Adaptive Independent Metropolis-Hastings by Fast Estimation of Mixtures of Normals

Paolo Giordani
Research Department, Sveriges Riksbank
paolo.giordani@riksbank.se

Robert Kohn
Australian School of Business
University of New South Wales

January 19, 2009

Abstract

Adaptive Metropolis-Hastings samplers use information obtained from previous draws to tune the proposal distribution automatically and repeatedly. Adaptation needs to be done carefully to ensure convergence to the correct target distribution because the resulting chain is not Markovian. We construct an adaptive independent Metropolis-Hastings sampler that uses a mixture of normals as a proposal distribution. To take full advantage of the potential of adaptive sampling our algorithm updates the mixture of normals frequently, starting early in the chain. The algorithm is built for speed and reliability and its sampling performance is evaluated with real and simulated examples. Our article outlines conditions for adaptive sampling to hold and gives a readily accessible proof that under these conditions the sampling scheme generates iterates that converge to the target distribution.

Keywords: Clustering; Gibbs sampling; Markov chain Monte Carlo; Semiparametric regression models; State space models.

1 Introduction

Bayesian methods using Markov chain Monte Carlo (MCMC) simulation have greatly influenced statistical practice over the past twenty years because of their ability to estimate complex models and produce finite sample inference. A key component in implementing MCMC simulation is the Metropolis-Hastings (MH) method (Metropolis et al. 1953; Hastings 1970), which requires the specification of one or more proposal distributions. The speed at which the chain converges to the posterior distribution and its ability to move efficiently across the state space depend crucially on whether the proposal distribution(s) provide good approximations to the target distributions, either locally or globally. Given the key role played by proposal distributions, it is natural to use experience from previous

draws to *adapt* the proposal to the target. Our article considers adaptive sampling that is subject to theoretical rules which ensure that the iterates converge to realizations from the correct target (posterior) distribution.

The literature on adaptive MCMC methods follows three main strands. Adaptation by *regeneration* stems from the work of Gilks et al. (1998). Adapting on the target distribution is proposed by Wang and Landau (2001) and generalized by Liang et al. (2007) and Atchade and Liu (2007). Our article focuses exclusively on *diminishing adaptation* schemes. Important theoretical advances in diminishing adaptation were made by Holden (1998), Haario et al. (2001), Andrieu and Robert (2001), Andrieu and Moulines (2006), Andrieu et al. (2005), Atchadé and Rosenthal (2005), Nott and Kohn (2005) and Roberts and Rosenthal (2007). The proofs of convergence for strict adaptive sampling are more complex than for the non adaptive case as the iterates are not Markovian because the MH kernel can depend on the entire history of the draws. Although more theoretical work on adaptive sampling can be expected, the existing body of results provides sufficient justification and guidelines to build adaptive MH samplers for challenging problems.

Research is now needed on how to design efficient and reliable adaptive samplers for broad classes of problems. This more applied literature mostly focuses on random walk Metropolis, see for example Roberts and Rosenthal (2006). Partial exceptions are Gåsemyr (2003) who uses normal proposals for an independent Metropolis-Hastings, but limits the tuning of the parameters to the burn-in period and Hastie (2005) who uses two step adaptation in a reversible jump context. By two step adaptation we loosely mean a sampling scheme in which a rather thorough exploration of the target density is carried out in the first part of the chain by a sampler other than independent MH (such as random walk Metropolis) before switching to a more efficient independent MH sampler with proposal density tuned on the first-stage draws. Independent MH schemes are implemented by Nott and Kohn (2005) to sample discrete state spaces in variable selection problems (e.g. to learn if a variable is in or out), and by Giordani and Kohn (2008) to learn about interventions, such as breaks or outliers, in dynamic mixture models.

Our paper contributes to the development of algorithms for adaptive independent MH sampling in continuous state spaces. Increased sampling efficiency is one important goal, particularly in cases where current best practice (typically some version of random walk Metropolis or Gibbs sampling) is less than satisfactory. But equally important achievements of adaptive schemes may be to expand the set of problems that can be handled efficiently by general purpose samplers and to reduce coding effort. In particular, adaptive schemes can reduce dependence on the use of conjugate priors. Such priors make it easier to implement MCMC schemes, but are less necessary when using adaptive sampling.

Our adaptive sampling approach is built on four main ideas. The first is to combine preliminary exploration of the target distribution and adaptive sampling into one estimation procedure. The second is to estimate mixtures of Gaussians from the history of the draws and use them as proposal distributions for independent MH in all parts of the estimation. The third is to perform this estimation frequently, particularly during the early part of the estimation, a strategy that we call *intensive adaptation*. The fourth is to ensure that the theoretical conditions for the correct ergodic behavior of the sampler are respected during adaptation. To apply these ideas successfully, estimation of the mixture parameters needs to be fast, reliable, and robust. We achieve a good balance of these goals by carefully selecting and tailoring to our needs algorithms developed in the clustering literature.

We study the performance of our adaptive sampler in two examples in which commonly used Gibbs schemes can be very inefficient and compare it with an adaptive random walk Metropolis sampler proposed by Roberts and Rosenthal (2006) that builds on the work of Haario et al. (2001).

Our paper also provides conditions and outlines a proof that our adaptive sampling scheme converges to the correct target distribution and gives convergence rates.

Our working paper available at http://arxiv.org/PS_cache/arxiv/pdf/0801/0801.1864v1.pdf contains several other applications and a more extensive discussion.

2 Some theory for adaptive sampling

A *diminishing adaptation* MH sampler performs the accept/reject step like a standard MH, but updates the proposal distribution using the history of the draws. This updating is 'diminishing' in the sense that changes in the proposal distribution tend to zero asymptotically in the number of iterations. In practice, this usually means that the proposal distribution itself settles down asymptotically.

This section outlines the theoretical framework for adaptive independent Metropolis-Hastings sampling as used in our article that gives some support for our practice. The appendix outlines proofs of the theoretical results, which extend similar results in Nott and Kohn (2005) for the case of a finite state space. Our aim is to provide simple accessible proofs that will help to popularize the adaptive methodology. All densities in this section are with respect to Lebesgue measure or counting measure, which we denote as $\mu\{\cdot\}$.

Let $\mathcal{Z}$ be a sample space and $\pi(z)$ a target density on $\mathcal{Z}$. We use the following adaptive MH scheme to construct a sequence of random variables $\{Z_n, n \geq 2\}$ whose distribution converges to $\pi(z)$. Z_0 and Z_1 are generated from $g_0(z)$ which is defined below. For $n \geq 1$, let $q_n(z; \lambda_n)$ be a proposal density for generating Z_{n+1}, where λ_n is a parameter vector that is based on $Z_0 = z_0, \ldots, Z_{n-1} = z_{n-1}$. Thus, given $Z_n = z$, we generate $Z_{n+1} = z'$ from q_n, and then with probability

$$\alpha_n(z, z') = \min\left(1, \frac{\pi(z')}{\pi(z)} \frac{q_n(z; \lambda_n)}{q_n(z'; \lambda_n)}\right) \tag{1}$$

we take $Z_{n+1} = z'$; otherwise we take $Z_{n+1} = z$. Our choice of $q_n(z; \lambda_n)$ is of the form

$$q_n(z; \lambda_n) = \omega_1 g_0(z) + (1 - \omega_1) g_n(z; \lambda_n) \tag{2}$$

where $0 < \omega_1 < 1$. The density $g_0(z)$ is constant and the density $g_n(z; \lambda_n)$ has parameter vector λ_n that evolves with the iterates. The form of both densities as used in our article is described more fully below.

We assume that there exists a constant $K > 0$ such that for all $z \in \mathcal{Z}$

$$\frac{\pi(z)}{g_0(z)} \leq K \qquad \text{and} \qquad \frac{g_n(z;\lambda_n)}{g_0(z)} \leq K \ , \tag{3}$$

and

$$\sup_{z\in\mathcal{Z}} \left| \Big(g_n(z;\lambda_n) - g_{n+1}(z;\lambda_{n+1}) \Big) / g_0(z) \right| = a_n \tag{4}$$

where $a_n = O(n^{-r})$ for some $r > 0$ almost surely. In relation to the dominance condition (3), we note that in the non-adaptive case, that is $q_n(z;\lambda_n) = q(z)$ for all n, Mengersen and Tweedie (1996) show that $\pi(z)/q(z) \leq K$ for all z is a necessary and sufficient condition for geometric ergodicity.

Under conditions (3) and (4), the following results are proved in Appendix 2.

