TI 2012-096/III
Tinbergen Institute Discussion Paper

The *R* Package MitISEM: Mixture of Student-t Distributions using Importance Sampling Weighted Expectation Maximization for Efficient and Robust Simulation

Nalan Basturk[1,3]
Lennart Hoogerheide[2]
Anne Opschoor[1]
Herman K. van Dijk[1,2,3]

[1] *Erasmus School of Economics, Erasmus University Rotterdam, and Tinbergen Institute;*
[2] *Faculty of Economics and Business Administration, VU University Amsterdam, and Tinbergen Institute;*
[3] *RCEA.*

Tinbergen Institute is the graduate school and research institute in economics of Erasmus University Rotterdam, the University of Amsterdam and VU University Amsterdam.

More TI discussion papers can be downloaded at http://www.tinbergen.nl

Tinbergen Institute has two locations:

Tinbergen Institute Amsterdam
Gustav Mahlerplein 117
1082 MS Amsterdam
The Netherlands
Tel.: +31(0)20 525 1600

Tinbergen Institute Rotterdam
Burg. Oudlaan 50
3062 PA Rotterdam
The Netherlands
Tel.: +31(0)10 408 8900
Fax: +31(0)10 408 9031

Duisenberg school of finance is a collaboration of the Dutch financial sector and universities, with the ambition to support innovative research and offer top quality academic education in core areas of finance.

DSF research papers can be downloaded at: http://www.dsf.nl/

Duisenberg school of finance
Gustav Mahlerplein 117
1082 MS Amsterdam
The Netherlands
Tel.: +31(0)20 525 8579

The *R* package MitISEM: Mixture of Student-t Distributions using Importance Sampling weighted Expectation Maximization for Efficient and Robust Simulation

Nalan Baştürk
Erasmus University
Rotterdam
& RCEA

Lennart Hoogerheide
Vrije Universiteit
Amsterdam
& Tinbergen Institute

Anne Opschoor
Erasmus University
Rotterdam
& Tinbergen Institute

Herman K. van Dijk
Erasmus University
Rotterdam
& Vrije Universiteit
Amsterdam
& Tinbergen Institute
& RCEA

Abstract

This paper presents the *R* package **MitISEM**, which provides an automatic and flexible method to approximate a non-elliptical target density using adaptive mixtures of Student-t densities, where only a kernel of the target density is required. The approximation can be used as a candidate density in Importance Sampling or Metropolis Hastings methods for Bayesian inference on model parameters and probabilities. The package provides also an extended MitISEM algorithm, 'sequential MitISEM', which substantially decreases the computational time when the target density has to be approximated for increasing data samples. This occurs when the posterior distribution is updated with new observations and/or when one computes model probabilities using predictive likelihoods. We illustrate the MitISEM algorithm using three canonical statistical and econometric models that are characterized by several types of non-elliptical posterior shapes and that describe well-known data patterns in econometrics and finance. We show that the candidate distribution obtained by MitISEM outperforms those obtained by 'naive' approximations in terms of numerical efficiency. Further, the MitISEM approach can be used for Bayesian model comparison, using the predictive likelihoods.

Keywords: finite mixtures, Student-t distributions, Importance Sampling, MCMC, Metropolis-Hastings algorithm, Expectation Maximization, Bayesian inference, R software.

1. Introduction

There exist classes of statistical and econometric models where the joint and marginal posterior distributions of the parameters have unknown analytical properties and non-elliptical Bayesian Highest Posterior Density (HPD) credible sets, see e.g. Berger (1985). In such cases it is not trivial to perform inference on the joint posterior distribution. Accurate estimation of such non-elliptical posterior distributions, e.g. with multimodal or skewed shapes, may be very important for the measurement of uncertainty in forecasts and policy measures and this phenomenon of non-elliptical shapes occurs frequently in empirical econometric analysis.

