Adaptive Mixture Modelling Metropolis Methods for Bayesian Analysis of Non-linear State-Space Models

Jarad Niemi, Duke University
Mike West, Duke University
Adaptive Mixture Modelling Metropolis Methods for Bayesian Analysis of Non-linear State-Space Models

Jarad Niemi and Mike West
Department of Statistical Science, Duke University, Durham, NC 27708-0251
Jarad Niemi: jarad@stat.duke.edu; Mike West: mw@stat.duke.edu

Abstract

We describe a strategy for Markov chain Monte Carlo analysis of non-linear, non-Gaussian state-space models involving batch analysis for inference on dynamic, latent state variables and fixed model parameters. The key innovation is a Metropolis-Hastings method for the time series of state variables based on sequential approximation of filtering and smoothing densities using normal mixtures. These mixtures are propagated through the non-linearities using an accurate, local mixture approximation method, and we use a regenerating procedure to deal with potential degeneracy of mixture components. This provides accurate, direct approximations to sequential filtering and retrospective smoothing distributions, and hence a useful construction of global Metropolis proposal distributions for simulation of posteriors for the set of states. This analysis is embedded within a Gibbs sampler to include uncertain fixed parameters. We give an example motivated by an application in systems biology. Supplemental materials provide an example based on a stochastic volatility model as well as MATLAB code.

Keywords
Bayesian computation; Forward filtering; backward sampling; non-linear state-space model; Regenerating mixture procedure; Smoothing in state-space models; Systems biology

1 Introduction

Motivated by problems of fitting and assessing structured, non-linear dynamic models to time series data arising in studies of cellular networks in systems biology, we revisit the problem of Bayesian inference on latent, time-evolving states and fixed model parameters in a state-space model context. In recent years, this general area has seen development of a number of customized Monte Carlo methods, but existing approaches do not yet provide the kind of comprehensive, robust, generally and automatically applicable computational methods needed for repeat batch analysis in routine application. The above criteria for effective statistical computation are central to our primary motivating context of dynamic cellular networks, an applied field that is beginning to grow rapidly as relevant high-resolution time series data on genetic circuitry becomes increasingly available in single-cell and related studies (Rosenfeld et al., 2005; Golightly and Wilkinson, 2005; Wilkinson, 2006; Wang et al., 2009) and synthetic

Supplemental materials
Supplemental materials are available for download via a single zipped file (am4-supplemental.zip). This file contains the following files:
- Non-linear non-Gaussian models: This file describes an extension of the AM4 framework for nonlinear non-Gaussian dynamic models. It contains an example based on a stochastic volatility model. (SV.pdf)
- MATLAB code for AM4 Folders contain all code and datasets necessary for executing the examples in this paper. A README.txt file contains all details about the code and datasets in those folders.
bioengineering (Tan et al., 2007). With this motivation, we have built on the best available analytic and Monte Carlo tools for non-linear dynamic models to generate a novel adaptive mixture modelling method that is embedded in an overall MCMC strategy for posterior computation; the resulting methodology satisfies the above criteria while also being computationally efficient, and has generated most effective MCMC analyses from the viewpoint of convergence in a range of example models we have studied.

Given a specified model and a series of observations over a fixed time interval, we are interested in summaries of the posterior distribution for the full series of corresponding state vectors as well as fixed model parameters; this is the batch analysis. Sequential filtering and retrospective smoothing using Monte Carlo is central to our general, non-linear forward-filtering, backward sampling (FFBS) approach that extends the profoundly useful FFBS methodology of conditionally linear, normal models introduced in Carter and Kohn (1994) and Frühwirth-Schnatter (1994); see also West and Harrison (1997), chapter 15. Filtering is the canonical setting for sequential particulate methods (West, 1992; Gordon et al., 1993; West, 1993b; Chen and Liu, 2000; Doucet et al., 2001; Liu and West, 2001); retrospective smoothing analysis has been explored in such contexts, in terms of both marginal (Kitagawa, 1996; Hürzeler and Künsch, 1998; Doucet et al., 2000) and joint ( Godsill et al., 2004) smoothing approaches. We do not, however, use particle filtering methods; though useful and interesting, the well-known shortcomings of these methods, including the key issue of particle attrition, are currently simply limiting from the viewpoint we have of robust, stable, and automatically applicable methods for a range of non-linear models. Previous approaches using MCMC in non-linear non-Gaussian state-space models include Carlin et al. (1992), Geweke and Tanizaki (1999), and Stroud et al. (2003). In particular, Stroud et al. (2003) suggest sampling latent states in blocks from an auxiliary mixture model for use as a Metropolis proposal. The main difference of our method is that our MCMC scheme does not condition on mixture component indicators. Closer to our perspective is Ravines et al. (2007) who, in the class of dynamic generalized linear models (West and Harrison, 1997, chapter 14), develop a global Metropolis-Hastings analysis where the proposal distribution for state vectors are generated from analytic approximations to filtering and smoothing distributions that are known to be accurate, and hence can be expected to lead to reasonable acceptance rates. We develop this perspective using adaptive mixture approximations to filtering distributions that apply widely. This leads to accurate, direct analytic approximations to the smoothed distributions for the full set of states in a batch analysis, and hence to effective Monte Carlo that uses these approximations as proposal distributions. This sampling strategy for latent states is embedded in an overall MCMC that couples in samplers for fixed model parameters to define a complete analysis.