Theorem 1 *For all measurable subsets A*

$$\sup_{A\subset\mathcal{Z}} \mid \Pr(Z_n \in A) - \pi(A) \mid \to 0 \quad as \quad n \to \infty. \tag{5}$$

Theorem 2 *Suppose that $h(z)$ is a measurable function that is square integrable with respect to the density g_0. Then, almost surely,*

$$\frac{1}{n}\sum_{j=1}^{n} h(Z_j) \to E_\pi(h(Z)) \quad as \quad n \to \infty. \tag{6}$$

We now describe the construction of $g_n(z;\lambda_n)$ in the article. Let $g_n^\star(z;\lambda_n^\star)$ be a mixture of normals with parameters $\lambda_n^\star$ which is obtained using k-harmonic means clustering as described in section 4 and appendix 1. Let $\tilde{g}_n^\star(z;\tilde{\lambda}_n^\star)$ be a second mixture of normals having the same component weights and means as $g_n^\star(z;\lambda_n^\star)$, but with its component variances inflated by a factor $k > 1$. We set

$$g_n(z;\lambda_n) = \omega_2' \tilde{g}_n^\star(z;\tilde{\lambda}_n^\star) + (1-\omega_2') g_n^\star(z;\lambda_n^\star) \ , \tag{7}$$

where $\omega_2' = \omega_2/(1-\omega_1)$ with ω_1 defined in (2), $\omega_1 > 0, \omega_2 > 0$, and $\omega_1 + \omega_2 < 1$. Thus,

$$q_n(z;\lambda_n) = \omega_1 g_0(z) + \omega_2 \tilde{g}_n^\star(z;\tilde{\lambda}_n^\star) + (1-\omega_1-\omega_2) g_n^\star(z;\lambda_n^\star) \ .$$

We note that $g_n(z;\lambda_n)$ is also a mixture of normals with parameters λ_n, and we say that it is obtained by 'stretching and fattening' the tails of $g_n^\star(z;\lambda_n^\star)$. This strategy for obtaining heavier tailed mixtures is used extensively in our paper.

Conditions (3) and (4) provide guidance for our algorithms and are checked informally in our work. The target $\pi(z)$ in our applications is the posterior density of the parameters and so is proportional to $\pi_0(z)\pi_1(z)$, where $\pi_0(z)$ is the prior and $\pi_1(z)$ is the likelihood. Thus, if $\pi_0(z)/g_0(z)$ is bounded for all n and all $z \in \mathcal{Z}$ and if the maximum likelihood estimator exists then $\pi(z)/g_0(z)$ is also bounded. If we choose $g_0(z)$ such that $\tilde{g}_n^\star(z;\tilde{\lambda}_n^\star)/g_0(z) \leq K$ for all n and $z \in \mathcal{Z}$ then $g_n(z;\lambda_n)$ will also be bounded. Finally, we check informally that (4) holds by checking that the iterates $g_n^\star(z;\lambda_n^\star)$ converge to a fixed proposal. In our experience, this almost always happens. However, we now show how to more formally ensure that the conditions (3) and (4) hold, while the practical performance of the algorithm remains very much as above. First, we add an extremely heavy tailed component to $g_0(z)$ with a probability that is extremely small. Second, let $\bar{g}_0(z) = g_0(z)$ and for $n \geq 1$ define $\bar{g}_n(z;\bar{\lambda}_n) = (1-a_n)\bar{g}_{n-1}(z;\bar{\lambda}_{n-1}) + a_n g_n(z;\lambda_n)$, with $a_n = (1-\gamma)/n^\gamma$ and γ very close to zero and positive. This means that for $n \geq 1$, $\bar{\lambda}_n$ is defined recursively in terms of $\bar{\lambda}_n$. Third, we redefine the proposal density as $q_n(z;\lambda_n) = \omega_1 g_0(z) + (1-\omega_1)\bar{g}_n(z;\lambda_n)$. This ensures that $\pi_0(z)/g_0(z)$ and $\bar{g}_n(z;\bar{\lambda}_n)$ are bounded in n and $z \in \mathcal{Z}$ and that the diminishing adaptation condition holds, i.e.

$$\sup_{z\in\mathcal{Z}} \left| \left(\bar{g}_n(z;\bar{\lambda}_n) - \bar{g}_{n+1}(z;\bar{\lambda}_{n+1}) \right) / g_0(z) \right| = a_n \ .$$

Section 7 and appendix F of Andrieu and Moulines (2006) give general convergence results for adaptive independent Metropolis-Hastings and Roberts and Rosenthal (2007) give an elegant proof of the convergence of adaptive sampling schemes. However, we believe that readers may find the conditions (3) and (4) and the proofs of Theorems 1 and 2 easier to understand for the methodology proposed in our article.

3 Implementation of the adaptive sampling scheme

This section describes the implementation of the sampling scheme. We anticipate that readers will use this as a basis for their own experimentation. The adaptive sampling scheme is run in two phases, a preliminary phase where the conditions for ergodicity are not enforced and a strict adaptive stage that enforces ergodicity. To make the structure of the algorithm clearer, we first describe it in pseudo-code and then discuss it in more detail.

Pseudo-code. Let $q_n(z) = \omega_1 g_0(z) + (1-\omega_1) g_n(z)$, $g_0(z) = 0.6\phi_0(z) + 0.4\tilde{\phi}_0(z)$,where $\phi_0(z)$ is a mixture of normals, initialized at iteration $n = 1$ by a Laplace expansion (in which case $\phi_0(z)$ is a multivariate normal), or by the prior or by a density estimated from a preliminary MCMC run if the Laplace approximation is unavailable. The density $\tilde{\phi}_0(z)$ is a mixture of normals with the same parameters as $\phi_0(z)$ except that the covariance matrices are multiplied by 25. Let $g_n(z) = \omega_2' g_n^\star(z) + (1-\omega_2')\tilde{g}_n^\star(z)$, where $g_n^\star(z)$ is a mixture of normals and $\tilde{g}_n^\star(z)$ is a mixture of normals with the same parameters except that the covariance matrices are multiplied by a scalar k. For notational convenience we omit to show in the pseudo-code the dependence of the densities $q_n(z), g_n(z)$ and $g_n^\star(z)$ on their parameters.

Let A_n denote the number of accepted MH draws up to but not including iteration n, and $S_n(M)$ the smallest MH acceptance probability in iterations $n-M$ to $n-1$.

1. At iteration n, $q_n(z)$ is used as a proposal to update z as in (2).

2. Define the *preliminary phase* to start at $n = 1$ and end when $S_n(M) > \alpha_M$. The density $\phi_0(z)$ is updated once at the end of the preliminary phase, and set equal to $g_{last}^\star$, the last estimated mixture of normals in the preliminary phase.

3. The mixture of normals $g_n^\star(z)$ is set equal to $g_{n-1}^\star(z)$ unless $A_n > 5\dim(z)$ and *either*

   - $n - n^\star$ belongs to a predetermined set of positive updating times given below, where $n^\star = n : A_n = 5\dim(z)$, or
   - The average MH probability in the last L iterations is lower than α_L (preliminary phase only).

4. Otherwise $g_n^\star(z)$ is updated as in section 4 and appendix 1.

We set $k = 16, \omega_1 = 0.05, \omega_2 = 0.15, \omega_2' = \omega_2/(1-\omega_1), L = 10, \alpha_L = 0.1, M = 20, \alpha_M = 0.02$.

End of pseudo-code

We find these values of $k, \omega_1, \omega_2, L, \alpha_L, M$ and α_M work well, but we do not claim they are optimal. We conjecture that the speed of convergence and efficiency of our sampler can be further improved with a more careful (and possibly adaptive) choice of these parameters. Any other value of k in the range 9 to 25 and of ω_1 and ω_2 in the range 0.05 to 0.3 worked well for the examples given in the paper.

During the preliminary phase, we *first* estimate the k-harmonic means mixture after a given number of *accepted* draws in order to ensure that the estimated covariance matrices are positive definite and estimated with sufficient accuracy. When there are 2 to 4 unknown parameters as in the inflation example we first estimate the k-harmonic means mixture after 20 *accepted* draws. If our parameter space is bigger then we would increase that number appropriately. The estimation after a given number of accepted draws is only done *once* throughout the sampling scheme. We then re-estimate the mixture after 50, 100, ..., 350, 400, 500, ..., 1000 and then every 1000 draws thereafter. We also recommend updating the proposal in the preliminary stage following a period of low acceptance probabilities in the MH step. Specifically, we re-estimate the mixture parameters if the average acceptance probability in the last L iterations is lower than α_L, where we set L and α_L as above. Low acceptance probabilities signal a poor fit of the proposal, and it is therefore sensible to update the proposal to give it a better chance of covering the area around the current parameter value. Since it is unclear that this does not violate any of the conditions required for the ergodicity of our adaptive sampler, we limit the updating of the proposal at endogenously chosen points to the preliminary phase, after which the proposal is updated only at predetermined intervals. The end of the preliminary adaptation phase could be set ex-ante, but we prefer to determine it endogenously by requiring the smallest acceptance probability in the last M iterations to be higher than α_M, where M and α_M are set as above. During the second phase (period of strict adaptation), we update the proposal every 1000 draws.