An important example is the class of instrumental variable models with weak instruments like some of the income-education models relevant for government agencies responsible for compulsory schooling laws. A second example is the class of mixture processes where one component is nearly non-identified since it corresponds to very few observations, which may occur in financial models with data that exhibit time varying volatility patterns and heavy tails. A detailed analysis of this literature is beyond the scope of the present paper. For more details on such econometric models we refer to Imbens and Angrist (1994) and Bos, Mahieu, and Van Dijk (2000) and the references cited there. An important issue is that it is non-trivial to simulate (pseudo-) random draws from such a non-elliptical joint posterior distribution in a numerically efficient way. Even if simulation from the conditional distributions is relatively easy, for example if application of the well-known Gibbs sampler is feasible, multi-modality and/or high correlations between the model parameters may cause the Gibbs sampler to converge extremely slowly or even yield erroneous results with a given sample of posterior draws.

This paper presents the *R* package **MitISEM**, which provides an automatic and flexible method to approximate a target density using adaptive mixtures of Student-t densities. The multivariate target density can be non-elliptical and only a kernel of the target density is required for the MitISEM method. The target density is usually a posterior or a predictive density in Bayesian inference. The approximation can – in a next stage – be used as a candidate density in Importance Sampling or Metropolis Hastings methods, in particular for Bayesian inference on model parameters and model probabilities. The MitISEM method has been introduced by Hoogerheide, Opschoor, and Van Dijk (2012) and it has been shown that the method provides substantial gains in computational efficiency in Bayesian estimation. The MitISEM method makes use of convex combinations of densities, and the approximation properties of such density combinations have been analyzed extensively in the literature. For instance Zeevi and Meir (1997) show that under certain conditions any density function may be approximated to arbitrary accuracy by a convex combination of 'basis' densities. The class of mixture of Student-t densities falls within this framework.

The basic algorithm in **MitISEM** iterates over importance weighted Expectation Maximization steps in order to efficiently construct a mixture of Student-t densities to achieve an accurate approximation of the target distribution. Starting with a single adapted Student-t density, a new mixture component is added iteratively until the required approximation is reached. At each iteration, parameters of each mixture component – including the mode, scale, degrees of freedom and mixing probability – are optimized such that the Kullback-Leibler divergence between target and mixture is minimized. The constructed mixture can then be used as a candidate density for an efficient and robust application of either Importance Sampling (IS) or the independence chain Metropolis-Hastings (MH) method.

We illustrate the MitISEM algorithm using a well-known statistical workhorse model from Gelman and Meng (1991) that is characterized by a very non-elliptical joint distribution while the conditional distributions are normal. We also use two classes of canonical econometric models: the Instrumental Variables (IV) regression model and the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model. Both classes of models yield non-elliptical posterior and predictive distributions for posterior and predictive densities. Furthermore, we show that the MitISEM approach can be used for the evaluation of model probabilities from predictive likelihoods, which are useful for Bayesian model comparison. We also introduce an R program for an adapted MitISEM algorithm as in Hoogerheide, Opschoor, and Van Dijk (2012), named 'sequential MitISEM', which substantially decreases the computational time required for the candidate density optimization, when the posterior distribution is updated using new observations or when one computes model probabilities with predictive likelihoods.

The remainder of this paper is organized as follows: Section 2 discusses the basic an sequential MitISEM methods, and briefly addresses the Importance Sampling and Expectation Maximization algorithms. Section 3 presents applications of the algorithm to several model structures and datasets. Section 4 concludes.

2. Mixture of Student-t Distributions by IS weighted EM (MitISEM)

The mixture of Student-t distributions constructed by Importance Sampling weighted Expectation Maximization is based on an iterative construction of a mixture of Student-t distributions (Hoogerheide, Opschoor, and Van Dijk 2012). The algorithm provides an automatic and flexible method to construct a proposal density minimizing the Kullback-Leibler divergence (or Cross-entropy distance) (Kullback and Leibler 1951) between two densities: the so-called target density and the proposal density. Each new Student-t component in the proposal density covers the areas of the target density that are not well-approximated by the previous proposal density. Parameters of the new Student-t component are quickly obtained using the Importance Sampling weighted Expectation Maximization method.

