Rejection Control and Sequential Importance Sampling
Author(s): Jun S. Liu, Rong Chen, Wing Hung Wong

Source: *Journal of the American Statistical Association*, Vol. 93, No. 443 (Sep., 1998), pp. 1022-1031
Published by: American Statistical Association
Stable URL: http://www.jstor.org/stable/2669846
Accessed: 26/03/2009 21:49

# Rejection Control and Sequential Importance Sampling

Jun S. LIU, Rong CHEN, and Wing Hung WONG

We discuss ways of combining rejection sampling and importance sampling methods in Monte Carlo computations and demonstrate their usefulness in updating dynamic systems. Specifically, we propose the *rejection controlled sequential importance sampling* (RC-SIS) algorithm, which is designed to simultaneously reduce Monte Carlo variation and retain independent samples in sequential importance sampling. The proposed method is demonstrated by three examples taken from econometrics, hierarchical Bayes analysis, and digital telecommunications. They all show significant improvements over previous results.

KEY WORDS: Blind deconvolution; Econometric disequilibrium model; Gibbs sampler; Markov chain Monte Carlo; Nonparametric Bayes; Sequential imputation; Weighted Markov chain Monte Carlo.

## 1. INTRODUCTION

A dynamic system is defined as a sequence of evolving probability distributions $\pi_t(\boldsymbol{x}_t)$ indexed by discrete time $t = 1, 2, \ldots$, where $\boldsymbol{x}_t = (\boldsymbol{x}_{t-1}, x_t)$. Here the word "evolving" emphasizes on the link between any two consecutive distributions and implies that the change from $\pi_{t-1}$ to $\pi_t$ is not too drastic. In this article we propose a new Monte Carlo method for studying such a system.

A typical dynamic system is the Bayesian learning procedure (or Bayesian expert system) with sequentially observed data. In such a system, the "learning" process is realized mathematically by the Bayes theorem. The corresponding sequence of evolving distributions $\pi_t$ are the posteriors of the parameters of interest conditioned on all the information collected up to time $t$. Such examples have been given by Berzuini, Best, Gilks, and Larissz (1997) and Spiegelhalter and Lauritzen (1990), who built probabilistic expert systems via graphical models for clinical monitoring.

A different avenue that leads to the consideration of a dynamic system, even when the problem is not sequential in nature, is in implementing a divide-and-conquer strategy for solving computational problems. Specifically, when a problem has a complex structure, it is often useful to decompose the target structure into a sequence of simpler but dynamically evolving substructures. Then one can use the information obtained by solving a sequence of easier and smaller problems to help solve the ultimate target problem. The key to the success of this method is the ability of gradually updating the system from the simplest structure to the target structure (Wong 1995). Two Markov chain Monte Carlo (MCMC) schemes that capitalize on this idea are the simulated tempering method (Geyer and Thompson 1995; Marinari and Parisi 1992) and the dynamic weighting method (Wong and Liang 1997). The sequential updating strategy has also been applied to solving difficult simulation/optimization problems, such as the genetic linkage problem (Irwin, Cox, and Kong 1994), and the ancestral inference problem (Geyer and Thompson 1995), and neural network training and the traveling salesman problem (Wong and Liang 1997).

In studying a dynamic system, we usually need to evaluate quantities that can be expressed as expectations of some functions with respect to $\pi_t(\boldsymbol{x}_t)$ at time $t$. In many situations (e.g., Bayesian missing-data problems), $\boldsymbol{x}_t$ also evolves to include more random variables as the system evolves. Because of the complexity of such dynamic systems, a closed-form solution is usually not available, and updating the quantities of interest requires analytical or numerical approximations, among which the Monte Carlo approach plays an important role.

To implement Monte Carlo for a dynamic system, we need to have random samples generated from $\pi_t(\boldsymbol{x}_t)$ at any given time $t$. Because the system is dynamic, static methods that include most of the popular MCMC schemes can be very inefficient. An alternative strategy, sequential imputation (SI), was proposed by Kong, Liu, and Wong (1994) for Bayesian missing-data problems. Similar methods have been used in other application areas, including protein structure simulation (Vasquez and Scheraga 1985), nonlinear state-space models (Gordon, Salmon, and Smith 1993), and econometrics (Hendry and Richard 1991). Although current Monte Carlo methods for dynamic systems are effective for many problems, they have limitations and difficulties. An inherent difficulty of SI is that when $t$ increases, the importance sampling weights get increasingly skewed, and the resulting Monte Carlo estimation tends to be less accurate. Liu and Chen (1995, 1998) developed resampling schemes and other generalizations to alleviate the problem, and Berzuini et al. (1997) suggested using MCMC to accomplish such resampling when calculating weights is difficult. But the resulting samples from these resampling schemes tend to be overcorrelated and quickly become de-

Jun S. Liu is Assistant Professor, Department of Statistics, Stanford University, Stanford, CA 94305. Rong Chen is Associate Professor, Department of Statistics, Texas A&M University, College Station, Texas 77843. Wing Hung Wong is Professor, Department of Statistics, UCLA, 8118 Math Sciences, 405 Hilgard Ave., Los Angeles, CA 90095. Liu's research was partially supported by National Science Foundation (NSF) grants DMS 94-04344 and DMS 95-01570 and the Terman fellowship from Stanford. Chen's research was partially supported by NSF grant DMS 96-26113. Wong's research was partially supported by earmarked grant CUHK 473/95p from the Research Grant Council of Hong Kong and by NSF grant DMS-9703918. The authors are grateful to Jerry Friedman, three referees, and the associate editor for insightful comments.

Journal of the American Statistical Association
September 1998, Vol. 93, No. 443, Theory and Methods

generate as the number of resamplings increases. Estimating Monte Carlo variations also becomes difficult.

In this article we propose incorporating rejection control steps in a sequential importance sampling scheme to improve performance. By rejecting those samples with small weights at an early stage and properly adjusting the weights for the remaining samples, the rejection control method provides an effective way to build up a good importance sampling distribution for dynamic system. We have applied the method to three examples from different application areas. All of them have shown significant improvements over the plain SI method.

Section 2 describes the main algorithm of sequential importance sampling with rejection control. Section 3 discusses issues related to actual implementations. Section 4 shows how the method works for several examples taken from econometrics, Bayesian nonparametric inference, and signal processing. The Appendix presents the necessary theoretical proofs.

## 2. THE MAIN ALGORITHMS

### 2.1 Sequential Importance Sampling

For a dynamic system $\pi_t(\boldsymbol{x}_t)$, We first reformulate the SI method of Kong et al. (1994) as a more general scheme, sequential importance sampling (SIS). Suppose that $X_1$ can be drawn from $g_1(x_1)$ (also denoted by $g_1(x_1|x_0)$ for notational simplicity), which is close to $\pi_1(x_1)$, and that $X_t$ can be drawn from $g_t(x_t \mid \boldsymbol{x}_{t-1})$, which mimics $\pi_t$. Then with $w_0 = 1$, the following steps can be implemented sequentially for $t = 1, \ldots, T$:

- Draw $X_t = x_t$ from distribution $g_t(x_t \mid \boldsymbol{x}_{t-1})$.
- Compute

$$u_t = \frac{\pi_t(\boldsymbol{x}_t)}{\pi_{t-1}(\boldsymbol{x}_{t-1}) g_t(x_t \mid \boldsymbol{x}_{t-1})}$$

and

$$w_t = w_{t-1} u_t. \tag{1}$$

Here $u_t$ is called an "incremental weight." At stage $t$, we obtain a sample $\boldsymbol{x}_t$ of the random vector $\mathbf{X}_t$ from the trial distribution

$$g_t(\boldsymbol{x}_t) = \prod_{s=1}^{t} g_s(x_s \mid \boldsymbol{x}_{s-1}). \tag{2}$$

Its importance sampling weight $\pi_t(\boldsymbol{x}_t)/g_t(\boldsymbol{x}_t)$ is easily shown to be $w_t$. The attractive feature of this scheme is that it allows us to build up an importance sampling distribution for $\boldsymbol{x}_t$ sequentially in a simple fashion. For Bayesian missing-data problems, Kong et al. (1994) chose $g_t$ as $\prod_{s=1}^{t} \pi_s(x_s \mid \boldsymbol{x}_{s-1})$. Liu and Chen (1998) provided a systematic study and several extensions of this and other related methods.