The updating schedule given above is also not optimal in any specific sense. The principle is to update more frequently in the initial phases of the chain. We would update more (less) frequently if the likelihood was very expensive (inexpensive) to compute compared

to updating the proposal. In our experience, the estimation frequency typically does not affect the performance of the algorithms dramatically. The fact that we update the proposal when it has performed poorly in recent iterates is very helpful in this regard since otherwise the MH can reject for long stretches if the updates are infrequent. Frequent updates are important if the initial proposal is poor and/or the target distribution is multimodal. Fortunately, updating the proposal when it is not performing well largely endogenizes the updating schedule.

The estimation of the mixture of normals can become slow when the number of iterations is large. To avoid this problem, after 1000 accepted draws we only add every j-th draw to the sample used to estimate the mixture parameters, where j is chosen so that the mixture is not estimated on more than 10000 observations.

4 A clustering algorithm for fast estimation of mixtures of normals in adaptive IMH

Finite mixtures of normals are an attractive option to construct the proposal density because they can approximate any continuous density arbitrarily well and are fast to sample from and evaluate. See McLachlan and Peel (2000) for an extensive treatment of finite mixture models.

However, estimating mixtures of normals is already a difficult problem when an independent and identically distributed sample from the target is given and estimation needs to be performed only once: the likelihood goes to infinity whenever a component has zero variance (an even more concrete possibility if, as unavoidable in IMH, some observations appear more than once), and its maximization, whether by the EM algorithm or directly, is plagued by local modes. Although several authors have made substantial advances in dealing with these problems (e.g. Figuereido and Jain 2002; Ueda, Nakano, Ghahramani, and Hinton 2000; Verbeek, Vlassis, and Krose 2003), in our experience the EM algorithm is not sufficiently reliable when the sample is small and contains a non-trivial share of rejected draws. The inevitable short runs of rejections give rise to small clusters with zero covariance matrix at which the EM algorithm frequently gets stuck. Moreover, EM-based algorithm

are computationally expensive and slow to converge, which makes them less attractive when the proposal is to be updated frequently.

We have therefore limited our attention to algorithms that estimate mixtures of normals quickly and without explicitly computing the covariance matrix of each component (for robustness). Within this class, the *k-means* algorithm is the most popular algorithm. We employ the *k-harmonic means* algorithm, an extension of the k-means algorithm that allows for soft membership. Degeneracies can be easily prevented, so the algorithm is remarkably robust even in the presence of a long series of rejections. The number of clusters is chosen with the BIC criterion. The increase in speed and reliability is paid for with a decreased fit to the target, as the family of k-means algorithms performs best when an optimal fit requires all components of the mixture to have the same probability and covariance matrix (see Bradly and Fayyad 1998, for a discussion). Hamerly and Elkan (2002) show that the performance of k-harmonic means deteriorates less rapidly than alternatives of similar computational cost with departures from these ideal conditions. An outline of the k-harmonic means algorithms is given in Appendix 1.

Although the k-harmonic means algorithm is less sensitive to initialization than either k-means or EM (Hamerly and Elkan 2002), in an unsupervised environment it is important to have good starting values. We have found the algorithm of Bradly and Fayyad (1998) to perform very well and at a low computational cost.

When most parameters are nearly normally distributed, fitting a mixture of normals to all the parameters is problematic in the sense that the chances of finding a local mode with all parameters normally distributed is quite high (though depending on the starting value of course). This is true of clustering algorithms and also of EM-based algorithms. To improve the performance of the sampler in these situations, we divide the parameter vector θ into two groups, θ_1 and θ_2, where parameters in θ_1 are classified as approximately normal and parameters in θ_2 are skewed. Our rule of thumb is to place a parameter in the 'normal' group if its marginal distribution has $|s| < 0.2$, where s is the skeweness. Our fattening the tails of the mixture should handle the kurtosis, though this would optimally be done with mixtures of more flexible distributions than the normal.

A normal is then fit to the first group and a mixture of p normals to the second. For

θ_1, we can compute the mean μ_{θ_1} and covariance matrix Σ_{θ_1} inexpensively from the draws. For θ_2, we fit a mixture of normals as detailed below, estimating probabilities $\pi_1, ..., \pi_p$, means $\mu_1, ..., \mu_p$, and covariance matrices $\Sigma_1, ..., \Sigma_p$. We then need to build a mixture for $\theta = \{\theta_1, \theta_2\}$. The mean is straightforward: for the normal parameters, all components have the same mean. The diagonal blocks of the covariance matrices Ω_i corresponding to $\text{var}(\theta_1)$ and $\text{var}(\theta_2)$ for component i are also straightforward. The off-diagonal blocks of Ω_i, corresponding to $\text{cov}(\theta_1, \theta_2)$ is computed as

$$\Omega_i^{12} = \sum_{t=1}^{n} \pi_{i,t}^* [(\theta_{1,t} - \mu_{\theta_1})(\theta_{2,t} - \mu_i)] / \sum_{t=1}^{n} \pi_{i,t}^*,$$

where $\pi_{i,t}^* = \Pr(K_t = i|\theta_{2,t})$ is the probability of $\theta_{2,t}$ coming from the i-th component.

If the proposal distribution is normal then it is computationally inexpensive to update it at every iteration. It is tempting to update a mixture of normals proposal with an on-line estimation procedure such as the on-line EM algorithm proposed in Andrieu and Moulines (2006). The advantage of on-line estimation is that the parameters of the mixture are updated recursively, so the proposal itself is updated at each iteration at a very small computational cost. However, on-line estimation of the mixture parameters in AIMH has a number of serious shortcomings. The estimates are inefficient compared to batch estimators because each data point is used only once, which corresponds to requiring a batch optimization to converge in one step. The loss of efficiency is more severe in small samples, that is, in the early phases of the chain. Direct estimation of the mixture component covariance matrices often leads to numerical problems in the early phases of the chain given that rejections in MH produce degenerate clusters. Finally, a limitation of on-line estimators is related to the fact that they are a form of stochastic gradient descent (see Spall (2003) for an introduction). When the function to be maximized is multimodal (as is typically the case with mixtures) on-line estimates are in general sensitive to the order of the draws, with initial draws having heavier influence than later draws in determining the mode at which estimates settle down. We have verified empirically that the quality of solutions given by several on-line algorithms deteriorates rapidly if the initial observations are not representative of the entire target distribution. This makes on-line algorithms unsuitable for use in the early, exploratory phases of the chain, though they can work well if the initial

proposal distribution already provides a reasonably good approximation of the target and the acceptance rates are sufficiently high.

Since we are opting for batch estimators, it is too costly to update the proposal at each iteration. We update it at predetermined numbers of iterations, more frequently in the earlier stages of the chain. Implementation details for the empirical examples are given in section 3.

We make two further comments on Andrieu and Moulines (2006). First, they propose to keep the number of components in the mixture constant, whereas we have found it advantageous to select the number adaptively as outlined in appendix 1. Second, they outline a proposed approach to using mixtures as proposal densities, but do not report on the empirical performance of their proposal.

5 Applications

State space models and nonparametric models are ideal initial applications for AIMH schemes. Although they can have a large number of parameters or latent variables, it often happens that conditional on a small subset of these, most parameters and latent variables can be integrated out or have known analytical form from which we can generate them. It is therefore often possible to draw all parameters in one or two blocks. Exploiting these features, it is also often inexpensive to find the posterior mode, possibly for a simplified version of the model, and therefore obtain a reasonable initialization of the proposal distribution. Finally, the standard approach based on Gibbs and Metropolis-within-Gibbs can be very inefficient, particularly for state space models (see Fruhwirth-Schnatter 2004).

For each of our applications we checked the results of the adaptive sampling scheme by re-running the sampler at a number (at least 5) of different starting points using a fixed proposal based on the last mixture of normals update in the second stage of the adaptive sampling. In all cases we got very similar results to those obtained using strict adaptation, where by 'very similar' we mean that the posterior means of the parameters were within three Monte Carlo standard deviations of those obtained using adaptive sampling.

For our examples we define the inefficiency of a sampling scheme as the factor by which the number of iterates needs to increase to give the same precision (standard error) as a

sampler generating independent draws. For two sampling schemes A and B, we define the inefficiency of scheme B relative to A as the factor by which it is necessary to increase the running time of B in order for it to obtain the same accuracy as A. It is computed as the inefficiency factor of B times its run time per iteration divided by the inefficiency factor of A times its run time per iteration.