Henceforth we use the notation $f(\theta)$ for the target density kernel of θ, the k-dimensional vector of interest. $f(\theta)$ can be a posterior density kernel of model parameters or a density kernel of data. In the former case we simplify the notation and remove the conditioning on data for convenience. $g(\theta)$ is the candidate/proposal density, a mixture of H Student-t densities such that:

$$g(\theta) = g(\theta|\zeta) = \sum_{h=1}^{H} \eta_h \, t_k(\theta|\mu_h, \Sigma_h, \nu_h), \tag{1}$$

where ζ is the set of modes μ_h, scale matrices Σ_h, degrees of freedom ν_h, and mixing probabilities η_h $(h = 1, \ldots, H)$ of the k-dimensional Student-t components with density:

$$t_k(\theta|\mu_h, \Sigma_h, \nu_h) = \frac{\Gamma\left(\frac{\nu_h+k}{2}\right)}{\Gamma\left(\frac{\nu_h}{2}\right)(\pi\nu_h)^{k/2}} |\Sigma_h|^{-1/2} \left(1 + \frac{(\theta-\mu_h)'\Sigma_h^{-1}(\theta-\mu_h)}{\nu_h}\right)^{-(k+\nu_h)/2}. \tag{2}$$

Here Σ_h is positive definite, $\eta_h \geq 0$ and $\sum_{h=1}^{H} \eta_h = 1$. We further restrict ν_h such that $\nu_h > 0$, but the user can adapt this lower bound.

The closest approach to MitISEM in the literature is the AdMit method (Hoogerheide, Kaashoek, and Van Dijk 2007b), implemented in Ardia, Hoogerheide, and Van Dijk (2009a,b). Both methods rely on the iterative construction of a mixture of Student-t densities as the candidate density, but there are three substantial differences between these methods. First, AdMit aims at minimizing the variance of the IS estimator, or the variance of the IS weights directly, whereas MitISEM aims at this goal indirectly by minimizing the Kullback-Leibler divergence. As a result, AdMit optimizes the mixture component weights using a non-linear optimization procedure that requires considerable computational effort. Second, in the AdMit method, means and scale matrices of the candidate components are chosen heuristically and are never updated when additional components are added to the mixture, whereas in MitISEM all mixture parameters are optimized jointly by means of the relatively quick EM algorithm. This implies a large reduction of the computing time in the approximation procedure, and is expected to lead to a better candidate in most applications. Third, as shown in Hoogerheide, Opschoor, and Van Dijk (2012), AdMit requires the joint target density kernel, whereas MitISEM requires candidate draws and importance weights. This implies that AdMit can not be applied partially to the marginal and conditional posterior distributions of subsets of parameters, whereas MitISEM can be used to approximate a marginal density of which no kernel is explicitly available.

2.1. Background on Importance Sampling

Importance Sampling (Hammersley and Handscomb 1975; Kloek and Van Dijk 1978) is a general method for estimating expectations of a function $h(\theta)$ of parameter θ where the probability density function of θ is possibly non-standard. Given a density kernel $f(\theta)$ for θ, Importance Sampling is based on draws from a candidate density $g(\theta)$ which is easy to simulate from, instead of direct simulations from $f(\theta)$, and it is a reasonable approximation to $f(\theta)$. The draws from the candidate density are *weighted* according to the Importance Sampling (IS) weights. For a consistent estimator of the expectation of the function of θ, $E(h(\theta))$, the candidate should cover the whole domain of θ values with $f(\theta) > 0$, and the finite sample properties of the estimator improve if $g(\theta)$ is a *good* approximation to the target kernel (Van Dijk 1984; Van Dijk, Hop, and Louter 1987; Geweke 1989; Hop and Van Dijk 1992). IS weights for parameter draws $\tilde{\theta}$ from $g(\theta)$ are calculated as:

$$\tilde{W}(\tilde{\theta}) = f(\tilde{\theta})/g(\tilde{\theta}), \tag{3}$$

i.e. draws with highest IS weights correspond to the region of the target which is covered relatively too little by the candidate density.

Cappé, Douc, Guillin, Marin, and Robert (2008) note that there is a renewed interest in Importance Sampling, due to the possibility of parallel processing implementation. Numerical efficiency in sampling methods is not only related to the efficient sample size or relative numerical efficiency, but also to the possibility to perform the simulation process in a parallel fashion. Unlike alternative methods such as the random walk Metropolis Hastings or the Gibbs Sampler, Importance Sampling makes use of independent draws from the candidate density, which in turn can be obtained from multiple core machines or computer clusters. See Durham and Geweke (2011) for a very novel approach. We also comment on this possibility in section 4.