If the SIS procedure is carried out independently for $m$ times, then $m$ iid draws, $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(m)}$, from $g_t$ are obtained with respective weights $w_t^{(1)}, \ldots, w_t^{(m)}$. Then for any function $h(\boldsymbol{x}_t)$, $\theta = E_{\pi_t}\{h(\mathbf{X}_t)\}$ can be approximated by

$$\hat{\theta}_m = \frac{\sum_{j=1}^{m} w_t^{(j)} h(\boldsymbol{x}_t^{(j)})}{\sum_{j=1}^{m} w_t^{(j)}}. \tag{3}$$

Henceforth we call the $\boldsymbol{x}_t^{(j)}$ *streams*. These streams are said to be properly weighted by the $w_t^{(j)}$ with respect to $\pi_t$ if $\hat{\theta}_m$ is $\sqrt{m}$ consistent for $\theta$ for any integrable function $h$.

It should be noted that the SIS method is valid only if all of the weights $w_t$ have finite variance. Otherwise, (3) may have infinite variance. With the Bayesian missing-data setting and the $g_t$ chosen as by Kong et al. (1994), one can show that the $w_t$ is proportional to the product of a sequence of predictive probabilities and is often bounded from above—and, thus has finite variance.

As $t$ increases, the distribution of $w_t$ typically becomes more skewed, implying that the chi-squared distance between $g_t$ and $\pi_t$ (equivalent to the coefficient of variation, $cv^2$, of $w_t$) increases. Consequently, many streams carried by the SIS will have minimal impact on the final estimation. It is thus desirable to discard them at an earlier stage. In the following section we propose a rejection scheme to achieve this goal without creating any bias. In doing so, we can continuously "guide" the SIS in light of $\pi_t$.

### 2.2 Rejection Control Operation

Suppose that at time $t$, streams $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(m)}$ are drawn from $g_t$ and

$$w_t^{(j)} \equiv w_t(\boldsymbol{x}_t^{(j)}) \propto \pi_t(\boldsymbol{x}_t^{(j)})/g_t(\boldsymbol{x}_t^{(j)}), \quad \text{for} \quad j = 1, \ldots, m.$$

For any given threshold value $c > 0$, we introduce the following operation to find rejection control at time $t$ (RC($t$)):

- For $j = 1, \ldots, m$, accept stream $\boldsymbol{x}_t^{(j)}$ with probability

$$r_t^{(j)} = \min\left\{1, \frac{w_t^{(j)}}{c}\right\}.$$

- If the $j$th steam $\boldsymbol{x}_t^{(j)}$ is accepted, then attach to it a new weight $w_t^{(*j)} = p_c w_t^{(j)}/r_t^{(j)}$, where

$$p_c = \int \min\left\{1, \frac{w_t(\boldsymbol{x}_t)}{c}\right\} g_t(\boldsymbol{x}_t)\, d\boldsymbol{x}_t.$$

Note that $p_c$ is maintained only for conceptual clarity, not for computational need. This is because $p_c$ is a constant across all of the streams and is not needed for the evaluation of the ratio estimate (3).

The foregoing RC scheme can be viewed as a technique for adjusting the trial density $g_t$ in light of current importance weights. The new trial density $g_t^*(\boldsymbol{x}_t)$ resulting from this adjustment is expected to be closer to the target function $\pi_t(\boldsymbol{x}_t)$. In fact, it can be seen that

$$g_t^*(x) = p_c^{-1} \min\{g_t(\boldsymbol{x}_t), \pi_t(\boldsymbol{x}_t)/c\}. \tag{4}$$

In Section 5 we show that $g_t^*$ is indeed closer to $\pi_t$ in terms of chi-squared distance. After applying rejection control, we typically will have fewer than $m$ samples. More samples can be drawn from either $g_t(\boldsymbol{x}_t)$ or $g_t^*(\boldsymbol{x}_t)$ (via rejection control) to make up for the rejected streams.

### 2.3 The Rejection Controlled–Sequential Importance Sampling Algorithm

For a given checking schedule $0 < t_1 < t_2 < \cdots < t_k < \ldots$ and a given threshold sequence $(c_1, c_2 \ldots)$, we can conduct the RC operation described in the previous subsection at each check point $t_k$ for the dynamic system. The complete procedure can thus be summarized as follows:

1. At each checkpoint $t_k$, start RC($t_k$) as described in Section 2.2 with the threshold value $c = c_k$. If stream $\boldsymbol{x}_{t_k}^{(j)}$ with weight $w_{t_k}^{(j)}$ passes this checkpoint, then proceeds the same way as in a standard SIS but with the old weight replaced by $w_{t_k}^{(*j)} = \max\{w_{t_k}^{(j)}, c_k\}$.
2. For each rejected stream, start a new stream from time $t = 0$ and let it pass through all the check points at $t_1, \ldots, t_k$, with threshold values $c_1, \ldots, c_k$. If rejection occurs in any middle checkpoint, then restart again.

Note that after the first rejection control [i.e., RC($t_1$)], the sampling distribution $g_t^*(\boldsymbol{x}_t)$ for $X_t^{(j)}$ is no longer the same as the one described in (2). In Section 5 we show that for any time $t$, the streams $\boldsymbol{x}_t^{(j)}$ resulting from rejection controlled sequential importance sampling (RC-SIS) are properly weighted with respect to $\pi_t$ by their modified weights $w_t^{(*j)}$.

### 2.4 Efficiency Issues

In light of the work of Casella and Robert (1996), one can view an importance sampler as a Rao–Blackwellization of the corresponding rejection method that is always *statistically* more efficient. Thus it is not obvious why incorporating a rejection control step in SIS can be useful. The following example shows that the key benefit of RC in SIS is to help simulate *future* samples more efficiently.

*Example 2.1.* Consider a simple dynamic system in which the evolving distribution is

$$\pi_t(x_1, \ldots., x_t) = \frac{1}{(2 + 2a_t)^t}$$

for

$$(x_1, \ldots, x_t) \in [-1 - a_t, 1 + a_t]^t.$$

Here $a_t = 1/t$ for $t = 1, \ldots, 19$ and $a_{20} = 0$. Hence at the end of process ($t = 20$), the target distribution $\pi_{20}(\boldsymbol{x}_{20})$ is uniform on the 20-dimensional cube $[-1, 1]^{20}$. At each stage, the "up-to-date" density $\pi_t$ approaches $\pi_{20}$ but has a longer tail. The symmetry for the $\pi_t$ is not essential, but all of the $\pi_t$ must contain the support of the target distribution.