In the examples below we compare the performance of the AIMH sampler to the following version of the Haario et al. (2001) adaptive random walk Metropolis sampler proposed on page 3 of Roberts and Rosenthal (2006). Specifically, let θ be the parameters in the model, $\hat{\theta}$ the posterior mode and V the variance covariance matrix of the Laplace approximation to the posterior. Then at iteration j the proposal distribution is given by

$$\begin{aligned} Q_j(\theta^c, \cdot) &= N(\theta^c, (0.1)^2 V/d) \quad \text{if} & j < 5d, \\ Q_j(\theta^c, \cdot) &= (1-\beta) N(\theta^c, (2.38)^2 \Sigma_j/d) + \beta N(\theta^c, (0.1)^2 I_d/d) \quad \text{if} & j \geq 5d, \end{aligned}$$

where $N(\theta, V)$ is the normal density with mean θ and covariance matrix V, θ^c is the current value of θ, d is the dimension of θ, $\beta = 0.05$ and Σ_j is the current empirical estimate of the covariance matrix of the target distribution based on the iterates thus far. In all cases we initialized this sampler at the posterior mode.

5.1 Time-varying parameter autoregressive models

Consider the following time-varying parameter first order autoregressive (AR(1)) process (the extension to a more general autoregressive process is straightforward):

$$y_t = c_t + \rho_t y_{t-1} + \sigma_\epsilon \epsilon_t \,, \; c_t = c_{t-1} + \lambda_0 \sigma_\epsilon u_t \text{ and } \rho_t = \rho_{t-1} + \lambda_1 v_t, \tag{8}$$

where ϵ_t, u_t, v_t are all $nid(0,1)$. The model has three parameters $(\sigma_\epsilon^2, \lambda_0^2, \lambda_1^2)$, while c_0 and ρ_0 can be treated either as parameters or (our choice) as states. Given conjugate priors (inverse gamma for the parameters, and normal for c_0 and ρ_0), Gibbs sampling is straightforward (Carter and Kohn 1994). Fruhwirth-Schnatter (2004) reports that based on the autocorrelations of the iterates, Gibbs sampling can be very inefficient for these models. In the following application we also find that the Gibbs draws are highly autocorrelated and,

by comparing posterior statistics from Gibbs sampling and from our AIMH, we also find that the autocorrelations do not reveal the full extent of the problem.

5.1.1 Application: US CPI inflation

We apply the model to quarterly U.S. CPI inflation for the period 1960-2005 (184 observations). The annualized quarterly CPI inflation is defined as $400(P_t/P_{t-1} - 1)$, where P_t is aggregated from monthly data (averages) on Consumer Price Index For All Urban Consumers: All Items, seasonally adjusted, Series ID CPIAUCSL, Source: U.S. Department of Labor: Bureau of Labor Statistics. We use rather dispersed inverse gamma priors for $\sigma_\epsilon^2, \lambda_0^2, \lambda_1^2$ with a common shape parameter of 1. The scale parameters are defined by setting the modes of the priors close to maximum likelihood estimates: σ_{OLS}^2 for σ_ϵ^2 (where σ_{OLS}^2 is the residual variance from an AR(1) model estimated by OLS), at $0.01\sigma_{OLS}^2$ for λ_0^2 and at 0.001^2 for λ_1^2. The modes of λ_0^2 and λ_1^2 are centered at the maximum likelihood estimates to ensure that the bimodality in the posterior distribution of the log of λ_1^2 documented in Figure 2 is not induced by the prior.

For given parameters, the likelihood is easily computed via the Kalman filter. It is therefore simple to find the posterior mode, at which the chain is initialized. Posterior mode values suggest that time variation is nearly absent.

Starting with Gibbs sampling, we draw 40 000 times after a burn-in of 5000. The recursive parameter means seem to settle down (not reported) and the posterior distributions are in line with a normal approximation taken at the mode, suggesting a persistent AR(1) with little sign of parameter variation (see Figure 1). It may therefore seem reasonable to assume that the chain has produced a sample representative of the entire posterior.

However, the AIMH scheme tells a different story. The proposal is initialized at a mixture of two normals $g_0(z) = 0.5\phi(z; \widehat{\mu}, \widehat{\Sigma}) + 0.5\phi(z; \widehat{\mu}, 16\widehat{\Sigma})$, where $\widehat{\mu}$ is the posterior mode and $-\widehat{\Sigma}$ is the inverse of the Hessian of the log-posterior evaluated at $\widehat{\mu}$. The AIMH soon discovers that the posterior distribution of $\log(\lambda_1^2)$, not to mention λ_1^2, is highly non-normal (see Figure 2), with substantial probability mass around a second mode corresponding to non-trivial amounts of time variation in ρ_t and a lower ρ_1.

We also ran the adaptive random walk Metropolis sampler outlined at the start of the

section. The sampler settles down to an acceptance rate of 20% and obtains the correct posterior distribution, and in particular finds both modes. Table 1 gives the inefficiency factors for all three samplers as well as the inefficiency factors of the Gibbs and ARWM relative to AIMH. The table shows that the AIMH sampler is appreciably more efficient than the other two samplers.

5.2 Additive semiparametric Gaussian models

In this example we consider the additive semiparametric regression model with Gaussian errors, with some of the covariates entering linearly and the others entering more flexibly

$$y_i = \sum_{j=1}^{m} \gamma_j z_{ji} + \sum_{h=1}^{H} f_h(x_{h,i}) + \sigma_\epsilon \epsilon_i \,; \tag{9}$$

the ϵ_i are $nid(0,1)$ and z is a vector of regressors that enter linearly. The x_h, $h = 1, ..., H$ are covariates that enter more flexibly by using the quadratic polynomial spline functions

$$f_h(x_{h,i}) = \beta_{0,h} x_{h,i} + \sum_{j=1}^{J} \beta_{h,j} (x_{h,i} - \tilde{x}_{h,j})_+^2 = \beta_{0,h} x_i + g_h(x_{h,i}), \tag{10}$$

where $x_+ = x$ if $x > 0$ and 0 otherwise and $\{\widetilde{x}_{h,1}, ..., \widetilde{x}_{h,k}\}$ are points (or 'knots') on the abscissae of x_h such that $\min(x_h) = \widetilde{x}_{h,1} < ... < \widetilde{x}_{h,J} < \max(x_h)$. In this paper we choose 30 knots so that each interval contains the same number of observed values of x_h. For a discussion of quadratic spline bases and other related bases see chapter 3 of Ruppert et al. (2003). We assume that a global intercept term is included in z in (9) and for simplicity we include the parameters $\beta_{h,0}$, $h = 1, \ldots, H$ in the vector γ and x_h, $h = 1, \ldots, H$ as part of the vector z. This transforms the nonparametric model into an highly parametrized linear model

$$y = \widetilde{Z}\widetilde{\gamma} + \epsilon. \tag{11}$$

The prior for the linear parameters γ is normal with a diagonal covariance matrix $\gamma \sim N(0, v_\gamma^2 I)$, where v_γ can be set to a large number. It is also convenient to assume a normal prior for the nonparametric part, with all parameters independent and $\beta_{h,j} \sim N(0, \tau_h^2)$, $, j =$

$1, ..., J$, $h = 1, ..., H$. However, with this prior there is a high risk of over-fitting if we simply set τ_h^2 to a large number. The variance τ_h^2 is often chosen by cross-validation, but in a fully Bayesian setting we can treat τ_h^2 as a parameter. To illustrate the advantage of AIMH in working with different priors, we experiment with two options for the prior τ_h^2. The first prior is log-normal and rather dispersed: $\ln(\tau_h^2) \sim N(0, 5^2)$, the second is inverse gamma with shape parameter 1 and scale parameter implied by setting the mode at 0.1^2. The prior for σ_ϵ^2 is inverse gamma with shape parameter one and scale parameter implied by setting the prior mode at the OLS residual variance estimated on (11). The prior for $\widetilde{\gamma} = (\gamma, \beta_1, ..., \beta_H)$ is therefore jointly normal conditional on $\tau^2 = \{\tau_1^2, .., \tau_H^2\}$, $\widetilde{\gamma}|\tau \sim N(\mathbf{0}, V_{\widetilde{\gamma}}(\tau))$, where $V_{\widetilde{\gamma}}(\tau) = \text{diag}(v_\gamma^2 I, \tau_1^2, I, \ldots, \tau_H^2 I)$ is a block diagonal matrix. One way to estimate the posterior density of the semiparametric model is to use Gibbs or Metropolis-within-Gibbs sampling as proposed by Wong and Kohn (1996). In this approach the parameters $\widetilde{\gamma} = \{\gamma, \beta_1, ..., \beta_H\}$ are conjugate given $\theta = \{\sigma_\epsilon^2, \tau_1^2, ..., \tau_H^2\}$, and σ_ϵ^2 is conjugate given $\widetilde{\gamma}$. Each variance τ_h^2 can be updated with a Gibbs step for the inverse gamma prior, or with a Metropolis-Hastings step for the log-normal prior. In this second case, we use a Laplace approximation of $p(\ln(\tau_h^2)|\beta_h)$, which is very fast to compute using analytical derivatives. However, the correlation between τ_h^2 and $\{\beta_{h,1}, .., \beta_{h,J}\}$ could be quite high using either prior for τ_h^2. In addition, using a log normal prior for τ_h^2 leads to high rejection rates in the Metropolis-Hastings step when generating the τ_h^2 in the Boston housing data example of section 5.2.1. Both problems are elegantly solved by integrating out $\widetilde{\gamma}$ and generating θ as a block using an efficient AIMH sampler.