2.2. Background on the Expectation Maximization Algorithm

The EM algorithm is a method (Dempster, Laird, and Rubin 1977) to achieve the Maximum Likelihood estimates of parameters θ in models with incomplete data or latent variables. An example of the latter case is the finite mixture model. For the use of the EM algorithm on finite mixture models, we refer to e.g. McLachlan and Peel (2000); McLachlan and Krishnan (2008).

If the latent variables would be observable, the computation of the Maximum Likelihood estimate of θ would be relatively straightforward, depending on the degree of nonlinearity of the first order conditions. The idea behind EM is to take the expectation of the objective function, in most cases the log-likelihood function, with respect to the latent variables. The expectation of the log-likelihood function is then maximized with respect to the model parameters. In most models, expectations of the latent variables depend on the model parameters θ, hence the two steps are repeated until convergence.

As shown in Hoogerheide, Opschoor, and Van Dijk (2012), in the MitISEM approach, the objective function corresponds to the logarithm of the candidate density $g(.|\zeta)$ evaluated at a set of draws θ^i from a previous candidate $g_0(\theta)$, where each candidate value $\log g(\theta^i|\zeta)$ is weighted by the Importance Sampling weights $W^i \equiv f(\theta^i)/g_0(\theta^i)$ of each draw θ^i from the previous candidate $g_0(\theta)$:

$$\frac{1}{N}\sum_{i=1}^{N} W^i \log g(\theta^i|\zeta)$$

where $g(.|\zeta)$ is the mixture of Student-t densities to be optimally chosen. Note that we use an *importance weighted* EM algorithm, because we have candidate draws and corresponding importance weights, rather than draws from the posterior density.

The mixture of Student-t densities (1) for θ^i is equivalent with the specification

$$\theta^i \sim N(\mu_h, w_h^i \Sigma_h) \qquad \text{if} \qquad z_h^i = 1,$$

where z^i is a latent H-dimensional vector indicating from which Student-t component the 'observation' θ^i stems: if θ^i stems from component h, then $z_h^i = 1$, $z_j^i = 0$ for $j \neq h$; $\Pr[z^i = e_h] = \eta_h$ with e_h the h-th column of the identity matrix; w_h^i has the Inverse-Gamma distribution $IG(\nu_h/2, \nu_h/2)$. For a more extensive explanation of this mixture, see e.g. Peel and McLachlan (2000).

2.3. The IS weighted EM algorithm

Here we stress that in literature the EM algorithm is typically used to find the optimal values of model parameters that maximize the log-likelihood for a given set of data. Here we could use EM to find the optimal mixture of Student-t distributions for a given set of posterior draws that were obtained by MH and draws from a previous candidate. However, the *IS-weighted* EM method has three advantages. First, we do not require a burn-in sample. Second, the use of all candidate draws (without the rejections of the MH method) helps to prevent numerical problems with estimating scale matrices of Student-t components; also draws with relatively small, but positive importance weights are helpful for this purpose. Third, the use of all candidate draws may lead to a better approximation.

As shown in Hoogerheide, Opschoor, and Van Dijk (2012), the L-th Expectation step for the mixture of Student-t distributions is achieved as follows:

$$\tilde{z}_h^i \equiv E\left[z_h^i \,\middle|\, \theta^i, \zeta = \zeta^{(L-1)}\right] = \frac{t_k(\theta^i|\mu_h, \Sigma_h, \nu_h)\,\eta_h}{\sum_{j=1}^{H} t_k(\theta^i|\mu_j, \Sigma_j, \nu_j)\,\eta_j}, \tag{4}$$

$$\widetilde{z/w}_h^{\,i} \equiv E\left[\frac{z_h^i}{w_h^i} \,\middle|\, \theta^i, \zeta = \zeta^{(L-1)}\right] = \tilde{z}_h^i\,\frac{k+\nu_h}{\rho_h^i + \nu_h}, \tag{5}$$

$$\begin{aligned}\xi_h^i &\equiv E\left[\log w_h^i \,\middle|\, \theta^i, \zeta = \zeta^{(L-1)}\right] = \\ &= \left[\log\left(\frac{\rho_h^i + \nu_h}{2}\right) - \psi\left(\frac{k+\nu_h}{2}\right)\right]\tilde{z}_h^i + \left[\log\left(\frac{\nu_h}{2}\right) - \psi\left(\frac{\nu_h}{2}\right)\right](1 - \tilde{z}_h^i),\end{aligned} \tag{6}$$