We choose $g_t(x_t|\boldsymbol{x}_{t-1}) = \pi_t(x_t|\boldsymbol{x}_{t-1})$, which is uniform on $[-1 - a_t, 1 + a_t]$. Because the weight for each stream is either 0 or 1, to study the efficiency of the plain SIS we need only count the number of streams with nonzero weights. Suppose that at stage $t - 1$ a stream $\boldsymbol{x}_{t-1}$ has nonzero weight. Then for it to have nonzero weight at stage $t$, we need all of its components $x_1, \ldots, x_{t-1}$ to fall in the interval $[-1 - a_t, 1 + a_t]$. Hence if we let $N_t$ denote the number of streams with nonzero weights left at time $t$, then

$$E(N_t) = E(N_{t-1}) \left( \frac{1 + a_k}{1 + a_{k-1}} \right)^{t-1}.$$

Consequently, at the end of process the expected number is

$$\begin{aligned} E(N_{20}) &= \frac{(1 + a_{20})^{20}}{(1 + a_1)(1 + a_2) \cdots (1 + a_{20})} E(N_0) \\ &= E(N_0)/20 = .05m, \end{aligned}$$

where $m$ is the number of streams with which we start. Note that with standard SIS, we must carry every stream to the end.

RC-SIS enables one to discard those streams with zero weights at an early stage. Let $T$ denote the length of a stream when either its weight first becomes 0 or it survives after 20 stages. Then $T$ is a random variable with probability distribution

$$P(T = t) = \frac{(1 + a_t)^{t-1}}{(1 + a_1) \cdots (1 + a_{t-1})} \left[ 1 - \left( \frac{1 + a_{t+1}}{1 + a_t} \right)^t \right]$$

for

$$2 \le t \le 19,$$

$P(T = 1) = (a_1 - a_2)/(1 + a_1)$, and $P(T = 20) = \prod_{t=1}^{19}(1 + a_t)^{-1}$. Hence the expected length of the stream is $E(T) = 6.27$. So using rejection control is equivalent to simulating streams of average length 6.27 compared to that of length 20 without using RC. The amount of computing needed is only about 30% of that needed by plain SIS.

This simple example illustrates two important requirements for RC-SIS to work well:

- The change from the current distribution $\pi_t$ to the future distribution $\pi_{t+1}$ should be modest. Indeed, if the sequence of $\pi_t$ is not well behaved, (e.g., with $\pi_t$ being uniform $[-2, 2]$ for $1, \ldots, 19$), then RC-SIS fails.
- The trial distribution $g_t$ must be reasonably similar to $\pi_t$.

## 3. IMPLEMENTATION OF REJECTION CONTROLLED SEQUENTIAL IMPORTANCE SAMPLING

It is usually difficult to prescribe a set of effective threshold values $c_1, \ldots, c_k, \ldots$ in advance. Our strategy is to make $c_k$ a function of the current weights $w_{t_k}^{(j)}$. A useful empirical formula is

$$c_k = p_1 \min_j(w_{t_k}^{(j)}) + p_2 \bar{w}_{t_k} + p_3 \max_j(w_{t_k}^{(j)}),$$
$$p_1 + p_2 + p_3 = 1, \qquad p_l \ge 0. \quad (5)$$

where $\bar{w}_{t_k}$ is the current average of the $w_t^{(j)}$. Percentiles of the weights can also be used as threshold values. Note that this approach introduces only negligible bias and correlation among the streams.

The values of $p_l$ can be adjusted to achieve a low-threshold or a high-threshold effect. When the threshold is low, the acceptance probability is high, and each RC step needs less time to complete, and vice versa. But with low threshold, the sampling distribution $g_s$ may not be sufficiently modified. We have implemented both methods and the results are comparable (see Sec. 4 for detailed examples). However, the low-threshold strategy seems to be more robust with respect to the total computing time spent. It is sensible to use a relatively high threshold at an early stage and gradually decrease it, because much more computation is required to make up the lost streams in a later stage than in an early stage.

The checking schedule can be either deterministic or dynamic. For deterministic schedule, we give values of $(t_1, \ldots, t_k)$ in advance. For example, we can design that $t_i = i \times L$, for $i = 1, 2, \ldots$, where $L$ is the lag value determined by how difficult the problem of interest is. Whereas a difficult problem demands that $L = 1$, in an easy one $L$ can be as large as 30 to 50.

The dynamic checking schedule can be achieved by prescribing a sequence of values $d_1, \ldots, d_t, \ldots$ and monitoring the coefficient of variation $cv^2(t)$ of the weights $w_t^{(j)}$ constantly. Once the event $cv^2(t) \geq d_t$ is observed, $t$ becomes a checkpoint, and we conduct a RC operation.

The $cv^2$ sequence $d_k$ should be adjusted to accommodate the threshold sequence $c_k$. When the $c_k$ are high, the number of times for rejection control should be small. Thus we choose a geometric sequence $d_t = a_t d_{t-1}$ to control the checking schedule, where $a_t = 1$ if $cv^2(t-1) < d_{t-1}$ and $a_t = \rho$ otherwise. The constant $\rho$ was chosen around $1.1 \sim 1.5$. When the $c_k$ are low, we use a sublinear function of time,

$$d_t = d_0 + t^r / e_0,$$

where $o < r < 1$, to control the checking schedule.

## 4. EXAMPLES

### 4.1 Econometric Disequilibrium Models

Fair and Jaffee (1972) first proposed the disequilibrium model as an alternative to the the postwar mainstream approach to economics, the equilibrium method. (See Quandt 1982, 1988 for reviews and discussions.) In this example we show how the simple dynamic disequilibrium model used by Hendry and Richard (1991) can be analyzed using the RC-SIS method. The conceptual framework illustrated here should be easy to generalize to more complicated settings.

Let $q_t' = (q_{1t}, q_{2t})$, $t = 0, 1, \ldots$, be a sequence of bivariate normal random variables with the conditional means and covariance matrices given by the recursive relationship

$$E(q_{it} \mid q_{t-1}) = \alpha_i q_{it-1}, \qquad \operatorname{cov}(q_t \mid q_{t-1}) = \mathbf{I},$$

for $t = 1, \ldots, T$, where $\mathbf{I}$ is the identity matrix and $0 \leq \alpha_i < 1$. The observed data for this model are $y_t = \min\{q_{1t}, q_{2t}\}$, for $t = 0, 1, \ldots, T$. For simplicity in presentation, the initial states $q_{11}$ and $q_{21}$ were taken to be 0 and assumed known. Of interest is the likelihood function or the posterior distribution of $\theta = (\alpha_1, \alpha_2)$.

