{t+1}. We want u_{t+1} to

be any point on Δ_{t+1} and not necessarily one of $z_{t+1,1}, \ldots, z_{t+1,1000}$. For this reason we take as u_{t+1} the maximizer of

$$b^*(x) = \sum_{i=1}^{1000} \hat{b}_t(z_{t,i}) \phi_2 \left(\Sigma_\eta^{-1/2} (x - z_{t,i}) \right), \tag{4.6}$$

with x lying on the circumference of Δ_{t+1}. In order to calculate the first posterior density $\hat{b}_2$ with the aid of (4.5) and to get u_2 we define a *dummy* circle Δ_1 of zero radius by taking $z_{1,1} = \ldots = z_{1,1000} = u_1$ and we get $\hat{b}_1 = 0.001$ for $i = 1, \ldots, 1000$. For the maximization of (4.6) we use the subroutine **e04bbf** of the NAG library for Fortran 77. This subroutine can only maximize univariate functions and we make (4.6) univariate by the following transformation. Each x that we examine lies on Δ_{t+1}. Each point on that circle can be represented by its polar coordinates with respect to u_t. Since they all lie at a distance r_{t+1} from it only the angle θ between the x-axis and the segment connecting them with u_t changes. Thus, (4.6) becomes a function of θ only, since r_{t+1} is known. After the optimal angle, θ_{opt}, say, has been found we easily calculate the coordinates of u_{t+1} as

$$\begin{aligned} u_{t+1,1} &= u_{t,1} + r_{t+1} \cos(\theta_{opt}), \\ u_{t+1,2} &= u_{t,2} + r_{t+1} \sin(\theta_{opt}). \end{aligned}$$

In fact, we use the same trick in approximating Δ_{t+1}. Since it is centred on u_t and has radius r_{t+1}, we can split $[0, 2\pi)$ into 1000 equal segments and the angles $\theta_i = (i-1)/(2\pi)$ that result from this give us the points $z_{t+1,i}$. Therefore, instead of having $\hat{b}_{t+1}(z_{t+1,i})$ we can equally well have $\hat{b}_{t+1}(\theta_i)$.

We also discretize B_{t+1}. All the angles are measured anticlockwise and therefore, the angle ψ_i between u_{t+1} and $z_{t+1,i}$ is

$$\psi_i = \theta_i + 2\pi - \theta_{opt}.$$

Then B_{t+1}, which, incidentally, after a few simple calculations can be expressed as

$$B_{t+1} = \frac{1}{2} - \frac{1}{2}\int_{x\in\Delta_{t+1}} b_{t+1}(x)\cos\left(\psi(x)\right)dx,$$

is approximated by

$$\hat{B}_{t+1} = \frac{1}{2} - \frac{1}{2}\sum_{i=1}^{1000}\hat{b}_{t+1}(\theta_i)\cos(\theta_i + 2\pi - \theta_{opt}).$$

Instead of 1000 one could choose more points to improve the approximation, because it is obvious that, the larger the radius, the more points we need to approximate the circle. However, even with just 1000 points the calculations are quite time consuming.

In order to simulate expectations we created 400 realizations of a true chain of extrema $x_t, t = 1, \ldots, 100$. Then, for each of them we approximated b_t and B_t and chose the chain of optimal u_t. We approximated expectations as

$$\hat{E}\left[||x_t - u_t||^2\right] = \frac{\sum_{i=1}^{400}||x_t(i) - u_t(i)||^2}{400}$$

$$\hat{E}\left[B_t||x_{t-1}-u_{t-1}||^2\right] = \frac{\sum_{i=1}^{400}\hat{B}_t(i)||x_{t-1}(i)-u_{t-1}(i)||^2}{400}$$
$$\hat{E}(\gamma_N) = \frac{\sum_{t=1}^{N}\hat{E}\left[||x_t-u_t||^2\right]}{N}.$$

The simulations were performed under three different variance regimes. In the first one, $\Sigma_1 = \Sigma_\eta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, in the second $\Sigma_1 = \Sigma_\eta = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ and in the third $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma_\eta = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$. The third case was designed so as to check the effect that correlation between the components of the disturbance has on the limit of the expected rate of loss. The results appear in Figure 4.7. In the first column we plot the estimated expected rates of loss and we see that they stabilize quickly in all cases. In the second column we present the evolution of the expected squared distances $\hat{E}[||x_t-u_t||^2]$. We notice that after a short initial transient period they reach a certain level (different in each case) around which they fluctuate. The third column shows the evolution of $\hat{\beta}_t$ in time. In all cases it quickly attains a level below 1 and fluctuates around it. In the fourth column we test the validity of (4.3). We see that the ratio

$$\frac{E[||x_t-u_t||^2]-\sigma_{\eta 1}^2-\sigma_{\eta 2}^2}{4E[B_t||x_{t-1}-u_{t-1}||^2]}$$

stays close to 1 with some variability. The fluctuations that the quantities experience around their constant levels are due, in our opinion, to the fact that they are sample estimates and to the discretizations of the continuous densities. In all cases the limit of the rate of loss agrees with its theoretical derivation. We can therefore be fairly sure that our theoretical analysis and

the conclusions based on it are sound.

A remarkable feature of the simulation results is that the limit of the expected rate of loss is lower in the third case than in the first one. In other words, the presence of correlation between the components of the disturbance has a beneficial effect as regards the controllability of the system. We think that when this correlation is strong the number of directions x_t can take during its random walk is greatly reduced and this makes the search for it easier.

4.4.3 Practical application

In a practical application noise will also be present. Nevertheless the principle behind the choice of u_{t+1} will be the same. Given the observations $\mathbf{y}_t$ and the control points $\mathbf{u}_t$ we will choose as u_{t+1} the mode of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. Unfortunately this distribution is not available in closed form and we will again have to use resampling starting with a random sample $(x_1(1), \ldots, x_1(n))$ from $N_2(\mathbf{0}, \Sigma_1)$, the prior distribution of x_1. We take $u_1 = \mathbf{0}$. When the sample $(x_t^*(1), \ldots, x_t^*(n))$ from $p(x_t|\mathbf{y}_{t-1}, \mathbf{u}_{t-1})$ is available and y_t is obtained at u_t, the weight $w_t(i)$ corresponding to $x_t^*(i)$ will be

$$w_t(i) = \exp\left[-\frac{1}{2\sigma_\epsilon^2}\left(y_t - \frac{1}{2}||x_t^*(i) - u_t||^2\right)^2\right].$$

The samples take the form of disks around the current control point and if the value of the observation is large they may look like "doughnuts". Sometimes the value of y_t may be so large that all the MLE's lie outside the range of the prior sample. This could make the samples collapse. In order to prevent this from happening we employ intervention, as we did in the univariate case.

More specifically, we replace the posterior sample by a sample taken as follows. Each of the MLE's can be represented by its distance $r_{t+1} = \sqrt{2y_t}$ from u_t and an angle θ. We sample n angles from a Uniform distribution over $[0, 2\pi)$ and then for each of them we sample a point from a bivariate Normal distribution with mean the corresponding MLE and a diagonal variance matrix with both diagonal elements equal to

$$\sigma^2 = \frac{2y_t + 4\sigma_\epsilon^2}{4}.$$

We again tried to maintain a correspondence with the univariate case. A posterior sample for x_t can be transformed to a prior sample for x_{t+1} by adding to each of its points a random draw from the distribution of the disturbance η_{t+1}. The resampling method used was smooth bootstrap.

In order to find the mode of $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$ when all we have is the sample from it we calculate its density estimate from that sample and maximize it using the NAG routine **e04lbf**. For the density estimation we use the optimal bandwidth given by (2.7) and the correction (2.8).

Our method can easily be modified if as well as x_t there are additional unknown quantities. However, we do not pursue this further.

4.4.4 Simulations

Here we present the results of a simulation study conducted in order to examine whether the system is controllable under several variance regimes. Figures 4.8(a) to 4.8(e) present estimated expected rates of loss for Titterington's

method with $K = 0$ for the following five cases:

- $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\Sigma_\eta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\sigma_\epsilon^2 = 1$, in 4.8(a).
- $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\Sigma_\eta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\sigma_\epsilon^2 = 3$, in 4.8(b).
- $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\Sigma_\eta = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, $\sigma_\epsilon^2 = 1$, in 4.8(c).
- $\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, $\Sigma_\eta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\sigma_\epsilon^2 = 1$, in 4.8(d).
- $\Sigma_1 = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$, $\Sigma_\eta = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$, $\sigma_\epsilon^2 = 1$, in 4.8(e).

The rates were once more simulated by taking 50 chains, each of 100 time points. We can observe the same features as in the univariate case. The system is controllable and we see that the level of the rates agrees with the theoretical case. We see furthermore that the limit of the rate depends more on Σ_η than on Σ_1. We finally note once more that when there is correlation between the components of x_t the limit of the expected rate of loss gets smaller.

4.5 Conclusions

In this chapter we dealt with two new forms of the extremum adaptation problem, one including unknown coefficients in the system and observation

equations and the other referring to multidimensional spaces. We saw that our adaptation of Titterington's method proposed in the last chapter can be straightforwardly modified so that it works here too.

In the multidimensional case we reached similar conclusions about the control problem to those of the unidimensional one. The case of unknown coefficients however, highlighted a disadvantage of the resampling techniques used. When we adopt the prior distribution as the importance sampler the likelihood solely affects the resampling weights. Sometimes prior sample points which have high likelihood but which do not come from the main support of the posterior distribution dominate resampling, resulting in very poor posterior samples. One solution is to increase the sample size so as to increase the chance that satisfactory points will be included in the posterior sample. It would however, be preferable to use a more efficient importance sampler which at the same time permits easy calculation of the resampling weights. The next chapter examines some alternatives that have been recently proposed in that respect.

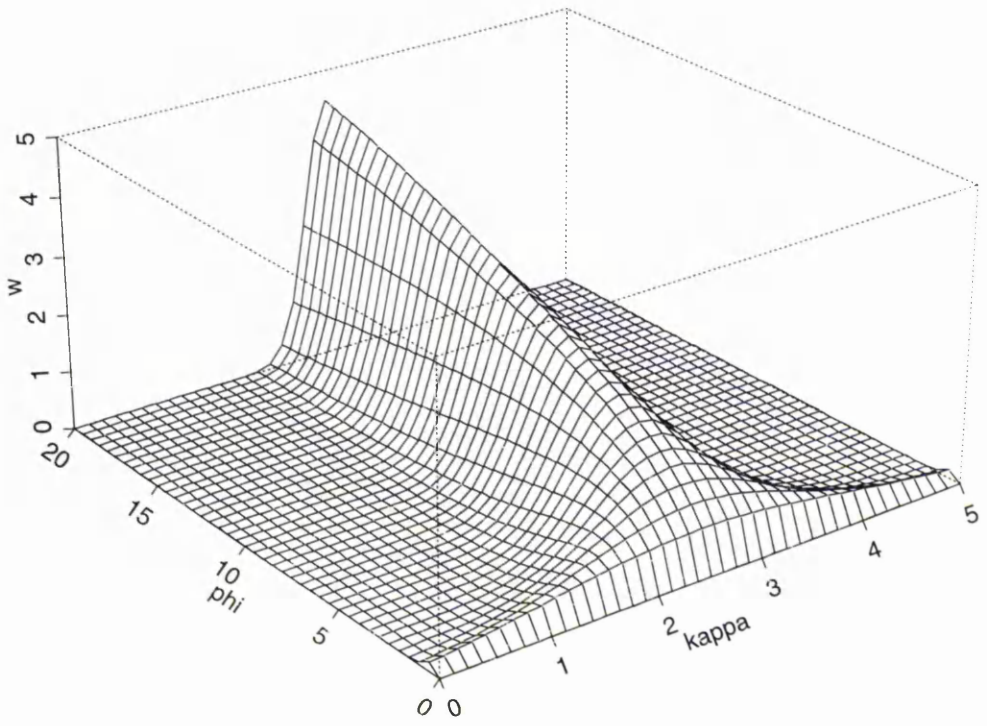


Figure 4.1: Graph of the function $w(\phi, \kappa) = \phi^{1/2} \exp\left[-\frac{\phi}{2}(y-\kappa)^2\right]$ for $y = 2.5$.

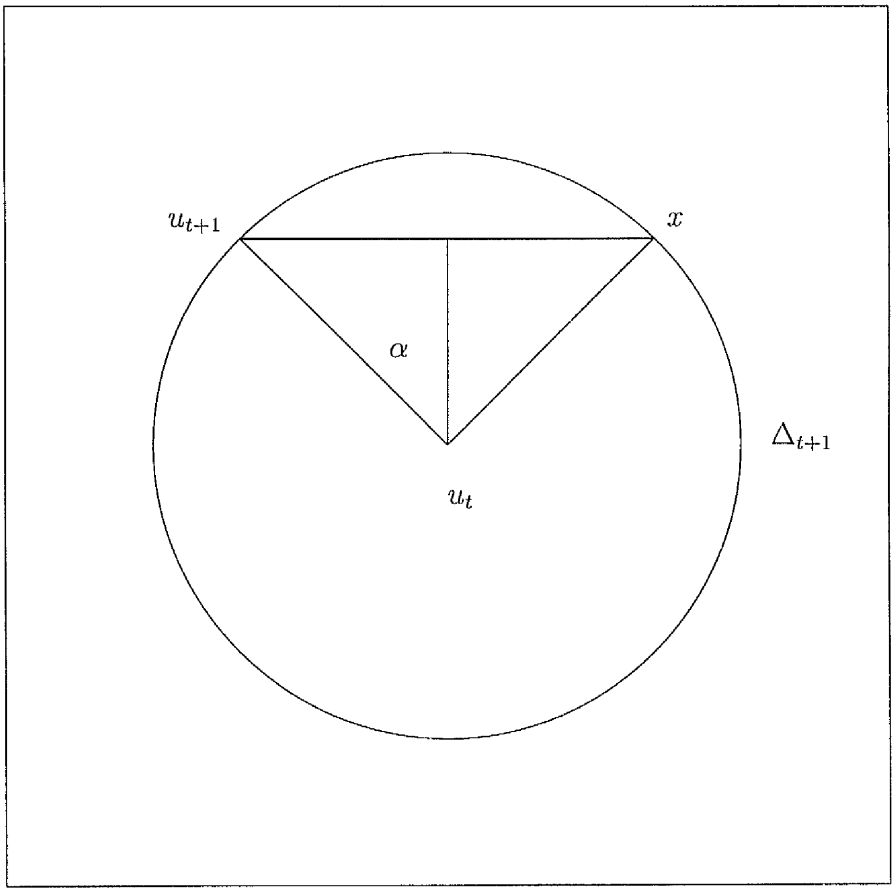


Figure 4.2: Illustration of the geometric ideas behind section 4.4.1.

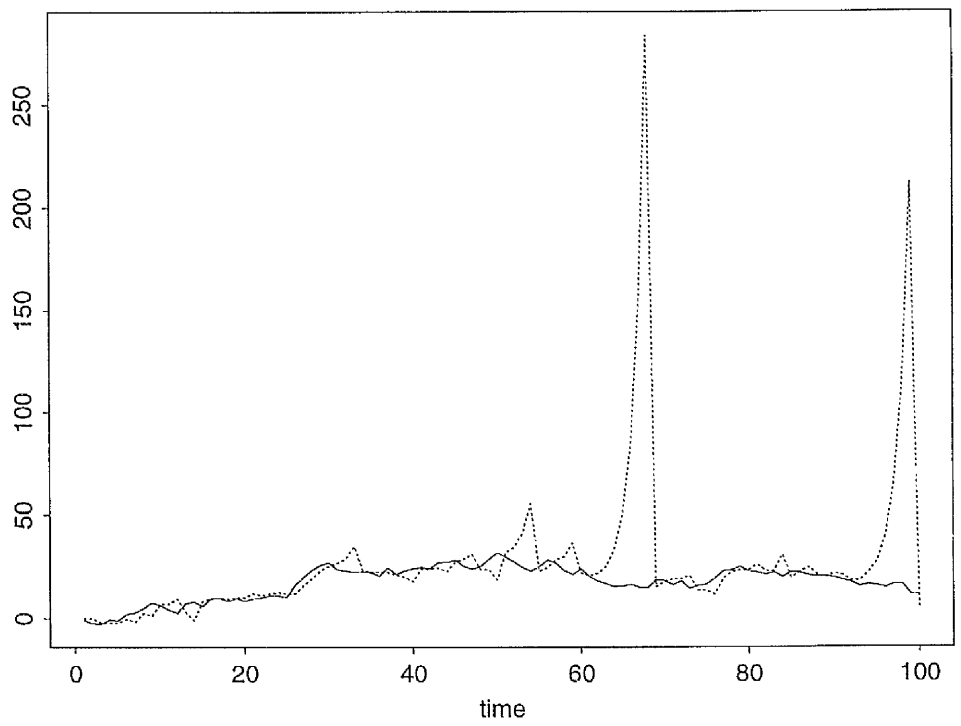


Figure 4.3: Trace of x_t and u_t for $t = 1, \ldots, 100$ from the simulation example of section 4.3. Solid line: x_t, broken line: u_t.