$$\delta_h^i \equiv E\left[\frac{1}{w_h^i}\middle|\theta^i, \zeta = \zeta^{(L-1)}\right] = \frac{k+\nu_h}{\rho_h^i + \nu_h}\,\tilde{z}_h^i + (1-\tilde{z}_h^i), \tag{7}$$

with $\rho_h^i \equiv (\theta^i - \mu_h)'\Sigma_h^{-1}(\theta^i - \mu_h)$, $\psi(.)$ the digamma function (the derivative of the logarithm of the gamma function $\log\Gamma(.)$), and all parameters $\mu_h, \Sigma_h, \nu_h, \eta_h$ elements of the set of candidate's parameters $\zeta^{(L-1)}$ optimized in the previous EM step $(L-1)$. Given the expectation of the latent variables in (4) to (7), parameters of each mixture component are updated using the first order conditions of the expectation of the objective function in the Maximization step:

$$\mu_h^{(L)} = \left[\sum_{i=1}^{N} W^i\ \widetilde{z/w}_h^i\right]^{-1}\left[\sum_{i=1}^{N} W^i\ \widetilde{z/w}_h^i\,\theta^i\right], \tag{8}$$

$$\hat{\Sigma}_h^{(L)} = \frac{\sum_{i=1}^{N} W^i\ \widetilde{z/w}_h^i\,(\theta^i - \mu_h^{(L)})(\theta^i - \mu_h^{(L)})'}{\sum_{i=1}^{N} W^i\ \tilde{z}_h^i}, \tag{9}$$

$$\eta_h^{(L)} = \frac{\sum_{i=1}^{N} W^i\,\tilde{z}_h^i}{\sum_{i=1}^{N} W^i}. \tag{10}$$

Further, $\nu_h^{(L)}$ is solved from the first order condition of ν_h:

$$-\psi(\nu_h/2) + \log(\nu_h/2) + 1 - \frac{\sum_{i=1}^{N} W^i\,\xi_h^i}{\sum_{i=1}^{N} W^i} - \frac{\sum_{i=1}^{N} W^i\,\delta_h^i}{\sum_{i=1}^{N} W^i} = 0. \tag{11}$$

Cappé *et al.* (2008) only update the expectations and scale structures of the Student-t distributions and not the degrees of freedom, because there is no closed-form solution for the latter. The degrees of freedom parameter ν_h during the EM procedure is optimized to obtain a better approximation of the target distribution. Furthermore, the resulting values of ν_h $(h = 1, \ldots, H)$ may provide information on the shape, e.g. kurtosis of the target distribution.

2.4. MitISEM: The basic algorithm

The Mixture of Student-t densities is constructed using the following steps:

Algorithm 1. *The MitISEM approach for obtaining an approximation to a target density:*

(0) **Initialization**: Simulate draws $\theta^1, \ldots, \theta^N$ from the naive proposal density g_{naive}. First, one obtains a Student-t distribution with a fixed degree of freedom. Second, the mode and scale matrix equal to the target distribution's mode and minus the inverse Hessian of the log-target density kernel evaluated at the mode. The mode and scale of this initial density are updated using the IS weighted EM algorithm.

(1) **Adaptation**: Estimate the target distribution's mean and covariance matrix using IS with the draws $\theta^1, \ldots, \theta^N$ from g_{naive}. Use these estimates as the mode and scale matrix of Student-t density $g_{adaptive}$. Draw a sample $\theta^1, \ldots, \theta^N$ from this adaptive Student-t distribution $g_0 = g_{adaptive}$, and compute the IS weights for this sample.

(2) Apply the **IS-weighted EM algorithm** given the latest IS weights and the drawn sample of step 1. The output consists of the new candidate density g with optimized

ζ, the set of $\mu_h, \Sigma_h, \nu_h, \eta_h$ ($h = 1, \ldots, H$). Draw a new sample $\theta^1, \ldots, \theta^N$ from this proposal density and compute corresponding IS weights.