Section 2 introduces the state-space model context, focusing on non-linear models with additive Gaussian noise. Section 3 reviews mixture model approximations in state-space models, and develops a regenerating procedure to improve the utility of Gaussian sum mixtures in adaptive mixture modelling for analytic approximation to sequential filtering and smoothing. Section 4 then embeds the mixture analysis in an overall MCMC as the novel Metropolis proposal method for latent states. Section 5 illustrates the analyses with non-linear models relevant in systems biological studies of dynamic cellular networks. Section 5.2 provides a full Bayesian analysis of a model with fixed parameters and a two-dimensional state vector in a systems biology example. Comments on, and comparisons with, prior methods are included throughout. Section 6 provides summary comments.

2 State-Space Model and FFBS Analysis

Begin with the Markovian state-space model (West and Harrison, 1997)
where \( x_t \) is the unobserved state of the system at time \( t \), \( y_t \) is the observation at time \( t \), \( f(\cdot) \) and \( g(\cdot) \) are known, non-linear observation and evolution functions, and \( \nu_t \sim N(0, V_t) \) and \( \omega_t \sim N(0, W_t) \) are independent and mutually independent Gaussian terms, the observation and evolution noise terms, respectively. Initially we assume any model parameters in the non-linear functions or noise variances are known. We note that the development can be extended well beyond our focus here on additive, normal noise terms, though for specificity we focus on that structure here.

We use \( s:t \) to denote consecutive times \( s, s + 1, \ldots, t \) for any \( s, t \geq s \), so that \( x_{s:t} = \{ x_s, \ldots, x_t \} \) and so forth. Based on the batch of data \( y_{1:T} \), the main goal is simulation of the set of states \( x_{0:T} \) from the implied posterior

\[
p(x_{0:T} \mid y_{1:T}) \propto p(x_0) \prod_{t=1}^{T} p(y_t \mid x_t) p(x_t \mid x_{t-1})
\]

(2)

where \( p(x_0) \) is the density of the initial state and \( p(y_t \mid x_t) \) and \( p(x_t \mid x_{t-1}) \) are defined by equation (1). This is done using the FFBS strategy:

- **FF**: For each \( t = 1:T \) in sequence, sequentially process the datum \( y_t \) to update numerical summaries of the filtering densities \( p(x_t \mid y_{1:t}) \) at time \( t \).
- **BS**: Simulate the joint distribution in equation (2) via the implied backward compositional form

\[
p(x_{0:T} \mid y_{1:T}) \propto p(x_T) \prod_{t=1}^{T} p(x_{t-1} \mid x_t, y_{1:(t-1)}).
\]

(3)

That is:

a. draw \( x_T \sim p(x_T \mid y_{1:T}) \) and set \( t = T \);

b. draw \( x_{t-1} \sim p(x_{t-1} \mid x_t, y_{1:(t-1)}) \);

c. reduce \( t \) to \( t - 1 \) and return to step b; stop when \( t = 0 \).

This generates the full joint sample \( x_{0:T} \) in reverse order. All steps of FFBS depend fundamentally on the structure of the joint densities

\[
p(y_{1:T}, x_{0:T}) = p(y_t \mid x_t) p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:(t-1)}).
\]

(4)

In particular, filtering relies on the ability to compute and summarize

\[
p(x_t \mid y_{1:t}) \propto \int p(x_t, x_{t-1}) p(x_{t-1} \mid y_{1:(t-1)}) dx_{t-1} = \int p(y_t, x_t, x_{t-1} \mid y_{1:(t-1)}) dx_{t-1}
\]

(5)
while backward sampling relies on the ability to simulate from

\[ p(x_{t-1} | x_t, y_{1:t-1}) \propto p(x_{t-1} | y_{1:t-1}) p(x_t | x_{t-1}) = p(x_t, x_{t-1} | y_{1:t-1}), \]

(6)

the bivariate margin of equation (4). In linear, Gaussian models, these distributions are all Gaussian; in non-linear models, the implied computations require some form of approximation.

3 Normal Mixture Model Approximations

3.1 Background and Notation

Our strategy is based on approximation of the sequentially updated distributions of states via mixtures of many, very precise normal components. Mixtures have been used broadly in dynamic modelling, for both model specification and computational methods, especially in adaptive multi-process models and to represent model uncertainty in terms of multiple models analyzed in parallel (chapter 12 and references in West and Harrison, 1997; Fearnhead and Meligkotsidou, 2007). The basic idea of normal mixture approximation in non-linear state-space models in fact goes back several decades to at least Harrison and Stevens (1971) in statistics and Sorenson and Alspach (1971) in engineering; the latter using the term Gaussian sum for direct analytic approximations to non-linear models; see also Alspach and Sorenson (1972) and Harrison and Stevens (1976). Our method here is a direct extension of the original Gaussian sum approximation idea now embedded in the Markov chain Monte Carlo framework. The approach builds on the concept of using mixtures of many precise normal components to approximate sequences of posterior distributions for sets of states as the conditioning data is updated; in essence, this revisits and revises earlier adaptive importance sampling approaches (West, 1992, 1993a,b) to be based on far more efficient – computationally and statistically – Metropolis accept/reject methods.

