Bayesian Analysis (2008) **3**, Number 4, p. 659-688

EM versus Markov chain Monte Carlo for Estimation of Hidden Markov Models: A Computational Perspective

Tobias Rydén*

Abstract. Hidden Markov models (HMMs) and related models have become standard in statistics during the last 15–20 years, with applications in diverse areas like speech and other statistical signal processing, hydrology, financial statistics and econometrics, bioinformatics etc. Inference in HMMs is traditionally often carried out using the EM algorithm, but examples of Bayesian estimation, in general implemented through Markov chain Monte Carlo (MCMC) sampling are also frequent in the HMM literature. The purpose of this paper is to compare the EM and MCMC approaches in three cases of different complexity; the examples include model order selection, continuous-time HMMs and variants of HMMs in which the observed data depends on many hidden variables in an overlapping fashion. All these examples in some way or another originate from real-data applications. Neither EM nor MCMC analysis of HMMs is a black-box methodology without need for user-interaction, and we will illustrate some of the problems, like poor mixing and long computation times, one may expect to encounter.

Keywords: hidden Markov model, incomplete data, missing data, EM, Markov chain Monte Carlo, trans-dimensional Monte Carlo, computational statistics

1 Introduction

Hidden Markov models are a class of statistical models that today have become standard in applied statistics, with applications in areas like speech processing (Levinson et al. 1983; Jelinek 1998), bioinformatics (Koski 2001), econometrics (Raj 2002), finance (Bhar and Hamori 2004), and many more. More general references to the subject include MacDonald and Zucchini (1997), Cappé et al. (2005) and Frühwirth-Schnatter (2006).

By a hidden Markov model we mean a discrete-time bivariate process $\{(X_k, Y_k)\}$ possessing the following properties: (i) the X-process is a finite-state Markov chain; (ii) the Y-variables are conditionally independent given all X-variables; (iii) given all X-variables, the conditional distribution of Y_k depends on X_k but not on any other X-variables; (iv) the X-process is non-observable (latent). Before proceeding we make a few remarks about this definition. Conditions (ii) and (iii) together stipulate a local dependence between the X- and Y-variables. This property is exactly the same as in a state-space model, and it is completely accurate to say that an HMM is a state-space model with finite state space. The conditional distributions of Y_k for various values

*Centre for Mathematical Sciences, Lund University, Lund, Sweden, mailto:tobias.ryden@matstat.lu.se

 DOI:10.1214/08-BA326

of X_k in the state space are typically taken from a common family of distributions, for instance Normal distributions with different means and/or variances, so that the marginal (unconditional) distribution of Y_k is over-dispersed relative to a single distribution in that family. This mechanism is the same as for mixture distributions, but in most contexts where the term 'mixture distribution' is used, the pairs (X_k, Y_k) are i.i.d. In an HMM they are not, so that an HMM can be viewed as a mixture model with dependence. We remark that the Y-process is in general not Markov however.

Many variants and extensions of the above definition are found in the literature, and we will also consider one in the present paper. A particularly common extension is when the conditional distribution of Y_k is allowed to depend, in addition to X_k, also on some lagged Y-variables $Y_{k-1}, Y_{k-2}, \ldots, Y_{k-r}$. Such models are often referred to as *Markov-switching autoregressions*, or *autoregressions with Markov regime* (see e.g. Krolzig 1997).

The first papers on HMMs appeared in the second half of the 1960's, authored by Leonard Baum, Ted Petrie and co-workers, and mostly dealing with the case when the output variables Y_k take values in a finite set as well. Results established in these early papers include consistency and asymptotic normality of the maximum-likelihood estimator (MLE) (Baum and Petrie 1966; Petrie 1969), and in particular a version of the EM algorithm (Baum et al. 1970), formulated for the particular case of HMMs before Dempster et al. (1977) coined the term 'EM' in general. Since then the MLE and the EM algorithm have been the main vehicles for inference in HMMs. The popularity of EM is explained by an efficient computational tool known as the *forward-backward algorithm*, that is used to implement EM for HMMs. Recent years have seen an increased interest in Bayesian inference in HMMs however, often implemented using Gibbs sampling, and the purpose of the present paper is indeed to discuss this approach, its strengths and weaknesses, and how it compares to frequentist approaches. At this point it should be stressed that EM is not *per se* a tool for frequentist (ML) inference, but a framework that can equally well be used for computing maximum aposteriori (MAP) estimates in Bayesian settings (Dempster et al. 1977, p. 6). Thus EM is best described as a *method for computing a point estimate*. The comparison we will make is hence EM vs. Gibbs, with bootstrap complementing EM for interval estimation and model selection, and Gibbs sampling possibly being replaced by more general MCMC samplers for model selection and more complex models. Our approach to this comparison will mainly be computational, and not on larger differences between the frequentist and Bayesian paradigms. Thus our focus is on *how*, in a computational sense, different models are analysed using the two approaches and what kind of computational efforts are required. Put differently, our perspective is that of a statistician who has no strong beforehand preference for either approach, but is pragmatic and wants to arrive at useful results at a reasonable computational cost.

The comparison will be based on three case studies of increasing complexity and difficulty. The first one is the model with conditionally Normal distributions for Y_k, as outlined above. The second case is similar, but such that the size of the state space, i.e. the number of hidden states, is unknown and needs to be estimated. In the third case finally the hidden Markov chain evolves in continuous time, and at many time-points

this chain affects several of the Y-variables, thus creating an overlap in the dependence structure; this obviously violates (iii) in the definition above.

A paper somewhat similar to the present one is that by Scott (2002), which contains a very readable and illuminating survey of the uses of the forward-backward and related recursive algorithms in Bayesian analysis of HMMs. Scott's focus is thus mainly on the application of such computational tools in a Bayesian context and not so much on comparing frequentist and Bayesian approaches to inference; we do mention however that his paper (Section 4.1) contains a nice analysis of marginal likelihoods vs. BIC for model selection.

2 Case I: A simple hidden Markov model

We will start with a simple hidden Markov model having $d = 3$ states and the data being conditionally Normal with common conditional variance; $Y_k|X_k = i \sim \mathrm{N}(\mu_i, \sigma^2)$. The parameters of this model are thus μ_i, σ^2 and the transition probabilities of the hidden chain; the latter will be denoted by a_{ij}. We will estimate these parameters based on a simulated set of data for which the true parameters values are

$$A = \{a_{ij}\} = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix},$$

$\mu = (\mu_1, \mu_2, \mu_3) = (-2, 0, 2)$, and σ being either 0.5, 1 or 1.5. The stationary distribution of the hidden chain is (0.2, 0.6, 0.2) and the chain is assumed stationary. It is however convenient to also include the initial probabilities $\rho_i = \mathbf{P}(X_1 = i)$ as separate parameters in the model, and we do so even if these are implicitly given by the stationarity assumption and A; this is because otherwise the distribution of the complete data does not form an exponential family as the stationary probabilities are non-linear functions of the transition probabilities.

The marginal distribution of Y_k is thus a mixture of Normals, and the different cases for σ^2 correspond to different degrees of overlap between the three components (Figure 1). For $\sigma = 0.5$ the components are reasonably separated and the marginal distribution is trimodal, for $\sigma = 1$ the marginal distribution is unimodal but its shape otherwise visually reveals the presence of multiple components, whereas for $\sigma = 1.5$ the overlap is considerable and it is difficult by eye-inspection to detect the presence of multiple components. For each of the three values of σ^2, a sample $\mathbf{y}_{1:n} = (y_1, y_2, \ldots, y_n)$ of size $n = 1,000$ was simulated.

For this HMM, implementation of the EM algorithm is standard (cf. Cappé et al. 2005, Sections 10.3.1–2). We do not quote the full details here, but notice that a key step is to compute the conditional probabilities $\mathbf{P}_\theta(X_k = i \,|\, \mathbf{y}_{1:n})$ for $k = 1, 2, \ldots, n$ and $\mathbf{P}_\theta(X_{k-1} = i, X_k = j \,|\, \mathbf{y}_{1:n})$ for $k = 2, 3, \ldots, n$. Here $\mathbf{P}_\theta$ denotes probability under a certain set θ of parameters. The computation of these conditional probabilities is typically carried out using the forward-backward algorithm, which amounts to computing the joint probabilities/densities, or *forward variables*, $p_\theta(X_k = i, \mathbf{y}_{1:k})$ for

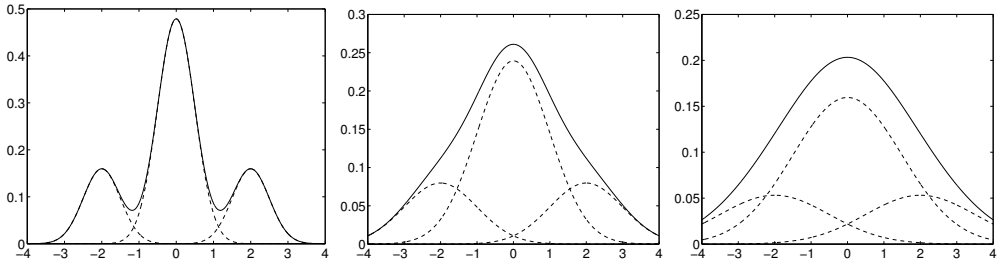

Figure 1: *Densities of the Normal components, weighted by the stationary probabilities (dashed lines) and marginal densities of the observations Y_k (solid lines) for the hidden Markov model of Case I with $\sigma = 0.5$ (left panel), $\sigma = 1$ (middle panel) and $\sigma = 1.5$ (right panel).*

$k = 1, 2, \ldots, n$ (the forward pass), and the joint conditional densities, or *backward variables*, $p_\theta(\mathbf{y}_{k+1:n}|X_k = i)$ for $k = n, n-1, \ldots, 1$ (the backward pass). In practice these forward and backward variables tend to zero or infinity exponentially fast in the recursions, whence any useful implementation applies some kind of normalisation (cf. Cappé et al. 2005, Sections 5.1.1.1–2, 3.2.2 and 3.4 for a thorough discussion on this topic). For instance one may, in each iteration, normalise the forward and backward variables to sum to one over i; in the forward pass this corresponds to computing filtered probabilities $\mathbf{P}_\theta(X_k = i \,|\, \mathbf{y}_{1:k})$.

We now turn to the Bayesian perspective. The by far most popular method for sampling from the posterior distribution in cases like this is to include the latent data $\mathbf{X}_{1:n}$ in the MCMC state space and to run the full Gibbs sampler, i.e. alternating between sampling model parameters and latent data from their respective full conditional distributions. This is because given the latent Markov chain and the data, the parameters are conditionally independent with distributions from standard parametric families (at least as long as the prior distribution is conjugate relative to the model specification) and, vice versa, given the parameters and the data the latent process is a non-homogeneous Markov chain and hence simple to sample.

In the present case the prior distribution of the parameters was taken conjugate to the complete data likelihood, as follows. Each row of the transition probability matrix as well as the initial distribution $(\rho_1, \rho_2, \ldots, \rho_d)$ were given an independent Dirichlet distribution prior $\mathrm{Dir}(1, 1, \ldots, 1)$, each μ_i was given an independent Normal prior $\mathrm{N}(\xi, \kappa^{-1})$ with $\xi = (\min y_k + \max y_k)/2$ and $\kappa = 1/R^2$ where $R = \max y_k - \min y_k$ is the data range, σ^{-2} was given a gamma prior $\Gamma(\alpha, \beta)$ with $\alpha = 2$, the hyperparameter β was given a gamma prior $\Gamma(g, h)$ with $g = 0.2$ and $h = 10/R^2$, and all parameters were assumed apriori independent. This prior specification is very much in line with what Richardson and Green (1997) did for mixture models.