The next example shows how to update all parameters in one block with an efficient AIMH sampler. We first note that, conditional on θ, $\widetilde{\gamma}$ can be integrated out, making it possible to compute $p(\theta|y) \propto p(y|\theta)p(\theta)$, where $y|\theta \sim N(\mathbf{0}, \sigma_\epsilon^2 I + \widetilde{Z} V_{\widetilde{\gamma}}(\tau) \widetilde{Z}')$.

5.2.1 Application: Boston housing data

We use the Gaussian semiparametric model to study the Boston housing data introduced by Harrison and Rubinfield (1978) and analyzed semiparametrically by Smith and Kohn (1996). The dataset is available at www.cs.utoronto.ca/~delve/data/boston and has 506 observations. The dependent variable is the log of MV, the median value of owner-occupied

homes. We use all 13 available covariates (see Smith and Kohn or the web-site for a full description) in the linear part and the following six in the nonparametric part (Smith and Kohn use only the first five): $X_5 = NOX$, nitrogen oxide concentration, $X_6 = RM$, average number of rooms, $X_8 = DIS$, logarithm of the distance from five employment centers, $X_{10} = TAX$, property tax rate, $X_{13} = STAT$, proportion of the population that is lower status, $X_1 = CRIM$, per capita crime rate by town.

The proposal distribution for the seven parameters $\theta = \{\ln(\sigma_\epsilon^2), \ln(\tau_5^2), ..., \ln(\tau_1^2)\}$ is initialized by fattening the tails of the Laplace approximation. To find the Laplace approximation, we simply apply Newton-Raphson optimization (with numerical derivatives) to $\ln p(y|\theta) + \ln p(\theta)$, which involves no extra coding effort since both densities are needed to compute the MH acceptance ratio. Figure 3 provides results for the case of a log-normal prior on τ_h^2, $h = 1, ..., H$ and shows that the acceptance rate quickly improves and stabilizes at around 60% when all seven parameters are updated jointly. Most parameters are approximately lognormally distributed, except those connected to the variables TAX and $CRIM$, which benefit from the added flexibility of mixtures. The correlation matrix of the smoothing parameters $\{\ln(\tau_5^2), ..., \ln(\tau_1^2)\}$ is nearly diagonal. This suggests that the AIMH could handle large numbers of smoothing parameters efficiently by updating them in blocks (with a different proposal density estimated adaptively on each block), since the blocks would be nearly independent of each other.

Table 1 reports the inefficiency factors for both the Gibbs sampler and the AIMH sampler for both inverse gamma and log normal priors, as well as the inefficiency of the Gibbs sampler relative to the AIMH sampler. The table shows that in terms of relative efficiency (defined at the beginning of section 5), the AIMH is about 40% more efficient than the Gibbs sampler when both samplers use the inverse gamma prior on τ_h^2, and nearly seven times more efficient when both samplers use the log-normal prior. Reported results are for the average inefficiency factors (over both h and i) of $f_h(x_{h,i})$. Looking at the autocorrelation of the log-parameters gives similar inefficiency ratios.

We also applied the adaptive random walk Metropolis sampler to this data set, but could not make it work well. With the sampler initialized at the posterior mode, the acceptance rate started at over 50%, but within a few hundred iterations fell to below 1% and stayed

there indefinitely. We do not report any inefficiency factors for this sampler because we do not believe that inference is reliable with such a low acceptance rate. We conjecture that the poor behavior of the ARWM sampler in this example compared to the other two examples is because this example has 7 parameters whereas the other two have 3 and 2 parameters. In addition, the second derivatives of the log posterior in this example are far from constant, so a unique covariance matrix may do very poorly. By contrast, a mixture of normals allows for local correlations between the parameter and therefore may be less affected.

Boston	mean $f_h(x_{h,i})$		**Inflation**	$\log(\sigma_\epsilon^2)$	$\log(\lambda_0^2)$	$\log(\lambda_1^2)$
AIMH, IG	2.6		AIMH	6.7	2.8	6.1
Gibbs, IG	6.3 (1.4)		Gibbs	9.4 (1.3)	113.3 (37.4)	156.4 (23.7)
AIMH, LN	1.6		ARWM	21.5 (3.1)	23.5 (8.3)	23.6 (3.8)
M-Gibbs, LN	18.4 (6.8)					

Table 1: Inefficiency factors for the semiparametric (Boston) and state space (inflation) models, together with the inefficiencies of the Gibbs sampler and the ARWM relative to the AIMH sampler in brackets. AIMH: adaptive independent Metropolis-Hastings; M-Gibbs: Metropolis-within-Gibbs; and ARWM: adaptive random walk Metropolis. IG and LN: inverse gamma and log-normal priors for the Boston data.

6 Discussion and conclusion

In order to understand the strengths and limitations of our sampler, we find it useful to consider two desirable qualities of an adaptive IMH scheme. First, given a sufficiently large sample drawn from the target, we wish to construct a proposal density which fits the target as well as possible. This is an approximating ability: we want to accurately 'map' the areas that we have already explored. Second, we wish the sampler to perform as well as possible when the initial proposal fails to cover part of the support of the target distribution. This is an exploring ability: when we propose in a region where our map is poor, we want to explore that region and quickly update our map.

For example, using a normal proposal when the target is highly non-normal results in little approximating ability. Updating the proposal only once or very rarely results in little exploring ability, since the proposal reacts slowly or not at all to the information that it is fitting poorly at a given point.

Our sampler has several characteristics designed to enhance its exploring ability. Frequent updating, particularly at early iterations, and updating following a sequence of low MH acceptance probabilities quicken the pace at which the proposal adapts to the information that it is not fitting well in a certain area. Long tails are useful not only to satisfy (3) and (4), but also to improve the chances of venturing into unexplored parts of the state space. Finally, mixtures are ideally suited for this exploration because a new component can be centered on a value causing a sequence of rejections. The long runs of rejections that can plague standard IMH are therefore much less likely using our AIMH sampling scheme because the proposal distribution is updated frequently and will accommodate the cluster of rejections by changing the mixture parameters or adding a new component. If our sampler makes a move in a region where the proposal fits poorly, it should therefore be able to explore it. Of course as the parameter dimension increases, if the initial proposal fails to cover a region we may never explore that region simply because the probability of making a proposal there is too small.

The most interesting applications arise when current best practice is inefficient or cumbersome and, in our opinion, when adaptation starts early. In work in progress we have worked successfully with 15 to 25 parameters in several real data applications, but in general, the number of parameters that it is possible to handle using our approach depends on the shape of the target distribution.

Acknowledgement

We would like to thank Luke Tierney, Christophe Andrieu, Antonietta Mira and two anonymous referees for helpful suggestions and questions that helped improve the accuracy and presentation of a previous version of the paper. Robert Kohn's research was partially supported by ARC grant DP0667069.

References

Andrieu, C. and Moulines, D. (2006), "On the ergodicity properties of some adaptive MCMC algorithms," *Annals of Applied Probability*, 16, 1462–1505.

Andrieu, C., Moulines, D., and Doucet, A. (2005), "Stability of stochastic approximation under verifiable conditions," *SICON*, 44, 283–312.

Andrieu, C. and Robert, C. P. (2001), "Controlled MCMC for optimal sampling," Technical report, University of Bristol.

Atchade, Y. and Liu, J. (2007), "The Wang-Landau algorithm for Monte carlo computation in general state spaces," Available at http://www.mathstat.uottawa.ca/ yatch436/gwl.pdf.

Atchadé, Y. and Rosenthal, J. (2005), "On adaptive Markov chain Monte Carlo algorithms," *Bernoulli*, 11, 815–828.

Bradly, P. and Fayyad, U. (1998), "Refining initial points for k-means clustering," *Proceedings of the 15th International Conference on Machine Learning*, 91–99.

Carter, C. and Kohn, R. (1994), "On Gibbs sampling for state-space models," *Biometrika*, 83, 589–601.

Figuereido, M. and Jain, A. (2002), "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 381–396.

Fruhwirth-Schnatter, S. (2004), "Efficient Bayesian parameter estimation," in *State space and unobserved component models*, eds. Harvey, A., Koopman, S., and Shephard, N., Cambridge: Cambridge University Press, pp. 123–151.

Gåsemyr, J. (2003), "On an adaptive version of the Metropolis-Hastings algorithm with independent proposal distribution," *Scandinavian Journal of Statistics*, 30, 159–173.

Gilks, W., Roberts, G., and Sahu, S. (1998), "Adaptive Markov chain Monte Carlo through regeneration," *Journal of the American Statistical Association*, 93, 1045–1054.