By way of notation we denote a normal mixture distribution for a random variate \( z \) by

\[ z \sim Nm(p_{i,j}, m_{i,j}, C_{i,j}) \text{ when } p(z) = \sum_{j=1}^{J} p_{j} N(m_{j}, C_{j}) \]

where \( N(\mu, \sigma^2) \) indicates a Gaussian distribution with mean \( \mu \) and standard deviation \( \sigma \).

3.2 Mixtures in State-Space Models

Suppose at time \( t-1 \) the density \( p(x_{t-1} | y_{1:t-1}) \) is – either exactly or approximately – given by

\[ (x_{t-1} | y_{1:t-1}) \sim Nm(p_{t-1,i,j}, m_{t-1,i,j}, C_{t-1,i,j}). \]

Then the key trivariate density of equation (4) is

\[ p(y_t, x_t, x_{t-1} | y_{1:t-1}) = \sum_{j=1}^{J} p_{t-1,j} p(y_t | x_t) p(x_t | x_{t-1}) N(x_{t-1} | m_{t-1,j}, C_{t-1,j}). \]

(7)
Suppose that component variances $C_{t-1,j}$ are very small relative to the variances $V_t$, $W_t$ and inversely related to the local gradients of the regression and evolution functions $f(\cdot)$, $g(\cdot)$ in equation (1); this generally requires a large value of $J$. Then variation of component $j$ of the summand in equation (7) is heavily restricted to the implied, small region around $x_t = m_{t-1,j}$ and we can accurately approximate $g(\cdot)$ and $f(\cdot)$ with local linearization valid in that small region. The two lead terms in summand $j$ are replaced by the local normal, linear forms

$$a_{t,j} = g_t(m_{t-1,j})$$

$$f_{t,j} = f_t(a_{t,j})$$

This immediately reduces equation (7) to a mixture of trivariate normals, so that all marginals and conditionals are computable as normal mixtures. In particular, the key distributions for filtering and smoothing are:

The equation (5) for forward filtering:

$$x_t|y_{1:t} \sim N(p_{t,j}, m_{t,j}, C_{t,j})$$

having elements $m_{t,j} = a_{t,j} + A_{t,j}f_t(y_t - f_t)$ and $C_{t,j} = R_{t,j} - A_{t,j}^2Q_{t,j}$ where

$$A_{t,j} = R_{t,j}f_t'(a_{t,j})/Q_{t,j}$$

$$R_{t,j} = C_{t-1,j} - A_{t,j}' C_{t-1,j} A_{t,j} + W_t$$

and

$$Q_{t,j} = R_{t,j}f_t'(a_{t,j})^2 + V_t$$

The component probabilities are updated via $p_{t,j} \propto p_{t-1,j}N(y_t|f_t, Q_{t,j})$.

The equation (6) for backward sampling:

$$x_{t-1}|x_t, y_{1:t-1} \sim N(m_{t-1,j}, h_{t-1,j}, H_{t-1,j})$$

having elements $h_{t,j} = m_{t-1,j} + B_{t,j}x_t - a_{t,j}$ and $H_{t,j} = C_{t-1,j} - B_{t,j}'R_{t,j}B_{t,j}$ where

$$B_{t,j} = C_{t-1,j}g_t'(m_{t-1,j})/R_{t,j}$$

and

$$R_{t,j} = g_t'(a_{t,j})^2 + V_t$$

The component probabilities are $q_{t,j} \propto p_{t-1,j}N(x_t|a_{t,j}, R_{t,j})$.

For large $J$ and small enough $C_{t-1,j}$, the implied filtering computations will provide good approximations to the true model analysis, and have been used quite widely in applications in engineering and elsewhere for some years. Smoothing computations based on the approximations are direct, but have been less widely used and exploited to date. Our strategy is to embed these mixture computations in an overall MCMC, using them to define a global Metropolis proposal distribution for $p(x_0:T|y_1:T)$. As a nice by-product, the observed Metropolis acceptance rates also provide an indirect assessment of the adequacy of the Gaussian method as a direct analytic approximation, though our interest is its use in obtaining exact posterior samples.

### 3.3 Two-State Example

To clearly fix ideas, Figure 1 shows aspects of an example with $g_t(x) = 0.1x^3 + \sin(5x)$, $W_t = 0.2$ and in which $(x_{t-1}|y_{1:t-1})$ is a $J = 50$ component mixture with resulting density graphed in the figure. The comparison between the bivariate contours of the exact and mixture approximation of $p(x_t, x_{t-1}|y_{1:t-1})$ demonstrates the efficacy of the method in this highly non-linear model. Evidently, the mixture model is an accurate representation of the true non-linear model, though the joint mixture density is in fact very slightly more diffuse – a good attribute from the viewpoint of the goal of generating a useful Metropolis proposal distribution.

Approximation accuracy increases with the number of normal mixture components used so long as the variances $C_{t-1,j}$ fall off appropriately as $J$ increases, as we discuss further below.
3.4 Regenerating Mixtures

The basic mixture approximation strategy can work well when the component means are spread out, the component variances are small, and the component probabilities are approximately equal. The component means being spread out leads to desirable wide-ranging evaluations of the non-linear evolution and observation functions. Very small component variances improve the validity of the local linear approximations to the non-linear functions over increasingly small regions. Balanced component probabilities ensures that each component contributes to the mixture after propagation through the non-linearities; if only a few components dominate, all other components are effectively irrelevant and the overall strategy will collapse.