Under this prior specification, the full conditional distributions are given by

$$(\rho_1, \rho_2, \ldots, \rho_d)|\ldots \sim \mathrm{Dir}(I\{X_1 = 1\} + 1, I\{X_1 = 2\} + 1, \ldots, I\{X_1 = d\} + 1) \qquad (1)$$

where $I\{\cdot\}$ denotes an indicator function,

$$(a_{i1}, a_{i2}, \ldots, a_{id})|\ldots \sim \mathrm{Dir}(n_{i1}+1, n_{i2}+1, \ldots, n_{id}+1) \tag{2}$$

where $n_{ij} = \#\{1 < k \leq n : X_{k-1} = i, X_k = j\}$ is the number of transitions from state i to j in the latent state sequence and with conditional independence across rows $i = 1, 2, \ldots, d$,

$$\mu_i|\ldots \sim \mathrm{N}\left(\frac{S_i + \kappa\xi\sigma^2}{n_i + \kappa\sigma^2}, \frac{\sigma^2}{n_i + \kappa\sigma^2}\right) \tag{3}$$

where $S_i = \sum_{k:\, X_k = i} y_k$, $n_i = \#\{1 \leq k \leq n : X_k = i\}$ is the number of visits to state i in the latent state sequence and with conditional independence across $i = 1, 2, \ldots, d$,

$$\sigma^{-2}|\ldots \sim \Gamma\left(\alpha + \frac{1}{2}n, \beta + \frac{1}{2}\sum_{k=1}^{n}(y_k - \mu_{X_k})^2\right), \tag{4}$$

and

$$\beta|\ldots \sim \Gamma(g + \alpha, h + \sigma^{-2}). \tag{5}$$

Here in all cases '...' denotes other parameters, the latent Markov chain and the data. Moreover, for the latent chain it holds that given parameters and data, this process is a non-homogeneous Markov chain with initial distribution

$$\mathbf{P}(X_1 = j \mid \ldots) \propto \rho_j\, \varphi(y_1; \mu_j, \sigma^2)\, p_\theta(\mathbf{y}_{2:n}|X_1 = j) \tag{6}$$

and transition probabilities

$$\mathbf{P}(X_k = j \mid X_{k-1} = i) \propto a_{ij}\, \varphi(y_k; \mu_j, \sigma^2)\, p_\theta(\mathbf{y}_{k+1:n}|X_k = j) \tag{7}$$

where φ is the density of a Normal distribution with the indicated mean and variance. Given that the relations are only up to proportionality in j, they need to be normalised in order to obtain the correct probabilities. We note that the densities of the partial data sequences are nothing but the backward variables. This leads to the common method *backward recursion forward sampling* for simulating the latent Markov chain conditional on the data (e.g. Chib 1996, Section 2.1). The opposite, i.e. forward recursion backward sampling, is also possible.

The full Gibbs sampler then amounts to alternating between updating the parameters conditional on the data and hidden Markov chain, and updating the hidden chain conditional on the data and parameters. In our implementation, this was done in the following order.

(a1) Update $(\mu_1, \ldots, \mu_d)$ by drawing independently from (3).

(a2) Update σ^2 by drawing from (4).

(a3) Update β by drawing from (5).

(a4) Update A by drawing $(a_{i1}, \ldots, a_{id})$ from (2), independently for $i = 1, \ldots, d$.

(a5) Update $(\rho_1, \ldots, \rho_d)$ by drawing from (1).

(b) Update $\{X_k\}_{k=1}^n$ by drawing X_1 from (6) and then X_k from (7) for $k = 2, 3, \ldots, n$.

One sequence of these steps (a) and (b) is typically referred to as a *sweep* of the Gibbs sampler.

2.1 Amount of missing information and rate of convergence/mixing

The EM algorithm was run for the model described above, for one sample of size $n = 1,000$ for each of the three different values for σ^2. The initial values of the parameters were computed as follows: (i) initial means were computed as $\mu_i = \min y_k + R/(2d) + (i-1)R/d$ with R as above, thus spreading the μ_i (uniformly) over the range of the data; (ii) an initial imputation of the x_k was computed by nearest distance, i.e. x_k was set to the argument i minimising $(y_k - \mu_i)^2$ over the initial means μ_i; (iii) an initial σ^2 was computed as $n^{-1}\sum_1^n (y_k - \mu_{x_k})^2$ for initial means and imputation; (iv) initial estimates of the a_{ij} were computed as n_{ij}/n_i where n_{ij} is the number of transitions from state i to j in the initial imputation and $n_i = \sum n_{ij}$; (v) the initial probabilities ρ_i of X_1 were set to $1/d$ for all i. Figure 2 (left) shows that the estimates of the means μ_i converge to the corresponding MLEs as EM is iterated. It also shows however, that convergence becomes slower the larger σ^2 gets. It is well known that the asymptotic rate of convergence of EM is linear, meaning that if ζ is any parameter, $\zeta^{(m)}$ its value in the m-th iteration of EM and ζ_{ML} its (unknown) ML estimate, then the error $e^{(m)} = \zeta^{(m)} - \zeta_{\mathrm{ML}}$ satisfies $e^{(m+1)} \approx ce^{(m)}$ for some $0 \le c < 1$ (e.g. Meilijson 1989, p. 132). This provided the EM sequence actually does converge to the MLE, which is not guaranteed (cf. Wu 1982). One

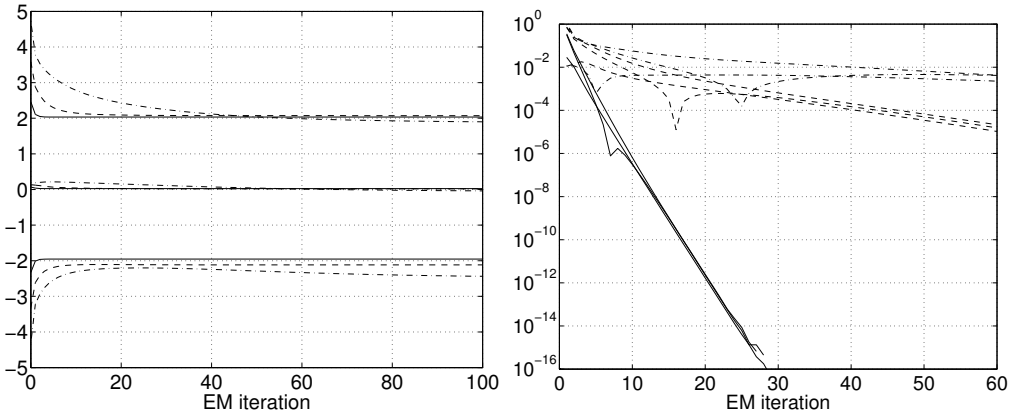

Figure 2: *Left plot: trajectories of* $\mu_i^{(m)}$ *for EM iterations* $m = 0, 1, \ldots, 100$*, component indices* $i = 1, 2, 3$ *(bottom to top curves) and data simulated with* $\sigma = 0.5$ *(solid lines),* $\sigma = 1$ *(dashed lines) and* $\sigma = 1.5$ *(dash-dotted lines). Right plot: absolute differences* $|\mu_i^{(m+1)} - \mu_i^{(m)}|$ *for EM iterations 1–60; same components and line symbols.*

should note that once the forward-backward algorithm has been run, the gradient of the log-likelihood at the present parameter estimate may be computed at little additional computational cost. This allows for switching from EM to e.g. quasi-Newtown methods in the vicinity of the MLE, hence speeding up convergence (e.g. Meilijson 1989). The cost of such approaches is the loss of EM's stability property—that an EM iteration can never decrease the likelihood.

The constant c is related to the amount of missing information in the model, and the more missing information, the larger becomes c; these heuristics can be made precise in terms of information matrices of the model (e.g. Meilijson 1989, p. 132). The asymptotics implies that $e^{(m)} \sim bc^m$ for some real b as $m \to \infty$, so that $\zeta^{(m+1)} - \zeta^{(m)} \sim c^m b(c-1)$. Plotting $\zeta^{(m+1)} - \zeta^{(m)}$ on a lin-log scale should thus produce roughly a straight line, and this is exactly what Figure 2 (right) shows. It also shows that the decay is faster when the amount of missing information is smaller ($\sigma = 0.5$), confirming the heuristics numerically. Crude estimates of c for the three cases $\sigma = 0.5$, 1 and 1.5 are 0.3, 0.89 and 0.97 respectively.

For the Gibbs sampler the parameters were initialised as for the EM algorithm, β was set to its prior mean g/h and then for each data set the sampler was run for 11,000 sweeps, of which the first 1,000 were discarded as a burn-in period. Plots of the remaining 10,000 samples of the μ_i are shown in Figure 3. It it obvious from the plots that the marginal variances of the sampled μ_i are larger for larger values of σ^2 in the data.

However, letting ζ be any model parameter, $\zeta^{[t]}$ its sampled value in the t-th sweep of the Gibbs sampler and using the empirical average $\overline{\zeta}^{[T]} = T^{-1} \sum_{t=1}^{T} \zeta^{[m]}$ as an estimate of the posterior mean of ζ, the variance of $\overline{\zeta}^{[T]}$ is not only governed by the marginal variance of $\zeta^{[t]}$ but is asymptotically equivalent to $T^{-1} \sum_{t=-\infty}^{\infty} \mathrm{Cov}(\zeta^{[t]}, \zeta^{[0]})$ for a stationary version of the Gibbs sampler. Figure 4 shows empirical autocorrelations for the samples of μ_2, and we see that the dependence across the Gibbs sweeps is much stronger for $\sigma = 1$ than for $\sigma = 0.5$. Indeed, the marginal sample variances of the sampled μ_2

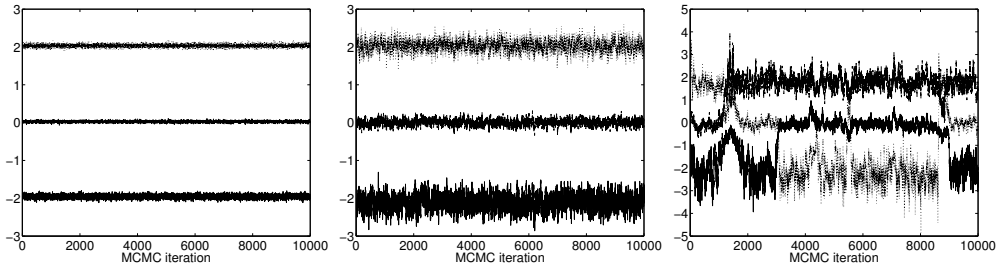

Figure 3: *Samples of μ_1, μ_2 and μ_3 in 10,000 sweeps of the full Gibbs sampler for the Normal HMM and data simulated with $\sigma = 0.5$ (left panel), $\sigma = 1$ (middle panel) and $\sigma = 1.5$ (right panel). Prior to the draws displayed here there was a burn-in of 1,000 sweeps (not shown). Note the different scales on the y-axes.*

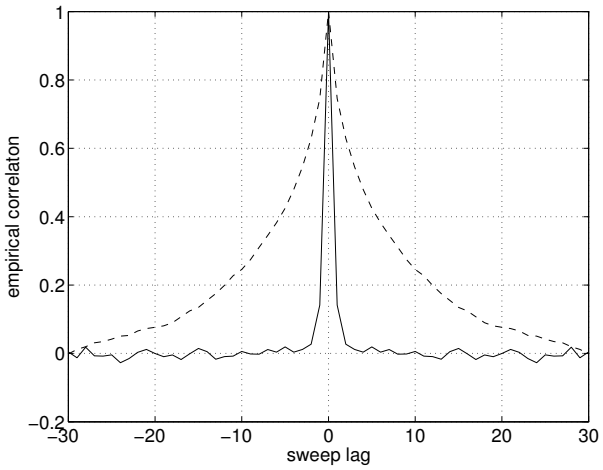

Figure 4: *Empirical autocorrelations for the sequences of sampled* μ_2 *computed from 10,000 sweeps of the full Gibbs sampler for the Gaussian HMM and data simulated with* $\sigma = 0.5$ *(solid line) and* $\sigma = 1$ *(dashed line).*

were 0.00049 and 0.0070 for $\sigma = 0.5$ and $\sigma = 1$ respectively, while crude estimates of $\sum_{t=-\infty}^{\infty} \mathrm{Cov}(\mu_2^{[t]}, \mu_2^{[0]})$ obtained by summing empirical autocovariances over the range $-30 \leq t \leq 30$ were 0.00063 and 0.093. The ratios are about 14 and 149 respectively, so that the stronger dependence causes a tenfold increase in asymptotic variance of the empirical posterior mean on top of the increase already caused by the larger marginal variance.