Giordani, P. and Kohn, R. (2008), "Efficient Bayesian inference for multiple change-point and mixture innovation models," *Journal of Business and Economic Statistics*, 26, 66–77.

Haario, H., Saksman, E., and Tamminen, J. (2001), "An adaptive Metropolis algorithm," *Bernoulli*, 7, 223–242.

Hamerly, G. and Elkan, C. (2002), "Alternatives to the k-means algorithm that find better clusterings," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, eds. Kalpakis, K., Goharian, N., and Grossmann, D., New York: Academic Press, pp. 600–607.

Harrison, D. and Rubinfield, D. (1978), "Hedonic prices and the demand for clean air," *Journal of Environmental Economics and Management*, 5, 81–102.

Hastie, D. (2005), "Towards automatic reversible jump Markov chain Monte Carlo," Unpublished PhD dissertation, Department of Mathematics, University of Bristol.

Hastings, W. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 57, 97–109.

Holden, L. (1998), "Adaptive chains," Manuscript, Norwegian Computing Center, Oslo.

Liang, F., Liu, C., and Carroll, R. J. (2007), "Stochastic approximation in Monte Carlo computation," *Journal of the American Statistical Association*, 102, 305–320.

McLachlan, G. and Peel, D. (2000), *Finite Mixture Models*, New York: Wiley.

Mengersen, K. L. and Tweedie, R. L. (1996), "Rates of convergence of the Hastings and Metropolis algorithms," *The Annals of Statistics*, 24, 101–21.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., H., T. A., and Teller, E. (1953), "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, 21, 1087–1092.

Nott, D. and Kohn, R. (2005), "Adaptive sampling for Bayesian variable selection," *Biometrika*, 92, 747–763.

Roberts, G. O. and Rosenthal, J. S. (2006), "Examples of adaptive MCMC," Preprint (http://probability.ca/jeff/ftpdir/adaptex.pdf).

— (2007), "Coupling and ergodicity of adaptive MCMC," *Journal of Applied Probability*, 44, 458–475.

Ruppert, D., Wand, M., , and Carroll, R. (2003), *Semiparametric regression*, Cambridge: Cambridge University Press.

Smith, M. and Kohn, R. (1996), "Nonparametric regression using Bayesian variable selection," *Journal of Econometrics*, 75, 317–343.

Spall, J. (2003), *Introduction to Stochastic Search and Optimization*, New York: Wiley.

Ueda, N., Nakano, R., Ghahramani, Z., and Hinton, G. (2000), "SMEM algorithm for mixture models," *Neural Computation*, 12, 2109–2128.

Verbeek, J., Vlassis, N., and Krose, B. (2003), "Efficient Greedy Learning of Gaussian Mixture Models," *Neural Computation*, 15, 469–485.

Wang, F. and Landau, D. P. (2001), "Efficient multiple-range random walk algorithm to calculate the density of states," *Physical Review*, 86, 2050–2053.

Wong, C. and Kohn, R. (1996), "A Bayesian approach to additive semiparametric regression," *Journal of Econometrics*, 74, 209–235.

Appendix 1: k-harmonic means clustering

We estimate the mixture of normal parameters using the k-harmonic means clustering algorithm which can be described as follows. (See Hamerly and Elkan 2002, for a discussion). Let p be the number of clusters.

1. Initialize the algorithm with $c_1, ..., c_p$, the component centers. The starting values are chosen with the procedure of Bradly and Fayyad (1998) . We depart slightly from Bradley and Fayyad in using the harmonic k-means algorithm (rather than k-means) in the initialization procedure.

2. For each data point θ_t, compute a weight function $w(\theta_t)$ and a membership function $m(c_i|\theta_t)$ for $t = 1, ..., n$ as

$$w(\theta_t) = \frac{\sum_{i=1}^{p} ||\theta_t - c_i||^{-p-2}}{(\sum_{i=1}^{p} ||\theta_t - c_i||^{-p})^2} \quad \text{and} \quad m(c_i|\theta_t) = \frac{||\theta_t - c_i||^{-p-2}}{\sum_{i=1}^{p} ||\theta_t - c_i||^{-p-2}},$$

where $||\theta_t - c_i||$ is the Euclidean or Mahalanobis distance. Following Bradly and Fayyad (1998), we put a lower boundary ϵ on $||\theta_t - c_i||$ (to avoid degeneracies when trying $||c_i - c_i||$). The membership function softens the sharp membership of the k-means algorithm, so one observation can belong to more than one cluster in differing degrees. The weight function gives more weight to observations that are currently covered poorly (i.e. that are far from the nearest center).

3. Update each center c_i

$$c_i = \frac{\sum_{t=1}^{n} m(c_i|\theta_t)w(\theta_t)\theta_t}{\sum_{t=1}^{n} m(c_i|\theta_t)w(\theta_t)}.$$

4. Repeat until convergence. This gives the cluster centers, which we take as estimates of the component means. The other mixture parameters can then be estimated for $i = 1, ..., k$ as

$$V_i = \frac{\sum_{t=1}^{n} m(c_i|\theta_t)w(\theta_t)(\theta_t - c_i)(\theta_t - c_i)'}{\sum_{t=1}^{n} m(c_i|\theta_t)w(\theta_t)} \quad \text{and} \quad \pi_i \propto \sum_{t=1}^{n} m(c_i|\theta_t)w(\theta_t).$$

5. The number of clusters is chosen with the BIC criterion given a maximum number (5 in our examples).

We notice that the covariance matrices V_i are only estimated once, after convergence. k-means type algorithms also differ from the EM algorithm in that they do not evaluate the likelihood $p(\theta|c_1, ..., \pi_1, V_1, ...)$. This sub-optimal use of information in fact turns out to be a great advantage for our purposes. Fewer iterations than for EM are needed for convergence, and each iteration is faster. Even more importantly, the algorithm does not get stuck in the small degenerate clusters caused by rejections in the sense that, unlike for the EM algorithm with freely estimated covariances, these small clusters are not absorbing. If k-harmonic means does find a degenerate cluster, this causes no trouble for convergence, and after convergence we can use a predefined matrix in place of any non-positive-definite covariance matrix (for example, if V_i is not positive definite we set it to $0.5^2 Var(\theta)$). If desired, the mixture parameters can be refined with a few steps of the EM algorithm. In this case, we recommend not updating the the covariance matrices for the reasons just discussed.

Appendix 2: Proofs

The one-step transition kernel for Z_{n+1} in section 2 is given by

$$T_n(z, dz') = \alpha_n(z, z')q_n(z')\mu\{dz'\} + \delta_z(dz')(1 - \nu_n(z)) \tag{12}$$

where $\delta_z(dz') = 1$ if $z \in dz'$ and is 0 otherwise, and

$$\nu_n(z) = \int_{\mathcal{Z}} \alpha_n(z, z')q_n(z')\mu\{dz'\}. \tag{13}$$

By the construction of the MH transition kernel,

$$\int_{\mathcal{Z}} \pi(z)T_n(z, dz')\mu\{dz\} = \pi(z')\mu\{dz'\} \ . \tag{14}$$

In this section K is a generic constant, independent of n, z and z'. It is convenient to write any function of z and λ of the form $h_n(z; \lambda_n)$ as $h_n(z)$. Without loss of generality we assume throughout this section that $\mathcal{Z}$ is a discrete space. Exactly the same proof goes through for the continuous case with summations replaced by integrals. We use the notation $z_{s:t}$ to mean $\{z_s, \ldots, z_t\}$ for $s \leq t$, with a similar interpretation for $Z_{s:t}$.

To prove Theorem 1 we first obtain the following two lemmas.

Lemma 1 *Under the assumptions of Section 2, for any $n, k > 0$ and $z, z' \in \mathcal{Z}$,*

(a) $q_n(z) \leq Kg_0(z)$.

(b) $\alpha_n(z, z')q_n(z') \leq Kg_0(z')$

(c) There exists an ϵ_1, $0 < \epsilon_1 < 1$, such that $\alpha_n(z, z')q_n(z') > \epsilon_1\pi(z')$ for all $z, z' \in \mathcal{Z}$.