(3) **Iterate on the number of mixture components**: Given the current mixture of H components with corresponding μ_h, Σ_h, ν_h and η_h ($h = 1, \ldots, H$), take $x\%$ of the sample $\theta^1, \ldots, \theta^N$ that correspond to the highest IS weights. Construct with these draws and IS weights a new mode μ_{H+1} and scale matrix Σ_{H+1} which are starting values for the additional component in the mixture candidate density. This choice ensures that the new component covers a region of the parameter space in which the previous candidate mixture had relatively too little probability mass. Given the latest IS weights and the drawn sample from the current mixture of H components, apply the IS-weighted EM algorithm to optimize *each* mixture component μ_h, Σ_h, ν_h and η_h with $h = 1, \ldots, H + 1$. Draw a new sample from the mixture of $H + 1$ components and compute corresponding IS weights.

(4) **Assess convergence using the IS weights** and return to step 3 unless the algorithm has converged.

In Step (0), we added a novel robustification by updating the initial proposal density using an IS weighted EM step. Step (1) can be seen as an intermediate step which quickly tries to improve the initial candidate distribution g_0. If during the EM algorithm, a scale matrix Σ_h of a Student-t component (with very small weight η_h) becomes (nearly) singular, then this h-th component is removed from the mixture.

In Step (4) convergence can be assessed by computing the relative change in Coefficient of Variation (CoV) of the IS weights, i.e. the standard deviation of the IS weights divided by their mean, as in Hoogerheide, Opschoor, and Van Dijk (2012), who use the candidate from MitISEM for Importance Sampling or the independence chain MH method. Zellner, Ando, Baştürk, Hoogerheide, and Van Dijk (2012), who use the MitISEM candidate for rejection sampling, propose an alternative criterion for the convergence of the MitISEM algorithm. They use the unconditional acceptance probability, which is a more natural and intuitive convergence criterion in this case. The default convergence in MitISEM is defined as the change of the CoV being smaller than 10%, but the user can specify convergence in terms of the acceptance probability. The convergence tolerance can also be altered by the user.

Starting values for η_{H+1} and ν_{H+1} are at each iteration set at 0.10 and 1, respectively. I.e. the new component has fat tails, and a relatively low probability ex-ante. Starting values for μ_h, Σ_h, and ν_h ($h = 1, \ldots, H$) are the optimal values in the previous mixture of H components, while η_h ($h = 1, \ldots, H$) is 0.90 times the previously optimal value. Alternative initial values for η_{H+1} and ν_{H+1} can be set by the user.

Finally, we introduce another novel robustification of the MitISEM method. With this robustification, the given number of candidate draws that is used to construct the candidate does not include draws for which the candidate density is 0. If the target density is concentrated in a restricted parameter space, for example for a mixture GARCH model, the number of 'useful' or 'effective' draws can be otherwise very smal,. especially the first steps of the MitISEM algorithm. This robust simulation is the default simulation method in the provided package, but can be disregarded by the user.

2.5. Sequential MitISEM

This section presents the algorithm for 'Sequential MitISEM', proposed by Hoogerheide, Opschoor, and Van Dijk (2012). Sequential MitISEM applies MitISEM in a sequential manner, so that the candidate distribution for posterior simulation is updated when new data become available. The alternative to this method is to repeatedly apply the basic MitISEM approach when new data become available. Such an 'ad hoc' approach, applying the whole MitISEM algorithm from scratch to achieve multiple estimates over time, can be computationally inefficient for example for daily Bayesian forecasts.

Sequential MitISEM relies on the possibility that the posterior for $y_{1:T+1} = \{y_1, \ldots, y_{T+1}\}$ is similar to the posterior for data $y_{1:T} = \{y_1, \ldots, y_T\}$. One can check if the same candidate can be used for the posterior density for the updated data, and 'recycle' the same candidate distribution if the previous candidate is a good approximation to the posterior of the updated data. Even if the 'old' candidate is not a good approximation to the updated data, it may suffice to perform an update using the IS-weighted EM algorithm, keeping the number H of Student-t components the same. If the resulting quality is still below a desired level, then one can start the MitISEM algorithm for the updated data, adding components until convergence. Note that the IS-weighted EM algorithm of MitISEM is much more suited to perform (either small or large) adaptations than the AdMit method, since all student-t components are updated.