The reason for the higher correlation when $\sigma = 1$ is that we have a model with a *centered parametrisation* (Papaspiliopoulos et al. 2007), and when $\sigma = 1$ the complete (observed and latent) data is much more informative about the model parameters than the observed data alone. This intuition can be made precise in terms of the *Bayesian fraction of missing information* (see Papaspiliopoulos et al. 2007, Section 2.2 for details). This fraction will stay the same as n increases, so that a larger sample is no solution to the problem. A potential remedy however is to convert to a *non-centered parametrisation*, that would for instance express the Markov chain trajectory as a function of its transition probabilities and some random variables with a distribution not depending on any further parameters. A useful such parametrisation has, to the author's knowledge, not been proposed however. A different remedy is to remove the latent chain from the state space of the MCMC sampler, which would then comprise the model parameters alone. This drastically reduces the dimensionality of the sampler's state space, but also makes convenient Gibbs sampling impossible. Instead one has to turn to Metropolis-Hastings sampling using e.g. random walk proposals, and designing such moves to provide acceptable mixing of the MCMC sampler is not always easy.

2.2 Label-switching

A further phenomenon often occurring in computational analysis of HMMs is seen in Figure 3 for the data with $\sigma = 1.5$. Here the sampled μ_i exhibit what is commonly referred to as *label-switching*; in some sweeps the current ordering of the μ_i (which are initially sorted in ascending order) is changed. This phenomenon may be problematic as soon as one goes beyond computing a point estimate, but indeed also when approximating posterior means with empirical averages from an MCMC sampler. The underlying reason for label-switching is that since the prior is invariant under permutation of state indices (labels) and so is the likelihood function, the same holds for the posterior. As a consequence, for instance, all μ_i have the same marginal posterior distribution. The reason why label-switching does not appear for the data with $\sigma = 0.5$ and $\sigma = 1$ is that for these cases the posterior modes arising from permutation of state labels are far enough apart that the ordering of the μ_i is never changed when updating these parameters independently according to (3). Label-switching may be dealt with in various ways. One way is to break the permutation invariance of the prior by introducing identifiability constraints. In the present case a natural such constraint is $\mu_1 < \mu_2 < \ldots < \mu_d$, truncating the prior to the region where the constraint holds. The posterior is then also zero outside this region, so that the ordering of the μ_i is unambiguous. A problem with identifiability constraints however is that constraints on different parameters, like the variances σ_i^2 if they are taken individual for each component, typically lead to different shapings of the posterior; Frühwirth-Schnatter (2001) gives a good description of such an example in detail. In the author's view the best approach is to use a permutation invariant prior for the MCMC simulations, thus avoiding any constraints there, and to invoke possible constraints afterwards as part of the post-processing of the MCMC output. Frühwirth-Schnatter (2001) proposed to end each sweep by randomly permuting the state labels. Jasra et al. (2005) give a useful and readable overview of the label-switching problem. We remark that label-switching is a potential problem also with EM, which we discuss further below.

2.3 Interval estimation

We now turn to interval estimation. Here Gibbs sampling gives credibility intervals for free whereas EM itself only provides a point estimate. The MLE is asymptotically normal however, i.e. the weak convergence $n^{1/2}(\theta_{\mathrm{ML}} - \theta_0) \to \mathrm{N}(0, \mathcal{J}_0^{-1})$ holds as the sample size $n \to \infty$, where θ_0 denotes the true parameter and $\mathcal{J}_0$ is the limiting Fisher information matrix at this point (Bickel et al. 1998, Theorem 1). Moreover, $\mathcal{J}_0$ can be estimated by the observed Fisher information, by which we mean the negative of the Hessian of the log-likelihood, evaluated at θ_{ML} and divided by n (Bickel et al. 1998, p. 1619). This Hessian can be obtained by numerical differentiation of the log-likelihood at the MLE, but there are also ways of doing an exact computation (Lystig and Hughes 2002). Having computed the observed Fisher information, $\hat{\mathcal{J}}_0$ say, in either way, one can thus obtain a two-sided confidence interval for the i-th component θ_i of θ, with approximate degree of confidence $1 - \alpha$, as $\theta_{i,\mathrm{ML}} \pm z_{1-\alpha/2}[\hat{\mathcal{J}}_0^{-1}]_{ii}/\sqrt{n}$ where $z_{1-\alpha/2}$ is the $(1 - \alpha/2)$-quantile of the standard Normal distribution and $[\hat{\mathcal{J}}_0^{-1}]_{ii}$ is the i-th

diagonal element of $\hat{\mathcal{J}}_0^{-1}$. Similarly, a confidence ellipsoid for θ with approximate degree of confidence $1-\alpha$ is obtained as the region satisfying $(\theta-\theta_{\mathrm{ML}})^\top \hat{\mathcal{J}}_0\,(\theta-\theta_{\mathrm{ML}}) \leq \chi^2_{1-\alpha}(q)$, where q is the dimension of the parameter space and $\chi^2_{1-\alpha}(q)$ is the $(1-\alpha)$-quantile of the $\chi^2(q)$ distribution.

If one does not want to rely on asymptotic normality, obtaining confidence intervals becomes more difficult. The obvious alternative is bootstrap, where resampled data is obtained either by parametric resampling (e.g. Rydén et al. 1998) or non-parametric resampling. In either case an MLE is to be computed for each resampled data series, resulting in an overall large number of EM iterations, and for non-parametric bootstrap the dependence in the data requires the use of e.g. block resampling (Carlstein et al. 1998) to which are attached tuning parameters such as block size. Bootstrap for HMMs is thus far from an automated procedure. In addition, when bootstrapping e.g. the Normal mixture model studied here, the MLEs computed for the different bootstrap replicates may have the μ_i appearing in different orders w.r.t. their numerical values. That is, for some bootstrap replicate $\mu_{1,\mathrm{ML}} < \mu_{2,\mathrm{ML}} < \mu_{3,\mathrm{ML}}$ whereas maybe $\mu_{2,\mathrm{ML}} < \mu_{1,\mathrm{ML}} < \mu_{3,\mathrm{ML}}$ for another replicate. This will in particular be the case if EM is run multiple times for each replicate, initialised from different random starting points. Thus the label-switching problem occurs here too. To compute sensible bootstrapped confidence intervals for the μ_i say, the MLEs for the different replicates need to have the μ_i sorted in a common way, e.g. in ascending order. The net effect of such a constraint is similar to that caused by an identifiability constraint on a prior in a Bayesian context, as discussed above.

To make a concrete comparison of bootstrapping with EM and credibility intervals from Gibbs sampling, let us consider computing the 95% percentile of the bootstrap and posterior distributions, respectively, for μ_2 in the above example from the data with $\sigma^2 = 1$. To make the meaning of μ_2 unique, we impose the identifiability constraint $\mu_1 < \mu_2 < \mu_3$. The purpose of computing this percentile could be e.g. as the upper bound of a 90% confidence or credibility interval for μ_2. Using the parameters estimated with EM, we generated 661 bootstrap series of size $n = 1,000$ by simulating an HMM with parameters given by the MLE computed from the original data. That is, we used parametric bootstrap. The number 661 is obtained from an argument assuming that the bootstrap distribution of μ_2 is approximately Normal (see the Appendix); Normal probability plots (not shown) revealed that such as assumption is reasonable. For each bootstrap replicate, EM was used to re-estimate the parameters. For both the original and the bootstrapped series, the EM algorithm was stopped when all parameters a_{ij}, μ_i and σ moved less that 10^{-3} in an iteration. Computing the 661 bootstrapped parameter estimates took 2,177 s of CPU time, i.e. 3.3 s per replicate, using a Matlab implementation running under Linux on a PC with an Intel Pentium IV 2.66 GHz CPU. The sample 95% percentile of μ_2 was 0.154. The 5% quantile was -0.103, so that $(-0.103, 0.152)$ is the bootstrap estimate of a 90% confidence interval for μ_2.

We then used the 10,000 sweeps (after burn-in) of the Gibbs sampler to obtain a sample 95% percentile of μ_2 equal to 0.141 (the sample 5% percentile was -0.129). We also computed empirical autocovariances of the indicator sequence $\zeta^{[t]} = I\{\mu_2^{[t]} \leq 0.141\}$

and summed them over the range $-30, \ldots, -1, 0, 1, \ldots, 30$ as above to obtain an estimate 5.47 of the constant C_α of the Appendix, accounting for the increased variance caused by the serial dependence in the Gibbs sampler. Thus about $5.57 \times 661 \approx 3,700$ samples from the Gibbs sampler would suffice to obtain the same precision in estimating the sample percentile as for the bootstrap, or in total 4,700 sweeps including burn-in. The CPU time per sweep was 50.5 ms, giving a total time of about 237 s for 4,700 sweeps. Thus, because the ratio 3300 ms/50.5 ms ≈ 65 of CPU times for a bootstrap replicate and Gibbs sweep respectively is much larger than the variance increase factor 5.47, Gibbs sampling is faster than bootstrapping. On the other hand, estimating a number like 5.47 requires a preliminary run to estimate covariances in the Gibbs sampler, and this factor C_α also depends on the chosen parameter and quantile. For instance, for the median of μ_2 this factor was estimated at 12.1 while for the 95% percentile of μ_3 it was estimated at 3.6. From this perspective, bootstrapping is more automatic. For the bootstrap analysis the computation time crucially depends on the stopping tolerance for EM however, which is somewhat arbitrary.

2.4 Summary

We now summarise this section. We have seen that both EM and full Gibbs sampling are viable inferential procedures for the model considered, which is thought of as being representative for an HMM with known number of states and component densities from some parametric family. Both approaches suffer from less informative data, EM in the way of slower convergence and Gibbs sampling in terms of mixing rate. In one pass of the EM algorithm the main computational expense is the forward-backward algorithm. One sweep of the Gibbs sampler uses the backward pass only but in addition forward sampling, rendering a computational cost similar to that of one EM iteration. However, typically one would run many more Gibbs sweeps than EM iterations, so that the total computational cost for Gibbs sampling is much higher. If only a point estimate is desired EM is thus the simplest and quickest way there. This is true also if a prior is placed on the parameters—at least as long as it is conjugate relative to the complete data likelihood—and the point estimate is then the MAP. Having said that we remark that since EM does not guarantee convergence to the MLE or MAP but may end up at a local maximum or even a saddle point of the likelihood function or posterior density, it is quite common to run EM from several initial points in the parameter space, often chosen randomly, and to select the point giving the overall largest objective function as the final estimate. Such multiple runs of EM may offset most of the computational savings of only one run, relative to Gibbs sampling for computing e.g. posterior means.