Let $\lambda_t = \max\{q_{1t}, q_{2t}\}$, and let $\delta_t$ be $i$ if $y_t = q_{it}$. If we write $x_t = (\lambda_t, \delta_t)$, $\boldsymbol{x}_t = (\theta, x_1, \ldots, x_t)$, and take a Bayesian approach, then the target distribution of interest at time $t$ is

$$\pi_t(\boldsymbol{x}_t) = p(\boldsymbol{x}_t \mid \boldsymbol{y}_t) \propto p(\theta) \prod_{s=1}^{t} p(x_s \mid \theta, y_s).$$

When the prior distribution for $\theta$ is uniform, the marginal $\pi_t(\theta)$ is also the likelihood function of $\theta$. If we have obtained multiple samples $\boldsymbol{x}_T^{(j)}$ from $\pi_t$, then the likelihood function of $\theta$ can be approximated in two ways: by using the marginal histogram of the $\theta$ (this is achievable because $\boldsymbol{x}_T = (\theta, x_1, \ldots, x_T)$), and by using a mixture of complete data posteriors (Rao–Blackwellization). The second method apparently is superior. Because obtaining direct samples from $\pi_T$ is difficult, we use the SIS strategy. That is, we simulate $X_T$ from a sequentially built-up trial distribution $p_T$ and then weight the $X_T$ properly. Finally, the likelihood function of $\theta$ can be approximated by a *weighted mixture* of complete-data posteriors.

The SIS can be carried out as follows: generate $\theta$ from its prior, (e.g., the uniform distribution); then for $t = 1, 2, \ldots$, we use

$$g_t(x_t \mid \boldsymbol{x}_{t-1}) = p(x_t \mid \theta, \boldsymbol{x}_{t-1}, \mathbf{y}_{t-1}, y_t).$$

The incremental weight $u_t$ can be computed up to a normalizing constant as

$$u_t \propto p(y_t \mid \theta, \boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}).$$

More details are given in Section 5.

With initial value $q_{11} = q_{21} = 0$, Liu and Chen (1998) simulated 50 observations from the model with $\alpha_1 = \alpha_2 = .6$. Assuming symmetry (i.e., $\alpha_1 = \alpha_2$), they performed the plain SIS method with $m = 10{,}000$ and resulted in a final $cv^2$=5.13. The procedure took 8.16 cpu seconds on a Silicon Graphics workstation with R10000 microprocessor. Here we first demonstrate how the method of RC-SIS with a dynamic checking schedule can be applied to the same setting, and then apply it to a more difficult case.

A geometric $cv^2$ threshold sequence with $d_1 = 1.0$ and $\rho = 1.1$ (see Sec. 3) were first used together with the rejection threshold sequence described by (5), with $p_2 = p_3 = .5$ and $p_1$=0 (high thresholding). The RC operated four times at time $t = 11, 32, 41, 46$. The algorithm took 13.8 seconds on the same Silicon Graphics workstation, and the resulting $cv^2$ is 1.05. Using a rule of thumb in survey sampling, the efficiency for using the rejection control increases at least threefold. When using $d_1 = 1.0$ and $\rho = 1.4$, RC operated twice at $t = 11$ and 32, and the resulting $cv^2 = 2.04$. It took 9.8 seconds on the same machine.

A more dramatic improvement shows up in the asymmetric case. Here we simulated 50 data points $y_1, \ldots, y_{50}$ from the model with $\alpha_1 = .9$ and $\alpha_2 = 0$. Because the parameters are unidentifiable, we constrained that $\alpha_1 > \alpha_2$. In this case the plain SIS method failed miserably, with a $cv^2$

Table 1. Histogram of the Tack Data

| Frequencies | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Counts | 0 | 3 | 13 | 18 | 48 | 47 | 67 | 54 | 51 | 19 |

hovering around 200 to 500 and resulting in a very inaccurate estimate of the quantity of interest. We applied RC-SIS with a $cv^2$ threshold sequence of $d_1 = 2.1$ and $\rho = 1.4$. The resulting final $cv^2$ is 3.9 with 92 seconds on the same machine.

### 4.2 Hierarchical Bayes Model Analysis

Empirical Bayes and hierarchical modeling have attracted much attention from statisticians since the mid-1970s. Its usefulness in data analysis has been demonstrated repeatedly (Maritz and Lwin 1989). The simplest general setting for a hierarchical model is as follows: The $Y_i$ are independent observations from $p_i(y \mid \zeta_i)$, the $\zeta_i$ are iid draws from an unknown distribution $F$, and $F$ is determined by another set of estimable parameters. In many cases, however, it is of interest to treat $F$ nonparameterically. (See Laird 1978, and Laird and Louis 1987 for early developments.) A Bayesian solution to the same problem can be obtained by putting a Dirichlet process prior on $F$. (See Escobar 1994 and Liu 1996 for recent discussions.)

To show how the RC-SIS works, we consider the nonparametric hierarchical Bayes model for binary data (Berry and Christensen 1979). The method illustrated here can be readily extended to other hierarchical Bayes models, either parametric or nonparametric ones. In a binomial-Dirichlet nonparametric Bayes model, each observation $(l_i, y_i)$ is treated as $y_i \sim \text{bin}(l_i, \zeta_i)$, and the $\zeta_i$ are iid samples from an unknown distribution $F$. Furthermore, $F$ is assumed to follow a Dirichlet process $\mathcal{D}_\alpha$ a priori, where $\alpha$ can be any nonnegative measure on $[0, 1]$ but is assumed to be Lebesgue for simplicity.

In this example, the infinite dimensional parameter $F$ can be integrated out via a Polya-urn argument. (See Ferguson 1973 for details.) Therefore, the involved dynamic variable is $\boldsymbol{x}_t = (\zeta_1, \ldots, \zeta_t)$, and the dynamic distribution of interest is

$$\begin{aligned}\pi_t(\boldsymbol{x}_t) &= p(\zeta_1, \ldots, \zeta_t \mid y_1, \ldots, y_t) \\ &\propto \prod_{s=1}^{t} \zeta_s^{y_s}(1-\zeta_s)^{l_s - y_s}\alpha(\zeta_1) \\ &\quad \times \prod_{s=2}^{t}\left[\alpha(\zeta_s) + \sum_{j=1}^{s-1}\delta_{\zeta_j}(\zeta_s)\right].\end{aligned}$$

With samples from such a distribution, one could approximate, say, the posterior expectation of the unknown density $dF$,

$$\begin{aligned}E[dF(z) \mid Y] &= \alpha(z) + E[(\delta_{\zeta_1}(z) + \cdots \delta_{\zeta_t}(z)) \mid Y] \\ &= \alpha(z) + \sum_{i=1}^{t} p_i(z \mid Y),\end{aligned}$$

where $Y$ is the collection of all observations and $p_i(| \, Y)$ is the marginal posterior distribution of $\zeta_i$ evaluated at $z$.

We consider the dataset of Beckett and Diaconis (1994), which comprises 320 binary strings of length 9 from rolls of thumbtacks. A 1 was recorded if the tack landed point up, and a 0 was recorded if the tack landed point down. In a nonparametric hierarchical Bayes setting, the nine outcomes in each string are treated as independent Bernoulli observations with a common parameter $\zeta_i$, and $\zeta_i \sim F$. The data of 320 9-tuples can be reduced to their respective sufficient statistics, the total number of ups in each string, and is shown in Table 1. Liu (1996) applied the plain SIS with $m = 10{,}000$ to the problem, and the resulting $cv^2$ was in the range of 30 to 50. We repeated the same SIS procedure 250 more times for the same dataset but with randomly shuffled orderings and obtained 250 $cv^2$s with minimum 23.6, median 68.8, mean 126.2, and maximum 2,016.6.