These properties are explicitly maintained using a novel regenerating procedure shown in Table 1. Suppose we wish to approximate an arbitrary density \( p(x) \) using an equally-weighted mixture of Gaussians with means set at the \( J+1 \)-quantiles of \( p(x) \) and component variances constant and chosen so that the variance of the mixture equals that of \( p(x) \). For large \( J \), this satisfies the above desiderata for mixture approximations in our context, and this idea is used to map any given mixture distribution to one with any number of components with these characteristics.

In our model of equation (1), suppose an initial prior \( p(x_0) \) is defined as a mixture, either directly or by applying the regeneration procedure in Table 1 to an original prior and using a large value \( J \). We proceed through the sequential updating analysis, now at each step using the regenerating procedure when necessary to revise, balance and hence improve the overall adequacy of the approximation at each stage. Depending on the model and data, this regeneration may be needed to approximate the prior \( p(x_{t+1:1}, y_{1:t-1}) \) and posterior \( p(x_t | y_{1:t}) \) at each step. Although equation (7) is not satisfied in these cases, equations (8) and (9) are still relevant and therefore a proposal can be drawn and the Metropolis acceptance probability computed.

4 Metropolis MCMC

4.1 Adaptive Mixture Model Metropolis for States

The mixture modelling strategy defines a computationally feasible method for evaluating and sampling from a useful approximation to the full joint posterior density of states of equation (2) and, in reverse form, equation (3). Forward filtering computations apply to sequentially update the mixture forms \( p(x_t | y_{1:t}) \) over \( t = 1:T \) using equation (8) and the regenerating procedure. This is followed by backward sampling over \( t = T, T-1, \ldots, 0 \) using the mixture forms of equation (9). Write \( q(x_{0:T} | y_{1:T}) \) for the implied joint density of states from this analysis; that is, \( q(\cdot | y_{1:T}) \) has the form of the reverse equation (3) in which each \( p(\cdot | \cdot) \) is replaced by the corresponding mixture density.

We treat the analysis via Metropolis-Hasting MCMC analysis. With a current sample of states \( x_{0:T} \), apply the FFBS to generate a new, candidate draw \( x_{0:T}^* \) from the proposal distribution with density \( q(x_{0:T}^* | y_{1:T}) \). This is assessed via the standard accept/reject test, accepting with probability

\[
\rho(x_{0:T}^* | x_{0:T}) = \min \left[ 1, \frac{w(x_{0:T}^*)}{w(x_{0:T})} \right]
\]

where \( w(\cdot) = p(\cdot | y_{1:T})/q(\cdot | y_{1:T}) \), \( p(\cdot | y_{1:T}) \) is given by equation (1), and \( q(x_{0:T}^* | y_{1:T}) \) is given by equation (9). With no unknown fixed parameters, \( q(x_{0:T} | y_{1:T}) \) is known from the previous MCMC iteration, otherwise the densities to evaluate \( q(x_{0:T} | y_{1:T}) \) will need to be calculated by repeating equation (9) for \( x_{0:T} \).
This is a global MCMC, applying to the full set of consecutive states, that will generally define an ergodic Markov chain on the \( x_{0:T} \) based on everywhere-positivity of both \( p(x_{0:T} | y_{1:T}) \) and \( q(x_{0:T} | y_{1:T}) \). As \( q \) is expected to provide a good global approximation, the resulting MCMC can be expected to perform well, and as mentioned above the acceptance rates provide some indication of the accuracy of the approximation. Evidently, acceptance rates can generally be expected to decay with increasing time series length \( T \). Experiences of Ravines et al. (2007) in the simpler DGLM context, of our own group in this and related model contexts, bear out the utility of the method. Some additional comments and numerical comparisons of acceptance rates appear below. The overall procedure for latent state sampling is termed *adaptive mixture modelling Metropolis method* (AM4) and is provided in Table 2. In this algorithm, regenerate refers to the procedure in Table 1. Depending on the model and prior, these regeneration procedures may be unnecessary.

### 4.2 Combined MCMC for States and Fixed Model Parameters

Practical applications involve models with fixed, uncertain parameters as well as the latent states, and a complete analysis embeds the above simulator for states within an overall MCMC that also includes parameters (e.g., see West and Harrison, 1997, section 15.2). With a vector of parameters \( \theta \), extend the model notation to

\[
y_t = f_t(x_t | \theta) + \nu_t, \quad x_t = g_t(x_{t-1} | \theta) + \omega_t,
\]

where, now, the initial prior \( p(x_0 | \theta) \) may involve elements of \( \theta \) as may the variances \( \nu_t, W_t \) (one key case being constant, unknown variances that are then elements of \( \theta \)).

The overall computational strategy is then to apply the above state sampler at each stage of an overall MCMC conditional on \( \theta \), and to couple this with sampling of \( \theta \) values using the implied distribution \( p(\theta | x_{0:T}, y_{1:T}) \) at each step of the chain. Since \( \theta \) is changing at each step in the MCMC, the filtered distributions, i.e. component probabilities, means, and variances in equation (8), are recomputed at each iteration of the MCMC for joint sampling of \( x_{0:T} \).