For interval estimates the quickest solution is confidence intervals based on the Normal distribution and the observed information, requiring negligible additional computation time compared to computing the point estimate itself. In the comparison of bootstrap vs. Gibbs sampling our conclusion is that the actual computation time is smaller for Gibbs sampling in most cases. The serial dependence of the sampler however creates a need for preparatory analyses, and working with the (conditionally) i.i.d. replicates of the bootstrap is a lot simpler from this perspective.

3 Case II: Unknown number of states

We now turn to a more challenging estimation problem, namely that of estimating the number d of hidden states along with the model parameters. In other words, we consider *model selection*. If the HMM is fully parametrised in the sense that the unknown parameters are the transition probabilities and parameters specifying each individual component—as in the previous section for instance—then the class of models with d states, $\Theta^{(d)}$ say, form a nested sequence in d; $\Theta^{(1)} \subset \Theta^{(2)} \subset \cdots$ in the sense that for each specific model in $\Theta^{(d)}$, there is a model in $\Theta^{(d+1)}$ governing the same distribution for $\{Y_k\}$ (the model is equivalent). With a frequentist approach, model selection in such settings is often carried out using generalised likelihood ratio tests (GLRTs), with another approach being penalised likelihood criteria like the Akaike or Bayesian information criteria (AIC/BIC). Here we do not consider AIC/BIC further however, but rather focus on methods that provide some kind of measure of confidence in the selected model; AIC/BIC obviously do select a model, but provide no information about the confidence in this model is relative to others.

A major problem with GLRTs in the case of HMMs is that the asymptotic distribution is not the usual χ^2 because if the true model is in $\Theta^{(d)}$, there is a continuum of equivalent models in $\Theta^{(d+1)}$. This difficulty occurs already for mixture distributions, (see e.g. McLachlan and Peel 2000, Section 6.4). Although theory explains the limit distribution in terms of a Gaussian process (e.g. Hansen 1992), this theory has not lent itself to any practically useful numerical approximations to critical levels or p-values. Rather, the approach usually taken in the literature is bootstrapping the GLRT. This bootstrap can be either non-parametric, leading to the same non-trivial design choices for dependent data as described in the previous section, or parametric.

In a Bayesian framework one places a prior distribution on the model size d, and then tries to infer the posterior distribution of d—and the model parameters—given data. In practice this again requires numerical computations using MCMC. There are two main approaches to this problem, with one being to run MCMC, typically a Gibbs sampler, for different d separately, and the other approach being to use MCMC algorithms that incorporate moves between models of different dimensionality, often called trans-dimensional MCMC. In the former case the central quantity one needs to estimate for fixed d is the marginal likelihood $p(\mathbf{y}_{1:n}|d)$, since $p(d\,|\,\mathbf{y}_{1:n}) \propto p(\mathbf{y}_{1:n}|d)p(d)$ where $p(d)$ is the prior on d. There are many ways to approximate marginal likelihoods, see e.g. Frühwirth-Schnatter (2004) and Frühwirth-Schnatter (2006, Sections 5.4 and 11.6.3) who in particular advocated using the *bridge sampler* (Meng and Wong 1996). With this method, $p(\mathbf{y}_{1:n}|d)$ is estimated as

$$\hat{p}(\mathbf{y}_{1:n}|d) = \frac{L^{-1}\sum_{\ell=1}^{L} \kappa(\tilde{\theta}^{[\ell;d]}) p^\star(\tilde{\theta}^{[\ell;d]}|\mathbf{y}_{1:n}, d)}{M^{-1}\sum_{m=1}^{M} \kappa(\check{\theta}^{[m;d]}) q(\check{\theta}^{[m;d]})},$$

where $p^\star(\theta|\mathbf{y}_{1:n}, d) = p(\mathbf{y}_{1:n}|\theta, d)p(\theta|d)$ is the unnormalised posterior density of θ on $\Theta^{(d)}$, κ is an arbitrary function on $\Theta^{(d)}$, q is an arbitrary probability density on $\Theta^{(d)}$, the $\check{\theta}^{[m;d]}$ are samples from the posterior $p(\theta|\mathbf{y}_{1:n}, d)$ obtained using some MCMC algorithm, and the $\tilde{\theta}^{[\ell;d]}$ are i.i.d. samples from q. This estimator contains particular Monte Carlo

schemes for estimating $p(\mathbf{y}_{1:n}|d)$ like importance sampling and reciprocal importance sampling as special cases, obtained by selecting κ appropriately. Meng and Wong (1996) showed that an optimal κ, in terms of asymptotic variance, is given by

$$\kappa(\theta) \propto \frac{1}{Lq(\theta) + Mp(\theta|\mathbf{y}_{1:n}, d)}.$$

Here $p(\theta|\mathbf{y}_{1:n}, d)$ is the *normalised* posterior density for θ on $\Theta^{(d)}$, which is not known. Nevertheless one can construct an iterative procedure in which $p^\star(\theta|\mathbf{y}_{1:n}, d)$ is normalised using the estimate of $p(d|\mathbf{y}_{1:n})$ from the previous iteration (see Frühwirth-Schnatter 2004, Eq. (8)), and this recursion typically converges in a few steps to a final estimate of $p(d|\mathbf{y}_{1:n})$. Thus this approach requires small additional efforts compared to Gibbs sampling for fixed d, since the same code can be used for a range of d. The density q can also be chosen based on output from the Gibbs sampler (see Frühwirth-Schnatter 2004, Section 3.4). In the application below we choose between d in the range from 1 to 8, so that a marginal likelihood analysis would amount to (a) running the Gibbs sampler for all d in the range 1–8 separately, (b) using e.g. the bridge sampling device to obtain estimates $\hat{p}(\mathbf{y}_{1:n}|d)$, and (c) computing $\hat{p}(d|\mathbf{y}_{1:n})$, $1 \leq d \leq 8$, as numbers proportional to $\hat{p}(\mathbf{y}_{1:n}|d)p(d)$ and summing to unity. What one could worry about in this application, and in general, is the mixing rate and how well the Gibbs sampler explores the posterior density for over-parametrised models, i.e. models with d larger than what is supported by the data.

Turning to trans-dimensional MCMC, in particular reversible jump MCMC (RJMCMC) due to Green (1995) has been used quite widely for HMMs. While computation of marginal likelihoods by sampling for different d separately requires little new computer code, trans-dimensional MCMC does require substantial additional coding efforts and can also be difficult to design so that dimension-changing moves achieve reasonable acceptance rates. However, trans-dimensional MCMC can sometimes improve mixing compared to fixed-dimensional MCMC by allowing the sampler to move between modes in the posterior distribution for some particular model size d through visits to models of other sizes; see Richardson and Green (1997, Section 6.2.2) for a somewhat contrived but still relevant illustration with finite mixtures. A small comparison of RJMCMC to marginal likelihoods computed using the bridge sampler, for a two-component Poisson mixture model, can be found in Frühwirth-Schnatter (2006); the conclusion there is a moderate advantage for the bridge sampler.

Finally we mention that there are also other hierarchical Bayesian approaches that do not explicitly put a prior on d, but in which a posterior distribution of d rather results as a function of the posterior distribution of other model parameters. One example is the *hierarchical Dirichlet process* (HDP) of Teh et al. (2006). In this model a random probability distribution G_0 is drawn from the Dirichlet process $\mathrm{DP}(\gamma, H)$, where γ and H are hyperparameters and H is a distribution on the state space of the latent Markov chain. It is usually most convenient to let this state space correspond not to $\{1, 2, 3, \ldots\}$, but to the space in which the *parameters* associated with the conditional distributions of the different states lie. Thus, in the $N(\mu_i, \sigma^2)$ model of the previous section we would take this space as $\mathcal{R}$, whereas in the example of this section, described below,

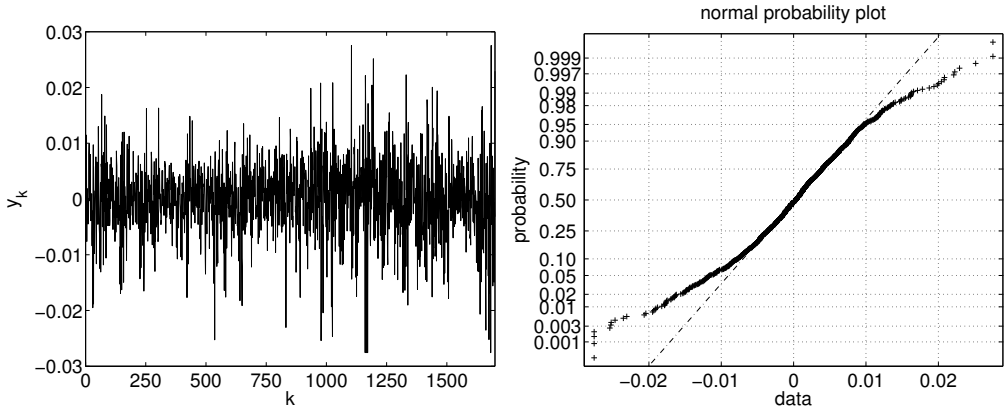

Figure 5: *Time series (left) and Normal probability plot (right) of S&P 500 data.*

we would take it as $(0, \infty)$. Since the realisation of a Dirichlet distribution is discrete (Ferguson 1973), the support of G_0, which is infinite, can be thought of as the set of possible states. Then for each such possible state the HDP contains another Dirichlet process, drawn from $\mathrm{DP}(\alpha_0, G_0)$, that governs the transition probabilities. In this way the set of possible states is always infinite, and we interpret d as the *realised number of states*, i.e. the number of distinct states visited within the time span the observations were taken. When doing posterior analysis of the model using MCMC, this number is a bi-product and its posterior distribution can hence be simulated. In this Bayesian model formulation the states can in fact be thought of as nuisance parameters, and the primary objective of the analysis is to cluster observations on one or more levels of hierarchy. Yet another paper in which the parameter d is thought of as the *realised* number of states during the time span of the observations, is that by Chopin (2007). There the approach is however closer to that described above, in that it starts from the traditional formulation of an HMM which is then reformulated as the latent chain gradually visiting an increasing number of distinct states.

3.1 Data and model

The data we will study here is a sequence of 1,700 daily log-returns from the S&P 500 stock index during the 1950's. The structure of the HMM is again conditional Normal, but this time with components having zero mean and individual variances; in other words, $Y_k | X_k = i \sim \mathrm{N}(0, \sigma_i^2)$. This HMM can be thought of as a stochastic volatility model, with a finite number of possible volatilities. The sample mean of the original data is -4.0×10^{-11}, which was subtracted off before further analyses were carried out. This particular dataset has previously been analysed by Rydén et al. (1998) (called 'subseries E') and Robert et al. (2000), and is displayed in Figure 5. Note in particular that the Normal probability plot reveals that the marginal distribution has tails heavier than those of a Normal distribution, and the mixture of several Normals is a way to

capture that feature of the data.