(d) $\nu_n(z) > \epsilon_1$ for all $z \in \mathcal{Z}$, where $\nu_n(z)$ is defined by (13)*.*

(e) For $k \geq 1$, let $\Delta_n(z, z') = \alpha_n(z, z')q_n(z') - \alpha_{n+1}(z, z')q_{n+1}(z')$. Then,

$$\mid \Delta_n(z, z') \mid \leq \ K\left(g_0(z') + \frac{\pi(z')}{\pi(z)}g_0(z)\right)a_n. \tag{15}$$

(f)

$$\mid \nu_n(z) - \nu_{n+1}z) \mid \leq K\left(1 + \frac{g_0(z}{\pi(z)}\right)a_n \tag{16}$$

Proof. (a) $q_n(z)/g_0(z) = \omega_1 + (1-\omega_1)g_n(z)/g_0(z)$ and the result follows from (3). (b) follows from (a) and $\alpha_n(z,z') \leq 1$. To show (c), note that $q_n(z)/\pi(z) \geq \omega_1 g_0(z)/\pi(z)$. From (3), there is an ϵ_1 such that $q_n(z)/\pi(z) > \epsilon_1$ for all $z \in \mathcal{Z}$. It is now straightforward to show that $\alpha_n(z,z')q_n(z')/\pi(z') > \epsilon_1$ for all $z, z' \in \mathcal{Z}$. (d) follows from

$$\nu(z) = \sum_{z'} \alpha_n(z,z')q_n(z') > \epsilon_1 \sum_{z'} \pi(z') = \epsilon_1$$

To obtain (e), it is necessary to consider the following four cases.

Case 1. $\alpha_n(z,z') = 1$ and $\alpha_{n+1}(z,z') = 1$. Then, $|\Delta_n| = \mid q_n(z') - q_{n+1}(z') \mid \leq Kg_0(z')a_n$ by (4).

Case 2. $\alpha_n(z,z') < 1$ and $\alpha_{n+1}(z,z') < 1$.

$$\mid \Delta_n(z,z') \mid = \frac{\pi(z')}{\pi(z)} \mid q_n(z) - q_{n+1}(z) \mid \leq K\frac{\pi(z')}{\pi(z)} g_0(z)a_n.$$

Case 3. $\alpha_n(z,z') = 1$ and $\alpha_{n+1}(z,z') < 1$. In this case $\Delta_n(z,z') = q_n(z') - \pi(z')q_{n+1}(z)/\pi(z)$. If $\Delta_n(z,z') \geq 0$, then

$$0 \leq \Delta_n(z,z') \leq \frac{\pi(z')}{\pi(z)}\Big(q_n(z) - q_{n+1}(z)\Big) \leq Kg_0(z)a_nc_k \ .$$

If $\Delta_n(z,z') < 0$, then

$$0 < -\Delta_n(z,z') = \frac{\pi(z')}{\pi(z)} q_{n+1}(z) - q_n(z') \leq q_{n+1}(z') - q_n(z')$$

Thus,

$$\mid \Delta_n(z,z') \mid \leq K\Big(g_0(z') + \frac{\pi(z')}{\pi(z)}\Big)a_n.$$

Case 4. $\alpha_n(z,z') < 1$ and $\alpha_{n+1}(z,z') = 1$. This case is similar to case 3.

To obtain (f), we note that

$$\mid \nu_n(z) - \nu_{n+1}(z) \mid \leq \sum_{z'} |\Delta_n(z,z')| \ ,$$

and the result follows from (e). ■

With ϵ_1 as in Lemma 1, choose $0 < \epsilon < \epsilon_1$ and let

$$R_n(z, z') = \frac{T_n(z, z') - \epsilon\pi(z')}{1 - \epsilon} \tag{17}$$

Then, $R_n(z, z')$ is a one-step transition kernel with the following properties.

Lemma 2 *(a)*

$$\sum_z \pi(z) R_n(z, z') = \pi(z') \ .$$

(b)

$$R_n(z, z') \leq K g_0(z_n) + \eta \delta_z(z')$$

where $0 < \eta < 1$.

(c)

$$| \ R_n(z, z') - R_{n+1}(z, z') \ | \ \leq \ K a_n \left\{ \left(g_0(z') + \frac{\pi(z')}{\pi(z)} g_0(z) \right) + \left(1 + \frac{g_0(z)}{\pi(z)} \right) \delta_z(z') \right\}$$

(d)

$$\sum_{z_{n-m+1}} \cdots \sum_{z_{n-1}} \prod_{k=1}^{m} R_{n-k}(z_{n-k}, z_{n-k+1}) \leq K g_0(z_n) + \eta^m \delta_{z_{n-m}}(z_n)$$

(e) For $1 \leq l \leq j-1$ *and* $j = 1, \ldots, n$,

$$\sum_{z_{n-l-1}} \cdots \sum_{z_{n-j+1}} \pi(z_{n-j+1}) \prod_{k=1+1}^{j-1} R_{n-j}(z_{n-k}, z_{n-k+1}) = \pi(z_{n-l})$$

(f) For $j = 1, \ldots, n$ *and* $l = 1, \ldots, j-1$,

$$| \sum_{z_{n-j+1}} \cdots \sum_{z_{n-1}} \pi(z_{n-j+1}) \prod_{k=l+1}^{j-1} R_{n-j}(z_{n-k}, z_{n-k+1})$$
$$\times \left(R_{n-l}(z_{n-l}, z_{n-l+1}) - R_{n-j}(z_{n-l}, z_{n-l+1}) \right) \prod_{k=1}^{l-1} R_{n-k}(z_{n-k}, z_{n-k+1}) \ | \ \leq \ K g_0(z_n) a_{n-j} (j-l)^{-1}$$

Proof. (a) follows from (17) and (14). (b) follows from (17). (c) follows from (15) and (16). (d) is true for $m = 1$ and is obtained in general by induction. (e) follows from part (a). (f) follows from parts (a) to (e). ■

Proof of Theorem 1. Let $\Delta_j, j = 1, 2, \ldots$ be an independent Bernoulli process such that $\Delta_j = 1$ with probability ϵ and $\Delta_j = 0$ with probability $1 - \epsilon$. From (17), $T_n(z, z') = (1-\epsilon)R_n(z, z') + \epsilon\pi(z')$ so that we can interpret $T_n(z, z')$ as a mixture of transition kernels, such that $T_n(z, z') = R_n(z, z')$ if $\Delta_n = 0$ and $T_n(z, z') = \pi(z')$ if $\Delta_n = 1$. For $j = 1, \ldots, n$, let $A_{n,j}$ be the event that $\Delta_{n-j+1} = 1, \Delta_k = 0, k = n - j + 2, \ldots, n$. Let B_n be the event that $\Delta_j = 0$ for $j = 1, \ldots, n$. Then $\Pr(A_{n,j}) = \epsilon(1-\epsilon)^{j-1}$ and $\Pr(B_n) = (1-\epsilon)^n$, and

$$\Pr(Z_n = z_n) = \sum_{j=1}^{n} \Pr(Z_n = z_n \mid A_{n,j}) \Pr(A_{n,j}) + \Pr(Z_n = z_n \mid B_n) \Pr(B_n).$$

As in the proof of Theorem 1 in Nott and Kohn (2005), we can write $\Pr(Z_n = z_n | A_{n,j}) = C_{0,j} + C_{1,j} + \cdots + C_{j-1,j}$, where

$$C_{l,j} = \sum_{z_0} \cdots \sum_{z_{n-1}} \Pr(Z_{0:n-j} = z_{0:n-j}) \pi(z_{n-j+1}) \prod_{k=l+1}^{j-1} R_{n-j}(z_{n-k}, z_{n-k+1})$$
$$\left(R_{n-l}(z_{n-l}, z_{n-l+1}) - R_{n-j}(z_{n-l}, z_{n-l+1}) \right) \prod_{k=1}^{l-1} R_{n-k}(z_{n-k}, z_{n-k+1})$$

From part (e) of Lemma 2, $C_{0,j} = \pi(z_n)$ and by part (f) of Lemma 2, $\mid C_{j,n} \mid \leq K g_0(z_n) a_{n-j}(j-l)^{-1}$ for $j > 1$. Using a similar argument to that in Nott and Kohn (2005), this implies that

$$\left| \sum_{l=0}^{j-1} C_{l,j} \right| \leq \pi(z_n) + K g_0(z_n)(n-j)^{-r_1} j^2 \ .$$

Thus,

$$\sum_{j=1}^{n} \Pr(Z_n = z_n | A_{n,j}) \Pr(A_{n,j}) = \pi(z_n) - (1-\epsilon)^n \pi(z_n) + \eta_n \quad \text{where}$$
$$\mid \eta_n \mid \leq \ K n^{-r_1} \sum_{j=1}^{n-1} \left(1 - \frac{j}{n}\right)^{-r_1} j^2 \epsilon (1-\epsilon)^{j-1}.$$

We also have that

$$\Pr(Z_n = z_n \mid B_n) = \sum_{z_0} \cdots \sum_{z_{n-1}} q_0(z_0) g_0(z_1) \prod_{k=1}^{n-1} R_k(z_k, z_{k+1})$$
$$\leq K g_0(z_n) + \eta^{n-1} g_0(z_n) \leq K g_0(z_n) \ .$$

using Lemma 2 (c) and (3). Hence,

$$\mid \Pr(Z_n = z_n) - \pi(z_n) \mid \leq \ K g_0(z_n) \Big((1-\epsilon)^n + n^{-r_1} \Big) \tag{18}$$

The proof of Theorem 1 follows. ■

The proof of Theorem 2 is similar to that in Nott and Kohn (2005) if we use (18).