With $m = 10{,}000$, the average computing time for one complete SIS process was 76 seconds on a Silicon Graphics R10000 microprocessor. The histogram of the logarithms of these $cv^2$'s is plotted in Figure 1. This figure shows that the original ordering was very favorable to the plain SIS and that Liu (1996) was lucky to have obtained a good numerical result. Quite often, however, the SIS with an unfavorable process ordering can result in a disastrous $cv^2$ and lead to a huge Monte Carlo variation. The rejection control method can make the SIS process much more stable.

We picked one of the 250 results (from different data ordering) at random and observed that the resulting $cv^2$ for the plain SIS was in the range of 120–170. We applied the RC-SIS to this new randomly ordered dataset, which took 550 seconds of computing time, and the resulting $cv^2$ was 7.6. It was a threefold efficiency gain compared to the plain SIS after the extra computing time was taken into account.

The following check schedules were used for low thresholding: $d_t = .5 + t^{.3}$ and $d_t = .5 + t^{.5}/3$. In high thresholding, the geometric sequence with $d_1 = 2.1$ and $\rho = 1.2$ was used. The rejection threshold value $c_k$ for each check-

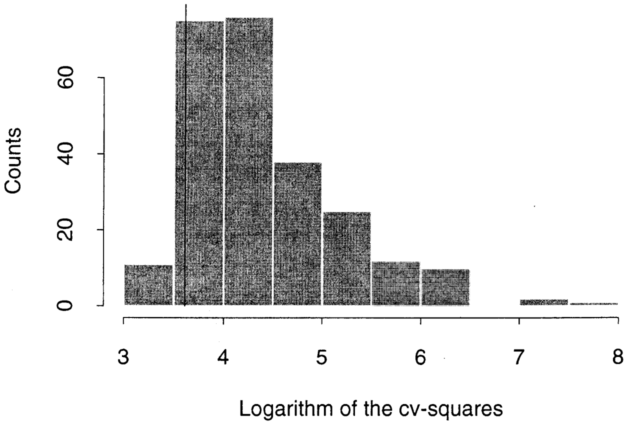

Figure 1. The Histogram of the log($cv^2$) Resulting From Processing 250 Randomly Permuted Tack Data of Beckett and Diaconis (1992) by SIS. The vertical line indicates the log($cv^2$) resulting from the original data ordering.

*Table 2. Three-Level Blind Deconvolution Simulation Results (MAP): RC-SIS With Deterministic and Dynamic Checking Schedules Compared to the SIS With Rejuvenation*

| | *Real-time MAP* | | | | | *Final MAP* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *RC-SIS* | | | | *SIS-r* | *RC-SIS* | | | | *SIS-r* |
| | *Schedule* | | | | | *Schedule* | | | | |
| *Errors* | *50* | *30* | *10* | *cv* | | *50* | *30* | *10* | *cv* | |
| 0–2 | 7 | 17 | 24 | 29 | 16 | 74 | 125 | 157 | 168 | 132 |
| 3–5 | 57 | 69 | 89 | 91 | 98 | 82 | 50 | 35 | 26 | 56 |
| 6–8 | 62 | 60 | 61 | 56 | 57 | 32 | 21 | 7 | 5 | 8 |
| 9–11 | 38 | 31 | 15 | 17 | 16 | 8 | 3 | 1 | 1 | 0 |
| 12–15 | 27 | 15 | 11 | 7 | 9 | 4 | 1 | 0 | 0 | 1 |
| 16–20 | 8 | 7 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 21–25 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 26–30 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

point $t_k$ was computed using (5) with $(p_1, p_2, p_3)$ depending on $t_k$. We took $p_1 = 0$, $p_2 = t_k/(t_k + 5)$, and $p_3 = 5/(t_k + 5)$ for high thresholding and took $p_1 = t_k/(t_k + 10)$, $p_2 = 8/(t_k + 10)$, and $p_3 = 2/(t_k + 10)$ for low thresholding.

## 4.3 Blind Deconvolution

In digital communications, the following linearly degraded moving-average system is common:

$$y_t = \sum_{i=0}^{q} h_i x_{t-i} + \varepsilon_t, \qquad t = 1, 2, \ldots,$$

where $\varepsilon_t$ is Gaussian white noise with constant variance $\sigma^2$. The $y_t$'s are sequentially observed output signals coming through the system. The main objective is to reconstruct the input signal $x_t$, which is discrete with known levels $s_1, \ldots, s_m$, without knowing the system coefficients $h_0, \ldots, h_m$ and without training data; that is, blind (see Donoho 1981; Giannakis and Mendel 1989; Lii and Rosenblatt 1982).

Chen and Li (1995) first proposed a Bayesian solution to the problem via Gibbs sampling. Because the algorithm is static, it is not applicable to real-time communication, though it is useful in seismology and underwater acoustics. Liu and Chen (1995) proposed an SIS method for real-time blind deconvolution and provided detailed formulas. In particular, they used a multivariate Gaussian prior for the coefficients $(h_i)$ and integrated them out in the SIS procedure. Hence the dynamic variables involved in this example are only those true signals, [i.e., $\boldsymbol{x}_t = (x_1, \ldots, x_t)$], and the dynamic system is

$$\pi_t(\boldsymbol{x}_t) \propto \int \cdots \int p(y_1, \ldots, y_t \mid \boldsymbol{x}_t, h_0, \ldots, h_m) \times p(h_0, \ldots, h_m) p(\boldsymbol{x}_t)\, dh_0 \ldots dh_m.$$

*Table 3. Three-Level Blind Deconvolution Simulation Results (Real-Time MAP Estimator) for Different Total Number of Streams*

| | *Number of streams* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *1,000* | | *500* | | *200* | | *100* | | *50* | |
| *Errors* | *RC* | *rj* | *RC* | *rj* | *RC* | *rj* | *RC* | *rj* | *RC* | *rj* |
| 0–2 | 29 | 16 | 27 | 24 | 26 | 20 | 17 | 14 | 12 | 13 |
| 3–5 | 91 | 98 | 95 | 74 | 80 | 64 | 73 | 57 | 58 | 38 |
| 6–8 | 56 | 57 | 57 | 64 | 61 | 44 | 57 | 38 | 51 | 20 |
| 9–11 | 17 | 16 | 14 | 21 | 20 | 21 | 31 | 21 | 33 | 12 |
| 12–15 | 7 | 9 | 6 | 10 | 12 | 19 | 14 | 13 | 19 | 11 |
| 16–20 | 0 | 1 | 1 | 4 | 1 | 5 | 7 | 10 | 16 | 8 |
| 21–25 | 0 | 0 | 0 | 1 | 0 | 3 | 1 | 6 | 7 | 5 |
| 26–30 | 0 | 1 | 0 | 2 | 0 | 4 | 0 | 7 | 0 | 6 |
| >30 | 0 | 2 | 0 | 0 | 0 | 20 | 0 | 34 | 4 | 87 |

NOTE: RC-SIS with dynamic schedule, denoted by "RC," compared to SIS-rejuvenation, denoted by "rj."

Noticing that the standard SIS often has extremely skewed weight as more and more signals are processed, Liu and Chen (1995) suggested using a rejuvenation approach that resamples the streams according to their normalized weight, when $cv^2$ gets large. Problems associated with such an approach are that the number of distinctive samples decreases after rejuvenation, and that evaluating Monte Carlo variation is difficult. In this and the next sections, we demonstrate how RC-SIS comes to a rescue, not only controlling the effective sample size (following the custom of survey sampling, this is defined as the ratio of the actual sample size over $1 + cv^2$), but also providing independent (hence distinct) samples.