3.2 Likelihood ratio testing with bootstrap and EM

For the likelihood ratio analysis we employed parametric bootstrap for testing d vs. $d+1$ hidden states. In other words we computed the MLEs $\hat{\theta}^{(d)}$ and $\hat{\theta}^{(d+1)}$ for models with d and $d+1$ states respectively, and the corresponding observed (log) LR-statistic $\mathrm{LR}_{\mathrm{obs}}^{(d)} = \log L(\hat{\theta}^{(d+1)}; \mathbf{y}_{1:n}) - \log L(\hat{\theta}^{(d)}; \mathbf{y}_{1:n})$ as the difference between the respective log-likelihoods. These MLEs were computed by starting EM at randomly chosen initial points—each row of A as well as the initial vector ρ was drawn from a Dirichlet $\mathrm{Dir}(1, 1, \ldots, 1)$ distribution and the σ_i were drawn uniformly on $(0, \max|y_k|/2)$, all independently—and EM was iterated until the difference of two successive log-likelihood values was less than 10^{-3}. No stopping criteria were put on the parameters themselves, as the log-likelihoods are the important numbers here. For each model 50 different random initial points were used, and the overall best log-likelihood was stored.

Furthermore we simulated bootstrapped series of size $n = 1,700$ from the model with parameter $\hat{\theta}^{(d)}$, and for each of these a bootstrapped LR-statistic $\mathrm{LR}_{\mathrm{boot},r}^{(d)}$ was computed by proceeding exactly as above for the original data (including subtraction of the sample mean). Denoting the total number of bootstrapped series by R, i.e. $r = 1, 2, \ldots, R$, an estimated p-value of the test for d vs. $d+1$ hidden states is then given by $(b+1)/(R+1)$ where b is the number of bootstrap samples r for which $\mathrm{LR}_{\mathrm{boot},r}^{(d)} > \mathrm{LR}_{\mathrm{obs}}^{(d)}$.

We performed this bootstrap procedure for testing $d = 2$ vs. $d = 3$ using $R = 200$ bootstrap series. The observed LR-statistic was 10.03, and the largest bootstrapped one was 9.85. Thus $b = 0$ and the estimated p-value is $1/201 \approx 0.005$. These results differ somewhat from those obtained by Rydén et al. (1998, Table III). The differences arise as Rydén et al. based their analysis of subsamples of length 800 observations, whereas here samples of full size 1,700 were used. The main point we want to make here however is that these computations were slow; on average each bootstrapped LR-statistic took 1805 s of CPU-time to produce using Matlab on the same machine as in Section 2, yielding a total CPU-time of about 100 h for all 200 series. Obviously these computations would have been must faster if implemented in e.g. C, but the main focus here is on the comparison to the RJMCMC computations described below. We also proceeded to testing $d = 3$ vs. $d = 4$ using another $R = 200$ bootstrap series. In this test the observed LR-statistic was 7.41 and $b = 6$ of the bootstrapped ones exceed this value, giving an estimated p-value of $7/201 \approx 0.035$. The average CPU-time per bootstrap replicate was 4722 s, giving a total computation time of about 262 h. We did not attempt to test $d = 4$ vs. $d = 5$.

One could cut run times by running less than 50 repeated runs of EM for each series. Making too few runs, say 10, however causes a considerable risk of not finding the maximal likelihood for the overparametrised model with $d+1$ states, and even ending up with negative LR-statistics (which are false and artefacts of unsuccessful optimisation over $\Theta^{(d+1)}$).

3.3 Implementation of the reversible jump MCMC sampler

For the Bayesian analysis we employed a reversible jump sampler similar to that described in Cappé et al. (2005, Example 13.2.2). This sampler incorporates moves that update the model parameters without changing the dimension of the model—just like in the previous section—but also moves that increase or decrease the number of hidden states. The latter are the *split move* which attempts to split one component into two, with different variances, and the *combine move* which attempts to combine two components into one. Using moves of this structure goes back to Richardson and Green (1997).

A uniform prior over $\{1, 2, \ldots, d_{\max}\}$ with $d_{\max} = 8$ was put on d and the σ_i were equipped with independent uniform $\mathrm{U}(0, \alpha)$ priors where the hyperparameter α had an exponential prior with mean $5 \max_k |y_k|$. This prior on the σ_i is the same as that used by Robert et al. (2000). The transition probabilities were parametrised by ω_{ij} for $i, j = 1, 2, \ldots, d$, where each ω_{ij} has an independent exponential prior with unit mean and $a_{ij} = \omega_{ij} / \sum_{j'} \omega_{ij'}$. This gives the same Dirichlet prior on the rows of A as in the previous section, but simplifies the construction of the RJMCMC sampler as the row sum constraint does not apply to the ω_{ij}.

One sweep of the RJMCMC sampler employed contains the following steps.

(i) Impute the hidden Markov chain using current parameters and backward recursion forward simulation, and sample each σ_i^{-2} from its conditional distribution given data and $\mathbf{x}_{1:n}$. This distribution has density (in v) proportional to $v^{(n_i-3)/2} e^{-v(S_i/2)} I\{v \geq \alpha^{-2}\}$, where $n_i = \#\{k : x_k = i\}$ and $S_i = \sum_{k:\, x_k = i} y_k^2$. Sampling from this density was carried out using the slice sampler of Damien et al. (1999).

(ii) Resample the ω_{ij} by a Metropolis-Hastings step with proposal $\omega'_{ij} = \omega_{ij} \varepsilon_{ij}$ for each $i, j = 1, 2, \ldots, d$, where $\log \varepsilon_{ij}$ are independent $\mathrm{N}(0, \tau_\omega)$. The joint proposal of all ω'_{ij} was either accepted or rejected. The acceptance ratio for this move is described in Cappé et al. (2005, Example 13.1.14).

(iii) Resample α from its full conditional distribution, which has density (in v) proportional to $v^{-d} e^{-Rv} I\{v \geq \max \sigma_i\}$. The same slice sampler as in (ii) was used.

(iv) Attempt a split of a component with probability $1/2$ if $1 < d < d_{\max}$ (otherwise 1 if $d = 1$ and 0 if $d = d_{\max}$), or attempt to combine two components with probability $1/2$, 0 and 1 for the cases $1 < d < d_{\max}$, $d = 1$ and $d = d_{\max}$ respectively. This move was designed as in Cappé et al. (2005, Example 13.2.2) with the exception that there are no Normal means to consider, while σ_{i_0}, where i_0 is the component to split, was split as $\sigma_{i_1} = \sigma_{i_0} \xi_\sigma$, $\sigma_{i_1} = \sigma_{i_0} / \xi_\sigma$ with $\log \xi_\sigma \sim \mathrm{N}(0, \tau'_\sigma)$. The Jacobian corresponding to this part of the move is $2\sigma_{i_0} / \xi_\sigma$, replacing the expression in (i) of the cited example. We refer to Cappé et al. (2005) for further details on this move.

A major difference between the sampler presented here and that used in Robert et al.

(2000) is that the latter includes the latent Markov chain as part of the RJMCMC state space, while here the imputation in (i) above is only used as a means to resample the σ_i from their full conditional distribution; when step (i) is finished, the imputed realisation of $\mathbf{X}_{1:n}$ is discarded. This dramatically reduces the dimensionality of the sampler's state space and suggests a faster mixing sampler (when moves are identical otherwise, which they are not here compared to Robert et al. 2000). A further aspect of our sampler is that we did not put any identifiability (ordering) constraints on the parameters.

3.4 Results from the reversible jump MCMC sampler

We ran this RJMCMC sampler for 100,000 sweeps with $\tau_\omega = 0.2$, $\tau'_\sigma = 0.6$ and $\tau'_\omega = 0.9$ respectively (the last variance being involved in splitting the ω_{ij} in (iv)). Using a Matlab-implementation, the average CPU-time for one sweep was 0.67 s with a total computation time of about 19 h on the same computer as above. Again this run time would have been shorter with an implementation in a different computer language, but the main point here is that this it was shorter than for the bootstrap analysis.

The mean acceptance probability was 44% for the Metropolis-Hastings step in (ii) above and 2.3% for the split-combine move (iv). The latter is obviously lower than desired. We did moderate attempts to increase it by varying the variances τ' involved in the split-combine move, and the rate reported here was the best obtained. We remark that obtaining satisfying acceptance rates for dimension-changing moves for HMMs is generally difficult; Robert et al. (2000) obtained the rate 4.4% for the same dataset, seemingly contradicting the above suggestion of faster mixing with a smaller RJMCMC state space. Robert et al. however employed a much more elaborate split-combine move being more carefully optimised, so the rates are not directly comparable. Here we did not attempt to optimise the move structure, but rather suggest that even with a not too involved structure one can obtain a sampler that works, although not optimally.

Some example output plots are presented in Figure 6. We note that apart for over the model size d, the sampler appears to mix well. We also note that there is label-switching within the subsequences of sweeps with $d = 2$ and $d = 3$ respectively, i.e. the σ_i do not stay in fixed order within these subsequences. This switching is mainly caused by visits to models of different sizes. Discarding the first half of the sample as burn-in, the estimated posterior probabilities of $d = 2$, 3, 4 and 5 were 0.427, 0.494, 0.067 and 0.011 respectively, with values below 0.2% for remaining values of d. These estimates are similar to those obtained by Robert et al. (2000, Table 2), but with a slightly higher value for $d = 3$ at the expense of $d = 2$. Comparing to the bootstrap analysis, the degree of belief in $d = 4$ vs. $d = 3$ is comparable to what was obtained with the GLRT, whereas the results for $d = 2$ vs. $d = 3$ is entirely different; here $d = 2$ comes out as a plausible model, though it was firmly rejected by the GLRT.

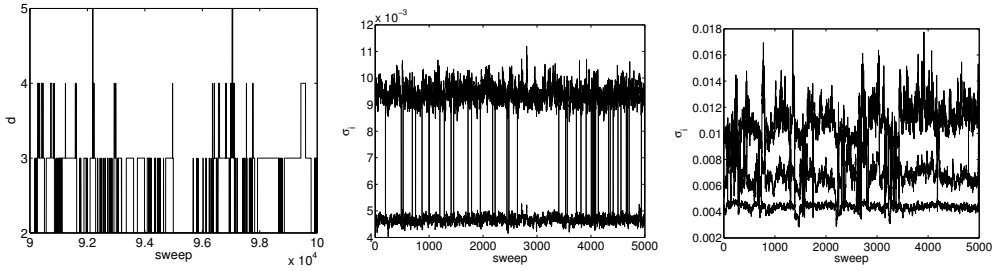

Figure 6: *Left plot: Sampled values of model size d in sweeps 90,001–100,000 of the RJMCMC sampler. Middle plot: Sampled values of σ_1 and σ_2 in the last 5,000 sweeps with $d = 2$. Right plot: Sampled values of σ_1, σ_2 and σ_3 in the last 5,000 sweeps with $d = 3$.*

3.5 Summary

Summing up this case, we first remark that if one wants to compute only the best model without any further information on how plausible it is relative to other ones, then the simplest solution is using EM to compute MLEs for all candidate models and then calculating and comparing their BICs or some other penalised likelihood criterion. We also remark however that BIC is based on approximating the distribution of the MLE by a Normal, and may be unreliable, as exemplified by Scott (2002, Section 4.1), for data of small or moderate size.