Depending on the model form and priors specified for \( \theta \), sampling fixed parameters will typically be performed in terms of a series of Gibbs sampling steps, perhaps with some blocking of subsets of parameters. Under a specified prior \( p(\theta) \), the complete conditional posterior for any subset of elements \( \theta_i \) given the remaining elements \( \theta_{-i} \) is

\[
p(\theta | \theta_{-i}, x_{0:T}, y_{1:T}) \propto p(\theta)p(x_0 | \theta) \prod_{t=1}^{T} p(y_t | x_t, \theta) p(x_t | x_{t-1}, \theta).
\]  

(11)

Sometimes this conditional can be sampled directly; a key example is when \( V = V_t \), \( W = W_t \) and \( \theta_i = (V, W) \) when, under independent inverse gamma priors, the above conditional posterior is also the product of independent inverse gammas. In other cases, resampling some elements \( \theta_i \) will use random-walk Metropolis-Hastings methods involving an accept/reject test to resample \( \theta_i \), i.e., a standard Metropolis-within-Gibbs series of moves. So long as the prior density \( p(\theta) \) that can be directly and easily evaluated up to a constant, such moves are easy to implement since the terms in equation (11) can be trivially evaluated at any point \( \theta \). Our example in Section 5.2 illustrates this overall strategy.
5 Examples

5.1 Illuminating Example

Consider the example model with

\[ y_t = \frac{x_t^2}{20} + \epsilon_t \]
\[ x_t = \frac{x_{t-1}}{2} + 25x_{t-1}/(1 + x_{t-1}^2) + \cos(1.2\pi) + \epsilon_t \]

(12)

where \( V_t = V = 10, W_t = W = 1 \) and, initially, \( x_0 \sim N(0, 10) \). This model was originally introduced by Andrade Netto et al. (1978) and has since been studied by Kitagawa (1987); West (1993b); Gordon et al. (1993); Hürzeler and Künsch (1998); Doucet et al. (2000). Interest in this model has stemmed from the non-linear nature found in both the observation and evolution equations. In addition, the squared term in the observation equation introduces a bimodal likelihood for \( x_t \) whenever \( y_t > 0 \). As will be seen, this causes multi-modality in the resulting smoothed distribution for the states.

Simulated states and observations with \( T = 100 \) appear in Figure 2. Using \( J = 1,000 \) and regenerating \( p(x_t|x_{t-1}, y_{1:t-1}) \) and \( p(x_t|y_{1:t}) \) at each \( t \), the resulting filtering distributions are shown for selected time points in Figure 3. This figure shows filtering densities that display markedly non-Gaussian behavior, the bimodality being induced by the lack of identification of the sign of \( x_t \) from the data alone.

AM4 applied to the 101-dimensional set of states \( x_0:T \) generated 100,000 samples following 100,000 burn-in steps, achieving acceptance rates of around 20%. Figure 4 shows histograms of the sampled states at the same time points in Figure 3, again evidence of the high degree of non-Gaussianity.

Figure 5 compares AM4 to the standard MCMC method of Carlin et al. (1992), where univariate states are sampled, and Kitagawa (1987), which evaluates the smoothed density on a grid. For state \( x_{58} \) all three methods achieve equivalent results, but for state \( x_{72} \) the Carlin et al. method never visits the left-most mode of the distribution. Therefore the autocorrelations for Carlin et al. are misleadingly low.

More interesting are the bivariate smoothing distributions for \( (x_t, x_{t-1}|y_{1:T}) \) for each \( t \), obtained simply from marginal samples from the MCMC. Figure 6 shows a scatterplot of the posterior samples for \( x_{71} \) versus \( x_{72} \), while Figure 7 shows a smoothed density estimate for this same data. These two figures display a high degree of non-Gaussian behavior and the reconstructions are simply not obtainable under standard linearization methods or easily, if at all, via other numerical approaches. Standard MCMC methods such as sampling univariate states or a multivariate random walk have difficulty escaping the modes shown in Figure 7. Particle filtering approaches such as Godsill et al. (2004) can produce results with similar multimodal posteriors. But, for models with unknown fixed parameters such as those to follow, these methods quickly suffer from the added dimensionality.

5.2 Example Motivated by Pathway Studies in Systems Biology

Discrete-time models have been gaining popularity for modelling biochemical pathways. Initial uses were aimed at approximate parameter inference of stochastic differential equations (Golightly and Wilkinson, 2005, 2006b,a; Wilkinson, 2006). More recently discrete-time models have been suggested as alternatives to ordinary and stochastic differential equations (Gadkar et al., 2005a,b; He et al., 2007, 2008). Here we analyze a discrete-time model of a biological system. Derivation of this model can be found in Niemi (2009).
Consider a biological system that has two proteins of interest: an activator and a target. Two molecules of the activator and two molecules of the target can combine to form a tetramer. This tetramer can then enhance the production of the target. This type of system is very common in human cells, and particularly in gene pathways that control cell developmental processes that play key roles in cancer when deregulated (Sears et al., 1997; Nevins, 1998; Bild et al., 2006). Activators themselves often have an oscillating pattern – related to progression through cell cycles. As a synthetic example that mirrors this structure, suppose that we obtain noisy measurements of the activator and target proteins

\[
y_i = \left( \begin{array}{c} a_i \\ x_i \end{array} \right) + \omega_i
\]