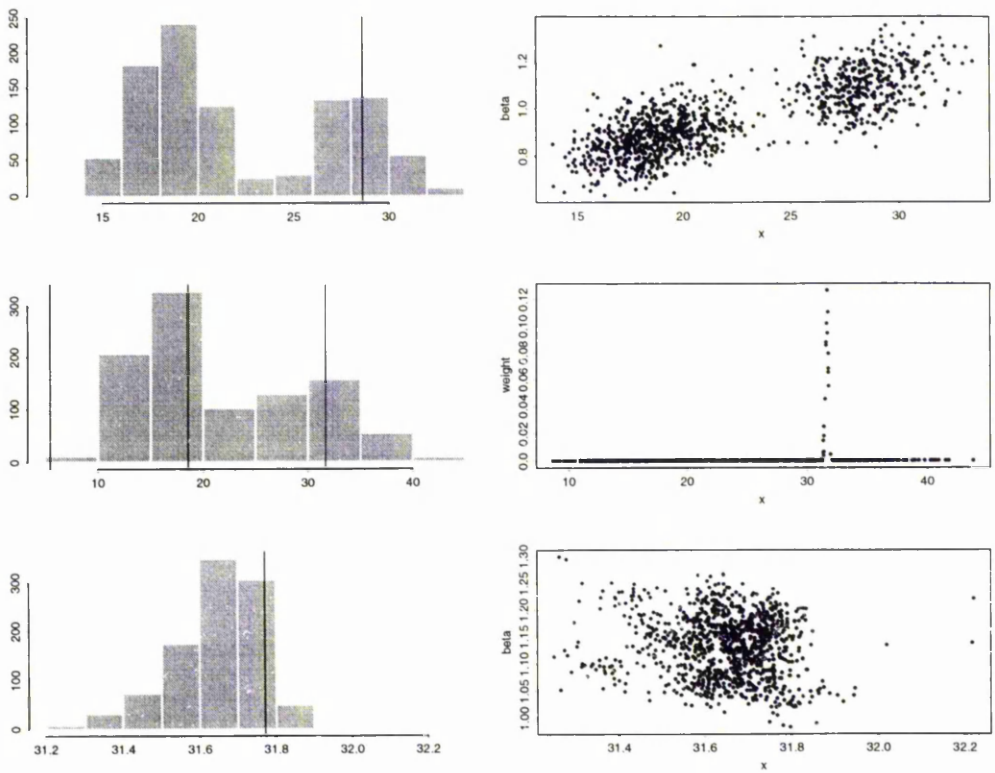


Figure 4.4: Top left: Posterior sample of x_{49}. The vertical line denotes the true location of x_{49}. Top right: Posterior sample of x_{49} and β. Middle left: Prior sample of x_{50}. The vertical lines denote the location of u_{50} and the two MLE's. Middle right: Un-normalized weights of prior x_{50} points. Bottom left: Posterior sample of x_{50}. The vertical line denotes the true location of x_{50}. Bottom right: Posterior sample of x_{50} and β. All for the simulation experiment of section 4.3.

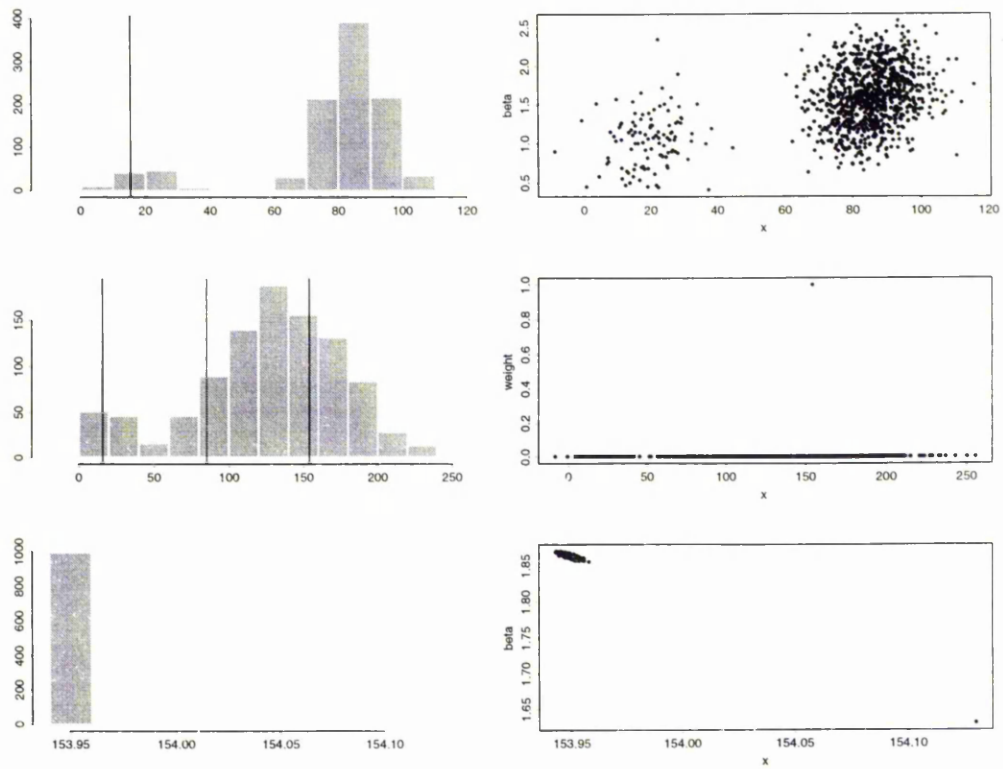


Figure 4.5: Top left: Posterior sample of x_{65}. The vertical line denotes the true location of x_{65}. Top right: Posterior sample of x_{65} and β. Middle left: Prior sample of x_{66}. The vertical lines denote the location of u_{66} and the two MLE's. Middle right: Un-normalized weights of prior x_{66} points. Bottom left: Posterior sample of x_{66}. Bottom right: Posterior sample of x_{66} and β. All for the simulation experiment of section 4.3.

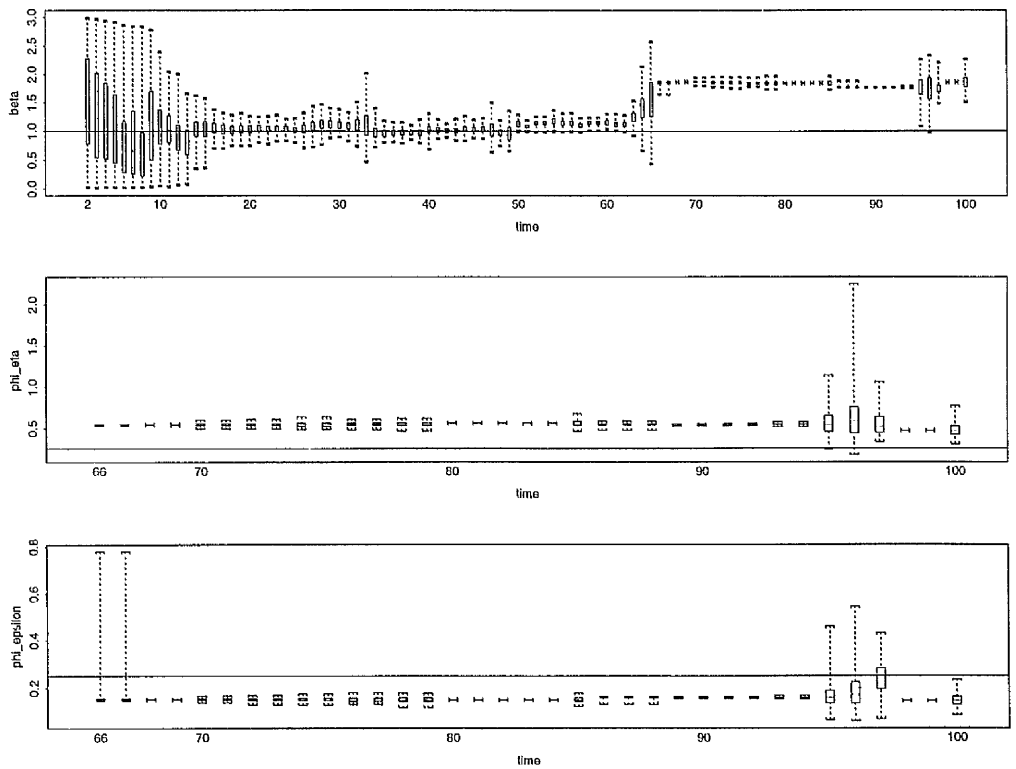


Figure 4.6: Top: Boxplots of the posterior samples of β for $t = 2, \ldots, 100$. Middle: Boxplots of the posterior samples of ϕ_η for $t = 66, \ldots, 100$. Bottom: Boxplots of the posterior samples of ϕ_ϵ for $t = 66, \ldots, 100$. All for the simulation example of section 4.3. The endpoints of each boxplot correspond to the smallest and largest value in the sample. The horizontal lines denote the true value of each parameter.

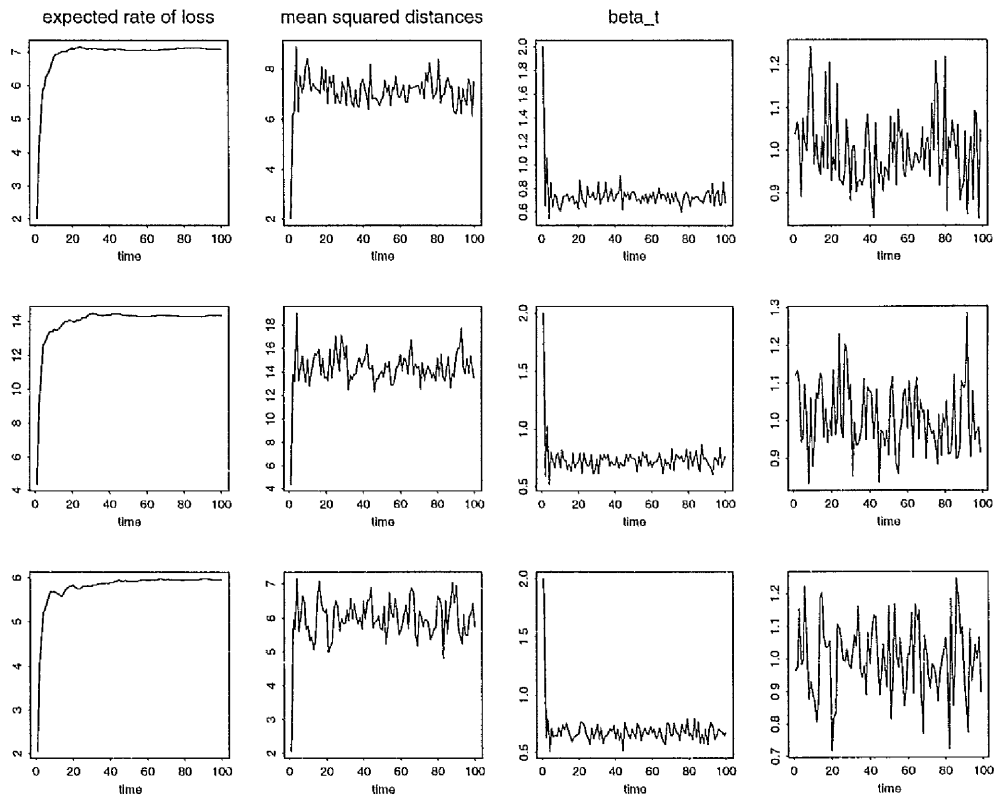


Figure 4.7: Results of the simulations in section 4.4.2. The first column shows expected rates of loss; the second column shows $\hat{E}[||x_t - u_t||^2]$; the third column shows $\hat{\beta}_t$; the fourth column shows $(\hat{E}[||x_t - u_t||^2] - (\sigma^2_{\eta 1} + \sigma^2_{\eta 2}))/(4\hat{E}[B_t||x_{t-1} - u_{t-1}||^2])$, all for $t = 1, \ldots, 100$. In the first row, $\Sigma_1 = \Sigma_\eta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. In the second row, $\Sigma_1 = \Sigma_\eta = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$. In the third row, $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma_\eta = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$.

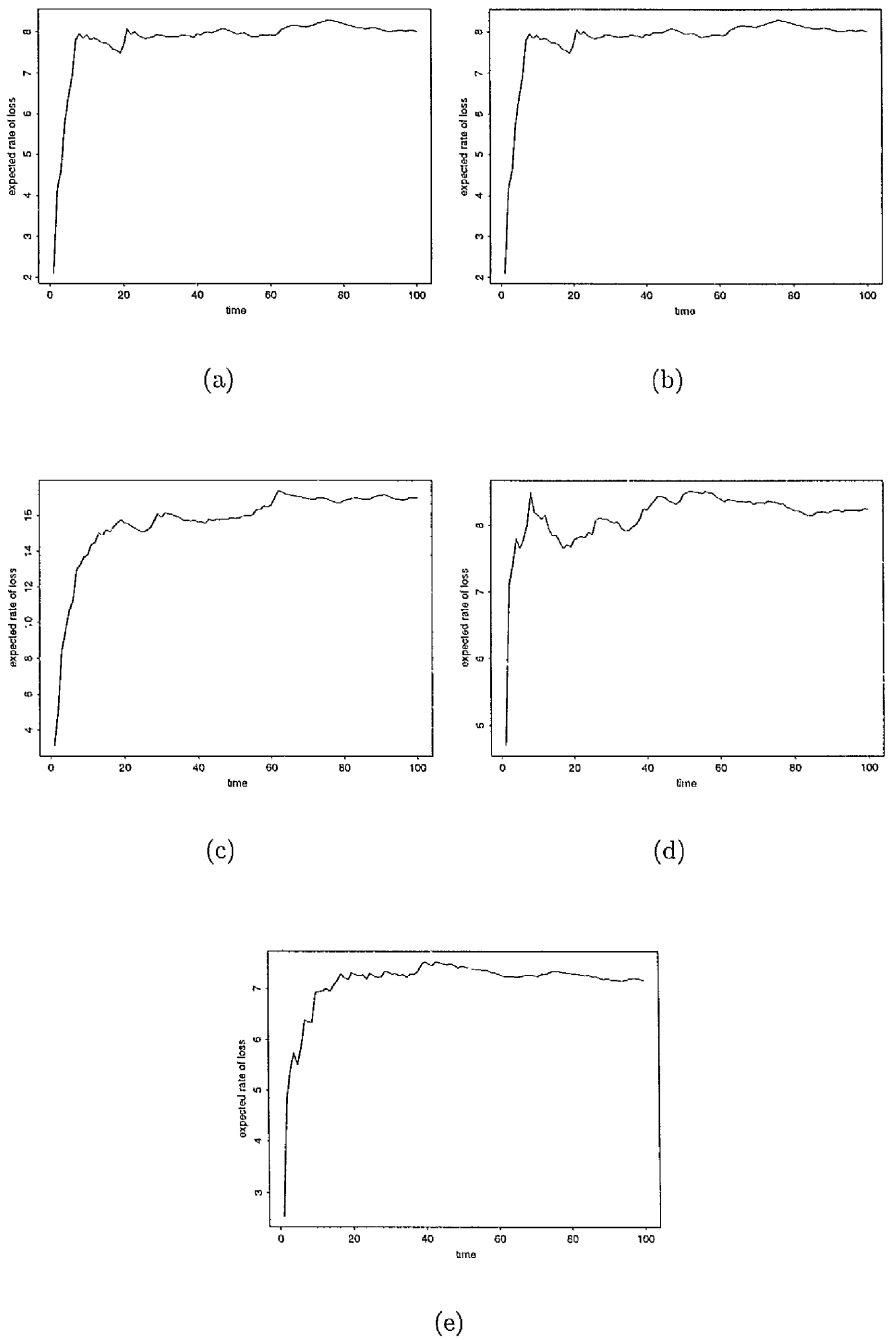


Figure 4.8: Estimated expected rates of loss from the simulations in section 4.4.4.

Chapter 5

More methods for the control problem

5.1 Introduction

The first part of the previous chapter demonstrated a well known fact about importance resampling. When the target of resampling is a posterior distribution and the importance sampler is the associated prior, the resampling weights are very sensitive to the presence of outliers in the data. The result of an outlying item of data may be a posterior sample totally unrepresentative of the distribution it is supposed to come from. If the resampling method used is weighted bootstrap, another side effect of outliers is that posterior samples may be very poor in the sense that they contain few distinct values. Recently, there have been proposed sampling methods which are designed to be immune to the presence of outliers.

A second problem concerns inference about unknown constants in dynamic model problems. As we have mentioned earlier, in such settings the effects of sample impoverishment are reversed by the system evolution equation. However, this is true only for the unknown states the system goes through. The posterior samples of unknown constants become poorer as time goes by because the system evolution equation leaves them unchanged. Methods that have appeared lately call for conditioning the analysis on these unknowns or integrating them out of the distributions involved.

The aim of this chapter is to present all these methods and to examine how they could be implemented in the control problem we have been studying. It is organised as follows. Section 5.2 presents all the methods in a general setting while section 5.3 shows how they can be adapted to the control problem. In section 5.4 the methods are compared in a simulation experiment. Section 5.5 presents two more methods for constructing good importance samplers which, we believe, are not yet suitable for dynamic model problems. Section 5.6 summarizes the chapter and gives a brief discussion. We close the chapter by re-evaluating our work on the control problem and pointing to directions for future research in section 5.7.

5.2 Presentation of the methods

In this section we present the methods in general dynamic model terms. We begin with the methods that give samplers which are not affected by outliers and then we present those that tackle the problems encountered by unknown constant quantities. It should be noted that all these methods have been developed with the weighted bootstrap in mind as the resampling technique

used. The use of smooth bootstrap would avoid some of the problems they are trying to solve, for example the problems concerning unknown constants. However, in presenting the methods we assume that resampling is implemented by weighted bootstrap.

5.2.1 Auxiliary variable particle filtering

This method was first used by Berzuini *et al* (1997) for the analysis of data arising from patient monitoring. It was independently presented in a general dynamic modelling context by Pitt and Shephard (1997), to whom its name is due. The name derives from **particle filtering**. This term, also used by Carpenter *et al* (1997), describes the approximation of the filtering distribution by a discrete probability function over a sample taken from the filtering or some other distribution. The members of the sample are called particles. What we have used in the two previous chapters is particle filtering implemented with resampling.