Here we demonstrate the RC-SIS procedure for the three-level example of Liu and Chen (1995) . The true blurring equation is

$$y_t = x_t + .8x_{t-1} - .4x_{t-2} + \varepsilon_t,$$

with input signals iid from the set {0, 1, 3} with equal probability. The system signal-to-noise ratio is fixed at 15 dB, which makes the standard deviation of the white noise around .3. The prior distribution for the coefficients is a product of independent N(0, 1,000)'s. We simulated 200 sequences, each with 100 observations from the system, and tested the RC-SIS procedure with different checking schedules. The total number of streams used was 1,000, and the rejection thresholds sequence used was $(70\%, 65\%, 60\%, \ldots, 20\%, \ldots, 20\%)$ quantiles of the current weights. We tested the deterministic checking schedule with $t_i = i \times L$ for $L = 10, 30, 50$ and the dynamic checking schedule using $cv^2$ threshold sequence with $d_1 = m/20$ (i.e., effective sample size is 6) and $\rho = 1.05$. Table 2 presents the number of misclassification using real-time MAP (i.e., maximum a posteriori) estimator with delay

Table 4. Quantiles of Number of Misclassification for the 16-Level Example

| | 1–200 | | | 51–200 | | | 101–200 | | |
|---|---|---|---|---|---|---|---|---|---|
| Percentile | 25% | 50% | 75% | 25% | 50% | 75% | 25% | 50% | 75% |
| | | | | Real-time (delay 3) | | | | | |
| No. of misclassifications | 40.00 | 66.50 | 92.25 | 6.00 | 29.00 | 56.25 | .75 | 6.50 | 30.00 |
| | | | | Final | | | | | |
| No. of misclassifications | 8.00 | 11.00 | 22.00 | 0 | 2.00 | 8.25 | 0 | 1.00 | 5.25 |

$d = 3$ (i.e., at time $t$ the signal $x_{t-3}$ is estimated with current weight using MAP) and final-weight MAP. For comparison, we also included the result from using Liu and Chen's (1995) method, the SIS with rejuvenation (henceforth, SIS-r), for the same sequences, with resampling invoked when the effective sample size is less than 3. Table 2 shows that dynamic scheduling performs better than deterministic scheduling, and RC-SIS was generally better than SIS-r.

To study the effect of total number of streams $m$, we applied RC-SIS and SIS-r to the aforementioned system, for different total number of streams $m$. Again, 200 simulated sequences of length 100 were used. Table 3 shows the number of misclassified signals using the real-time (delay 3) MAP estimator. Dynamic scheduling with $d_1 = m/20$ and $\rho = 1.05$ was used for RC-SIS. The effective sample size threshold for SIS-r was 3. Table 3 reveals robust performances of RC-SIS for small $m$, in which case the SIS-r behaved very badly. We see that RC-SIS with $m = 100$ performed as good as SIS-r with $m = 1{,}000$. In fact, there were cases for the SI-r where the error rate exceeded 30%, implying that there was not even one "good" stream. In such cases, resampling from the existing streams could not help, whereas RC-SIS helped by regenerating new streams from the beginning, and it hence works well with small number of streams and small memory.

### 4.4 A Tougher Test

A more convincing demonstration of the usefulness of the method comes from solving the blind deconvolution problem when input signals have 16 levels. The main difficulty that this problem entails is the lack of information at the beginning stage of the SIS. Because of the large number of possible input signal levels, the initial streams sampled based on the prior distribution tend to be too diffused, and most of the streams are "bad." Liu and Chen (1995) used the Gibbs sampler at the beginning stage of the signal processing to finesse such a difficulty. But this extra MCMC step sacrifices real-time feature of the scheme and also induces dependencies among the streams.

We implemented the RC-SIS for the 16-level system,

$$y_t = .9162x_t - .1833x_{t-1} + .4812x_{t-2} - .1987x_{t-3} + \varepsilon_t,$$

where the input signals $x_t$ are uniformly distributed on the set $\{-15, -13, \ldots, -1, 1, \ldots 13, 15\}$. The noise $\varepsilon_t$ is independent Gaussian. The signal-to-noise ratio is controlled at 30 dB, which makes the noise standard deviation around .31.

Table 5. Percentiles of cpu Time Used in the Example in Section 4.4

| Percentile | 10% | 25% | 50% | 75% | 90% |
|---|---|---|---|---|---|
| cpu | 40.28 | 45.76 | 52.46 | 62.45 | 73.43 |

Along with RC, we adopted a multistage processing strategy. First, we applied the RC-SIS procedure to the first 50 observed output signals, pretending that they were generated from a simpler four-level system with levels at $\{-12, -4, 4, 12\}$. A nearly noninformative prior on the system coefficients was used. Second, we retained the resulting posterior mean and 100 × (posterior covariance matrix) of the system coefficients and used them as the prior mean and prior covariance matrix of the system coefficients for processing the 16-level system. This two-stage processing scheme consumed much less computing time than processing the 16-level system directly (with nearly noninformative prior for the system coefficients) for the same performance.

For the four-level preprocessing, we set the checking schedule as $\{10, 20, \ldots, 50\}$ with rejection thresholds at $\{95\%, 90\%, \ldots, 75\%\}$ percentiles of the updated weights. For the final processing of the 16-level problem, we set the checking schedule as $\{20, 40, \ldots, 200\}$ with rejection thresholds at $\{80\%, 70\%, \ldots, 20\%, \ldots, 20\%\}$ percentiles of the weights. The real-time classification delay was set at 3, and the number of streams $m$ was given as 100.

We simulated 200 signal sequences, each with 200 blurred signals. Table 4 displays the quartiles of the numbers of misclassification for real-time restoration and final restoration. The numbers of misclassification were counted for three time intervals 1–200, 51–200 and 101–200. Table 5 shows the percentiles of cpu time used (on a SunSparc 20) for processing each simulated series. Our method performed very well for this simulated case. In contrast, the plain SIS method failed to produce any meaningful result even with $m = 2{,}000$, and we had to use the Gibbs sampler to initialize the signal restoration.

## 5. DISCUSSION

From what we have seen in previous sections, the RC procedure is effective when combined with SIS. It not only saves memory space, but also enables the trial distribution to approach the target more quickly. Many problems can be formulated as a dynamic system and solved using techniques described in this article; for example, those examples of Berzuini et al. (1997), Irwin et al. (1992), and Wong and Liang (1997). We hope that the results reported in this article can stimulate further interest in and efforts on this type of problem.

There are other, more sophisticated ways of doing rejection control. For example, one may allow several redraws of a new stream starting from the previous checkpoint to obtain an acceptable sample, or allow resampling of a new stream from existing good streams. Numerical and theoretical exploration of such "partial rejection" schemes are in progress. Efficient procedures for on-line estimation of common parameters that influence all of the data (such as in a nonlinear state-space model with unknown parameters) are also under development.

## APPENDIX: PROOFS AND COMPUTATION

### A.1 Proof for the Reduction of Chi-Squared Distance by Rejection Control

Here we prove that the rejection control method indeed reduces the chi-squared distance between the target distribution and the modified trial distribution. In other words, we prove that

$$\mathrm{var}_{g*}\{\pi(X)/g^*(X)\} \le \mathrm{var}_g\{\pi(X)/g(X)\}, \qquad \text{(A.1)}$$

where $g$, $g^*$, and $\pi$ are as defined in (4) with $t$ omitted. For simplicity, we suppress all the use of subscript $t$ in the following derivation.