We have discussed several methods to provide not only a model that is in some sense best-fitting, but also a number indicating how likely this model is in relation to other candidates. For simulating p-values of generalised likelihood ratio tests using parametric bootstrap and EM, as good as no extra code compared to Case I is required, but we have seen that computation times can be substantial even when carrying out only a few tests; in the example above just 2 vs. 3 states and 3 vs. 4 states were tested. Likewise, little extra code is required for computing marginal likelihoods using the bridge sampler, and computation times will typically be much less than for the bootstrap approach. A possible problem with this method is, as noted above, potentially poor mixing of the Gibbs sampler with over-parametrised models. As long as the Gibbs sampler mixes reasonably for all candidate models, this approach should however be feasible and have a relatively low computational cost. Reversible jump MCMC on the other hand requires considerable amounts of additional code. Its problems with mixing do not lie in mixing within models of fixed size, which can be improved relative to a fixed-d Gibbs sampler, but typically in small probabilities of moving between models of different sizes. Designing good dimension-changing moves often demands experimentation and experience. The HMM considered here is in fact quite simple, and with more complex structures such as involved dependencies between the latent and observed data, autoregressions in the Y_k etc., designing RJMCMC samplers can be extremely difficult or almost impossible. Similarly, in such models a fixed-d Gibbs sampler may spend hours of computation time in minor modes and fail to deliver a correct picture of the posterior density surface, while

EM can repeatedly end up in local maxima. Model selection in more complex models can thus be intrinsically difficult, irrespective of the inferential and computational approach taken.

Finally it should be pointed out that with a Bayesian analysis, regardless of the computational approach taken, some other aspects require attention too. One is the influence of the prior distribution, which can be particularly difficult to understand in model selection settings with a prior on d. Aitkin (2001) is a good and critical review of several earlier papers all using Normal mixtures to analyse a common set of data, and in particular it focuses on the quite different results obtained; see also Frühwirth-Schnatter (2006, Section 5.3.2).

4 Case III: Overlapping dependence structure

In this third and last case we will study a model that involves a continuous-time (hidden) Markov chain, and also a more complex structure for the dependence of the output w.r.t. this latent process.

4.1 Biological background

The model is motivated by array comparative genomic hybridisation data. The genetic material of a cell is encoded in the form of DNA. In eukaryotic cells, such as a human cell, the DNA lies inside the cell nucleus. Each cell (nucleus) carries a complete copy of the full DNA. In human cells the DNA is organised into 23 pairs of chromosomes; thus 46 chromosomes in total. The chemical structure of DNA is usually described as a double helix, consisting of two *strands* wound around each other. Each strand consists of a sequence of four possible bases, coded using the alphabet {A, C, G, T}. The two strands are complementary in the sense that an A on one strand is matched by a T on the other strand, with a similar match between C and G. For each chromosome pair the base sequences of the two chromosomes are identical for the major part of their lengths (with the exception for one pair, the sex chromosomes). Therefore it is common to say that humans have *two copies* of the DNA. Functionally it also makes sense to think of each chromosome as one long string of base pairs. We will refer to a position in such a string as a *genomic location*, measured in the unit of *base pairs* (bps), relative to the beginning of the string. However, the functional view of a chromosome as a single string does not reflect the way that it is physically organised inside a cell nucleus. Rather, the DNA is split into many shorter segments. Also, parts of the DNA may exist in more or less copies (segments) than two. Thus the conception of the DNA existing in two copies is a simplified view, as, in parts of the genome, it can exist in only one copy (then carried by only one chromosome), in three or more copies, or in no copies at all. The number of copies of the DNA at a given genomic location is called the *copy number*. Deviations from the normal two copies are often referred to as *aberrations*. Aberrations are of medical interest, as they may cause or increase the risk of diseases, for instance different forms of cancer (e.g. Albertson et al. 2003). As another example, it has been

found that homozygotic twins may have different DNA, not regarding the actual base sequence but in the copy number at certain genomic locations (Bruder et al. 2008).

For reasons as indicated above, there is a large medical interest in studying copy number variations, and trying to relate copy number aberrations to e.g. diseases. To carry out such studies, techniques to measure copy numbers are needed. One such technique is *array comparative genomic hybridisation* (aCGH). The following is a brief and incomplete description of this data acquisition procedure; complete descriptions can be found e.g. in Snijders et al. (2001) or Albertson and Pinkel (2003). The basic principle of aCGH analyis is to compare, at selected genomic locations, the copy number of a sample DNA to that of a reference DNA. The first step of the procedure is to prepare spots on a micro array (carrying thousands of spots) with *clones*, or *reporters*. A clone is a short segment of base pairs that is complementary, in the sense A-T and C-G, to a specific chosen subsequence of the genome. In this way, each clone corresponds to a certain part of the genome. The sample DNA is then labelled with a flourescent dye of one colour, the reference DNA is labelled by a dye of a different colour, and a mixture of both is hybridised onto the micro array. DNA segments matching the clone on a particular spot will then bind to that spot. By irradiating the spots with laser light of colours corresponding to the two dyes, one can measure the amount of DNA from the two samples that has bound to each spot. Assuming also that the amount of DNA bound to each spot is proportional to the fraction of DNA containing the base sequence matching the corresponding clone, one obtains a figure representing the copy number of the sample DNA relative to that of the reference DNA (which is two) at the genomic locations specified by the clone. This number is typically transformed onto $\log_2$-scale, so that the actual data y_k for clone k on the microarray can be thought of as $y_k = \mu_{x_k} + \varepsilon_k$ where x_k is the copy number for clone k, μ_{x_k} is the corresponding mean level and ε_k is noise. In an ideal setting the μ_i would equal $\log_2(m/2)$ for a selection of $m = 0, 1, 2, \ldots$, but for various reasons of experimental bias this ideal relationship does not hold and one must treat the μ_i as unknown parameters. A plot of the data for one chromosome is found in Figure 7.

4.2 Data and model

From now on we will assume that the data is from some single chromosome. As each clone k corresponds to a subsequence of the genome of that chromosome, we can associate with it a starting location t_k^{start} and a stopping location t_k^{stop}, where the unit of these numbers is bp. We will also assume that the clones are sorted according to increasing starting location, i.e. t_k^{start} increases with k. As is obvious from Figure 7, the copy numbers are not independent across the clones. Many authors, e.g. Picard et al. (2005), have addressed statistical analysis of aCGH data using change point techniques. A different approach is to model the copy number process as a latent stochastic process. Fridyland et al. (2004) modelled $\{X_k\}$ as a finite-state Markov chain, resulting in $\{Y_k\}$ being an HMM. There are some arguments against such a model though, including that (i) clones are of unequal lengths and are separated by unequal distances, and that (ii) in many array designs clones overlap by as much as up to 30% of their lengths. Here *over-*

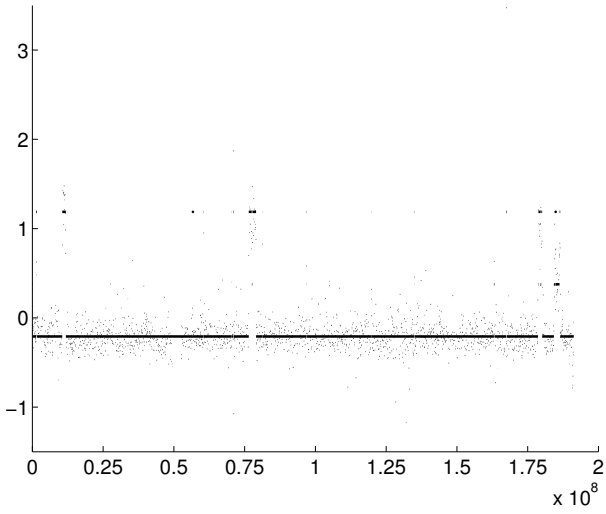

Figure 7: *aCGH data (thin lines) and a possible reconstruction of the latent Markov chain (thick lines) for chromosome 4 from the human breast cancer cell line BT-474. Each thin line segment represents the location and extension (x-axis) and measured copy number relative to the normal two, on $\log_2$-scale (y-axis), for one clone. For further information on this data we refer to Stjernqvist et al. (2007). The reconstruction is a trajectory of the hidden Markov chain obtained with the RJMCMC sampler described in the text and model parameters being those produced in the final iteration of the Monte Carlo EM algorithm.*

lap means that one clone may start before the previous one has ended, and indeed even more than two clones may overlap simultaneously (see Figure 8). These observations lead to the conclusion that a simple HMM with constant transition probabilities may not be a realistic model for the data. Stjernqvist et al. (2007) rather modelled the copy number process as a continuous-time Markov chain, with time unit being bp. This also allows for a copy number change within a clone. In fact the term 'time' is not entirely appropriate as the index is not time but rather location in the DNA in a chromosome; Stjernqvist et al. (2007) used the term 'continuous index' but we will stick to 'time' here for convenience. Letting $\{X(t)\}_{0\leq t\leq T}$ denote this latent process, where T is the length of a chromosome (different chromosomes are analysed separately), the model proposed for the observed Y_k is

$$Y_k \sim N\left(\frac{1}{t_k^{\text{stop}} - t_k^{\text{start}}} \int_{t_k^{\text{start}}}^{t_k^{\text{stop}}} \mu_{X(t)}\, dt, \sigma^2\right), \tag{8}$$

where μ_i as before is the mean level of observations when the latent Markov process is in state i. Hence the conditional expectation of Y_k is modelled as a weighted mean of the μ_i, with weights equal to the proportions of the clone that the copy number process spent in the respective states. Furthermore the Y_k are assumed conditionally

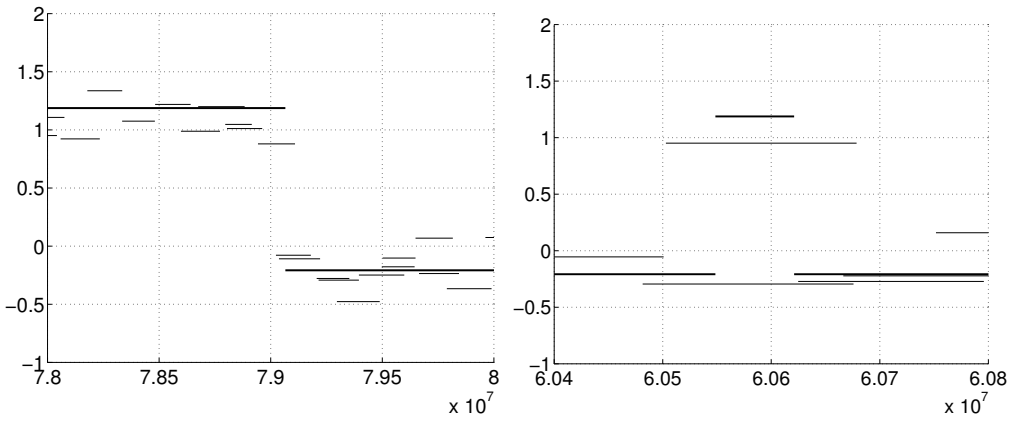

Figure 8: *Data (thin lines) and a possible reconstruction of the latent Markov chain (thick lines) for two shorter subsequences of the same data and reconstruction as in Figure 7.*

independent given the latent process $\{X(t)\}$. The dynamics of the Markov chain is expressed by transition intensities q_{ij}, $i, j = 1, 2, \ldots, d$, $i \neq j$. It would have been possible to formulate the model in discrete time, with a Markov chain evolving on bp's rather than clones. Since the number of bp's in a chromosome is very large (2×10^8 in the example below), such a Markov chain would have transition probabilities being either very small (for transitions between states) or very close to one (for staying in a state). Working with parameters that close to the boundaries of the parameter space is awkward, and we find the above continuous-time formulation more appealing.

4.3 Monte Carlo EM and MCMC algorithms

We note that the above model extends the basic HMM assumptions of Section 1 in several ways. The most obvious one is that the latent Markov chain evolves in continuous time. Moreover, whenever there is clone overlap the corresponding Y_k do not depend on disjoint parts of the trajectory of $\{X(t)\}$, violating a continuous-time counterpart to assumption (iii) of Section 1. This second extension has the major effect of precluding the design of an efficient forward-backward algorithm, as such algorithms always make implicit use of the ordered and disjoint dependence of the data on the latent process. This in turn leaves us without an efficient E-step in the EM algorithm for this model. There are many examples of EM algorithms for continuous-time HMMs, see e.g. Roberts and Ephraim (2008), but only in models in which the above-mentioned assumption (iii) holds in some way.