where \( i \in \{ \text{on}, \text{off} \} \), \( y_i \sim N(0, \sigma^2_m I) \), \( \omega_i \sim N(0, \text{diag}(\sigma^2_a, \sigma^2_x)) \) again independent and mutually independent, and \( p(x_0 | D_0) = N(1.5, 0.5) \); here \( a_t \) and \( x_t \) indicate mean fluorescence levels of the activator and target proteins, respectively, at time \( t \). The tetramer binding of these proteins is represented through the \( a^2 x^2 \) term since the exponents are determined from the number of molecules of each component. To recreate the oscillating pattern seen in activators, experimental conditions are controlled such that the activator can be modeled as an autoregressive process with two distinct means, \( \mu_{\text{on}} \) and \( \mu_{\text{off}} \), where the state \( i_t \) is known at all time points. Figure 8 provides a pictorial representation of the experimental setup and synthetic gene circuit.

Typically, the evolution equation is addressed separately through their production and decay functions. Consider first the evolution equation for the target protein, \( \psi \) indicates that from one time point to the next 100(1 − \( \psi \))% of the protein decays on average; decay is linear in \( x \) and independent of the activator. On the other hand, the production function, \( (k + a x^2)/(\beta + a^2 x^2) \), has a logistic form in \( x \) and also varies depending on the activator. This logistic form implies a minimal level of target protein production to be \( k/\beta \) and a maximal amount to be \( \alpha \). Figure 9 shows examples for high, medium, and low levels of the activator. The logistic nature of the activator’s evolution equation is a simple reparameterization of constitutive production, \( \mu_i (1 − \phi) \), and decay, \( (1 − \phi) \), linear in \( a \). This parameterization has the interpretation that \( \mu_i \) is the steady-state mean of the activator under the on and off experimental conditions which can be accurately estimated in steady-state experiments.

Of particular interest in these systems is the true level of the target since it may affect genes downstream in the overall biological pathway. In order to provide accurate estimates of the target protein, we need to account for uncertainty present in fixed parameters as well as the activator level. The analysis performed through MCMC is decomposed into Gibbs and Metropolis steps. The steps are all univariate with the exception of the draws for the latent states of the activator \( a_{0:T} \) and the target \( x_{0:T} \). With appropriate priors, most fixed parameters are available as Gibbs steps. The full conditional for \( \beta \) is unavailable and hence we use a random walk Metropolis. The joint draw for \( a_{0:T} \) is available through the standard FFBS augmented with a Metropolis-Hastings step to account for the target’s evolution equation. The diagonal evolution error structure allows for sampling \( x_{0:T} \) through AM4.

\[
y_i = \left( \begin{array}{c} a_i \\ x_i \end{array} \right) + \omega_i
\]
Informative priors are used for all fixed parameters either to truncate the parameters to reasonable regions or to provide information on plausible biological knowledge. For example, parameters in the evolution of the target protein are products of chemical kinetic reaction parameters. These parameters are all positive and therefore their products are also positive. Truly informative priors are provided for some parameters including steady-state means of the activator in the on and off states, which could be measured with accuracy in steady-state experiments.

The full MCMC analysis was performed using $J = 10$ mixture components and regeneration of both the prior and posterior at each time step and saving 50,000 iterations after 5,000 burn-in. The Metropolis steps achieved acceptance probabilities of 34%, 46%, and 8% for $\beta$, $x_0$, and $a_0$ respectively. Figure 10 provides posterior marginal histograms for fixed model parameters as well as their priors and true values. Figure 11 provides posterior median and pointwise 95% credible intervals for the activator and target protein. In more realistic scenarios, many of the proteins of interest may have no observations. In these situations, the methodology will work, but generally more information will need to be provided through the priors for meaningful inferences.

### 6 Further Discussion

The adaptive mixture modelling Metropolis method developed and exemplified here represents an efficient, effective, and relatively easily implemented computational strategy for Bayesian inference in non-linear state-space models. For implementation, the method as presented requires only availability of first-order derivatives of the evolution and observation equation similar to the extended Kalman filter, coupled with simulation routines. Extensions to non-Gaussian models require the first two derivatives of the log-likelihood. The overall approach represents a nice update on the use of Gaussian mixtures as direct analytic approximations and in a real sense a completion of a line of computational development for state-space models that stretches back nearly forty years. Many old ideas are good ideas, and the simple strategy of “Metropolizing” a global, analytic approximation to the full posterior distribution of a set of states adds a modern computational touch to a good, older idea. Critically, however, the mixture regeneration concept and strategy introduced here is simply fundamental to practical utility, as otherwise mixture approximations can and often will degenerate as discussed. The broader utility of the overall approach is clear once the state sampler is embedded in a larger MCMC that couples in fixed parameter sampling, as illustrated in our examples.

We have compared this method to other standard methods for analysis of state-space models including univariate state sampling and a multivariate random walk. In the example of Section 5.1, the Carlin et al. method failed to sufficiently explore the posterior over the set of states due to getting caught in modes and similar results are obtained with random walk proposals. Even after adjusting for computational time, AM4 outperformed these standard methods. AM4 is a widely applicable and effective methodology for MCMC analysis of state-space models.