The notation we use is mostly the same as that we have used for dynamic models throughout this thesis. Suppose that x_t is the vector of all the unknowns at time t. This means that some of its components can be quantities that remain constant in time. After receiving data y_t and processing them we obtain a sample $x_t(1), \ldots, x_t(n)$ from the filtering distribution $p(x_t|\mathbf{y}_t)$. We approximate this distribution by a discrete one assigning probability $1/n$ to each $x_t(i), i = 1, \ldots, n$. The predictive distribution of x_{t+1} can then be approximated by

$$\hat{p}(x_{t+1}|\mathbf{y}_t) = \frac{1}{n}\sum_{i=1}^{n} p(x_{t+1}|x_t(i)). \tag{5.1}$$

After receiving y_{t+1} the next filtering distribution can be approximated by

$$\hat{p}(x_{t+1}|\mathbf{y}_{t+1}) \propto p(y_{t+1}|x_{t+1})\hat{p}(x_{t+1}|\mathbf{y}_t) \tag{5.2}$$

and the objective is to obtain a sample from it in the most efficient way. The introduction of an auxiliary variable makes this task easier. Pitt and Shephard (1997) advocate using rejection sampling, importance resampling or MCMC simulation while Berzuini *et al* (1997) use only the latter. We are going to present the resampling and MCMC implementations.

5.2.1.1 Resampling implementation

In our resampling algorithms so far we have ignored the approximations (5.1) and (5.2). Our importance sampler has been $p(x_{t+1}|\mathbf{y}_t)$ and the way to sample from it has been outlined in section 1.3.3. If we want to have (5.1) as the importance sampler we can obtain a sample $x^*_{t+1}(1), \ldots, x^*_{t+1}(n)$ from it by repeating n times the following two steps.

For obtaining $x^*_{t+1}(j), j = 1, \ldots, n$:

- Pick at random a number i between 1 and n with equal probability of selection for all i.
- Draw $x^*_{t+1}(j)$ from $p(x_{t+1}|x_t(i))$.

From (5.2) it is easy to see that the resampling weight of each point $x^*_{t+1}(i)$ is just its likelihood $p(y_{t+1}|x^*_{t+1}(i))$, i.e. the same as if the importance sampler had been $p(x_{t+1}|\mathbf{y}_t)$. Therefore both these samplers are very prone to

failure if an outlying y_{t+1} is observed. The samples resulting from resampling may contain few distinct values and may be away from the main support of $p(x_{t+1}|\mathbf{y}_{t+1})$.

A more efficient importance sampler will be one that is not affected by outliers and which still permits easy calculation of the resampling weights. This precludes $p(x_{t+1}|\mathbf{y}_{t+1})$ from being the target since

$$p(x_{t+1}|\mathbf{y}_{t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|\mathbf{y}_t)$$

and $p(x_{t+1}|\mathbf{y}_t)$ is usually not available in closed form. Formula (5.2) is not very good either because for each point $x^*_{t+1}(j)$ from the importance sampler the calculation of the weight will involve the evaluation of $p(x^*_{t+1}(j)|x_i(i))$ for $i = 1, \ldots, n$. Usually n is large and this can be very time consuming. Moreover, $p(x_{t+1}|x_t)$ may not be available in closed form either.

If $p(x_{t+1}|x_t)$ is at least available for sampling we can find an efficient importance sampler. We modify (5.2) with the introduction of an auxiliary variable k that takes values from among $1, \ldots, n$. The filtering distribution becomes

$$\hat{p}(x_{t+1}, k|\mathbf{y}_{t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t(k)). \tag{5.3}$$

Note that (5.3) approximates $p(x_{t+1}, x_t|\mathbf{y}_{t+1})$ by restricting the x_t variable to take a value from among the sample points $x_t(1), \ldots, x_t(n)$ with uniform probability. The marginal distribution of x_{t+1} is (5.2) and therefore we can sample from it by obtaining a sample from (5.3) and dropping the k part of it.

If the importance sampler is $g(x_{t+1}, k)$ and we generate a sample $(x^*_{t+1}(1), k^*_1)$, $\ldots, (x^*_{t+1}(n), k^*_n)$ from it, the weight of $(x^*_{t+1}(i), k^*_i)$ will be

$$w_i = \frac{p(y_{t+1}|x^*_{t+1}(i))p(x^*_{t+1}(i)|x_t(k^*_i))}{g(x^*_{t+1}(i), k^*_i)}.$$

If $p(x_{t+1}|x_t)$ cannot be evaluated the importance sampler has to be of the form

$$g(x_{t+1}, k) \propto g(k|\mathbf{y}_{t+1})p(x_{t+1}|x_t(k)). \tag{5.4}$$

The weight of $(x^*_{t+1}(i), k^*_i)$ will be

$$w_i = \frac{p(y_{t+1}|x^*_{t+1}(i))}{g(k^*_i|\mathbf{y}_{t+1})}.$$

Note that taking $g(k|\mathbf{y}_{t+1}) = 1/n$ for $k = 1, \ldots, n$ gives us the importance sampler that we presented at the beginning of the section. Pitt and Shephard (1997) suggest taking

$$g(k|\mathbf{y}_{t+1}) \propto p(y_{t+1}|\mu_{t+1}(k)), \tag{5.5}$$

where $\mu_{t+1}(k)$ can be any parameter of $p(x_{t+1}|x_t(k))$, or even a random draw from it as long as it depends only on $x_t(k)$. The weight of $(x^*_{t+1}(i), k^*_i)$ becomes

$$w_i = \frac{p(y_{t+1}|x^*_{t+1}(i))}{p(y_{t+1}|\mu_{t+1}(k^*_i))}.$$

To draw (x^*_{t+1}, k^*) from (5.4) when $g(k|\mathbf{y}_{t+1})$ is of form (5.5) we have to

- choose k^* with

$$\Pr(k^* = i) = \frac{p(y_{t+1}|\mu_{t+1}(i))}{\sum_{j=1}^{n} p(y_{t+1}|\mu_{t+1}(j))}, i = 1, \ldots, n,$$

- draw x^*_{t+1} from $p(x_{t+1}|x_t(k^*))$.

This choice of importance sampler makes the procedure more resistant to outliers. This is so because the $x_t(i)$ are not chosen with uniform probability but those that give high $p(y_{t+1}|\mu_{t+1}(i))$ generate x^*_{t+1} more often than the rest. This ensures that the weights are more evenly distributed among the $x^*_{t+1}(i)$ points and the sample resulting from resampling cannot usually be very poor. However, if y_{t+1} is an outlier even the "best" $x_t(i)$ point may not be good enough and then the x^*_{t+1} points may have large weights without corresponding to the main support of $p(x_{t+1}|\mathbf{y}_{t+1})$. In such a case the resulting sample will be unsatisfactory even if it contains many distinct values.

5.2.1.2 MCMC implementation

If we want we can obtain the sample from (5.3) by an MCMC sampling algorithm. We will see that the calculations required are similar to those needed for resampling.

Gibbs sampling

The procedure begins by selecting, in any way we want, a pair of initial values $x^{(0)}_{t+1}, k^{(0)}$ from the support of (5.3). We then iterate $M+N$ times the following two steps.

At iteration m:

- Obtain $k^{(m)}$ from the discrete distribution with

$$\begin{aligned} \Pr(k^{(m)} = i) &= \Pr(k^{(m)} = i | x_{t+1}^{(m-1)}, \mathbf{y}_{t+1}) \propto p(x_{t+1}^{(m-1)} | x_t(i)) \\ &= \frac{p(x_{t+1}^{(m-1)} | x_t(i))}{\sum_{j=1}^{n} p(x_{t+1}^{(m-1)} | x_t(j))}. \end{aligned} \tag{5.6}$$

- Sample $x_{t+1}^{(m)}$ from

$$p(x_{t+1} | k^{(m)}, \mathbf{y}_{t+1}) \propto p(y_{t+1} | x_{t+1}) p(x_{t+1} | x_t(k^{(m)})). \tag{5.7}$$

Sampling from (5.6) is easy but it is necessary to recalculate the probabilities at each iteration and this may prove to be time consuming for large n. Perhaps this is why Berzuini *et al* (1997) use Metropolis-within-Gibbs. In other words they simulate one transition of a Metropolis-Hastings sampler designed to sample from (5.6). At iteration m they choose a value k' uniformly among $1, \ldots, n$ and set $k^{(m)} = k'$ with probability r and $k^{(m)} = k^{(m-1)}$ with probability $1 - r$, where

$$r = \min \left\{ 1, \frac{p(x_{t+1}^{(m-1)} | x_t(k'))}{p(x_{t+1}^{(m-1)} | x_t(k^{(m-1)}))} \right\}.$$

Metropolis-within-Gibbs is probably necessary for sampling from (5.7) too.

Metropolis-Hastings sampling

In Metropolis-Hastings sampling we update both x_{t+1} and k together at each iteration. We again start with an initial pair $x_{t+1}^{(0)}, k^{(0)}$ chosen in an arbitrary way and iterate $M + N$ times the following steps.

At iteration m:

- Draw x'_{t+1}, k' from a proposal distribution $g(x_{t+1}, k)$.
- Set $(x_{t+1}^{(m)}, k^{(m)}) = (x'_{t+1}, k')$ with probability r or retain $(x_{t+1}^{(m)}, k^{(m)}) = (x_{t+1}^{(m-1)}, k^{(m-1)})$ with probability $1 - r$, where

$$r = \min\left\{1, \frac{p(y_{t+1}|x'_{t+1})p(x'_{t+1}|x_t(k'))g(x_{t+1}^{(m-1)}, k^{(m-1)})}{p(y_{t+1}|x_{t+1}^{(m-1)})p(x_{t+1}^{(m-1)}|x_t(k^{(m-1)}))g(x'_{t+1}, k')}\right\}.$$

It is clear that the ratio involved in r is the ratio of the resampling weights that (x'_{t+1}, k') and $(x_{t+1}^{(m-1)}, k^{(m-1)})$ would have if they had been drawn from importance sampler g. Therefore, if $g(x_{t+1}, k) \propto p(x_{t+1}|x_t(k))$ then

$$r = \min\left\{1, \frac{p(y_{t+1}|x'_{t+1})}{p(y_{t+1}|x_{t+1}^{(m-1)})}\right\},$$

or if, as Pitt and Shephard (1997) advocate, we take g to be given by (5.4) and (5.5) then

$$r = \min\left\{1, \frac{p(y_{t+1}|x'_{t+1})p(y_{t+1}|\mu_{t+1}(k^{(m-1)}))}{p(y_{t+1}|x_{t+1}^{(m-1)})p(y_{t+1}|\mu_{t+1}(k'))}\right\}.$$

General comments

Whichever method we use we take M large enough so that after M iterations the algorithm has converged to the target distribution. The results of these first M iterations are discarded and as our sample we keep the results of the next N (in which case $N = n$) or of every (N/n)th of them if we want to eliminate the serial correlation between consecutive draws.

The discussion about resistance against outliers in the end of the last section applies here too since the target is again (5.3). Moreover, before storing the samples we have to wait for convergence. If y_{t+1} is an outlier it is likely that only a few values from among $1, \ldots, n$ will be selected for k and we may get stuck in any of them for a long time. This means that fast convergence is not guaranteed. Finally, as we have mentioned previously the need to monitor for convergence makes the methods unsuitable for the automatic analysis of dynamic systems.

MCMC algorithms like those presented here are also described in Liu and Chen (1997).

5.2.2 Stratified particle filtering

This method has been proposed by Carpenter *et al* (1997). It is similar to auxiliary variable particle filtering and its aim is to create an importance sampler resistant to outliers. The motivation for it is entirely different however.

Expanding $\hat{p}(x_{t+1}|\mathbf{y}_t)$ in (5.2) we get

$$\hat{p}(x_{t+1}|\mathbf{y}_{t+1}) \propto \sum_{i=1}^{n} p(y_{t+1}|x_{t+1}) p(x_{t+1}|x_t(i)) \tag{5.8}$$

and we can easily see that

$$\hat{p}(x_{t+1}|\mathbf{y}_{t+1}) = \sum_{i=1}^{n} \frac{\int p(y_{t+1}|x_{t+1}) p(x_{t+1}|x_t(i)) dx_{t+1}}{\sum_{j=1}^{n} \int p(y_{t+1}|x_{t+1}) p(x_{t+1}|x_t(j)) dx_{t+1}}$$

$$\cdot \quad \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t(i))}{\int p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t(i))dx_{t+1}}$$

$$= \sum_{i=1}^{n} \beta_i p_i(x_{t+1}). \tag{5.9}$$

This is a mixture of densities with unequal mixing weights. The weights β_i reflect the different "predictive likelihood" of each point $x_t(i)$. If we assume that the variance of x_{t+1} is the same under each p_i the most efficient way to sample n values from (5.9) is to sample $n_i = n\beta_i$ values from each component p_i. Unfortunately, usually neither p_i nor β_i is available in closed form.

Carpenter *et al* (1997) suggest taking an importance sampler

$$g(x_{t+1}) = \sum_{i=1}^{n} \hat{\beta}_i \hat{p}_i(x_{t+1}), \tag{5.10}$$

where $\hat{\beta}_i$ and $\hat{p}_i$ are approximations to β_i and p_i respectively and $\sum_{i=1}^{n} \hat{\beta}_i = 1$. Moreover $\hat{p}_i$ are such that we can easily sample from them. From each $\hat{p}_i$ we draw $\hat{n}_i = n\hat{\beta}_i$ points x^*_{t+1} and the weight of point $x^*_{t+1}(j)$ is

$$w_j = \frac{p(y_{t+1}|x^*_{t+1}(j))p(x^*_{t+1}(j)|x_t(i))}{\hat{\beta}_i \hat{p}_i(x^*_{t+1}(j))} \tag{5.11}$$

if $x^*_{t+1}(j)$ has been drawn from $\hat{p}_i$. The form of the weight can be explained if we imagine an auxiliary variable k taking values in $1, \ldots, n$. Then (5.8) is replaced by

$$\hat{p}(x_{t+1}, k|\mathbf{y}_{t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t(k)), \tag{5.12}$$

which has (5.8) as the marginal distribution of x_{t+1}, and (5.10) is replaced by

$$g(x_{t+1}, k) = \hat{\beta}_k \hat{p}_k(x_{t+1}). \tag{5.13}$$

Then (5.11) is the appropriate weight when the target is (5.12) and the importance sampler is (5.13). Of course, the x^*_{t+1} are not obtained exactly from (5.13) since each $\hat{p}_i$ is sampled a fixed number of times. However, we have been doing the same thing in all previous chapters, where our importance sampler has effectively been (5.13) with $\hat{\beta}_i = 1/n$ for all i and $\hat{p}_i = p(x_{t+1}|x_t(i))$ and each $\hat{p}_i$ has been sampled once.

It is very unlikely that $\hat{n}_i = n\hat{\beta}_i$ will be an integer quantity for all i. For this reason Carpenter *et al* (1997) have developed an algorithm that generates integers $n_1, \ldots, n_n$ such that

$$\sum_{i=1}^{n} n_i = n \quad , \quad E(n_i) = n\hat{\beta}_i \quad \text{and} \quad |n_i - n\hat{\beta}_i| \leq 1 \quad \text{for all} \quad i.$$

As with auxiliary variable particle filtering, stratified particle filtering can also be implemented with MCMC sampling but we do not pursue this further.

Stratified importance sampling is robust against outliers but like the previously mentioned methods, it does not eliminate the possibility of producing a sample that is far from the main support of the distribution it is supposed to represent. In Carpenter *et al* (1997) the weights are accumulated from time point to time point and resampling is not performed at all. This requires a straightforward modification of (5.8),(5.9) and (5.11) to accommodate for the fact that now $\hat{p}(x_t(i)|\mathbf{y}_t)$ is not $1/n$ for all i. At each time point the

current weights will reflect posterior probability and will counterbalance the likelihood. However, when some weights, as time goes by, become very small resampling will have to be performed. It is plausible that the effect of outliers will be less severe but it will not disappear.

It should be noted that none of the methods presented thus far offers any improvement as far as the posterior samples of unknowns that remain constant in time are concerned. This is the objective of the methods that we are going to present now.

5.2.3 Stratification

Now we make a distinction between the unknowns that change value in time and those that do not. We denote the former by x_t and the latter by θ. The filtering density at time t is $p(x_t, \theta|\mathbf{y}_t)$. Any of the methods mentioned so far can be used to get samples from it. The problem is that each time we move to a new filtering distribution the sample from it contains fewer distinct θ values than the previous one. Stratification was proposed in Pitt and Shephard (1997) as a way of dealing with this problem.

To implement the method we draw a sample $\theta_1, \ldots, \theta_m$ from $p(\theta)$, the prior distribution of θ. Then for each θ_i separately we start with a sample from $p(x_1|\theta_i)$ and as observations y_t come along we update it with any of the methods mentioned so far, so that, after having observed y_t, for each θ_i we have $x_t(i,1), \ldots, x_t(i,n)$ from $p(x_t|\mathbf{y}_t, \theta_i)$. These samples can be used to draw conclusions about x_t and θ. We propose some ways of doing this.