Suppose that the MitISEM candidate is optimized for the data until time T and the data set now includes observations upto time $T + \tau$ ($\tau = 1, 2, \ldots$). Define $y_{1:T+\tau} = \{y_1, \ldots, y_{T+\tau}\}$. For the updated data until $T + \tau$ the Sequential MitISEM steps are as follows:

Algorithm 2. *The Sequential MitISEM approach for obtaining a candidate density for the posterior density for data* $y_{1:T+\tau}$*:*

(1) Compute $\text{CoV}^r_{T+\tau}$, the CoV value (Coefficient of Variation of the IS weights) that is based on the posterior density kernel for data $y_{1:T+\tau}$ and the current, *reused* candidate density.

(2) Compare $\text{CoV}^r_{T+\tau}$ with CoV_T, the CoV value for the same candidate and the posterior for data, the last time when the candidate was updated. If the change is below a certain threshold (10%), stop. Otherwise go to step (3).

(3) Run the IS-weighted EM algorithm with the current mixture of H Student-t densities as starting values. Sample from the new distribution (with the same number of components H) and compute IS weights and the corresponding $\text{CoV}^u_{T+\tau}$, the CoV value with only an EM *update*. Since the IS-weighted EM algorithm updates all mixture components, it can easily perform a useful shift of the candidate density.

(4) Compare $\text{CoV}^u_{T+\tau}$ and $\text{CoV}^r_{T+\tau}$. If the change of quality is below a certain threshold (10%), stop. Otherwise go to step (5).

(5) Iterate on the number of components until the CoV value has converged.

Note that the change in CoV value can be substantial if the new observation y_{T+1} is an outlier. Steps (3) and (5) in that case will typically be required. A Student-t component is deleted from the mixture if the weight of this component is minimal. Hence the number of Student-t components is not necessarily monotonically increasing over time. The default tolerance for the required mixing probability is 10%, but this can be altered by the user. Further, in step (2) $\text{CoV}^{r}_{T+\tau}$ is compared with CoV_T rather than the CoV for the posterior at time $y_{T+\tau-1}$, since in the latter case a series of small increases of the CoV may eventually lead to a much worse candidate density.

3. Applications

In the following subsections, we apply the MitISEM and the sequential MitISEM methods to non-elliptical distributions arising from different canonical model structures. We first analyze the conditionally normal distribution of Gelman and Meng (1991), which can have non-elliptical shapes depending on the specific values of the function parameters. Second, we consider a standard GARCH model and a mixture of GARCH model (for S&P 500 data), which are classes of models extensively used in financial practice. Third, we consider an instrumental variables (IV) model (for income-education data). Both GARCH and IV models yield non-elliptical distributions for posterior and predictive densities. For this reason, obtaining a good candidate density, for example for Importance Sampling or the independence chain Metropolis Hastings method, is crucial for Bayesian estimation of the model parameters as well as model probabilities.

We summarize the application of the *R* package **MitISEM**, and compare the performance of the MitISEM method with a single, relatively 'naive' Student-t candidate density. The 'naive' density is still an adapted density, obtained by the IS weighted EM algorithm, with degrees of freedom set as 1. The fat tails of the 'naive' candidate distribution (due to the low degrees of freedom parameter 1) reduce the probability that relevant parts of the target distribution are not covered by the 'naive' candidate. Still, despite the optimized mode and scale, this density is expected to lead to a relatively poor approximation in particular for multimodal target densities. In section 3.4, we also compare the performance of the AdMit and MitISEM methods in terms of the approximation accuracy and computing time.

3.1. The Gelman-Meng distribution with banana shape

In this section we illustrate the MitISEM approach with a non-elliptical, bivariate distribution proposed by Gelman and Meng (1991). The target density kernel is:

$$f\left(x_1, x_2\right) = \exp\left\{-0.5\left(Ax_1^2 + x_1^2 + x_2^2 - 2Bx_1x_2 - 2C_1x_1 - 2C_2x_2\right)\right\}, \qquad (12)$$

where $(x_1, x_2)'$ plays the role of the vector of interest θ.