It is worth noting that there are various other uses of adaptive mixture approximations in time series and state-space models (West and Harrison, 1997). One related line of development to note is that of Chen and Liu (2000), who use mixtures of Gaussians to approximate the filtering densities for each $x_t$. They then proceed with a particulate, sequential Monte Carlo approach: conditioning on state particles to induce a linearized system and adjusting the weights to create the filtered densities for $x_{t+1}$. That approach adds particle filtering uncertainties to the problem. The approach is closely allied to the earlier sequential adaptive importance sampling approach of West (1992, 1993a,b), though replaces the uses of iteratively refined mixture approximations to posteriors for states at each time point with particle filtering. Though based on mixtures, the computational focus and method of Chen and Liu (2000) is clearly very different to our work;
again, we generate an accurate, global approximation to the full set of posteriors over states using deterministic mixture approximations, and use it as a global proposal for a Metropolis step within an MCMC framework. This accomplishes fully Bayesian inference for filtering and smoothing analyses in an over-arching framework that includes fixed parameters as well as dynamic states.

On matters of statistical and computational efficiency, we are encouraged by the high acceptance rates in the examples here, and a range of other studies with similar models. A nice by-product of the Metropolis strategy is that the acceptance rates can, in any specific application, be viewed as benchmarks on the inherent accuracy of the underlying analytic mixture-based posterior approximation. The tradeoffs, in terms of statistical versus computational efficiencies, relate to the number of mixture components chosen and the length of the observed time series, among other things. As noted, filtering approximations with mixtures can be made arbitrarily accurate, theoretically, by increasing the number of mixture components, at the cost of cpu time increasing linearly with the number of components. Also, as is true much more widely, Metropolis acceptance rates will decrease roughly linearly on a log scale with the time series sample size; while this generic issue can be addressed with longer MCMC run lengths and sub-sampling – a standard response to decreased acceptance rates – modifying the underlying proposal distribution by simply increasing the number of mixture components helps too. Tables 3 and 4 provide some insights into this, with a set of empirical estimates of acceptance rates and cpu times in a simple simulated model context. Extensions and novel strategies are clearly needed to extend analyses to much longer time series while maintaining acceptance rates at practically useful levels, and some of our current research is exploring some new directions for this.

**Code**

The web page http://ftp.stat.duke.edu/WorkingPapers/08-21.html provides freely available Matlab code that implements the method described here. This includes support functions and the examples from this paper as templates for other more general models. (The version of this code used to produce this article is available on the JCGS webpage, see Supplemental Materials Section for details.)

**Supplementary Material**

Refer to Web version on PubMed Central for supplementary material.

**Acknowledgments**

The authors thank the editor and two anonymous reviewers for their constructive suggestions that helped improve this work. We are grateful to Lingchong You and Chee-Meng Tan for discussion of dynamic models in systems biology. We acknowledge support of the National Science Foundation (grants DMS-0342172 and BES-0625213) and the National Institutes of Health (grants P50-GM081883-01 and NCI U54-CA-112952-01). Any opinions, findings and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the NSF or NIH.

**References**


Figure 1.
Bivariate and marginal distributions for two-states in a model with $g_t(x) = 0.1x^3 + \sin(5x)$ and $W_t = 0.2$, and where $p(x_{t-1}|y_{1:t-1})$ is a $J = 50$ component mixture with density graphed on the horizontal axis of each frame. The left pane shows exact bivariate density contours and, on the vertical axis, the implied margin for $(x_t|y_{1:t-1})$. The right pane shows the corresponding 50-component mixture approximations. In each frame, the dashed line shows the evolution function $g_t(\cdot)$. (A color version of this figure is available in the electronic version of this article.)
Figure 2.
Model of equation (12): Simulated latent states and observations.
Figure 3.
Model of equation (12): Filtering densities for selected time points that display markedly non-Gaussian behavior.
Figure 4.
Model of equation (12): Histograms of the smoothed samples for the same time points shown in Figure 3.
Figure 5.
Model of equation (12): Traceplots, autocorrelation functions, and histograms for two different states comparing AM4 (‘o’), Carlin et al. (1992) (‘+’), and Kitagawa (1987) (‘-’). (A color version of this figure is available in the electronic version of this article.)
Figure 6.
Model of equation (12): Scatterplot of MCMC samples for $x_{71,72}$. 
Figure 7.
Model of equation (12): Reconstruction of bivariate density $p(x_{71}, x_{72} | y_{1:100})$. 
Figure 8.
Model of equation (13): Depiction of a tetramer experiment. A. Experimental setup with switched glass burets containing chemical solutions to control the on-off state of the activator. The desired solution flows into a channeled microscopy slide. Bacteria containing plasmids with a synthetic gene circuit are adhered to the gray area of the channel. B. A synthetic gene circuit containing three chemical species: an activator, a target, and a tetramer, $A_2X_2$. Arrows indicate that the activator and target are produced, degraded, and can form tetramers. Tetramers can decay back to the activator and target or can enhance the production of the target.
Figure 9.
Model of equation (13): Production and decay functions for various levels of the activator. (A color version of this figure is available in the electronic version of this article.)
Figure 10.
Model of equation (13): Histograms of marginal posterior estimates (shaded histogram), prior (superimposed curve), and true value (‘x’) for fixed model parameters. (A color version of this figure is available in the electronic version of this article.)
Figure 11.
Model of equation (13): Pointwise median (solid line) and 95% credible interval (dashed line) results for the underlying state (dots) and the observed data (circles). The activator is shown in the top pane and the target protein in the bottom. (A color version of this figure is available in the electronic version of this article.)
Table 1

Regenerating procedure to approximate an arbitrary density $p(x)$ with a mixture of Gaussians $Nm(p_{1:j}, m_{1:j}, C_{1:j})$.