If we need inference for $p(\theta|\mathbf{y}_t)$ we can use importance sampling ideas. We have $\theta_1, \ldots, \theta_m$ from $p(\theta)$ and the target is $p(\theta|\mathbf{y}_t) \propto p(\mathbf{y}_t|\theta)p(\theta)$. Therefore,

each θ_i receives importance weight

$$w_{\theta,t}^{(i)} = p(\mathbf{y}_t|\theta_i) = p(y_1|\theta_i) \prod_{j=2}^{t} p(y_j|\mathbf{y}_{j-1}, \theta_i).$$

We do not usually have $p(y_t|\mathbf{y}_{t-1}, \theta)$ in closed form but we can approximate it as part of the sampling process. Note that

$$p(y_t|\mathbf{y}_{t-1}, \theta_i) = \int p(y_t|x_t, \theta_i) p(x_t|\mathbf{y}_{t-1}, \theta_i) dx_t.$$

We have a sample $x_t^*(i,1), \ldots, x_t^*(i,n)$ from $p(x_t|\mathbf{y}_{t-1}, \theta_i)$ as a by-product of the sampling process or we can easily obtain one. Then

$$\hat{p}(y_t|\mathbf{y}_{t-1}, \theta_i) = \frac{1}{n} \sum_{j=1}^{n} p(y_t|x_t^*(i,j), \theta_i).$$

This assumes that we have $p(y_t|x_t, \theta)$ in closed form. Then $p(\theta|\mathbf{y}_t)$ can be approximated by the discrete distribution $\hat{p}(\theta|\mathbf{y}_t)$ over $\theta_1, \ldots, \theta_m$ with probability function

$$\Pr(\theta = \theta_i|\mathbf{y}_t) = \frac{w_{\theta,t}^{(i)}}{\sum_{j=1}^{m} w_{\theta,t}^{(j)}} = \pi_{\theta,t}^{(i)}.$$

If we want inference about $p(x_t|\mathbf{y}_t)$ we have, among others, the following options.

a) We already have a sample of x_t values but it is not a random sample from $p(x_t|\mathbf{y}_t)$ because of the weighting due to θ_i. However, $p(x_t|\mathbf{y}_t) =$

$\int p(x_t|\theta, \mathbf{y}_t)p(\theta|\mathbf{y}_t)d\theta$. We can get a random sample from $p(x_t|\mathbf{y}_t)$ by repeating the two following steps as many times as the desired sample size:

- Generate $\theta = \theta_k$ with probability $\pi_{\theta,t}^{(k)}, k = 1, 2, \ldots, m$. This approximates a draw from $p(\theta|\mathbf{y}_t)$.
- Sample a point from among $x_t(k,1), \ldots, x_t(k,n)$ with equal probability of selection for all points. This approximates a draw from $p(x_t|\theta, \mathbf{y}_t)$.

b) If we want to estimate an expectation, $E\left(h(x_t)|\mathbf{y}_t\right)$ say, we observe that

$$\begin{aligned} E\left(h(x_t)|\mathbf{y}_t\right) &= E\left[E\left(h(x_t)|\theta, \mathbf{y}_t\right)|\mathbf{y}_t\right] = \int \left(\int h(x_t)p(x_t|\theta, \mathbf{y}_t)dx_t\right) p(\theta|\mathbf{y}_t)d\theta \\ &= \int h(\theta, \mathbf{y}_t)p(\theta|\mathbf{y}_t)d\theta = E\left[h(\theta, \mathbf{y}_t)|\mathbf{y}_t\right]. \end{aligned}$$

We can now estimate $h(\theta, \mathbf{y}_t)$ and $E\left(h(x_t)|\mathbf{y}_t\right)$ from the available samples by

$$\hat{h}(\theta_i, \mathbf{y}_t) = \frac{1}{n}\sum_{j=1}^{n} h(x_t(i,j)) \quad , \tag{5.14}$$

$$\hat{E}\left(h(x_t)|\mathbf{y}_t\right) = \sum_{i=1}^{m} \pi_{\theta,t}^{(i)} \hat{h}(\theta_i, \mathbf{y}_t). \tag{5.15}$$

However, as time goes by and we obtain more observations y_t the weights $w_{\theta,t}^{(i)}$ for most of the θ_i become very small so it would be better perhaps to resample from time to time from among them in order to get a sample, from $p(\theta|\mathbf{y}_t)$, in which all its members have the same weight. We suggest doing this by sampling with replacement from among $\theta_1, \ldots, \theta_m$ with probabilities $\pi_{\theta,t}^{(1)}, \ldots, \pi_{\theta,t}^{(m)}$

respectively. Each θ_i is accompanied by a sample $\mathbf{x}_t^{(i)} = (x_t(i,1), \ldots, x_t(i,n))$, and when this θ_i is selected its accompanying sample stays with it. The justification for this is as follows.

We want samples from $p(\mathbf{x}_t, \theta|\mathbf{y}_t)$ and we have samples $\mathbf{x}_t^{(i)}, \theta_i$ from an importance sampler defined by $p(\theta)p(\mathbf{x}_t|\theta, \mathbf{y}_t)$. If we want to resample from among them the probabilities of selection have to be proportional to

$$w^{(i)} = \frac{p(\mathbf{x}_t^{(i)}, \theta_i|\mathbf{y}_t)}{p(\theta_i)p(\mathbf{x}_t^{(i)}|\theta_i, \mathbf{y}_t)} = \frac{p(\theta_i|\mathbf{y}_t)}{p(\theta_i)} \propto p(\mathbf{y}_t|\theta_i) \approx w_{\theta,t}^{(i)}.$$

This resampling may generate a poor sample for θ, creating the problem that we are trying to avoid. However, as with stratified particle filtering the effects of outliers may be less strong than if resampling was applied at each time point. Stratification is also very expensive in terms of storage requirements because usually m will have to be quite large for the state space of θ to be covered adequately and each θ_i will be accompanied by n values of x_t.

5.2.4 Rao-Blackwellization

Rao-Blackwellization is another way of dealing with unknown parameters that remain constant in time. If the filtering distribution is $p(x_t, \theta|\mathbf{y}_t)$, it is better, if possible, to integrate θ out and work with $p(x_t|\mathbf{y}_t)$. As we have said, an indicator of good performance of importance sampling is the variance of the weights. If $g(x_t, \theta)$ is the importance sampler used for $p(x_t, \theta|\mathbf{y}_t)$ and $g(x_t)$ is the marginal density of x_t obtained from g, it is true (Doucet (1998)) that

$$\mathrm{var}_{g(x_t)}\left(\frac{p(x_t|\mathbf{y}_t)}{g(x_t)}\right) \leq \mathrm{var}_{g(x_t,\theta)}\left(\frac{p(x_t,\theta|\mathbf{y}_t)}{g(x_t,\theta)}\right).$$

However, integrating θ out is not always easy.

Another form of Rao-Blackwellization is concerned with the estimation of $p(\theta|\mathbf{y}_t)$ when a sample $(x_t(1),\theta_1),\ldots,(x_t(n),\theta_n)$ from $p(x_t,\theta|\mathbf{y}_t)$ is available. The usual estimator would be the histogram of $\theta_1,\ldots,\theta_n$. The Rao-Blackwellized estimator takes advantage of the knowledge of the relationship between x_t and θ and the availability of the samples $x_t(1),\ldots,x_t(n)$, and has the form

$$\hat{p}(\theta|\mathbf{y}_t) = \frac{1}{n}\sum_{i=1}^{n} p(\theta|x_t(i),\mathbf{y}_t). \tag{5.16}$$

This implies that $p(\theta|x_t,\mathbf{y}_t)$ is available in closed form.

Finally, Rao-Blackwellization is a method of enrichment of a sample $(x_t(1),\theta_1)$, $\ldots,(x_t(n),\theta_n)$ from $p(x_t,\theta|\mathbf{y}_t)$ which is very poor in terms of θ. This method is presented in Liu and Chen (1998) and McEachern et al (1998). It is argued that since the sample already comes from $p(x_t,\theta|\mathbf{y}_t)$ it will not matter if each θ_i is replaced by a point coming from $p(\theta|x_t(i),\mathbf{y}_t)$. If this is not available for sampling we can replace each θ_i by a draw from the transition kernel of a Markov chain which has $p(\theta|x_t(i),\mathbf{y}_t)$ as its invariant distribution. In fact each θ_i can be replaced by more than one draw, $\theta_{i,1},\ldots,\theta_{i,k}$, for example. Each one will form a pair with a copy of $x_t(i)$. Moreover we can use Rao-Blackwellization before resampling. If we have $(x_t^*(1),\theta_1^*),\ldots,(x_t^*(n),\theta_n^*)$ from an importance sampler we can replace each pair $(x_t^*(i),\theta_i^*)$ by pairs $(x_t^*(i),\theta_{i,1}^*),\ldots,(x_t^*(i),\theta_{i,k}^*)$ with $\theta_{i,1}^*,\ldots,\theta_{i,k}^*$ drawn from $p(\theta|x_t^*(i),\mathbf{y}_t)$ or from

an appropriate Markov chain. Each of the new pairs will receive the resampling weight that $(x_t^*(i), \theta_i^*)$ had, and resampling can go on as usual. McEachern et al (1998) stress that the time points t at which we employ Rao-Blackwellization must have been decided in advance otherwise the sampler will be illegitimate.

It may not be easy to apply Rao-Blackwellization in practice. Otherwise, on its own or in combination with stratification it would be a very useful tool for providing inference about unknown constants.

5.3 Application of the methods to the control problem

In this section we examine the applicability of the methods we have discussed to the control problem that we have been studying. The notation used for the problem parameters is the same as that in the previous two chapters.

5.3.1 Auxiliary variable particle filtering

5.3.1.1 Resampling implementation

At first we concentrate on the univariate case and we assume that at any time point t the only unknown is the minimum x_t of the curve. The importance sampler we have been using, put into an auxiliary variable context, takes the form

$$g(x_{t+1}, k) \propto p(x_{t+1}|x_t(k)) \propto \exp\left[-\frac{1}{2\sigma_\eta^2}(x_{t+1} - x_t(k))^2\right]. \tag{5.17}$$

If $y_{t+1} > 0$ and if $u_{t+1} - \sqrt{2y_{t+1}}$ and $u_{t+1} + \sqrt{2y_{t+1}}$, the maximum likelihood estimates (MLE's) of x_{t+1}, are outside the range of the x_{t+1} part of the sample drawn from (5.17) we have considered y_{t+1} to be an outlier. When y_{t+1} is an outlier we have used intervention because resampling would have given very poor samples consisting of very few distinct values and possibly being distant from the main support of $p(x_{t+1}|\mathbf{y}_{t+1}, \mathbf{u}_{t+1})$. Instead of intervention we could have used an importance sampler of the form

$$g(x_{t+1}, k) \propto p(y_{t+1}|\mu_{t+1}(k))p(x_{t+1}|x_t(k))$$

with $\mu_{t+1}(k) = x_t(k)$ for example. However, we have already said that such samplers only enrich posterior samples but do not ensure that they will fall in the main support of $p(x_{t+1}|\mathbf{y}_{t+1}, \mathbf{u}_{t+1})$ if y_{t+1} is an outlier. We prefer to use a sampler like (5.17) but with an increased variance when necessary, so that among the x_{t+1}^* points it produces there are always some close to the MLE's.

We have arrived at the following approach. The distance between the MLE's, provided that $y_{t+1} > 0$, is $2\sqrt{2y_{t+1}}$. A Normal distribution with mean $x_t(k)$ which lies between the MLE's needs standard deviation σ such that $2\sigma > \sqrt{2y_{t+1}}$, so that it has a good chance of generating values close to at least one of the MLE's. We take an importance sampler of the form

$$g(x_{t+1}, k|\mathbf{y}_{t+1}, \mathbf{u}_{t+1}) \propto \exp\left[-\frac{1}{2\sigma^2}(x_{t+1} - x_t(k))^2\right] \tag{5.18}$$

with

$$\sigma^2 = \max\left\{\sigma_\eta^2, \frac{2y_{t+1}}{1.5^2}\right\}. \tag{5.19}$$

This means that if σ_η is not large enough (5.17) "stretches" itself out to cover the support of $p(x_{t+1}|\mathbf{y}_{t+1}, \mathbf{u}_{t+1})$. It is easy to see that the resampling weight of a point $(x^*_{t+1}(i), k^*_i)$ drawn from (5.18) will be

$$\begin{aligned} w_i &= \exp\left[-\frac{1}{2\sigma_\epsilon^2}\left(y_{t+1} - \frac{1}{2}(x^*_{t+1}(i) - u_{t+1})^2\right)^2\right] \\ &\cdot \exp\left[\left(\frac{1}{2\sigma^2} - \frac{1}{2\sigma_\eta^2}\right)(x^*_{t+1}(i) - x_t(k^*_i))^2\right]. \end{aligned}$$

The second exponential becomes 1 if $\sigma^2 = \sigma_\eta^2$. The denominator of σ^2 in (5.19) must be smaller than 4, but apart from that its choice is arbitrary. Note that with such an importance sampler we do not need intervention.

In exactly the same way we can devise an importance sampler for the multivariate case, multiplying the disturbance variance matrix Σ_η by an appropriate factor when necessary.

When there are additional unknowns as well as x_t it may not be possible to take such an approach. For example, to return to a case from the previous chapter, if the observation equation has the form $y_t = \alpha(x_t - u_t)^2 + \epsilon_t$ and α is unknown there are infinite MLE's and no way of finding an importance sampler that will generate points close to each one of them.

5.3.1.2 MCMC implementation

Both Gibbs and Metropolis-Hastings sampling are very straightforward to apply but as we have said Gibbs sampling can be very time consuming. Therefore we concentrate our attention on Metropolis-Hastings sampling only. In the univariate case when only x_t is unknown we can use (5.18) with variance given by (5.19) as the proposal density. As we have seen the acceptance ratio will be equal to the ratio of the weights that the proposed and the current point would have if (5.18) were used as an importance sampler.

If x_t is not the only unknown then, as it is not easy to find an effective importance sampler, neither is it easy to find a good proposal density.

5.3.2 Stratified particle filtering

We again start with the univariate case and with x_t being the only unknown. The filtering density can be written, as we have seen, as a mixture

$$\begin{aligned} p(x_{t+1}|\mathbf{y}_{t+1}, \mathbf{u}_{t+1}) &= \sum_{i=1}^{n} \beta_i p_i(x_{t+1}) \\ &= \sum_{i=1}^{n} \frac{\int p(y_{t+1}|x_{t+1}, u_{t+1}) p(x_{t+1}|x_t(i)) dx_{t+1}}{\sum_{j=1}^{n} \int p(y_{t+1}|x_{t+1}, u_{t+1}) p(x_{t+1}|x_t(j)) dx_{t+1}} \\ &\quad \cdot \frac{p(y_{t+1}|x_{t+1}, u_{t+1}) p(x_{t+1}|x_t(i))}{\int p(y_{t+1}|x_{t+1}, u_{t+1}) p(x_{t+1}|x_t(i)) dx_{t+1}}, \end{aligned}$$

where $x_t(1), \ldots, x_t(n)$ is a sample from $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$. We use simulation to estimate β_i since the integrals are analytically intractable because of the non-linearity of the observation equation. For each $x_t(i)$ we draw m points

$\hat{x}_{t+1}(i,j)$ from $p(x_{t+1}|x_t(i))$. Then

$$\int p(y_{t+1}|x_{t+1},u_{t+1})p(x_{t+1}|x_t(i))dx_{t+1} \simeq \frac{1}{m}\sum_{j=1}^{m} p(y_{t+1}|\hat{x}_{t+1}(i,j),u_{t+1})$$

and

$$\hat{\beta}_i = \frac{\sum_{j=1}^{m} p(y_{t+1}|\hat{x}_{t+1}(i,j),u_{t+1})}{\sum_{k=1}^{n}\sum_{j=1}^{m} p(y_{t+1}|\hat{x}_{t+1}(k,j),u_{t+1})}.$$

As for $p_i(x_{t+1}) \propto p(y_{t+1}|x_{t+1},u_{t+1})p(x_{t+1}|x_t(i))$ we simply take $\hat{p}_i(x_{t+1}) \propto p(x_{t+1}|x_t(i))$. The resampling weight of a point $x^*_{t+1}(j)$ generated from $\hat{p}_i(x_{t+1})$ will be

$$w_j = \frac{p(y_{t+1}|x^*_{t+1}(j),u_{t+1})}{\hat{\beta}_i}.$$

Of course we could use other forms of $\hat{p}_i$. For example we could have

$$\hat{p}_i(x_{t+1}) \propto \exp\left[-\frac{1}{2\sigma^2}(x_{t+1}-x_t(i))^2\right] \tag{5.20}$$

with σ^2 given by (5.19).

A similar approach can be taken for a multivariate setting or when there are more unknowns. If for example the precisions ϕ_ϵ and ϕ_η are also unknown and $(x_t(1),\phi_\eta(1),\phi_\epsilon(1)),\ldots,(x_t(n),\phi_\eta(n),\phi_\epsilon(n))$ is a sample from $p(x_t,\phi_\eta,\phi_\epsilon|\mathbf{y}_t,\mathbf{u}_t)$ we can estimate $\hat{\beta}_i$ as

$$\hat{\beta}_i = \frac{\sum_{j=1}^{m} p(y_{t+1}|\hat{x}_{t+1}(i,j), \phi_\epsilon(i), u_{t+1})}{\sum_{k=1}^{n}\sum_{j=1}^{m} p(y_{t+1}|\hat{x}_{t+1}(k,j), \phi_\epsilon(k), u_{t+1})},$$

with $\hat{x}_{t+1}(i,1), \ldots, \hat{x}_{t+1}(i,m)$ being random draws from $p(x_{t+1}|x_t(i), \phi_\eta(i))$, and we can use $p(x_{t+1}|x_t(i), \phi_\eta(i))$ as $\hat{p}_i(x_{t+1})$.

Stratified particle filtering can be combined with intervention. When we observe y_{t+1} we generate a temporary prior sample from $p(x_{t+1}|\mathbf{y}_t, \mathbf{u}_t)$. If the MLE's of x_{t+1} are outside this sample's range we employ intervention as outlined in Chapter 3, otherwise we discard the prior sample and apply stratified particle filtering. Of course, if $\hat{p}_i$ is of form (5.20) this process is not required.

5.3.3 Stratification

Stratification is also very straightforward to apply. Suppose that as well as x_t we also do not know ϕ_1, ϕ_η and ϕ_ϵ. According to the notation of section 5.2.3, $\theta = (\phi_\eta, \phi_\epsilon)$, although for the first two time points θ also includes ϕ_1 which later on is dropped. From the prior distribution $p(\phi_1, \phi_\eta, \phi_\epsilon)$ we draw m points $(\phi_1(1), \phi_\eta(1), \phi_\epsilon(1)), \ldots, (\phi_1(m), \phi_\eta(m), \phi_\epsilon(m))$, and then for each $(\phi_1(i), \phi_\eta(i), \phi_\epsilon(i))$ we draw $(x_1^*(i,1), x_2^*(i,1)), \ldots, (x_1^*(i,n), x_2^*(i,n))$ from $p(x_1, x_2|\phi_1(i), \phi_\eta(i), \phi_\epsilon(i))$. Subsequently the analysis proceeds as usual for each $(\phi_1(i), \phi_\eta(i), \phi_\epsilon(i))$ separately.