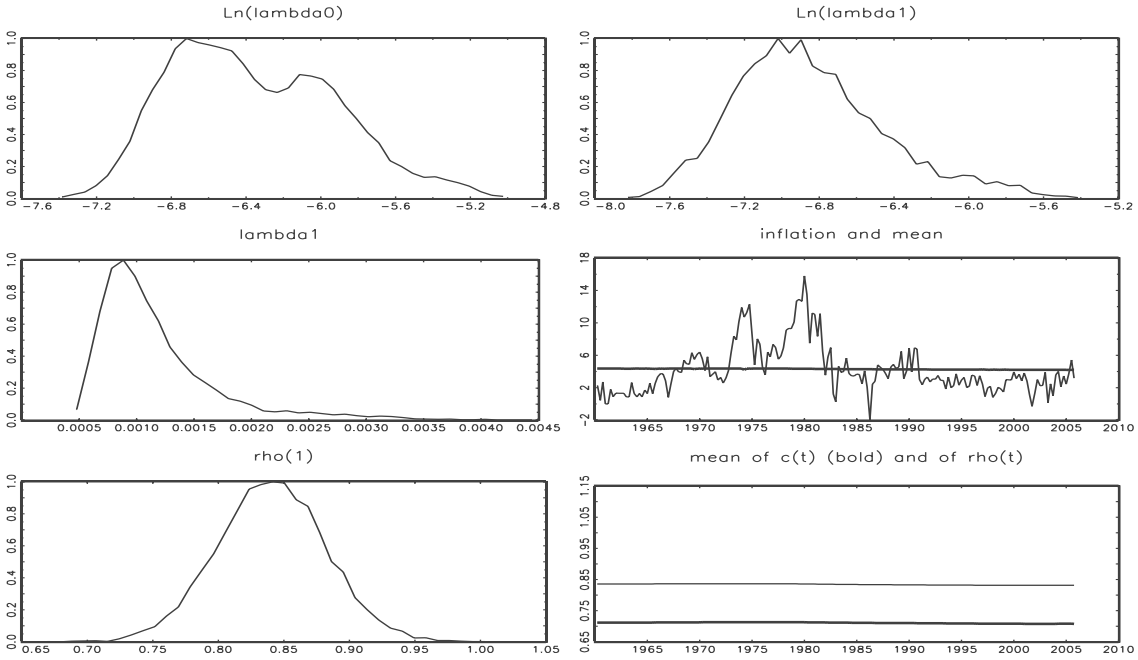

Figure 1: Inference for a time varying parameter AR(1) model for US inflation by Gibbs sampling. (a) marginal distribution of $\ln(\lambda_0)$ (b) marginal distribution of $\ln(\lambda_1)$ (c) marginal distribution λ_1 (d) inflation plot and mean, estimated as $E[(c_t/1-\rho_t)|y]$ (e) marginal distribution of $\rho_0|y$ (f) $E(c_t|y)$ (bold line) and $E(\rho_t|y)$.

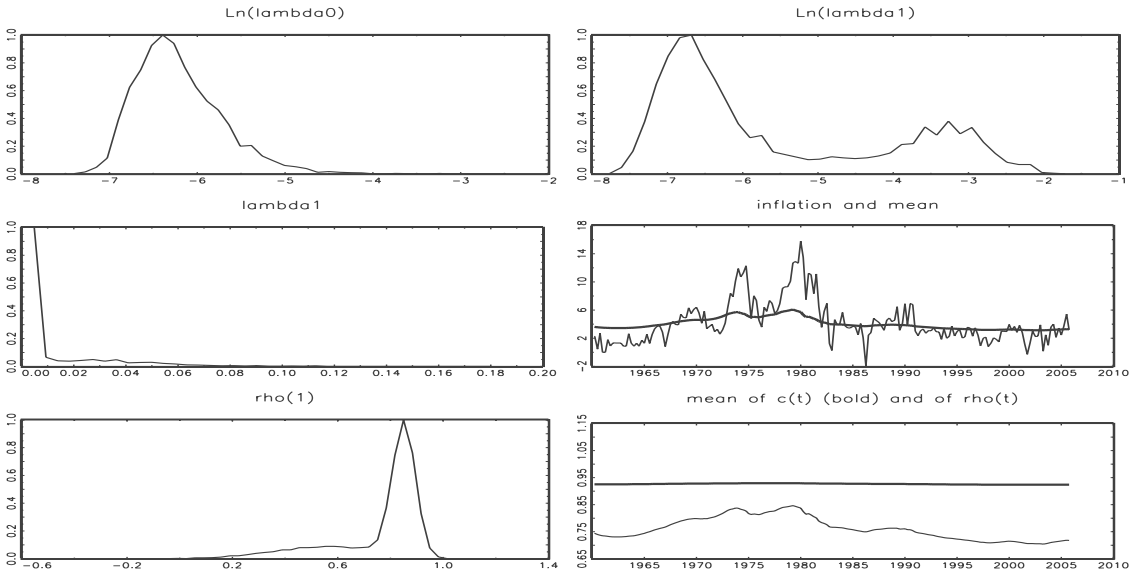

Figure 2: Inference for the model of figure 1 by adaptive IMH. The interpretation of the panels is the same as in figure 1.

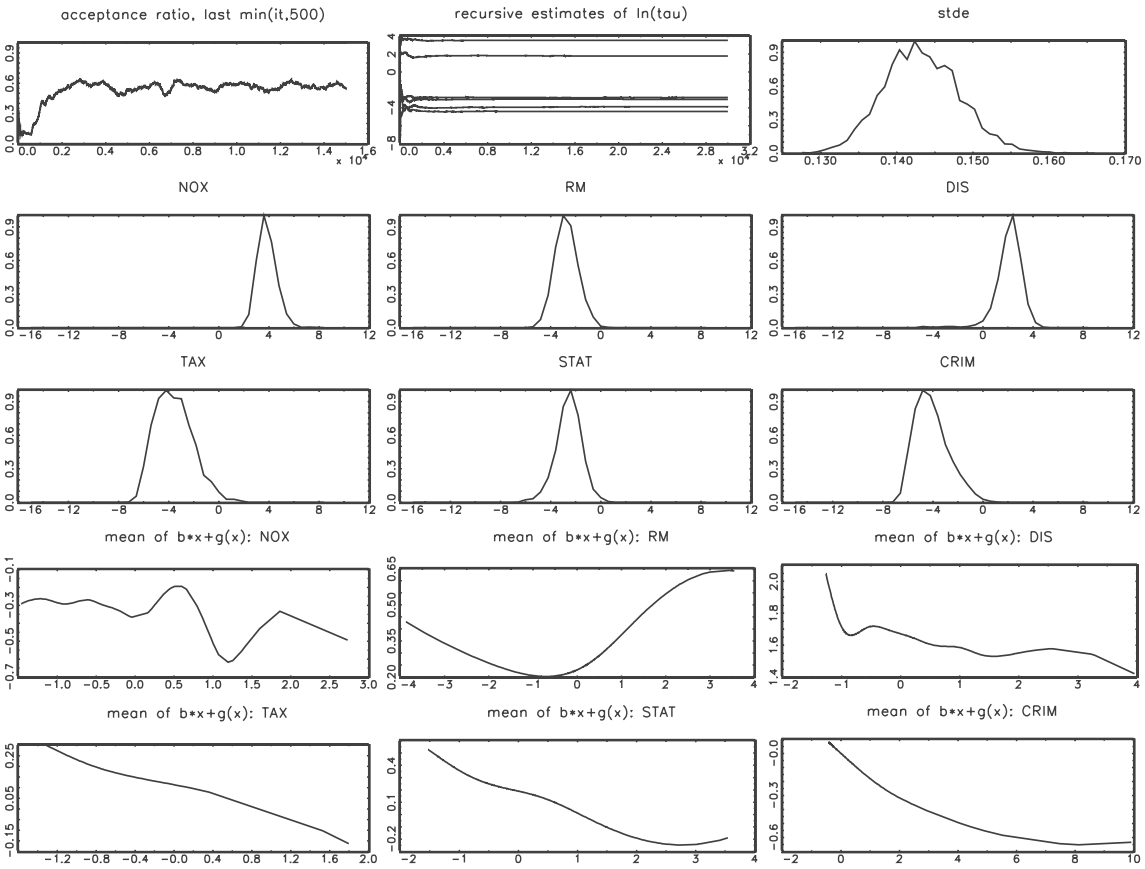

Figure 3: Inference for semiparametric model of housing prices by adaptive IMH. First row: recursive acceptance rate for the last min(it,500) iterations, recursive means of $\ln(\tau_i)$, marginal of σ_ϵ. Second and third rows: marginals of $\ln(\tau_i)$. Fourth and fifth rows: means of $\beta_i x + g_i(x)$.