The M-step of the EM algorithm for the model above is quite simple however, see Stjernqvist et al. (2007, Suppl. info.). In that paper the authors used a Monte Carlo EM (MCEM) algorithm to compute an approximation to the MLE. In other words, in

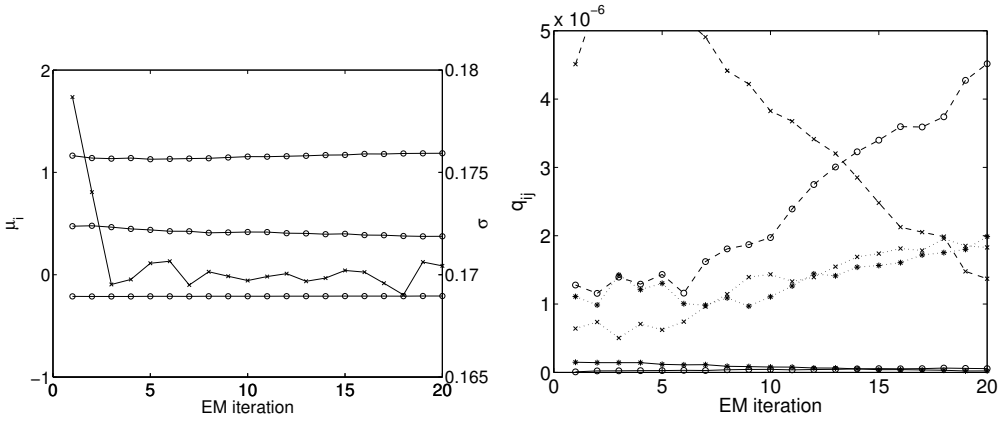

Figure 9: *Parameter estimates as a function of EM iterations for the continuous-time HMM with 3 states and the same aCGH data as shown in Figure 7. Left plots: means* μ_i *(*$\circ$*, left y-scale) and* σ *(*$*$*, right y-scale). Right plot: transition intensities* q_{ij} *for* $(i,j) = (1,2)$ *(solid,* $*$*);* $(i,j) = (1,3)$ *(solid,* $\circ$*);* $(i,j) = (2,1)$ *(dashed,* $\times$*);* $(i,j) = (2,3)$ *(dashed,* $\circ$*);* $(i,j) = (3,1)$ *(dotted,* $\times$*);* $(i,j) = (3,2)$ *(dotted,* $*$*).*

EM iteration m, letting $\theta^{(m)}$ denote the current estimate of the parameter (vector), one simulates R trajectories $\{x_r(t)\}_{0\le t\le T}$, $r = 1, 2, \ldots, R$, of the latent Markov chain from the conditional distribution $\mathbf{P}_{\theta^{(m)}}(\{X(t)\}_{0\le t\le T} \in \cdot \,|\, \mathbf{y}_{1:n})$ of this process given data, and approximate any expectations regarding the latent process by empirical averages over these replicates. The trajectories were in turn simulated using an RJMCMC algorithm due to Ball et al. (1999).

For an HMM with $d = 3$ states, this MCEM algorithm was run for 19 iterations with $R = 1400m^2$ trajectories $\{x_r(t)\}$ sampled in the m-th iteration; results for the parameters are found in Figure 9. We see that the estimates of μ_1, μ_2 and μ_3 appear to have converged reasonably over these iterations, and also that the estimate of σ appears to have reached a neighbourhood where it remains although with some oscillations. For the transition intensities q_{ij} the situation is different. Indeed, we see that q_{23}, q_{31} and q_{32} grow continually. State 3 appearing here is the one with the largest μ_i (≈ 1.19), and by making short visits to this state the model can fit the Normal mean in (8) closely to the observed y_k, as a suitable small weight is then attached to μ_3. In particular this happens for clones with large y_k; an example is displayed in the right plot of Figure 8. By making transition intensities to and from state 3 large such short visits become increasingly likely, and the shorter the visits the larger the estimates of these q_{ij} become in the next MCEM iteration, and so on. Thus, as MCEM proceeds there is a continuing trend of overfitting the data by making many jumps so as to match the mean of (8) closely to the observations.

This is an obvious problem with frequentist estimation of this model. One may argue that the main interest in this model does not lie in the estimation of model parameters

but rather in reconstruction of the hidden Markov chain, as from a biological and/or medical perspective the locations of the aberrations are of primary importance. This is true, but on the other hand using fixed—and perhaps rather arbitrarily set—transition intensities in the analysis is not a good solution either. In a Bayesian approach one would put a prior on the q_{ij}, preventing them from becoming overly large and thus elegantly solving the situation. A different frequentist approach would be to view the locations of jumps as parameters and their number as a model dimension and use penalised likelihood criteria like AIC/BIC to discriminate between different models. This would however be problematic because of difficulties in maximising the likelihood over the jump locations, and also because the number of jumps can be quite large. Thus we advocate a Bayesian approach as the most appropriate.

As an example we put independent vague priors $\mathrm{N}(\xi, \kappa^{-1})$ on the μ_i, with ξ being the sample mean of $\mathbf{y}_{1:n}$ and κ small, an improper prior with density $1/\sigma^2$ on σ^2 and independent exponential priors $\mathrm{Exp}(\beta)$ on each q_{ij}. Here $\beta = 4 \times 10^7$, which with a chromosome length of about 2×10^8 base-pairs (cf. Figure 7) gives a prior mean of about $2 \times 10^8/\beta = 5$ jumps from state i to j for the full chromosome. Of course the latent process does not reside in state i for all of the chromosome, but this calculation gives an idea about the prior dynamics. It is now possible to build an MCMC sampler similar in spirit to that of Section 2, alternating between updating model parameters and the latent Markov chain. The model parameters were again sampled from their full conditional distributions, while the Markov chain was updated using the same RJMCMC moves as in the MCEM algorithm described above. Two examples of the output are found in Figure 10. We see that the data provides information about the transition intensities as the posterior and prior densities are different, but a closer look at the tails also reveals that in the far right of these diagrams the prior and posterior densities are essentially equal. This illustrates that the prior prevents the transition intensities from becoming very large, and as the prior and posterior look alike here one may say that the inference is subjective for the tails.

4.4 Summary

Summing up this case, it has not been as clear a comparison of EM vs. MCMC as in the previous cases, as also the non-Bayesian model formulation requires MCMC samples to implement MCEM. Using MCEM we also repeatedly, as the EM iterations proceed, need to perform long runs of the MCMC sampler which in itself is quite complex. Thus we see a distinct advantage in using the Bayesian approach, which rather requires a single long run of the sampler. In our view the Bayesian approach also handles the problem of potential overfitting of the Markov trajectory to the data in a simpler way, as part of the model structure, than what can be done with a frequentist approach. Although we noted above that the prior has a definite impact on the tails of the posterior distribution of the transition intensities, it is equally clear that the centre of the same posteriors are very different from the priors, thus showing that the data has the major influence there.

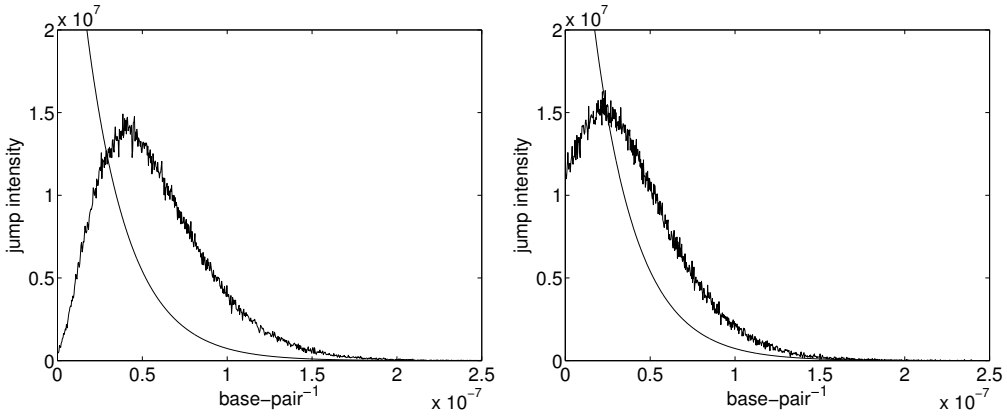

Figure 10: *Exponential prior densities and estimated posterior densities of transition intensities q_{23} (left) and q_{31} (right) of the continuous-time HMM of Case III, computed as rescaled histograms from the sampled q_{ij} in 300,000 sweeps of the RJMCMC algorithm.*

5 Discussion

We have compared inference using the EM algorithm and bootstrap on one hand, and Gibbs sampling or other MCMC algorithms on the other hand, in hidden Markov models of different degrees of complexity regarding model structure and inferential questions. In situations where one wants only a point estimate, or it is sufficient to compare models only using their (penalised) maximal likelihoods, then EM is typically the simplest and quickest way to go. When a point estimate is not sufficient, the comparison between bootstrap/EM and Gibbs/MCMC sampling is not so simple however. In the examples above we saw that Gibbs sampling and other MCMC algorithms, working in a Bayesian context, tended to require less computation time, whereas the (conditionally) i.i.d. replicates provided by bootstrap require no analyses of correlations etc. in order to assess the precision of the results. To some extent the choice is thus a matter of taste; whether one prefers to let the computer work longer, or to cut computation times but rather spend more time on the usually more manual parts of the analysis.

Having said that, it is important to be aware that inference in HMMs, whether frequentist or Bayesian, is not always an easy and far from automated task. In simple models such as that of case I there may be multimodality of the likelihood, potentially causing EM to converge to local maxima and poor mixing of MCMC samplers. A further problem illustrated is slow convergence of EM and slow mixing of the Gibbs sampler caused by imbalance between information about the parameters in the complete and observed data respectively. In Bayesian model selection the design of e.g. a reversible jump MCMC sampler is never trivial, and when running the Gibbs sampler to approximate marginal likelihoods it can spend most of its time in irrelevant modes of the posterior if the problem is ill-posed.

A Bayesian approach does show some advantages in the more complex models and

inferential problems, and indeed the critical aspect of both Cases II and III is model selection although less explicitly in Case III where the number of jumps of the latent Markov chain plays this role. In particular in Case III it is the Bayesian approach that deals with this aspect more efficiently and as an integral part of the model description. Care must however always be taken with issues like influence of prior distributions, when evaluating a Bayesian analysis.

Although a Bayesian approach to HMM analysis may be appealing from several perspective, it is the author's experience that users of HMMs often consider writing the computer code necessary to implement such procedures a prohibitive exercise. In particular reversible jump MCMC algorithms have, in the author's view, a somewhat unjustified reputation for being difficult to derive and implement. Still it is clear that readily available software packages would be extremely beneficial for making such methods available to a wider audience of researchers and users in statistics and other scientific fields.

Appendix: Review of some asymptotics of quantile estimation

Let $Z_1, Z_2, \ldots, Z_n$ be real random variables with some common distribution function F having density f. For some $\alpha \in (0,1)$, let ξ_α be the α-quantile of f, and let $\hat{\xi}_{\alpha,n}$ be the sample α-quantile computed from the sample Z_k, $k = 1, 2, \ldots, n$. Assume that $f(\xi_p) > 0$.