At any time point t the estimates of the conditional likelihoods are

$$\hat{p}(y_t|\phi_\eta(i), \phi_\epsilon(i), \mathbf{y}_{t-1}) = \frac{1}{n}\sum_{j=1}^{n} p(y_t|x_t^*(i,j), \phi_\epsilon(i), u_t),$$

where $x_t^*(i,1),\ldots,x_t^*(i,n)$ is a sample from $p(x_t|\mathbf{y}_{t-1},\mathbf{u}_{t-1},\phi_\eta(i),\phi_\epsilon(i))$.

In order to choose u_{t+1} we need an estimate of the mode of $p(x_t|\mathbf{y}_t,\mathbf{u}_t)$. Therefore we need a sample from it and we can obtain it in the way outlined in section 5.2.3.

Although in theory the method can be applied we believe that it is impractical because we will have to generate a large number m of points from the initially three-dimensional state space of θ and for each one of them a large number n of x_t samples. For this reason we do not pursue it in the simulations section.

5.3.4 Rao-Blackwellization

It is very difficult to employ Rao-Blackwellization, in any of the forms mentioned in the previous section, in our control problem. The reason is twofold.

First, it is not possible to integrate θ out of the filtering distribution because all the densities involved are non-linear with respect to it.

Secondly, we cannot even obtain $p(\theta|x_t,\mathbf{y}_t)$ in closed form because the calculations come to grief if we approximate the filtering densities by discrete densities based on samples. To see this more clearly consider the following:

$$\begin{aligned} p(\theta|x_t,\mathbf{y}_t) &\propto p(\theta,x_t|\mathbf{y}_t) \propto p(y_t|\theta,x_t)p(\theta,x_t|\mathbf{y}_{t-1}) \\ &= p(y_t|\theta,x_t)p(x_t|\theta,\mathbf{y}_{t-1})p(\theta|\mathbf{y}_{t-1}). \end{aligned} \tag{5.21}$$

If $(\theta_1,x_{t-1}(1)),\ldots,(\theta_n,x_{t-1}(n))$, is a sample from $p(\theta,x_{t-1}|\mathbf{y}_{t-1})$ then $p(\theta|\mathbf{y}_{t-1})$

is approximated by a discrete density over the θ part of that sample because it is not available in closed form. This means that any kernel used to enrich the θ part of the sample from $p(\theta, x_t|\mathbf{y}_t)$ will have to have the same discrete state space that gave the poor sample in the first place. Alternatively, we can replace $p(\theta|\mathbf{y}_{t-1})$ in (5.21) with a kernel density estimate $\hat{p}(\theta|\mathbf{y}_{t-1})$ based on the θ part of the sample from $p(\theta, x_{t-1}|\mathbf{y}_{t-1})$. Because of the non-linearities in the problem it is very likely that Metropolis-Hastings sampling will have to be used in order to sample from (5.21). This will require the evaluation of $\hat{p}(\theta|\mathbf{y}_{t-1})$ for every θ value considered for acceptance and can be very time consuming. As with stratification we will not pursue Rao-Blackwellization in the simulations section.

Furthermore, since $p(\theta|x_t, \mathbf{y}_t)$ is unavailable it is not possible to use an estimator like (5.16).

5.4 Simulations

In this section we examine how well some of the new methods perform in the analysis of the control problem and we compare them with the resampling algorithm we have used in the two previous chapters.

5.4.1 General comments

The set-up is the same as in Chapter 3. For the very first minimum, x_1, we assume that $x_1 \sim N(0, \sigma_1^2)$ and for the movement of the curve we assume that it is such that $[x_t|x_{t-1}] \sim N(x_{t-1}, \sigma_\eta^2)$. The observation satisfies $[y_t|x_t, u_t] \sim N((x_t - u_t)^2/2, \sigma_\epsilon^2)$. The methods are compared under five different scenarios,

each one defined by a combination of variance values:

- $\sigma_1^2 = \sigma_\eta^2 = 1, \sigma_\epsilon^2 = 0.01$
- $\sigma_1^2 = \sigma_\eta^2 = \sigma_\epsilon^2 = 1$
- $\sigma_1^2 = \sigma_\eta^2 = 1, \sigma_\epsilon^2 = 4$
- $\sigma_1^2 = \sigma_\eta^2 = 4, \sigma_\epsilon^2 = 1$
- $\sigma_1^2 = \sigma_\eta^2 = \sigma_\epsilon^2 = 4.$

Under each scenario we have generated 50 chains of true x_t for $t = 1, 2, \ldots, 100$. These are the ones also used in Chapter 3. Each method produced 50 chains of control points u_t and thus for each method at any time point $N \leq 100$ we have a simulated expected rate of loss

$$\hat{\gamma}_N = \frac{\sum_{t=1}^{N} \sum_{i=1}^{50} (x_{ti} - u_{ti})^2}{50 \cdot N},$$

where x_{ti} and u_{ti} are the locations of the minimum and the control point for chain i at time t.

We also discriminate between two cases, one in which the variances are known and one in which they are all unknown. When they are known the new methods we consider are auxiliary variable particle filtering in both resampling and MCMC implementations and stratified particle filtering. When the variances are unknown we consider stratified particle filtering and an MCMC implementation of the ordinary particle filtering we have used in the previous chapters. The methods were used in order to provide samples necessary for the application of our adaptation of Titterington's method with $K = 0$. For

comparison we also present for each scenario the results of our adaptation of Titterington's method with $K = 0$ from Chapter 3. There the method was implemented with resampling. All resampling required by the methods was performed with smooth bootstrap.

We now present the implementational details for each method.

5.4.2 Known variances

Auxiliary variable particle filtering with resampling was implemented with importance sampler (5.18) and variance (5.19).

For the MCMC implementation we considered two different proposal densities, one for sampling from $p(x_1|y_1, u_1)$ and one for any of the subsequent filtering densities. For $p(x_1|y_1, u_1)$ we did not use auxiliary variables, and the proposal density g at iteration m of the algorithm was given by

$$[x_1'|x_1^{(m-1)}] \sim N(x_1^{(m-1)}, \sigma_1^2).$$

For the subsequent filtering densities, $p(x_t|\mathbf{y}_t, \mathbf{u}_t)$, the proposal density was (5.18) with variance (5.19). Plots of the generated values revealed that the Markov chains converge quite quickly and that autocorrelations become very small very quickly. We took a "burn-in" period of length $M = 5000$ and from the subsequent 30000 iterations we stored the output of every thirtieth.

Finally, stratified particle filtering was employed as outlined in section 5.3.2 with $m = 10$ and combined with intervention. All the samples from prior and posterior densities had size $n = 1000$. For all the methods we took $u_1 = 0$,

the mean of the prior of x_1.

Figure 5.1 presents the simulated expected rates of loss for all the methods. We see that the system is controllable whichever method we use. We also see that none of them performs clearly better than the rest. If we compare them on grounds of practicality the ordinary particle filtering of Chapter 3 is the simplest to use while the MCMC implementation of auxiliary variable particle filtering is the most awkward.

5.4.3 Unknown variances

In this setting we do not use auxiliary variable particle filtering because it is difficult, if not impossible, to find an importance sampler or proposal density that will efficiently cover the whole support of the filtering distribution. For all the methods that we present the first two observations are considered as a block so that we will get a good sample for ϕ_η. Therefore, the first filtering density is

$$p(x_1, x_2, \phi_1, \phi_\eta, \phi_\epsilon | \mathbf{y}_2, \mathbf{u}_2).$$

We consider the precisions to be independent *a priori* and their priors are taken to be Ga(0.1,0.1) unless $\phi_\epsilon = 100$, in which case its prior is taken to be Ga(1000,10). The prior of x_1 given the precisions is $N(0, \phi_1^{-1})$ and for x_2 given the precisions and x_1 is $N(x_1, \phi_\eta^{-1})$.

For demonstrative reasons we combine the simple particle filtering of Chapter 3 with MCMC sampling. As in the case of known variances we had two

different proposal distributions, one for the first filtering distribution and one for the subsequent ones. When we worked with the first filtering distribution the starting values of the precisions were taken at random from a Uniform density on the interval $(0, 3)$ except that when $\phi_\epsilon = 100$ its starting value was taken from a Uniform distribution on $(90, 110)$. This was done because when we tried drawing starting values from the priors sometimes they were so extreme that the Markov chain would get indefinitely stuck far from the main body of the posterior otherwise there would be overflow problems. Given starting values for the precisions the starting values for x_1 and x_2 were drawn from the appropriate prior distributions. The proposal densities for each parameter at iteration m were

$$\begin{aligned} x_1' &\sim N(x_1^{(m-1)}, 0.5) \\ x_2' &\sim N(x_2^{(m-1)}, 0.5) \\ \log(\phi_1') &\sim N(\log(\phi_1^{(m-1)}), 0.3) \\ \log(\phi_\eta') &\sim N(\log(\phi_\eta^{(m-1)}), 0.3) \\ \log(\phi_\epsilon') &\sim N(\log(\phi_\epsilon^{(m-1)}), 0.3). \end{aligned}$$

For each of the subsequent filtering distributions the proposal density was similar to (5.17). An index k' between 1 and n was chosen at random with equal probabilities of selection for all possible values, and then x_{t+1}' was drawn from $p(x_{t+1}|x_t(k'), \phi_\eta(k'))$ where $(x_t(1), \phi_\eta(1)), \ldots, (x_t(n), \phi_\eta(n))$ was a sample from $p(x_t, \phi_\eta|\mathbf{y}_t, \mathbf{u}_t)$. Graphs of the generated values showed that the Markov chains converge quite fast and the autocorrelations in the samples diminish quickly. However, because this is a multivariate problem we took

a "burn-in" period of length $M = 10000$ and from the next 90000 iterations we stored the output of every ninetieth. Sometimes very small acceptance probabilities (between 1% and 10%) were reported but, as the graphs show, this did not affect the performance of the method. Because the acceptance ratio corresponds to a ratio of weights, small observed acceptance probabilities reflect cases where most of the resampling weights would have been very small and the posterior sample, had resampling been applied, would consist of very few distinct values.

Note that the samples for ϕ_η and ϕ_ϵ become poor as time goes by and, for this reason, after we had obtained $(x_t(1), \phi_\eta(1), \phi_\epsilon(1)), \ldots, (x_t(n), \phi_\eta(n), \phi_\epsilon(n))$ from $p(x_t, \phi_\eta, \phi_\epsilon | \mathbf{y}_t, \mathbf{u}_t)$ we augmented the sample, combining the method of section 2.4.1 with West's correction from section 2.2.

Stratified particle filtering was applied as outlined in section 5.3.2 with $m = 10$ and combined with intervention. For all methods all samples from prior and filtering distributions had size $n = 1000$. The only exception was in the two resampling based methods where the sample from the very first prior had size 10000. For all the methods we took $u_1 = u_2 = 0$, the mean of the prior of x_1.

Figure 5.2 presents the simulated expected rates of loss for all the methods. As in the case of known variances they all perform well with none standing out. As for the issue of practicality the comments of the previous section apply here too.

5.5 Two non-dynamic methods

We now present two other methods which were devised for use in static contexts. They both aim at finding a good importance sampler for any given target density. As we will see they are not well suited for use in the analysis of dynamic systems. In the presentation we follow the references in which they appeared.

5.5.1 Annealed importance sampling

Annealed importance sampling was presented in Neal (1998). Suppose that the target density is $p_0(x) \propto f_0(x)$, that we can compute $f_0(x)$ for any x but that we cannot generate samples from it. On the other hand we have another density $p_m(x) \propto f_m(x)$ from which we can draw random samples. We can also compute $f_m(x)$ at any point x. The density p_m is not guaranteed to be a good approximation to p_0, however, and therefore, cannot be used as an importance sampler.

For this reason we take a sequence of intermediate densities $p_1, p_2, \ldots, p_{m-1}$. This sequence is defined by us as follows. We take $p_i(x) \propto f_i(x)$, for $i = 1, 2, \ldots, m-1$, where

$$f_i(x) = f_0(x)^{\beta_i} f_m(x)^{1-\beta_i}$$

with $1 = \beta_0 > \beta_1 > \ldots > \beta_m = 0$. For each $i = 1, 2, \ldots, m-1$ we must have a Markov chain transition probability density T_i that leaves p_i invariant. The only other requirement for T_i is that we are able to sample from it. It

can have any form; for example it can be a sequence of Metropolis-Hastings transitions. Let $T_i(\tilde{x}|x)$ be the probability or probability density of the chain moving from x to $\tilde{x}$.

To draw a sample $x(1), \ldots, x(n)$ with corresponding weights $w(1), \ldots, w(n)$ we must do the following for each $x(j)$:

$$\begin{array}{lll} \text{generate} & x_{m-1} & \text{from } p_m, \\ \text{generate} & x_{m-2} & \text{from } T_{m-1}(\cdot|x_{m-1}), \\ & \ldots & \\ \text{generate} & x_1 & \text{from } T_2(\cdot|x_2), \\ \text{generate} & x_0 & \text{from } T_1(\cdot|x_1). \end{array}$$

Then let $x(j) = x_0$ and

$$w(j) = \frac{f_{m-1}(x_{m-1})}{f_m(x_{m-1})} \frac{f_{m-2}(x_{m-2})}{f_{m-1}(x_{m-2})} \cdots \frac{f_1(x_1)}{f_2(x_1)} \frac{f_0(x_0)}{f_1(x_0)}. \tag{5.22}$$

The form of the weight can be explained if we consider an m-dimensional state space with points $(x_0, \ldots, x_{m-1})$ and define on it the density

$$f(x_0, \ldots, x_{m-1}) = f_0(x_0) \prod_{i=1}^{m-1} \tilde{T}_i(x_i|x_{i-1}), \tag{5.23}$$

where $\tilde{T}_i$ is the reversal of T_i given by

$$\tilde{T}_i(x_i|x_{i-1}) = T_i(x_{i-1}|x_i) \frac{p_i(x_i)}{p_i(x_{i-1})} = T_i(x_{i-1}|x_i) \frac{f_i(x_i)}{f_i(x_{i-1})} \tag{5.24}$$

and $\tilde{T}_i(\tilde{x}|x)$ is the probability or probability density of the chain defined by

$\tilde{T}_i$ moving from x to $\tilde{x}$.

Notice that (5.23) has f_0 as the marginal density of x_0. The sample points have been generated from the importance sampler

$$g(x_0, \ldots, x_{m-1}) = \left[\prod_{i=1}^{m-1} T_i(x_{i-1}|x_i)\right] f_m(x_{m-1}). \tag{5.25}$$

Substituting (5.24) in (5.23) and dividing by (5.25) gives weights (5.22). Since they are the correct weights for the extended space they are also correct for the marginal space of x_0.

If an independent sample from p_0 is required we can resample from among $x(1), \ldots, x(n)$ with probabilities of selection proportional to the weights. Also we can use each $x(j)$ as the starting point of a Markov chain that leaves p_0 invariant. According to the theory of Rao-Blackwellization the points generated from this chain can take weight $w(j)$ and can also be used in a resampling scheme.

Neal (1998) argues that the larger m is the better the method performs because the variance of the weights is reduced. Moreover, for any given m the best way of choosing the actual values of β_i is to place $\ln(\beta_i), i = 0, \ldots, m$ at equidistant points in $[0, 1]$.

The virtue of the method is obvious. It can give importance samplers as good as we want for any target density. However it is very time consuming even in static settings. This feature will be even worse if we apply it in a dynamic problem for sampling in real time from the filtering densities. Of course, as computing equipment becomes faster, the method will prove more useful in the future.

5.5.2 Chaining via annealing

This method has some similarities with annealed importance sampling. It was proposed by Evans (1991) and its aim again is to find an importance sampler g that approximates a target density proportional to f in the best possible way. If we decide to look for g in a class $\mathcal{G}$ we can find it by **adaptive importance sampling**. This term is used by Evans (1991) and it is an entirely different technique from that presented in Chapter 2. Suppose that m^* is a vector of characteristics of f that can be calculated via expectations. Also suppose that $m(\lambda)$ is the corresponding vector for $g_\lambda \in \mathcal{G}$. We can find g as follows.

1. Choose a starting density $g_{\lambda_1} \in \mathcal{G}$.

2. Sample $x(1), \ldots, x(n)$ from g_{λ_1}. Then estimate m^* by $\hat{m}_1$ using the sample. For example, if x is univariate and its expectation is an element of m^*, the corresponding element of $\hat{m}_1$ will be

$$\frac{\sum_{i=1}^{n} x(i) f(x(i))/g_{\lambda_1}(x(i))}{\sum_{i=1}^{n} f(x(i))/g_{\lambda_1}(x(i))}. \tag{5.26}$$

3. Find the $g_\lambda \in \mathcal{G}$ that minimizes $||m(\lambda) - \hat{m}_1||^2$ and call it g_{λ_2}.

4. Generate $x(n+1), \ldots, x(2n)$ from g_{λ_2} and get a new estimate $\hat{m}_2$ of m^* by combining both samples. For example (5.26) becomes

$$\frac{\sum_{i=1}^{n} x(i) f(x(i))/g_{\lambda_1}(x(i)) + \sum_{i=n+1}^{2n} x(i) f(x(i))/g_{\lambda_2}(x(i))}{\sum_{i=1}^{n} f(x(i))/g_{\lambda_1}(x(i)) + \sum_{i=n+1}^{2n} f(x(i))/g_{\lambda_2}(x(i))}.$$

We then repeat steps 3 and 4 until the estimates stabilize. The final g_λ is the g we want. However, the method will not perform well if the starting density

g_{λ_1} is not good. This is why we use chaining. It involves setting up a chain of target densities. The chain is set up in such a way that for the first of the targets there is a good starting density for adaptive importance sampling. We apply adaptive importance sampling to each target density in turn, with the resulting best fitting density for a target being used as starting density for the next one. The final target in the chain is f itself.