With $w(x) = \pi(x)/g(x)$, the rejection probability $p_c$ in (4) can be expressed as

$$p_c = \int \min\left\{g(x), \frac{\pi(x)}{c}\right\}\, dx = \frac{1}{c}\, E_g[\min\{w(\mathbf{X}), c\}]. \qquad \text{(A.2)}$$

On the other hand, we have

$$\begin{aligned} 1 + \mathrm{var}_{g*}&\left\{\frac{\pi(\mathbf{X})}{g^*(\mathbf{X})}\right\} \\ &= \int \frac{\pi^2(x)}{g^*(x)}\, dx \\ &= p_c \int \frac{\pi(x)}{\min\{g(x), \pi(x)/c\}}\, \pi(x)\, dx \\ &= \int p_c \max\{w(x), c\}\pi(x)\, dx \qquad \text{(A.3)} \\ &= p_c E_g[\max\{w(X), c\}w(X)]. \qquad \text{(A.4)} \end{aligned}$$

Now we show that for any $w_1 > 0, w_2 > 0$,

$$\begin{aligned} h(w_1, w_2) &= [\min\{w_1, c\} - \min\{w_2, c\}] \\ &\quad \times [w_1 \max\{w_1, c\} - w_2 \max\{w_2, c\}] \\ &\ge 0. \end{aligned}$$

There are three cases: when $w_1 > c$ and $w_2 > c$, $h(w_1, w_2) = 0$; when $w_1 < c$ and $w_2 < c$, $h(w_1, w_2) = c(w_1 - w_2)^2 \ge 0$; and when $c$ is between $w_1$ and $w_2$, we assume without loss of generality that $w_1 \le c \le w_2$. Then

$$h(w_1, w_2) = (c - w_1)(w_2^2 - cw_1) \ge 0.$$

Hence the two random variables $\min\{w(X), c\}$ and $w(X)\max\{w(X), c\}$ are positively correlated. Together with the fact that $\min\{w(x), c\}\max\{w(x), c\} = cw(x)$, formulas (A.2) and (A.4), we have

$$\begin{aligned} c&\left[1 + \mathrm{var}_{g*}\left\{\frac{\pi(X)}{g^*(X)}\right\}\right] \\ &= E_g[\min\{w(X), c\}]E_g[\max\{w(X), c\}w(X)] \\ &\le E_g[\min\{w(X), c\}\max\{w(X), c\}w(X)] \\ &= cE_g[w^2(X)] = c\left[1 + \mathrm{var}_g\left\{\frac{\pi(X)}{g(X)}\right\}\right]. \end{aligned}$$

Hence we prove the result (A.1).

### A.2 Proof That Rejection Control–Sequential Importance Sampling Generates Properly Weighted Streams

Consider an individual stream up to time $t$: $\mathbf{x}_t = (x_1, \ldots, x_t)$, and has passed through $k$ checkpoints at time $0 < t_1 < \cdots < t_k < t$ with threshold values $(c_1, \ldots, c_k)$ . We introduce a notation for $s < t$:

$$g_{[s:t]}(\mathbf{x}_{[s:t]} \mid \mathbf{x}_{s-1}) = \prod_{\imath=s}^{t} g_\imath(x_\imath \mid \mathbf{x}_{\imath-1}).$$

Note that the sampling distribution for $\mathbf{x}_{t_1}$ is

$$g_{t_1}^*(\mathbf{x}_{t_1}) = \frac{1}{p_1}\, \min\left\{g_{t_1}(\mathbf{x}_{t_1}), \frac{\pi_{t_1}(\mathbf{x}_{t_1})}{c_1}\right\},$$

and generally for any $t$ such that $t_k < t < t_{k+1}$, we have

$$g_t^*(\mathbf{x}_t) = g_{t_k}^*(\mathbf{x}_{t_k})g_{[(t_k+1):t]}(\mathbf{x}_{[(t_k+1):t]} \mid \mathbf{x}_{t_k}).$$

Hence the importance sampling weight for $\mathbf{x}_t$ at stage $t$ is

$$w_t^*(\mathbf{x}_t) = \frac{\pi_t(\mathbf{x}_t)}{g_t^*(\mathbf{x}_t)} = \frac{\pi_{t_k}(\mathbf{x}_{t_k})}{g_{t_k}^*(\mathbf{x}_{t_k})} \prod_{s=t_{k-1}}^{t-1} u_{s+1},$$

where $u_\imath$ is defined in (1). Therefore, if $\mathbf{x}_{t_k}$ is properly weighted with $w_{t_k}^*(\mathbf{x}_{t_k})$ with respect to $\pi_{t_k}$, then $\mathbf{x}_t$ must also be properly weighted with $w_t^*(\mathbf{x}_t)$.

Now let us examine whether $\mathbf{x}_{t_k}$ is properly weighted. Because the sampling distribution for $\mathbf{x}_{t_k}$ is

$$\begin{aligned} g_{t_k}^*(\mathbf{x}_{t_k}) = \frac{1}{p_k}\, \min&\left\{g_{t_{k-1}}^*(\mathbf{x}_{t_{k-1}})g_{[(t_{k-1}+1):t_k]}\right. \\ &\left. \times\ (\mathbf{x}_{[(t_{k-1}+1):t_k]} \mid \mathbf{x}_{t_{k-1}}), \frac{\pi_{t_k}(\mathbf{x}_{t_k})}{c_k}\right\}, \end{aligned}$$

where the acceptance probability $p_k$ is the same for all $\mathbf{x}_{t_k}$ and the induced importance weight is

$$\begin{aligned} &w_{t_k}^*(\mathbf{x}_{t_k}) \\ &= p_k \max\left\{\frac{\pi_{t_k}(\mathbf{x}_{t_k})}{g_{t_{k-1}}^*(\mathbf{x}_{t_{k-1}})g_{[(t_{k-1}+1):t_k]}(\mathbf{x}_{[(t_{k-1}+1):t_k]} \mid \mathbf{x}_{t_{k-1}})}, c_k\right\} \\ &= p_k \max\left\{w_{t_{k-1}}^* \prod_{s=t_{k-1}}^{t_k-1} u_{s+1}, c_k\right\}. \end{aligned}$$

Hence the weights computed in Section 2 for RC-SIS are correct.