1. Set $p_j = 1/J$ for $j \in \{1, \ldots, J\}$.
2. Set $m_j$ equal to the $j/J + 1$ quantile of $p(x)$ for $j \in \{1, \ldots, J\}$. That is
   $$\frac{j}{J+1} = \int_{m_j}^{\infty} p(x)dx.$$
3. Set $C_j = C$ such that
   $$\text{Var} X = \sum_{j=1}^{J} p_j [C + (m_j - \bar{m})^2]$$
   where $\bar{m} = \sum_{j=1}^{J} p_j m_j$. 
Table 2

AM4 algorithm to sample from the full posterior of states given in (2) for the model of equation (1).

1. Regenerate \( p(x_0) \) to obtain \( p(x_0) \approx \text{Nm}(p_{0,1}, C_{0,1}) \).

2. For \( t = 1:T \), perform the forward filtering steps,
   a. For all \( j \), set
      \[
      \begin{align*}
      a_t, j &= \tilde{a}_t, j = g_t(m_{t-1}, j), \\
      R_t, j &= C_{t-1, j} g_t'(m_{t-1}, j)^2 + W_t, \\
      \tilde{p}_t, j &= \tilde{p}_{t-1, j}.
      \end{align*}
      \]
   b. Regenerate \( \text{Nm}(\tilde{a}_t, j, R_t, j) \) to obtain
      \[
      p(x_t \mid x_{t-1}, y_{1:(t-1)}) \approx \text{Nm}(\tilde{p}_{t-1,1:J, \tilde{a}_t, j, R_t, j}).
      \]
   c. For all \( j \), set
      \[
      \begin{align*}
      f_t, j &= f_t(a_t, j), \\
      Q_t, j &= R_t, j f_t'(a_t, j)^2 + V_t, \\
      m_t, j &= a_t, j + A_t, j(y_t - f_t, j), \\
      C_t, j &= R_t, j - A_t, j Q_t, j, \\
      \tilde{p}_t, j &= \tilde{p}_{t-1, j} \text{Nm}(y_t \mid f_t, j, Q_t, j).
      \end{align*}
      \]
      where \( A_{t,j} = R_{t,j} f_t(a_{t,j})/Q_{t,j} \)
   d. Regenerate \( \text{Nm}(\tilde{p}_{t,1:J, \tilde{m}_t, j, \tilde{C}_t, j}) \) to obtain
      \[
      p(x_t \mid y_{1:t}) \approx \text{Nm}(p_{t,1:J, \tilde{m}_t, j, \tilde{C}_t, j}).
      \]

3. Draw \( x^*_t \sim \text{Nm}(p_{T,1:J, m_{T,1:J}, C_{T,1:J}}) \).

4. For \( t = (T-1):0 \), perform the backward sampling draws,
   a. For all \( j \), set
      \[
      \begin{align*}
      h_t, j &= m_{t-1, j} + B_t, j(x_t - \tilde{a}_t, j), \\
      H_t, j &= C_{t-1, j} - B_t, j R_t, j, \\
      q_t, j &= \tilde{p}_{t-1, j} H(x_t \mid \tilde{a}_t, j, R_t, j).
      \end{align*}
      \]
      where \( B_{t,j} = C_{t-1,j} g_t(m_{t-1,j})/R_{t,j} \)
   b. Draw \( x^*_t \sim \text{Nm}(q_{t,1:J, h_{t,1:J}, H_{t,1:J}}) \).

5. Accept \( x^*_0 \) with probability \( p(x^*_0 \mid x_{0:T}) \) given in equation (10).
Table 3
Empirical mean (sd) % acceptance rates from 100 simulations of the model defined by: $f_t(x) = x$, $g_t(x) = \sin(x)$, $V_t = 1$, $W_t = 1$ and $x_0 \sim N(0, 10)$.

<table>
<thead>
<tr>
<th>Number of Components</th>
<th>Length of time series</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>0 (4)</td>
</tr>
<tr>
<td>5</td>
<td>62 (8)</td>
</tr>
<tr>
<td>10</td>
<td>66 (7)</td>
</tr>
<tr>
<td>50</td>
<td>74 (6)</td>
</tr>
<tr>
<td>100</td>
<td>79 (12)</td>
</tr>
</tbody>
</table>
Table 4
Relative empirical mean (sd) computation time in minutes for 10,000 iterations using the model in Table 3. Analysis was performed using Matlab R2007b on a 3.4 Ghz Intel Pentium 4 processor.

<table>
<thead>
<tr>
<th>Number of Components</th>
<th>Length of time series</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>0.4 (.0)</td>
</tr>
<tr>
<td>5</td>
<td>0.6 (.1)</td>
</tr>
<tr>
<td>10</td>
<td>0.6 (.1)</td>
</tr>
<tr>
<td>50</td>
<td>0.8 (.1)</td>
</tr>
<tr>
<td>100</td>
<td>1.1 (.2)</td>
</tr>
</tbody>
</table>