The chain consists of target densities proportional to

$$f_{(t,u,\lambda_1)}(x) = f(x)^{1/t} g_{\lambda_1}(x)^{1/u}, \tag{5.27}$$

with t and u being positive parameters. g_{λ_1} is a density from which we can draw samples. For large t and $u = 1$ (5.27) resembles g_{λ_1}, while for $t = 1$ and large u it resembles f. We select $u = 1$ and a large value t_1 for t for the first target density and g_{λ_1} as the corresponding starting density for adaprtive importance sampling. After adaptive importance sampling is finished we decrease t slightly and use the result of adaptive importance sampling as starting density for the new target. We stop when t reaches 1. We proceed in the same fashion by forming more target densities, but now we keep $t = 1$ and we increase u slowly until it becomes very large. The result of the final adaptive importance sampling is the density that can be used to perform importance sampling for f.

The comments concerning the applicability of annealed importance sampling in dynamic systems are valid here too. In fact, chaining will be slower than an annealed importance sampling scheme that uses the same number of intermediate densities. The choice of how to describe the densities by a vector of characteristic expectations is an awkward one. If the target density is of a

strange form a very large number of characteristics may be necessary to describe it adequately. Then trying to find an importance sampler that matches it in all of them may not be an easy task.

5.6 Discussion

In this chapter we dealt with recently proposed methods for obtaining samples from the filtering densities that arise in the analysis of dynamic models. They have arisen as improvements of the resampling based methods that we have been using so far. Some aim to be robust in the presence of outliers and others to avoid the gradual impoverishment of the posterior samples of unknown constants.

Auxiliary variable and stratified particle filtering are indeed more robust than the simple particle filtering we used in the two previous chapters. However, even they cannot guarantee that the posterior samples they give will come from the main body of the posterior distribution if an outlier is observed. This is an inherent problem of all particle filtering techniques because they use discrete probability functions over finite-sized samples in order to approximate continuous distributions. From a practical point of view it is better to implement them by resampling instead of by MCMC sampling so as to avoid the need to monitor the convergence of the Markov chains involved in the latter.

As far as unknown constants are concerned neither of the two methods just mentioned nor stratification offers any substantial improvement because they all use resampling. At least stratification accumulates the resampling weights

over many observations, thus alleviating to some extent the effects of outliers. Another corrective measure is to implement any necessary resampling with smooth bootstrap. Rao-Blackwellization looks more capable of achieving its aims but it requires the availability for sampling of the full conditional distribution of the unknown constants given all the other unknowns and the data. This requirement may not always be satisfied, and this is the case with our control problem.

Annealed importance sampling may be impractical for dynamic model problems for the time being but it will benefit from more powerful computing equipment in the future. It still relies on resampling, however, and therefore it too approximates continuous densities with discrete distributions.

Finally, chaining via annealing is very unwieldy and since annealed importance sampling can give equally good results with smaller computational cost we cannot see it becoming a widespread statistical tool.

5.7 Reassessment of our work on control

In the last three chapters we have devoted our attention to a problem from the area of stochastic control. We had a double motive for doing this. First, it is an interesting problem in its own right that can find useful applications in industry. However, it is difficult to solve with the usual theory of control because of the non-linearities involved. Secondly, if we express the problem in dynamic model terms and try to use statistical methods, the arising distributions are intractable. We wanted, therefore, to see how resampling would cope with sampling from them.

We suggested three ways of solving the problem which are almost equivalent. The one that, according to simulation results, turned out to be the best consists of setting the current control point equal to the mode of the posterior distribution of the minimum's location for the previous time point. The mode is estimated from a sample from the posterior. Assuming decreasing degrees of knowledge about the parameters of the problem the method managed to give good results for all but the most extreme case. It also coped well with an extension of the problem in a multidimensional space. We also managed to analyse theoretically a simpler version of the problem and gain useful insight into the workings of the method.

Because of the dynamical nature of the problem the samples required by the method were obtained with resampling and the resampling algorithm used was smooth bootstrap. We took the simplest choice of importance sampler, namely the prior associated with each posterior. This means that the most favoured points during resampling were the ones with the highest likelihood. Sometimes an outlier would be observed which would have led to a bad posterior sample, possibly consisting of few distinct values and being away from the main support of the posterior. Fortunately, we were able to use intervention and to correct things in all variants of the control problem apart from that with the most unknowns.

The new resampling algorithms of this chapter have as their strategy to develop importance samplers that generate points which are bound to receive large weights. We believe that this strategy does not really improve things and we attempt to explain why.

In dynamic models in general, if the posterior distributions are not available in closed form then neither are the priors. A resampling algorithm that has

a posterior as its target must then ensure that the prior forms part of the importance sampler. This leads to the weights being affected solely by the likelihood. However, high likelihood does not necessarily mean high posterior density, especially if an outlier is observed. For this reason we think that if the prior is available in closed form it should be left out of the importance sampler. Then it will be part of the numerator of the weights and will counterbalance the effect of the likelihood. This is, in our opinion, the only way to have a resampling algorithm totally immune to the presence of outliers. One could argue that a kernel density estimate of the prior based on a sample from it could be used in the calculation of weights. However, kernel density estimates are effective only for distributions of very few variables. We believe that the creation of importance samplers which are not affected by outliers is still an open challenge.

A course of action that we have not taken in this thesis would be not to perform resampling at each time point but to accumulate the weights. Resampling could be performed every time the effective sample size became very small, for example. The accumulated weights will then play the role of the prior distribution and points with high likelihood will not necessarily receive large weights.

The problem studied in the last three chapters and the models considered in Chapter 2 involved Normally distributed noise and disturbance variables. This was done only for convenience and does not imply that in non-Gaussian cases resampling is not effective. The **bearings only tracking** problem for example, which is a well known non-linear and non-Gaussian problem has been studied by many researchers with the aid of resampling; see among others Gordon *et al* (1993) and Carpenter *et al* (1997). Different noise dis-

tributions lead to different distributions of weights and different disturbance distributions lead to different propagation effects on the samples. The general properties of the method however, remain the same. The most important thing is to be able at least to sample from the disturbance distribution and to evaluate the likelihood.

Another interesting area of study in resampling into which we have not ventured is the development of algorithms which may not be as good asymptotically as the methods already presented but which may be more effective with finite sample sizes. If this turns out to be possible it will have significant implications since in practice sample sizes are always finite.

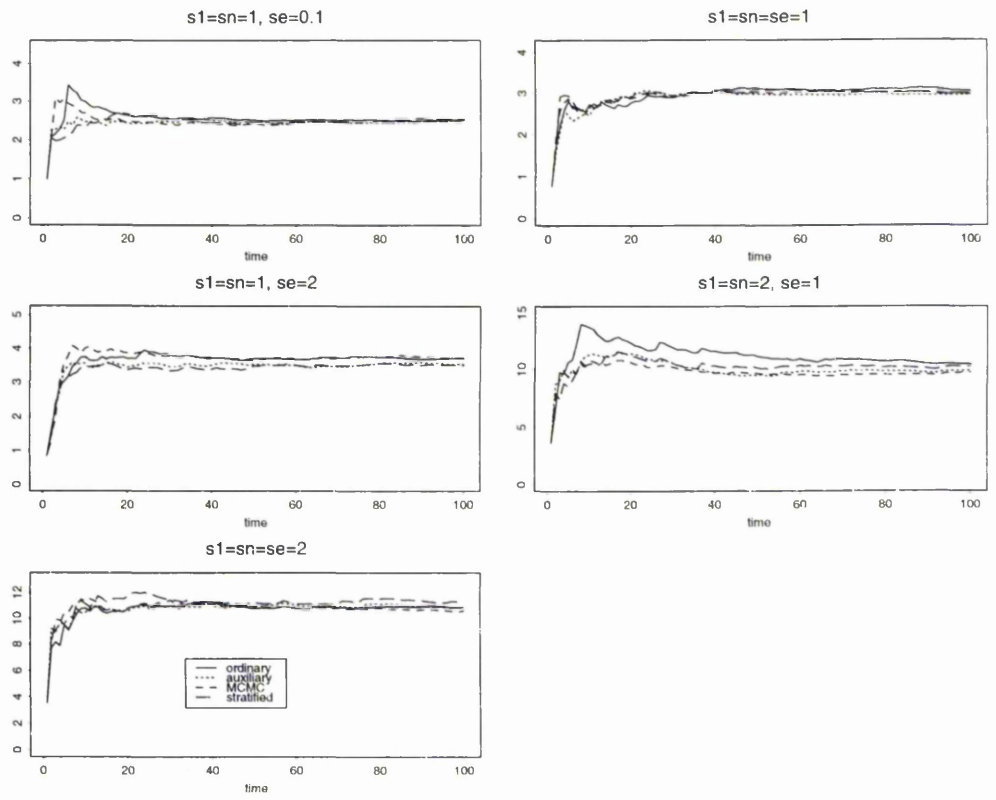


Figure 5.1: Simulated expected rates of loss for Titterington's method with $K = 0$ from the experiment of section 5.4.2. Five different combinations of σ_1, σ_η and σ_ϵ are considered. The standard deviations are assumed known. The method has been implemented with four different ways of obtaining the samples it requires. **Ordinary** signifies the simple particle filtering of Chapter 3. **Auxiliary** and **MCMC** mean the auxiliary variable particle filtering in its resampling and MCMC implementations respectively. Finally, **stratified** is the stratified particle filtering.

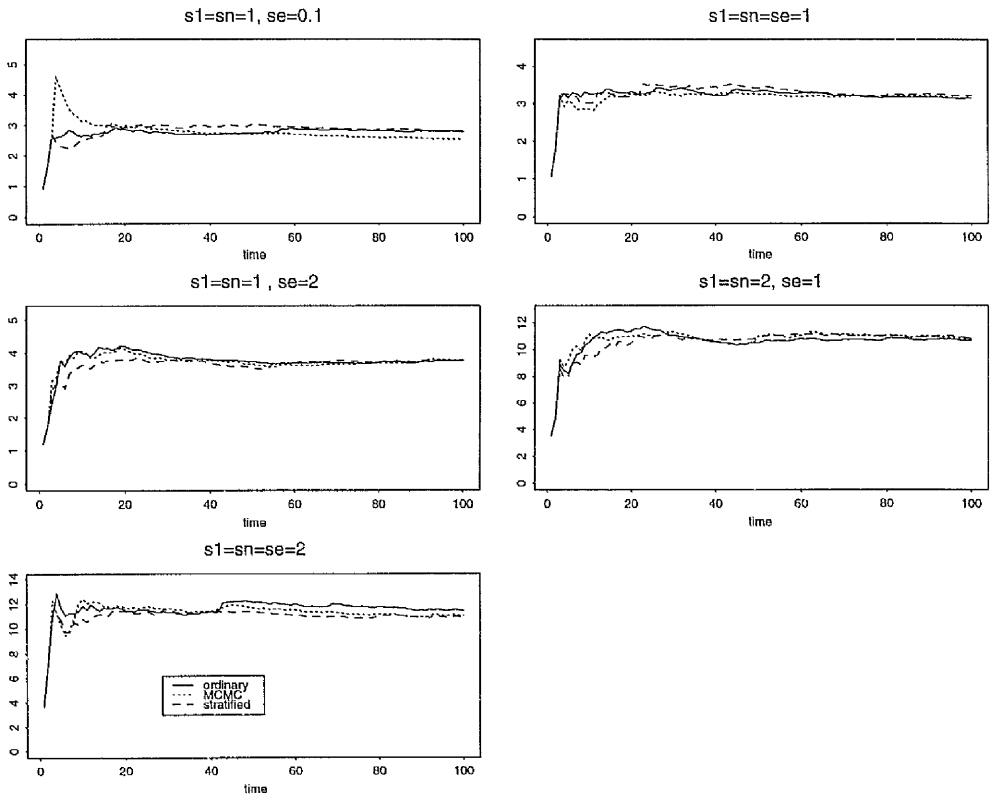


Figure 5.2: Simulated expected rates of loss for Titterington's method with $K = 0$ from the experiment of section 5.4.3. Five different combinations of σ_1, σ_η and σ_ϵ are considered. The standard deviations are assumed unknown. The method has been implemented with three different ways of obtaining the samples it requires. **Ordinary** signifies the simple particle filtering of Chapter 3. **MCMC** means the MCMC implementation of the simple particle filtering of Chapter 3. Finally, **stratified** is the stratified particle filtering.

Bibliography

[1] Acklam, P. J. (1996). Monte Carlo Methods in State Space Estimations. Cand. Scient. thesis, University of Oslo.

[2] Alspach, D. L. and Sorenson, H. W. (1972). Nonlinear Bayesian Estimation Using Gaussian Sum Approximations. *IEEE Trans. Auto. Control*, 17, 439-448.

[3] Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering.* Englewood Cliffs: Prentice-Hall.

[4] Aoki, M. (1967). *Optimization of Stochastic Systems.* New York: Academic Press.

[5] Bar-Shalom, Y. (1981). Stochastic Dynamic Programming: Caution and Probing. *IEEE Trans. Auto. Control*, 26, 1184-1195.

[6] Beadle, E. R. and Djuric, P. M. (1997). A Fast Weighted Bayesian Bootstrap Filter for Nonlinear Model State Estimation. *IEEE Trans. Aerosp. Elec. Syst.*, 33, 338-343.

[7] Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997). Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods. *J. Amer. Statist. Assoc.* , 92, 1403-1412.

[8] Besag, J. (1986). On the Statistical Analysis of Dirty Pictures (with discussion). *J. R. Statist. Soc.* B, 48, 259-302.

[9] Besag, J. and Green, P. J. (1993). Spatial Statistics and Bayesian Computation. *J. R. Statist. Soc.* B, 55, 25-37.

[10] Carpenter, J., Clifford, P. and Fearnhead, P. (1997). An Improved Particle Filter for Non-linear Problems. Technical report, University of Oxford.

[11] Doucet, A. (1998). On Sequential Simulation-Based Methods for Bayesian Filtering. Technical report, University of Cambridge.

[12] Drenick, R. F. and Shaw, L. (1964). Optimal Control of Linear Plants with Random Parameters. *IEEE Trans. Auto. Control*, 9, 236-244.

[13] Escobar, M. D. and West, M. (1995). Bayesian Density Estimation and Inference using Mixtures. *J. Amer. Statist. Assoc.*, 90, 577-588.

[14] Evans, M. (1991). Chaining via Annealing. *Ann. Statist.*, 19, 382-393.

[15] Fahrmeir, L. (1992). Posterior Mode Estimation by Extended Kalman Filtering for Multivariate Dynamic Generalized Linear Models. *J. Amer. Statist. Assoc.*, 87, 501-509.

[16] Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1997). *Bayesian Data Analysis.* London: Chapman and Hall.

[17] Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6, 721-741.

[18] Geweke, J. (1989). Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica* , 57, 1317-1339.

[19] Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (Eds.) (1996). *Markov Chain Monte Carlo in Practice.* London: Chapman and Hall.

[20] Givens, G. H. and Raftery, A. E. (1996). Local Adaptive Importance Sampling for Multivariate Densities with Strong Nonlinear Relationships. *J. Amer. Statist. Assoc.*, 91, 132-141.

[21] Gordon, N. J. (1993). Bayesian Methods for Tracking. Ph. D. Thesis, University of London.

[22] Gordon, N. J. (1997). A hybrid bootstrap filter for target tracking in clutter. *IEEE Trans. Aerosp. Elec. Syst.*, 33, 353-358.

[23] Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993). Novel Approach to Nonlinear / Non-Gaussian Bayesian State Estimation. *IEE Proc.-F*, 140, 107-113.

[24] Gordon, N. J. and Whitby, A. (1995). A Bayesian Approach to Target Tracking in the Presence of Glint, in *SPIE vol 2561, Signal and Data Processing of Small Targets* (ed. O.E. Drummond), 472-483.

[25] Hainsworth, T. J. and Mardia, K. V. (1992). A Markov Random Field Restoration of Image Sequences, in *Markov Random Fields. Theory and Applications*, 409-445, MOD.

[26] Hurzeler, M. and Kunsch, H. R. (1998). Approximations for General State Space Models. *J. Comp. Graph. Statist.*, 7, 175-193.

[27] Isard, M. and Blake, A. (1996). Contour Tracking by Stochastic Propagation of Conditional Density, in *Computer Vision - ECCV' 96* (eds. B. Buxton and R. Cipolla) 343-356.

[28] Julier, S. J. and Uhlmann, J. K. (1997). New Extension of the Kalman Filter to Nonlinear Systems, in *Proc. SPIE vol 3068*, 182-193.

[29] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME, J. Basic Eng.*, 82D, 35-45.

[30] Kendall, M. G. and Stuart, A. (1963). *The Advanced Theory of Statistics, 2nd Ed., Vol 1.* London: Charles Griffin and Co.

[31] Kitagawa, G. (1987). Non-Gaussian State Space Modelling of Non-Stationary Time Series. *J. Amer. Statist. Assoc.*, 82, 1032-1063.

[32] Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *J. Comp. Graph. Statist.*, 5, 1-25.

[33] Kong, A., Liu, J. S. and Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *J. Amer. Statist. Assoc.*, 89, 278-288.

[34] Kramer, S. C. and Sorenson, H. W. (1988). Recursive Bayesian Estimation using Piecewise Constant Approximations. *Automatica*, 24, 789-801.

[35] Lee, D. S. (1996). New Results for the SIR Filter. Technical report, Johns Hopkins University.

[36] Liu, J. S. and Chen, R. (1995). Blind Deconvolution via Sequential Imputations. *J. Amer. Statist. Assoc.*, 90, 567-576.

[37] Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *J. Amer. Statist. Assoc.*, 93, 1032-1044.

[38] MacEachern, S. N., Clyde, M. and Liu, J. S. (1998). Sequential Importance Sampling for Nonparametric Bayes Models: The Next Generation. *Canad. J. Statist.* , to appear.

[39] Masreliez, C. J. (1975). Approximate Non-Gaussian Filtering with Linear State and Observation Relations. *IEEE Trans. Auto. Control*, 20, 107-110.