Write $\hat{F}_n$ for the empirical distribution function computed from the sample, and write ϕ for the functional that maps a distribution function into its α-quantile. This functional is Hadamard differentiable at F with derivative $\phi'_F(h) = -h(\xi_\alpha)/f(\xi_\alpha)$ (van der Vaart 1998, Lemma 21.3). Thus we may approximate

$$\begin{aligned} \hat{\xi}_\alpha - \xi_\alpha &= \phi(\hat{F}_n) - \phi(F) \\ &= \phi(F + (\hat{F}_n - F)) - \phi(F) \\ &\approx \phi'_F(\hat{F}_n - F) \\ &= -(\hat{F}_n(\xi_\alpha) - F(\xi_\alpha))/f(\xi_\alpha), \end{aligned}$$

so that $n^{1/2}(\hat{\xi}_\alpha - \xi_\alpha)$ is asymptotically equivalent to $-n^{1/2}(\hat{F}_n(\xi_\alpha) - F(\xi_\alpha))/f(\xi_\alpha)$.

From this we conclude that for an i.i.d. sample, for which $\hat{F}_n(\xi_\alpha)$ is distributed like n^{-1} times a binomial random variable with parameters n and α, $n^{1/2}(\hat{\xi}_\alpha - \xi_\alpha)$ converges in distribution to a Normal law with zero mean and variance $\alpha(1-\alpha)/f(\xi_\alpha)^2$. This result is often proved using entirely different techniques (e.g. Ferguson 1996, Chapter 13). The advantage of the present approach is that it immediately extends to dependent samples. Indeed, if the Z_k are weakly dependent such that the sequence $(I\{Z_k \le \xi_\alpha\} - \alpha)$ of centered indicator functions satisfies a central limit theorem with asymptotic variance $C_\alpha \alpha(1-\alpha)/n$, then $n^{1/2}(\hat{\xi}_\alpha - \xi_\alpha)$ again has a limiting Normal distribution, with zero mean and variance $C_\alpha \alpha(1-\alpha)/f(\xi_\alpha)^2$. Here C_α is thus a constant accounting for the

amount of extra variance caused by the dependence in $\{Z_k\}$, relative to the i.i.d. case.

Now return to the i.i.d. setting and assume that we wish to estimate the quantile ξ_α with a standard error of at most say 5% relative to the actual value of $\xi_\alpha - m_F$, where m_F is the mean of F. That is, we should choose n large enough that the standard deviation $\sqrt{\alpha(1-\alpha)/n}/f(\xi_\alpha) \leq 0.05(\xi_\alpha - m_F)$. Assume also that F is Normal, or at least approximately so. Then $\xi_\alpha = m_F + z_\alpha \sigma_F$ where σ_F^2 is the variance of F and z_α is the α-quantile of the standard Normal distribution, and $f(\xi_\alpha) = 1/(\sqrt{2\pi}\sigma_F) \times e^{-z_\alpha^2/2}$. The inequality above thus becomes $\sqrt{2\pi\alpha(1-\alpha)/n}e^{z_\alpha^2/2} \leq 0.05 z_\alpha$, or $n \geq 400\, z_\alpha^{-2} e^{z_\alpha^2} 2\pi\alpha(1-\alpha)$. For $\alpha = 0.95$ we find $z_\alpha = 1.64$ and $n \geq 661$. Finally, if the Z_k are dependent as above these calculation look the same, except that an additional factor C_α multiplies the bound on n.

References

Aitkin, M. (2001). "Likelihood and Bayesian analysis of mixtures." *Statistical Modelling*, 1: 287–304. 677

Albertson, D. G., Collins, C., McCormick, F., and Gray, J. W. (2003). "Chromosome aberrations in solid tumors." *Nature Genetics*, 34: 369–376. 677

Albertson, D. G. and Pinkel, D. (2003). "Genomic microarrays in human genetic disease and cancer." *Human Molecular Genetics*, 12: R145–R152. 678

Ball, F. G., Cai, Y., Kadane, J. B., and O'Hagan, A. (1999). "Bayesian inference for ion-channel gating mechanisms directly from single-channel recordings, using Markov chain Monte Carlo." *Proceedings of the Royal Society of London - Series A*, 455: 2879–2932. 681

Baum, L. E. and Petrie, T. (1966). "Statistical inference for probabilistic functions of finite state Markov chains." *Annals of Mathematical Statistics*, 37: 1554–1563. 660

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. A. (1970). "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains." *Annals of Mathematical Statistics*, 41: 164–171. 660

Bhar, R. and Hamori, S. (2004). *Hidden Markov models: Applications to Financial Economics*. Boston, MA: Kluwer Academic Publishers. 659

Bickel, P. J., Ritov, Y., and Rydén, T. (1998). "Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models." *Annals of Statistics*, 26: 1614–1635. 667

Bruder, C. E. G., Piotrowski, A., Gijsbers, A. A. C. J., Andersson, R., Erickson, S., Diaz de Ståhl, T., Menzel, U., Sandgren, J., von Tell, D., Poplawski, A., Crowley, M., Crasto, C., Partridge, E. C., Tiwari, H., Allison, D. B., Komorowski, J., van Ommen, G.-J. B., Boomsma, D. I., Pedersen, N. L., den Dunnen, J. T., Wirdefeldt, K., and Dumanski, J. P. (2008). "Phenotypically concordant and discordant monozygotic

twins display different DNA copy-number-variation profiles." *American Journal of Human Genetics*, 82: 763–771. 678

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. New York: Springer. 659, 661, 662, 674, 6

Carlstein, E., Do, K.-A., Hall, P., Hesterberg, T., and Künsch, H. R. (1998). "Matched-block bootstrap for dependent data." *Bernoulli*, 4: 305–328. 668

Chib, S. (1996). "Calculating posterior distributions and modal estimates in Markov mixture models." *Journal of Econometrics*, 75: 79–97. 663

Chopin, N. (2007). "Inference and model choice for sequentially ordered hidden Markov models." *Journal of the Royal Statistical Society - Series B*, 69: 269–284. 672

Damien, P., Wakefield, J., and Walker, S. (1999). "Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables." *Journal of the Royal Statistical Society - Series B*, 61: 331–344. 674

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). "Maximum likelihood from incomplete data via the EM algorithm (with discussion)." *Journal of the Royal Statistical Society - Series B*, 39: 1–38. 660

Ferguson, T. S. (1973). "A Bayesian analysis of some nonparametric problems." *Annals of Statistics*, 1: 209–230. 672

— (1996). *A Course in Large Sample Theory*. London: Chapman & Hall. 684

Fridyland, J., Snijders, A. M., Pinkel, D., Albertson, D. G., and Jain, A. N. (2004). "Hidden Markov models approach to the analysis of array CGH data." *Journal of Multivariate Analysis*, 90: 132–153. 678

Frühwirth-Schnatter, S. (2001). "Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models." *Journal of the American Statistical Association*, 96: 194–209. 667

— (2004). "Estimating marginal likelihoods for mixture and Markov switching models using bridge sampling techniques." *Econometrics Journal*, 7: 143–167. 670, 671

— (2006). *Finite Mixture and Markov Switching Models*. New York: Springer. 659, 670, 671, 677

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82: 711–732. 671

Hansen, B. (1992). "The likelihood ratio test under non-standard conditions: Testing the Markov switching model of GNP." *Journal of Applied Econometrics*, 7: S61–S82. 670

Jasra, A., Holmes, C. C., and Stephens, D. A. (2005). "Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modelling." *Statistical Science*, 20: 50–67. 667, 5

Jelinek, F. (1998). *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press. 659

Koski, T. (2001). *Hidden Markov Models for Bioinformatics*. Dordrecht: Kluwer Academic Publishers. 659

Krolzig, H.-M. (1997). *Markov-switching vector autoregressions. Modelling, statistical inference, and application to business cycle analysis*, volume 454 of *Lecture Notes in Economics and Mathematical Systems*. Berlin: Springer-Verlag. 660

Levinson, S. E., Rabiner, L. R., and Sondhi, M. M. (1983). "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition." *Bell System Techical Journal*, 62: 1035–1074. 659

Lystig, T. and Hughes, J. P. (2002). "Exact computation of the observed information matrix for hidden Markov models." *Journal of Computational and Graphical Statistics*, 11: 678–689. 667

MacDonald, I. L. and Zucchini, W. (1997). *Hidden Markov and Other Models for Discrete-valued Time Series*. London: Chapman & Hall. 659

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. New York: Wiley. 670

Meilijson, I. (1989). "A fast improvement to the EM algorithm on its own terms." *Journal of the Royal Statistical Society - Series B*, 51: 127–138. 664, 665

Meng, X.-L. and Wong, W. H. (1996). "Simulating ratios of normalizing constants via a simple identity: A theoretical exploration." *Statistica Sinica*, 6: 831–860. 670, 671

Papaspiliopoulos, O., Roberts, G., and Sköld, M. (2007). "A general framework for the parametrization of hierarchical models." *Statistical Science*, 22: 59–73. 666

Petrie, T. (1969). "Probabilistic functions of finite state Markov chains." *Annals of Mathematical Statistics*, 40: 97–115. 660

Picard, F., Robin, S., Lavielle, M., Vaisse, C., and J., D. J. (2005). "A statistical approach for array CGH data analysis." *BMC Bioinformatics*, 6: 27. 678

Raj, B. (2002). "Asymmetry of business cycles: the Markov-switching approach." In Ullah, A., Wan, A., and Chaturvedi, A. (eds.), *Handbook of Applied Econometrics and Statistical Inference*, 687–710. New York: Marcel Dekker. 659

Richardson, S. and Green, P. (1997). "On Bayesian analysis of mixtures with an unknown number of components." *Journal of the Royal Statistical Society - Series B*, 59: 731–792. 662, 671, 674

Robert, C. P., Rydén, T., and Titterington, D. M. (2000). "Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method." *Journal of the Royal Statistical Society - Series B*, 62: 57–75. 672, 674, 675

Roberts, W. J. J. and Ephraim, Y. (2008). "An EM algorithm for ion-channel current estimation." *IEEE Transactions on Signal Processing*, 56: 26–33. 680

Rydén, T., Teräsvirta, T., and Åsbrink, S. (1998). "Stylized facts of daily return series and the hidden Markov model of absolute returns." *Journal of Applied Econometrics*, 13: 217–244. 668, 672, 673

Scott, S. L. (2002). "Bayesian methods for hidden Markov models: Recursive computing in the 21st century." *Journal of the American Statistical Association*, 97: 337–351. 661, 676

Snijders, A. M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., Hamilton, G., Hindle, A. K., Huey, B., Kimura, K., Law, S., Myambo, K., Palmer, J., Ylstra, B., Yue, J. P., Gray, J. W., Jain, A. N., Pinkel, D., and Albertson, D. G. (2001). "Assembly of microarrays for genome-wide measurement of DNA copy number." *Nature Genetics*, 29: 263–264. 678

Stjernqvist, S., Rydén, T., Sköld, M., and Staaf, J. (2007). "Continuous-index hidden Markov modelling of CgH copy number data." *Bioinformatics*, 23: 1006–1014. 679, 680

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). "Hierarchical Dirichlet processes." *Journal of the American Statistical Association*, 101: 1566–1581. 671

van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge: Cambridge University Press. 684

Wu, J. C. F. (1982). "On the convergence properties of the EM algorithm." *Annals of Statistics*, 11: 95–103. 664

Acknowledgments

Programming, simulations and analyses underlying the results presented in Section 4 were skillfully carried out by postgraduate student Susann Stjernqvist. The author thanks the reviewer, the Associate Editor and the Editor who all provided valuable comments that greatly improved the contents and presentation of this paper.