### A.3 Computations For the Example in Section 4.1

We provide the following details only for completeness. For the weights, we must compute

$$p(y_t \mid \theta, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) = \phi(y_t - \alpha_1 q_{1(t-1)})[1 - \Phi(y_t - \alpha_2 q_{2(t-1)})] + \phi(y_t - \alpha_2 q_{2(t-1)})[1 - \Phi(y_t - \alpha_1 q_{1(t-1)})],$$

and for imputing missing data we need

$$p(\delta_t = 1 \mid \theta, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}, y_t) = \frac{\phi(y_t - \alpha_1 q_{1(t-1)})[1 - \Phi(y_t - \alpha_2 q_{2(t-1)})]}{p(y_t \mid \theta, \mathbf{x}_{t-1}, y_{t-1})},$$

$$p(\lambda_t \mid \delta_t = 1, y_t, \theta, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) = \frac{\phi(\lambda_t - \alpha_2 q_{2(t-1)})}{1 - \Phi(y_t - \alpha_2 q_{2(t-1)})} I(\lambda_t < y_t),$$

and

$$p(\lambda_t \mid \delta_t = 2, y_t, \theta, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) = \frac{\phi(\lambda_t - \alpha_1 q_{1(t-1)})}{1 - \Phi(y_t - \alpha_1 q_{1(t-1)})} I(\lambda_t < y_t).$$

Suppose that the prior distribution for $\theta$ is $p_0(\alpha_1, \alpha_2)$. Then, given complete observations, the posterior is

$$p(\theta \mid q_1, \ldots, q_T) \propto p_0(\theta) \times \exp\left\{ -\sum_{t=2}^{T} \frac{(q_{1t} - \alpha_1 q_{1(t-1)})^2 + (q_{2t} - \alpha_2 q_{2(t-1)})^2}{2} \right\}.$$

Without loss of generality, we take $p_0(\theta)$ to be uniform on $[0, 1]^2$. Then the posterior distribution is simplified as

$$p(\theta \mid q_1, \ldots, q_T) \propto \exp\left\{ -\frac{(\alpha_1 - a_1)^2}{2b_1^2} - \frac{(\alpha_2 - a_2)^2}{2b_2^2} \right\}, \qquad 0 \le a_1, \quad a_2 < 1,$$

where

$$b_1 = \left( \sum_{t=2}^{T} q_{1(t-1)}^2 \right)^{-1/2},$$

$$a_1 = b_1^2 \sum_{t=2}^{T} q_{1t} q_{1(t-1)},$$

$$b_2 = \left( \sum_{t=2}^{T} q_{2(t-1)}^2 \right)^{-1/2},$$

and

$$a_2 = b_2^2 \sum_{t=2}^{T} q_{2t} q_{2(t-1)}.$$

When assuming that $\alpha_1 > \alpha_2$ and the complete data are observed, the marginal distribution of $\alpha_1$ is

$$p(\alpha_1 \mid q_1, \ldots, q_T) \propto \left[ \Phi\left( \frac{\alpha_1 - a_2}{b_2} \right) - \Phi\left( -\frac{a_2}{b_2} \right) \right] \times \exp\left\{ -\frac{(\alpha_1 - a_1)^2}{2b_1^2} \right\}.$$

We use an acceptance-rejection strategy described by Geweke (1991) and Robert (1995) to simulate from the foregoing truncated normal distribution.

*[Received August 1996. Revised November 1997.]*

## REFERENCES

Beckett, L., and Diaconis, P. (1994), "Spectral Analysis for Discrete Longitudinal Data," *Advances in Mathematics*, 103, 107–128.

Berry, D., and Christensen, R. (1979), "Empirical Bayes Estimation of a Binomial Parameter Via Mixture of Dirichlet Processes," *The Annals of Statistics*, 7, 558–568.

Berzuini, C., Best, N. G., Gilks, W. R., and Larissz, C. (1997), "Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods," *Journal of the American Statistical Association*, 92, 1403–1412.

Casella, G., and Robert, C. P. (1996), "Rao–Blackwellisation of Sampling Schemes," *Biometrika*, 83, 81–94.

Chen, R., and Li, T. (1995),"Blind Restoration of Linearly Degraded Discrete Signals by Gibbs Sampler," *IEEE Transactions on Signal Processing*, 43, 2410–2413

Donoho, D. (1981), "On Minimum Entropy Deconvolution," in *Applied Time Series Analysis II*, ed. D. F. Findley, New York: Academic Press.

Escobar, M. D. (1994), "Estimating Normal Means With a Dirichlet Process Prior," *Journal of the American Statistical Association*, 89, 268–277.

Fair, R. C., and Jaffee, D. M. (1972), "Methods of Estimation for Markets Disequilibrium," *Econometrica*, 40, 497–514.

Ferguson, T. S. (1973), "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics*, 1, 209–230.

Geyer, C. J., and Thompson, E. A. (1995), "Annealing Markov Chain Monte Carlo With Applications to Ancestral Inference," *Journal of the American Statistical Association*, 90, 909–920.

Gordon, N. J., Salmon, D. J., and Smith, A. F. M. (1993), "A Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEEE Proceedings on Radar and Signal Processing*, 140, 107–113.

Giannakis, G. B. and Mendel, J. M. (1989), "Identification of Nonminimum Phase Systems Using Higher-Order Statistics," *IEEE Transactions on Acoustic, Speech and Signal Processing*, 37, 665–674.

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57, 97–109.

Hendry, D. F., and Richard, J-F. (1991), "Likelihood Evaluation for Dynamic Latent Variables Models," in *Computational Economics and Econometrics*, eds. H. M. Amman, D. A. Belsley, and L. F. Pau, Dordrecht: Kluwer.

Irwin, M., Cox, N., and Kong, A. (1994), "Sequential Imputation for Multilocus Linkage Analysis," *Proceedings of the National Academy of Science*, 91, 11684–11688.

Kong, A., Liu, J. S., and Wong, W. H. (1994), "Sequential Imputations and Bayesian Missing-Data Problems," *Journal of the American Statistical Association*, 89, 278–288.

Laird, N. M. (1978), "Nonparametric Maximum Likelihood Estimation of a Mixing Distribution," *Journal of the American Statistical Association*, 73, 805–811.

Laird, N. M., and Louis, T. A. (1987), "Empirical Bayes Confidence Intervals Based on Bootstrap Samples," *Journal of the American Statistical Association*, 82, 739–757.

Lii, K. S., and Rosenblatt, M. (1982), "Deconvolution and Estimation of Transfer Function Phase and Coefficients for Non-Gaussian Linear Processes," *The Annals of Statistics*, 10, 1195–1208.

Liu, J. S. (1996), "Nonparametric Hierarchical Bayes Via Sequential Imputations," *The Annals of Statistics*, 24, 911–930.

Liu, J. S., and Chen, R. (1995), "Blind Deconvolution Via Sequential Imputations," *Journal of the American Statistical Association*, 90, 567–576.

——— (1998), "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, 93, 1032–1044.

Marinari, E., and Parisi, G. (1992), "Simulated Tempering: A New Monte Carlo Scheme," *Europhysics Letters*, 19, 451–458.

Maritz, J. S., and Lwin, T. (1989), *Empirical Bayes Methods* (2nd ed.), New York: Chapman and Hall.

Quandt, R. E. (1982), "Econometric Disequilibrium Models" (with discussion), *Econometric Review*, 1, 1–96.

——— (1988), *The Econometrics of Disequilibrium*, New York: Basil

Blackwell.

Robert, C. P. (1995), "Simulation of Truncated Normal Variables," *Statistics and Computing*, 5, 121–125.

Spiegelhalter, D. J., and Lauritzen, S. L. (1990), "Sequential Updating of Conditional Probabilities on Directed Graphical Structures," *Networks*, 20, 579–605.

Vasquez, M., and Scheraga, H. A. (1985), "Use of Build-Up and Energy-Minimization Procedures to Compute Low-Energy Structures of the Backbone of Enkephalin," *Biopolymers*, 24, 1437–1447.

Wong, W. H. (1995), Comment on "Bayesian Computation and Stochastic Systems" by Besag et al., *Statistical Science*, 10, 52–53.

Wong, W. H., and Liang, F. (1997), "Dynamic Weighting in Monte Carlo and Optimization," *Proceedings of the National Academy of Science* 94, 14220–14224.