[40] Masreliez, C. J. and Martin, R. D. (1977). Robust Bayesian Estimation for the Linear Model and Robustifying the Kalman Filter. *IEEE Trans. Auto. Control*, 22, 361-371.

[41] Meinhold, R. J. and Singpurwalla, N. D. (1989). Robustfication of Kalman Filter Models. *J. Amer. Statist. Assoc.*, 84, 479-486.

[42] Mueller, P. (1992). Posterior Integration in Dynamic Models. *Comp. Science Stat.*, 24, 318-324.

[43] Neal, R. M. (1998). Annealed Importance Sampling. Technical report, Univ. of Toronto.

[44] Oehlert, G. W. (1998). Faster Adaptive Importance Sampling in Low Dimensions. *J. Comp. Graph. Statist.*, 7, 158-174.

[45] Pitt, M. K. and Shephard, N. (1997). Filtering via Simulation: Auxiliary Particle Filters. Technical report, Imperial College, Univ. of London.

[46] Pole, A. and West, M. (1990). Efficient Bayesian Learning in Dynamic Models. *J. Forecasting* , 9, 119-136.

[47] Ripley, B. D. (1987). *Stochastic Simulation.* New York: Wiley.

[48] Ripley, B. D. and Sutherland, A. I. (1990). Finding Spiral Structures in Images of Galaxies. *Phil. Trans. Roy. Soc. Lond. A*, 332, 477-485.

[49] Rubin, D. B. (1987). The SIR Algorithm - A discussion of Tanner and Wong's: The Calculation of Posterior Distributions by Data Augmentation. *J. Amer. Statist. Assoc.*, 82, 543-546.

[50] Rubin, D. B. (1988). Using the SIR Algorithm to Simulate Posterior Distributions, in *Bayesian Statistics 3* (eds. J. M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith), 395-402.

[51] Salmond, D. J. (1990). Mixture Reduction for Target Tracking in Clutter, in *SPIE vol 1305, Signal and Data Processing of Small Targets*, 434-445.

[52] Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

[53] Smith, A. F. M. and Gelfand, A. E. (1992). Bayesian Statistics without Tears: A Sampling Resampling Perspective. *Amer. Statistician*, 46, 84-88.

[54] Smith, A. F. M. and Roberts, G. O. (1993). Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods (with discussion). *J. Roy. Statist. Soc. B*, 55, 3-102.

[55] Sorenson, H. W. and Stubberud, A. R. (1968). Recursive Filtering for Systems with Small but Non-Negligible Nonlinearities. *Int. J. Control*, 7, 271-280.

[56] Srinivasan, K. (1970). State Estimation by Orthogonal Expansion of Probability Distributions. *IEEE Trans. Auto. Control*, 15, 3-10.

[57] Sutherland, A. I. and Titterington, D. M. (1994). Bayesian Analysis of Image Sequences. Technical report 94-3, Dept. of Statistics, Univ. of Glasgow.

[58] Tanizaki, H. and Mariano, R. S. (1994). Prediction, Filtering and Smoothing in Non-Linear and Non-Normal Cases using Monte Carlo Integration. *J. Appl. Econometrics*, 9, 163-179.

[59] Titterington, D. M. (1973). A Method of Extremum Adaptation. *J. Inst. Maths Applics*, 11, 297-315.

[60] West, M. (1990). Bayesian Kernel Density Estimation. Technical report, Duke University.

[61] West, M. (1993). Approximating Posterior Distributions by Mixtures. *J. R. Statist. Soc.* B, 55, 409-422.

[62] West, M., and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models.* New York: Springer Verlag (2nd ed).

[63] Whittle, P. (1969). A View of Stochastic Control Theory. *J. R. Statist. Soc. A*, 132, 320-334.

[64] Wishart, D. M. G. (1969). A Survey of Control Theory. *J. R. Statist. Soc. A*, 132, 293-319.

Appendix A

Expected values of weighted bootstrap and smooth weighted bootstrap sample statistics

A.1 Weighted bootstrap samples

Suppose that a sample from a distribution g is available and let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be that sample. We apply weighted bootstrap and get a sample $(Y_1, Y_2, \ldots, Y_n)$ in order to study another distribution h with $h(x) = \frac{f(x)}{\int f(x)dx}$. The probabilities of selection are, as usual, $q_i = \frac{w_i}{\sum_{j=1}^{n} w_j}$ with $w_i = \frac{f(x_i)}{g(x_i)}$. Now let $\bar{Y}$ and $\hat{\sigma}_Y^2$ be the mean and variance of the weighted bootstrap sample.

The expectations of these two estimators can only be calculated with respect to (w.r.t.) g since they are actually based on the sample from it. For example, it is obvious that

$$E(\bar{Y}|\mathbf{X}) = E(Y_i|\mathbf{X}) = \sum_{j=1}^{n} q_j X_j = \frac{\sum_{j=1}^{n} \frac{f(X_j)}{g(X_j)} X_j}{\sum_{j=1}^{n} \frac{f(X_j)}{g(X_j)}}.$$

By taking the expectation w.r.t. g of the rightmost part of the above equality we get the unconditional expectation of $\bar{Y}$:

$$E_g(\bar{Y}) = E_g\left(\sum_{j=1}^{n} q_j X_j\right) = E_g\left(\frac{\sum_{j=1}^{n} \frac{f(X_j)}{g(X_j)} X_j}{\sum_{j=1}^{n} \frac{f(X_j)}{g(X_j)}}\right) = E_g\left(\frac{S_1}{S_2}\right).$$

Let $S_1 = E_g(S_1) + \delta_1$ and $S_2 = E_g(S_2) + \delta_2$, where $E_g(\delta_1) = E_g(\delta_2) = 0$. Also let $\mu_1 = E_g(S_1)$ and $\mu_2 = E_g(S_2)$.

Then

$$\begin{aligned} \frac{S_1}{S_2} &= \frac{\mu_1 + \delta_1}{\mu_2 + \delta_2} = \frac{\mu_1 + \delta_1}{\mu_2}\left(1 + \frac{\delta_2}{\mu_2}\right)^{-1} \\ &= \frac{\mu_1 + \delta_1}{\mu_2}\left(1 - \frac{\delta_2}{\mu_2} + \frac{\delta_2^2}{\mu_2^2} - \cdots\right) \approx \frac{\mu_1}{\mu_2} + \frac{\delta_1}{\mu_2} - \frac{\mu_1\delta_2}{\mu_2^2} - \frac{\delta_1\delta_2}{\mu_2^2} + \frac{\mu_1\delta_2^2}{\mu_2^3}. \end{aligned}$$

Higher powers of δ_2 have been omitted. Thus, we get

$$E_g\left(\frac{S_1}{S_2}\right) = \frac{\mu_1}{\mu_2} - \frac{1}{\mu_2^2} E_g(\delta_1\delta_2) + \frac{\mu_1}{\mu_2^3} E_g(\delta_2^2),$$

$$\begin{aligned}\mu_1 &= E_g(S_1) = E_g\left(\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i\right) = nE_g\left(\frac{f(X)}{g(X)}X\right)\\ &= n\int\frac{f(x)}{g(x)}xg(x)dx = n\int xf(x)dx,\end{aligned}$$

$$\begin{aligned}\mu_2 &= E_g(S_2) = E_g\left(\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}\right) = nE_g\left(\frac{f(X)}{g(X)}\right)\\ &= n\int\frac{f(x)}{g(x)}g(x)dx = n\int f(x)dx.\end{aligned}$$

Therefore,

$$\frac{\mu_1}{\mu_2} = \frac{n\int xf(x)dx}{n\int f(x)dx} = \int x\frac{f(x)}{\int f(x)dx}dx = \int xh(x)dx = E_h(X).$$

$$\begin{aligned}E_g(\delta_2^2) &= Var(S_2) = Var\left(\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}\right) = nVar\left(\frac{f(X)}{g(X)}\right)\\ &= n\left\{E_g\left[\left(\frac{f(X)}{g(X)}\right)^2\right] - \left[E_g\left(\frac{f(X)}{g(X)}\right)\right]^2\right\}\\ &= n\left[\int\left(\frac{f(x)}{g(x)}\right)^2 g(x)dx - \left(\int f(x)dx\right)^2\right]\\ &= n\left[\int\frac{(f(x))^2}{g(x)}dx - \left(\int f(x)dx\right)^2\right] = n\left[\int\frac{(f(x))^2}{g(x)}dx - \frac{1}{n^2}\mu_2^2\right].\end{aligned}$$

So,

$$\begin{aligned}
\frac{\mu_1}{\mu_2^3}E_g(\delta_2^2) &= n\frac{\mu_1}{\mu_2^3}\int\frac{(f(x))^2}{g(x)}dx - \frac{1}{n}\frac{\mu_1}{\mu_2} \\
&= nE_h(X)\frac{1}{n^2\left(\int f(x)dx\right)^2}\int\frac{(f(x))^2}{g(x)}dx - \frac{1}{n}E_h(X) \\
&= \frac{1}{n}E_h(X)\left[\int\frac{h(x)}{g(x)}h(x)dx - 1\right] = \frac{1}{n}E_h(X)\left[E_h\left(\frac{h(X)}{g(X)}\right) - 1\right].
\end{aligned}$$

$$\begin{aligned}
E_g(\delta_1\delta_2) &= Cov(S_1, S_2) = Cov\left(\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i, \sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}\right) \\
&= nCov\left(\frac{f(X)}{g(X)}X, \frac{f(X)}{g(X)}\right) \\
&= n\left\{E_g\left[\left(\frac{f(X)}{g(X)}\right)^2 X\right] - E_g\left(\frac{f(X)}{g(X)}X\right)E_g\left(\frac{f(X)}{g(X)}\right)\right\} \\
&= n\left[\int\frac{(f(x))^2}{g(x)}xdx - \frac{1}{n^2}\mu_1\mu_2\right].
\end{aligned}$$

So,

$$\begin{aligned}
\frac{1}{\mu_2^2}E_g(\delta_1\delta_2) &= \frac{n}{n^2\left(\int f(x)dx\right)^2}\int\frac{(f(x))^2}{g(x)}xdx - \frac{1}{n}\frac{\mu_1}{\mu_2} \\
&= \frac{1}{n}\left[E_h\left(\frac{h(X)}{g(X)}X\right) - E_h(X)\right].
\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{\mu_1}{\mu_2^3}E_g(\delta_2^2) - \frac{1}{\mu_2^2}E_g(\delta_1\delta_2) &= \frac{1}{n}\left[E_h(X)E_h\left(\frac{h(X)}{g(X)}\right) - E_h\left(\frac{h(X)}{g(X)}X\right)\right] \\ &= -\frac{1}{n}Cov_h\left(X, \frac{h(X)}{g(X)}\right),\end{aligned}$$

and finally,

$$E_g(Y_i) \approx E_h(X) - \frac{1}{n}Cov_h\left(X, \frac{h(X)}{g(X)}\right).$$

The conclusion is that

$$E_g(\bar{Y}) = E_g\left(\sum_{i=1}^{n} q_i X_i\right) \approx E_h(X) - \frac{1}{n}Cov_h\left(X, \frac{h(X)}{g(X)}\right), \qquad \text{(A.1)}$$

where X is a random variable with distribution g. The result shows that the mean of a weighted bootstrap sample is a biased estimator, but the bias goes to zero as the sample size increases.

Concerning the variance of the same sample we have the following calculations:

$$\hat{\sigma}_Y^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} Y_i^2 - n\bar{Y}^2\right).$$

$$\bar{Y}^2 = \frac{1}{n^2}\left(\sum_{i=1}^{n} Y_i^2 + \sum\sum_{i\neq j} Y_i Y_j\right)$$

$$
\begin{aligned}
E\left(\sum_{i=1}^{n} Y_i^2|\mathbf{X}\right) &= n\frac{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i^2}{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}}. \\
E(\bar{Y}^2|\mathbf{X}) &= \frac{1}{n^2}\left[E\left(\sum_{i=1}^{n} Y_i^2|\mathbf{X}\right) + E\left(\sum_{i\neq j}\sum Y_iY_j|\mathbf{X}\right)\right] \\
&= \frac{1}{n^2}\left[nE(Y_i^2|\mathbf{X}) + n(n-1)E(Y_iY_j|\mathbf{X})\right] \\
&= \frac{1}{n}\left\{E(Y_i^2|\mathbf{X}) + (n-1)\left[E(Y_i|\mathbf{X})\right]^2\right\} \\
&= \frac{1}{n}\left[\frac{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i^2}{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}} + (n-1)\left(\frac{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i}{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}}\right)^2\right].
\end{aligned}
$$

So, after a few straightforward calculations we arrive at

$$
E(\hat{\sigma}_Y^2|\mathbf{X}) = \frac{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i^2}{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}} - \left(\frac{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i}{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}}\right)^2.
$$

By a procedure similar to the one used for the mean $\bar{Y}$ we get that

$$
E_g\left(\frac{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i^2}{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}}\right) \approx E_h(X^2) - \frac{1}{n}Cov_h\left(X^2, \frac{h(X)}{g(X)}\right). \tag{A.2}
$$

From the calculations concerning the mean we have that

$$
\frac{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i}{\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}} = \frac{S_1}{S_2} \approx \frac{\mu_1+\delta_1}{\mu_2}\left(1 - \frac{\delta_2}{\mu_2} + \frac{\delta_2^2}{\mu_2^2}\right)
$$

and, after some approximations, we have that

$$E_g\left[\left(\frac{S_1}{S_2}\right)^2\right] = E_g\left[(\sum_{i=1}^{n} q_i X_i)^2\right] \approx \frac{\mu_1^2}{\mu_2^2} + \frac{3\mu_1^2}{\mu_2^4}E_g(\delta_2^2) + \frac{1}{\mu_2^2}E_g(\delta_1^2) - \frac{4\mu_1}{\mu_2^3}E_g(\delta_1\delta_2).$$

Using results from previous calculations we get the following:

$$\frac{\mu_1^2}{\mu_2^2} = [E_h(X)]^2,$$

$$\begin{aligned}\frac{\mu_1^2}{\mu_2^4}E_g(\delta_2^2) &= E_h(X)\frac{\mu_1}{\mu_2^3}E_g(\delta_2^2) \\ &= \frac{1}{n}[E_h(X)]^2\left[E_h\left(\frac{h(X)}{g(X)}\right) - 1\right],\end{aligned}$$

$$\begin{aligned}E_g(\delta_1^2) &= Var_g(S_1) = Var_g\left(\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}X_i\right) = nVar_g\left(\frac{f(X)}{g(X)}X\right) \\ &= n\left\{E_g\left[\left(\frac{f(X)}{g(X)}X\right)^2\right] - \left[E_g\left(\frac{f(X)}{g(X)}X\right)\right]^2\right\} \\ &= n\int\frac{(f(x))^2x^2}{g(x)}dx - \frac{\mu_1^2}{n}.\end{aligned}$$

So,

$$\frac{1}{\mu_2^2}E_g(\delta_1^2) = \frac{1}{n}E_h\left(\frac{h(X)X^2}{g(X)}\right) - \frac{1}{n}[E_h(X)]^2,$$

$$E_g(\delta_1\delta_2) = n\left[\int \frac{(f(x)^2)}{g(x)}xdx - \frac{1}{n^2}\mu_1\mu_2\right],$$

$$\frac{\mu_1}{\mu_2^3}E_g(\delta_1\delta_2) = \frac{1}{n}E_h(X)\left[E_h\left(\frac{h(X)}{g(X)}X\right) - E_h(X)\right].$$

So, after some simple calculations,

$$\begin{aligned} E_g\left[\left(\frac{S_1}{S_2}\right)^2\right] &= E_g\left[(\sum_{i=1}^{n} q_i X_i)^2\right] \\ &\approx [E_h(X)]^2 - \frac{3}{n}E_h(X)Cov_h\left(X, \frac{h(X)}{g(X)}\right) \\ &+ \frac{1}{n}Cov_h\left(X, \frac{h(X)}{g(X)}X\right). \end{aligned} \tag{A.3}$$

Then,

$$\begin{aligned} E_g(\hat{\sigma}_Y^2) &= Var_h(X) - \frac{1}{n}Cov_h\left(X^2, \frac{h(X)}{g(X)}\right) \\ &+ \frac{3}{n}E_h(X)Cov_h\left(X, \frac{h(X)}{g(X)}\right) - \frac{1}{n}Cov_h\left(X, X\frac{h(X)}{g(X)}\right). \end{aligned}$$

Again we see that the estimator is biased but the bias goes to zero as the sample size increases.

For the variance of the sample mean we have

$$V_g(\bar{Y}) = \frac{1}{n^2}\left[\sum_{i=1}^{n} V_g(Y_i) + \sum\sum_{j\neq i} Cov_g(Y_i, Y_j)\right].$$

$$V_g(Y_i) = V_g[E(Y_i|\mathbf{X})] + E_g[V(Y_i|\mathbf{X})]$$

and

$$Cov_g(Y_i, Y_j) = Cov_g(E[Y_i|\mathbf{X}], E[Y_j|\mathbf{X}]) + E_g[Cov(Y_i, Y_j|\mathbf{X})].$$

But given $\mathbf{X}$ the members of the weighted bootstrap sample are independent. Therefore $Cov(Y_i, Y_j|\mathbf{X}) = 0$. They are also identically distributed given $\mathbf{X}$ and so, $E[Y_i|\mathbf{X}] = E[Y_j|\mathbf{X}]$ for all i and j. These lead to

$$Cov_g(Y_i, Y_j) = V_g[E(Y_i|\mathbf{X})].$$

Then,

$$\begin{aligned} V_g(\bar{Y}) &= \frac{1}{n^2}\left[nV_g[E(Y_i|\mathbf{X})] + nE_g[V(Y_i|\mathbf{X})] + n(n-1)V_g[E(Y_i|\mathbf{X})]\right] \\ &= V_g[E(Y_i|\mathbf{X})] + \frac{1}{n}E_g[V(Y_i|\mathbf{X})] \\ &= V_g\left(\sum_{i=1}^{n} q_i X_i\right) + \frac{1}{n}E_g\left[\sum_{i=1}^{n} q_i X_i^2 - \left(\sum_{i=1}^{n} q_i X_i\right)^2\right] \end{aligned}$$

$$= \frac{n-1}{n} E_g\left[\left(\sum_{i=1}^{n} q_i X_i\right)^2\right] - \left[E_g\left(\sum_{i=1}^{n} q_i X_i\right)\right]^2 + \frac{1}{n} E_g\left(\sum_{i=1}^{n} q_i X_i^2\right).$$

All the quantities involved have been calculated earlier. We put them in the equation and simple calculations lead to

$$\begin{aligned} V_g(\bar{Y}) &= \frac{1}{n} V_h(X) + \left(\frac{3}{n^2} - \frac{1}{n}\right) E_h(X) Cov_h\left(X, \frac{h(X)}{g(X)}\right) \\ &+ \frac{n-1}{n^2} Cov_h\left(X, X\frac{h(X)}{g(X)}\right) - \frac{1}{n^2}\left[Cov_h\left(X, \frac{h(X)}{g(X)}\right)\right]^2 \\ &- \frac{1}{n^2} Cov_h\left(X^2, \frac{h(X)}{g(X)}\right). \end{aligned}$$

We see that there are non-leading terms which go to zero at the same rate as the leading one and therefore this variance is not for any n as small as the variance of the mean of a random sample from h.

A.2 Smooth bootstrap samples

Here again we start with a sample $\mathbf{X} = (X_1, \ldots, X_n)$ from $g(x)$. Each point X_i again receives weight $w_i = \frac{f(X_i)}{g(X_i)}$ which is transformed into a probability $q_i = \frac{w_i}{\sum_{j=1}^{n} w_j}$. Smooth bootstrap amounts to sampling n values $Y_1, \ldots, Y_n$ from a mixture

$$\hat{p}(y) = \sum_{i=1}^{n} q_i K(y; m_i, b_n^2), \tag{A.4}$$

of symmetric densities with means m_i and variance b_n^2. Recall that $m_i = \alpha X_i + (1-\alpha)\bar{X}$ and $\bar{X} = \sum_{i=1}^n q_i X_i$. Also recall that for a suitably chosen value of α, (A.4) has mean $\bar{X}$ and variance $s^2 = \sum_{i=1}^n q_i (X_i - \bar{X})^2 = \sum_{i=1}^n q_i X_i^2 - (\sum_{i=1}^n q_i X_i)^2$ which is an estimator of the variance $V_h(X)$ of the target distribution $h(x) \propto f(x)$. Define the sample statistics $\bar{Y} = \frac{1}{n}\sum_{i=1}^n Y_i$ and $\hat{\sigma}_Y^2 = \frac{1}{n-1}\sum_{i=1}^n (Y_i - \bar{Y})^2$.

Then,

$$E_g(\bar{Y}) = E_g\left[E(\bar{Y}|\mathbf{X})\right] = E_g(\bar{X}) = E_g\left(\sum_{i=1}^n q_i X_i\right),$$

which is the same result as in the case of weighted bootstrap.

For the variance of the sample we have,

$$E_g(\hat{\sigma}_Y^2) = E_g[E(\hat{\sigma}_Y^2|\mathbf{X})] = E_g(s^2) = E_g\left(\sum_{i=1}^n q_i X_i^2\right) - E_g\left[\left(\sum_{i=1}^n q_i X_i\right)^2\right],$$

which from (A.2) and (A.3) becomes

$$\begin{aligned} E_g(\hat{\sigma}_Y^2) &= Var_h(X) - \frac{1}{n} Cov_h\left(X^2, \frac{h(X)}{g(X)}\right) \\ &+ \frac{3}{n} E_h(X) Cov_h\left(X, \frac{h(X)}{g(X)}\right) - \frac{1}{n} Cov_h\left(X, X\frac{h(X)}{g(X)}\right). \end{aligned}$$

This is again the same as in the case of a weighted bootstrap sample.

For the variance of the mean notice that

$$V_g(\bar{Y}) = \frac{1}{n^2}\left[\sum_{i=1}^{n} V_g(Y_i) + \sum\sum_{j\neq i} Cov_g(Y_i, Y_j)\right].$$

$$V_g(Y_i) = V_g[E(Y_i|\mathbf{X})] + E_g[V(Y_i|\mathbf{X})]$$

and

$$Cov_g(Y_i, Y_j) = Cov_g(E[Y_i|\mathbf{X}], E[Y_j|\mathbf{X}]) + E_g[Cov(Y_i, Y_j|\mathbf{X})].$$

Here too the members of the smooth bootstrap sample are independent given $\mathbf{X}$. Therefore following the same logic as in the weighted bootstrap case we get

$$V_g(\bar{Y}) = V_g[E(Y_i|\mathbf{X})] + \frac{1}{n}E_g[V(Y_i|\mathbf{X})].$$

Because of the use of correction (2.8) for the means with suitable α, we have that $E(Y_i|\mathbf{X}) = \sum_{i=1}^{n} q_i X_i$ and $V(Y_i|\mathbf{X}) = \sum_{i=1}^{n} q_i X_i^2 - (\sum_{i=1}^{n} q_i X_i)^2$. Therefore from now on everything is the same as for weighted bootstrap.

Appendix B

The algorithm that locates the rectangle

For a clarification of the ideas presented in this section refer to Fig B.1. A rectangle lies on the square $N \times N$ lattice. We only know the halfwidth h of the rectangle and the coordinates (x_1, y_1) and (x_2, y_2) of the midpoints of its two short sides. The point $(0, 0)$ of the coordinate system coincides with the lower left corner of the lattice. The x-axis is the lower border of the lattice and the y-axis is the left border of the lattice. We want to find which pixels in the lattice belong to the rectangle. We begin by calculating $d_i = x_2 - x_1, d_j = y_2 - y_1, l = \sqrt{d_i^2 + d_j^2}, \cos\phi = d_i/l$ and $\sin\phi = d_j/l$. We also find the coordinates $(i(1), j(1)), (i(2), j(2)), (i(3), j(3)), (i(4), j(4))$ of the four corners of the rectangle. For example,

$$i(1) \;=\; x_1 - h\cos\left(\frac{\pi}{2} - \phi\right) = x_1 - h\sin\phi,$$

$$j(1) = y_1 + h\sin\left(\frac{\pi}{2} - \phi\right) = y_1 + h\cos\phi.$$

We will also use the coordinate system $x'O'y'$. (i, j) denotes the coordinates of a pixel with respect to the original coordinate system and (i', j') its coordinates with respect to $x'O'y'$. Their relationship is

$$\begin{aligned} i' &= (i - x_1)\cos\phi + (j - y_1)\sin\phi, \\ j' &= (j - y_1)\cos\phi - (i - x_1)\sin\phi. \end{aligned}$$

Obviously in the rectangle belong the pixels whose coordinates (i', j') satisfy

$$0 \leq i' \leq l \quad \text{and} \quad -h \leq j' \leq h.$$

To avoid examining all pixels in the lattice we try to find a "search area" within the lattice such that we are certain that all pixels outside the search area do not belong to the rectangle. We have two separate search areas depending on whether $|d_i| \geq |d_j|$ or $|d_i| < |d_j|$.

If $|d_i| \geq |d_j|$ our search area consists of the pixels in the square surrounding the rectangle. Their coordinates (i, j) satisfy the inequalities

$$\begin{aligned} \min(i(1), i(2), i(3), i(4)) &\leq i \leq \max(i(1), i(2), i(3), i(4)) \quad \text{and} \\ \min(j(1), j(2), j(3), j(4)) &\leq j \leq \max(j(1), j(2), j(3), j(4)). \end{aligned}$$

If $|d_i| < |d_j|$ our search area consists of the pixels in the parallelogram $ABB'A'$, see Fig. B.2. Their coordinates (i, j) satisfy

$$\begin{aligned} \min(j(1), j(2), j(3), j(4)) &\leq j \leq \max(j(1), j(2), j(3), j(4)), \\ x_1 - b + a &\leq i \leq x_1 + b + a, \end{aligned} \tag{B.1}$$

where $b = h/\sin\phi$ and $a = (j - y_1)\cos\phi/\sin\phi$. In other words, for any j satisfying (B.1) we take pixels between lines A'' and B'' and displace them to the right by a. If a is negative the displacement is to the left.

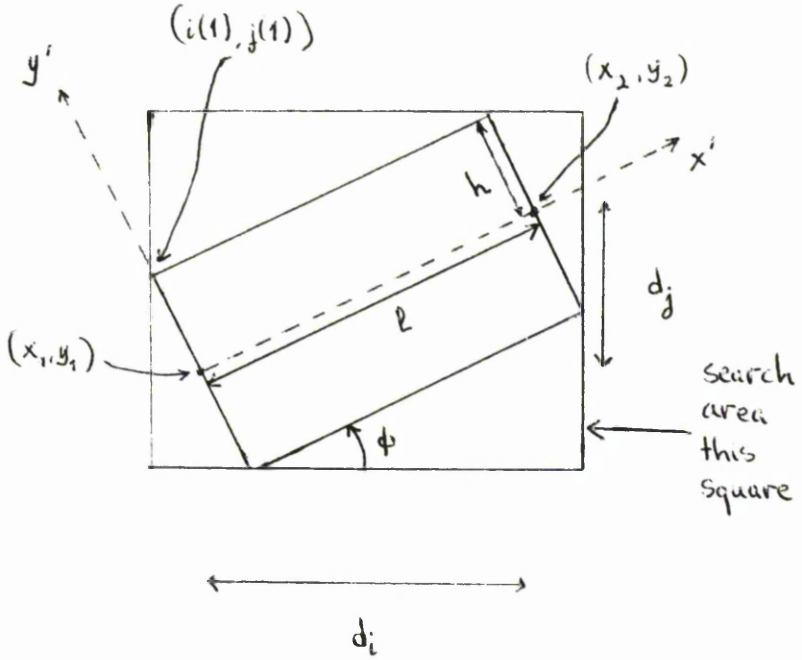


Figure B.1: Locating the rectangle: case of $|d_i| \geq |d_j|$.

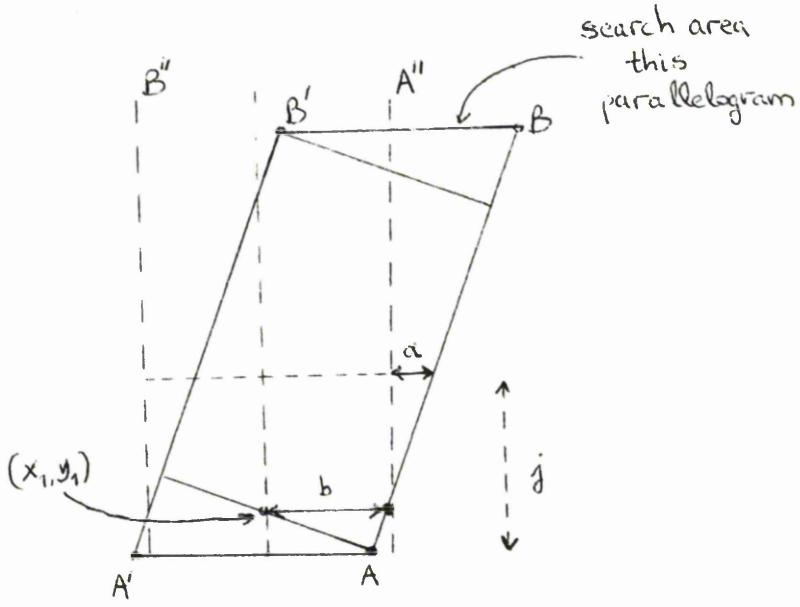


Figure B.2: Locating the rectangle: case of $|d_i| < |d_j|$.

Appendix C

Derivation of the formula for β_3

Here we give the steps that lead to the formula of β_3. After observing y_1 there were two candidates for the location of x_1, both equally plausible. Therefore, u_2 was randomly chosen from between them. The other candidate is denoted by u_2'. Then we observe y_2 and this gives us two candidates for x_2. The one lying on the same side of u_2 as u_2' has the higher posterior probability of being x_2. This point is termed u_3 and the other is u_3' for which, as we have said,

$$\Pr(x_2 = u_3' | \mathbf{y}_2, \mathbf{u}_2) = b_3.$$

Then

$$\beta_3 = \frac{4E\left[b_3(x_2 - u_2)^2\right]}{E\left[(x_2 - u_2)^2\right]}.$$

$E\left[(x_2 - u_2)^2\right] = 2\sigma_1^2 + \sigma_\eta^2$ and all that remains is to find $E\left[b_3(x_2 - u_2)^2\right]$. In

the following we again use the notation $\phi(a|b,c)$ for the value of the p.d.f of $N(b,c^2)$ at point a. Then

$$b_3 = \frac{[\phi(u_3'|u_2,\sigma_\eta) + \phi(u_3'|u_2',\sigma_\eta)]}{[\phi(u_3|u_2,\sigma_\eta) + \phi(u_3|u_2',\sigma_\eta) + \phi(u_3'|u_2,\sigma_\eta) + \phi(u_3'|u_2',\sigma_\eta)]},$$

if we take into account that $b_2 = 0.5$ always. In the remainder we ignore the term $(2\pi\sigma_\eta^2)^{-1/2}$ because it cancels out. Deriving the densities we get

$$\phi(u_3|u_2,\sigma_\eta) = \phi(u_3'|u_2,\sigma_\eta) = \exp\left[-\frac{1}{2\sigma_\eta^2}(u_3-u_2)^2\right] = \exp\left[-\frac{1}{2\sigma_\eta^2}(x_2-u_2)^2\right]$$

because $|u_3 - u_2| = |u_3' - u_2| = \sqrt{2y_2} = |x_2 - u_2|$. Also,

$$\begin{aligned}
&\phi(u_3|u_2',\sigma_\eta) \\
&= \exp\left[-\frac{1}{2\sigma_\eta}(u_3-u_2')^2\right] \\
&= \exp\left\{-\frac{1}{2\sigma_\eta^2}\left[(u_3-u_2)^2 + (u_2-u_2')^2 + 2(u_3-u_2)(u_2-u_2')\right]\right\} \\
&= \exp\left[-\frac{1}{2\sigma_\eta^2}(x_2-u_2)^2\right]\exp\left\{-\frac{1}{2\sigma_\eta^2}\left[(u_2-u_2')^2 + 2(u_3-u_2)(u_2-u_2')\right]\right\}.
\end{aligned}$$

Similarly we derive $\phi(u_3'|u_2',\sigma_\eta)$ and, if we recall the relative locations of u_2, u_2', u_3, u_3' it is not difficult to see that $(u_3-u_2)(u_2-u_2') = -(u_3'-u_2)$ (u_2-u_2). Taking all these remarks into account, after some trivial calculations we get

$$b_3 = \frac{1 + \exp\left[-\frac{1}{2\sigma_\eta^2}(u_2 - u_2')^2\right] \exp\left[-\frac{1}{\sigma_\eta^2}(u_3' - u_2)(u_2 - u_2')\right]}{2 + \exp\left[-\frac{1}{2\sigma_\eta^2}(u_2 - u_2')^2\right]\left\{\exp\left[-\frac{1}{\sigma_\eta^2}(u_3' - u_2)(u_2 - u_2')\right] + \exp\left[\frac{1}{\sigma_\eta^2}(u_3' - u_2)(u_2 - u_2')\right]\right\}}.$$

Again because of the relative location of the points we can say that $(u_3' - u_2)(u_2 - u_2') = \alpha\beta$ where $\alpha = |x_2 - u_2|$ and $\beta = |u_2 - u_2'|$. (This β should not be confused with the limit of β_t). Then,

$$b_3(x_2 - u_2)^2 = \frac{\alpha^2\left[1 + \exp\left(-\frac{\beta^2}{2\sigma_\eta^2}\right)\exp\left(-\frac{\alpha\beta}{\sigma_\eta^2}\right)\right]}{2 + \exp\left(-\frac{\beta^2}{2\sigma_\eta^2}\right)\left[\exp\left(-\frac{\alpha\beta}{\sigma_\eta^2}\right) + \exp\left(\frac{\alpha\beta}{\sigma_\eta^2}\right)\right]}. \tag{C.1}$$

The random quantities affecting (C.1) are α and β. We will get the densities $p(\alpha|\beta)$ and $p(\beta)$ and we will multiply (C.1) by each one of them and integrate in turns to get

$$E\left[b_3(x_2 - u_2)^2\right] = E\left\{E\left[b_3(x_2 - u_2)^2|\beta\right]\right\}.$$

Given β we can see that

$$x = x_2 - u_2|\beta \sim \frac{1}{2}N(0, \sigma_\eta^2) + \frac{1}{2}N(u_2' - u_2, \sigma_\eta^2).$$

Using the change of variables rule we finally get that

$$p(\alpha|\beta) = \frac{\exp\left(-\frac{1}{2\sigma_\eta^2}\alpha^2\right)}{2\sqrt{2\pi\sigma_\eta^2}}\left\{2 + \exp\left(-\frac{\beta^2}{2\sigma_\eta^2}\right)\left[\exp\left(-\frac{\alpha\beta}{\sigma_\eta^2}\right) + \exp\left(\frac{\alpha\beta}{\sigma_\eta^2}\right)\right]\right\}.$$

We see that, given their distance, the locations of u_2 and u_2' do not affect the distribution of α.

For β we observe that $x_1 - u_1 \sim N(0, \sigma_1^2)$ and therefore $\beta = 2\sigma_1\chi_1$, where χ_1 is random variable following the χ-distribution with one degree of freedom. Therefore,

$$p(\beta) = \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left(-\frac{\beta^2}{8\sigma_1^2}\right).$$

All that is needed from here are careful integrations.

Note: For deriving β_4, β_5 and other values the technique is the same, but each time the number of integrations that have to be performed increases by one. In general one has to keep in mind that the random quantities affecting b_t are all the deviations $|x_l - u_l|, l = 1, 2, \ldots, t-1$.