In order to obtain the MitISEM approximation to the density $f(x_1, x_2)$ one first defines the target density kernel (see also Ardia, Hoogerheide, and Van Dijk (2009b)) and an initial point for the optimization in step (0):

```
> set.seed(1111)
>  GelmanMeng <- function(x, A = 1, B = 0, C1 = 3, C2 = 3, log = TRUE){
+     if (is.vector(x))
+       x <- matrix(x, nrow = 1)
+     r <- -0.5 * (A*x[,1]^2*x[,2]^2 + x[,1]^2 + x[,2]^2 - 2*B*x[,1]*x[,2] -
+          2*C1*x[,1] - 2*C2*x[,2])
+     if (!log)
+       r <- exp(r)
+     as.vector(r)
+   }
>   mu0<-c(3,4)
>   App.GM <- MitISEM(KERNEL=GelmanMeng,mu0=mu0)
```

The output of the function `MitISEM` is a list. The first component is `CV`, a vector containing the coefficient of variation at each step of the adaptive fitting procedure. The second component is `mit`, a list consisting of the modes (`mu`), scale matrices (`Sigma`), degrees of freedom parameters (`df`) and mixing probabilities (`p`) of the mixture of Student-t distributions constructed by MitISEM. The third component is `summary`, a data frame containing information on the adaptive fitting procedure, which will be explained in the GARCH example.

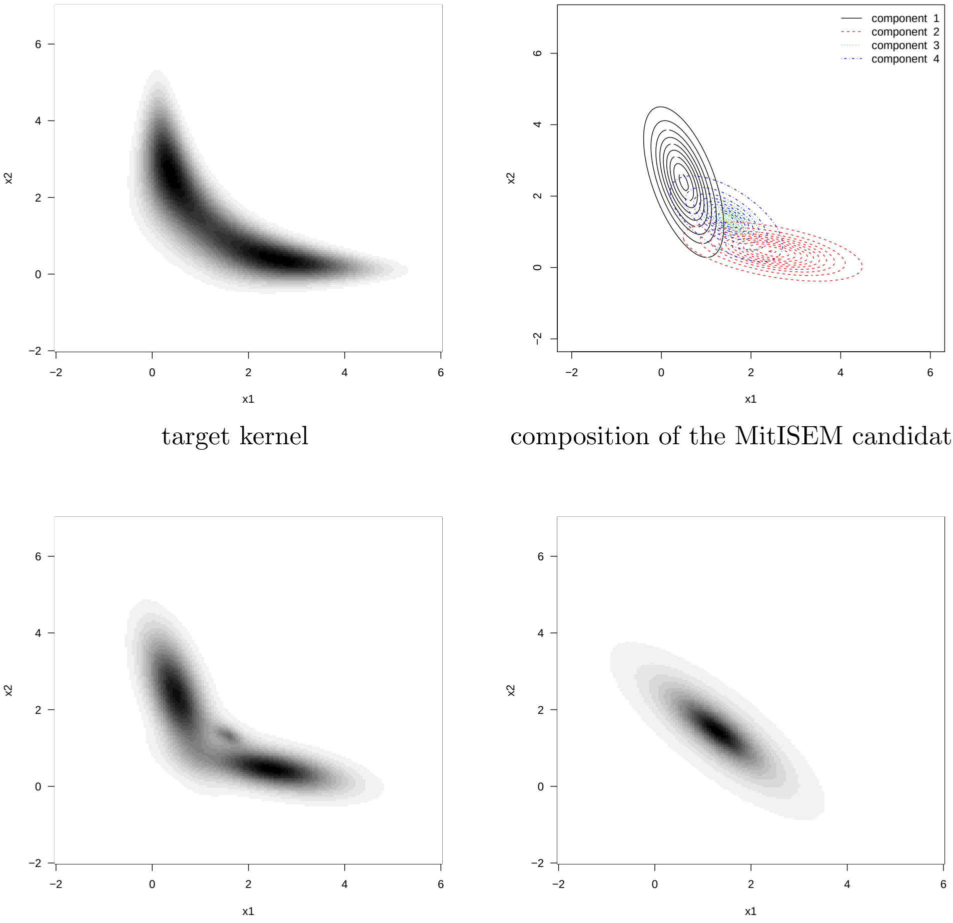
Similarly, the 'naive' approximation results are obtained by restricting the candidate density in the MitISEM approximation to have a single multivariate Student-t component where the degrees of freedom parameter is 1 by default and it is not optimized in the MitISEM algorithm:

```
>   control   <- list(optim.df=FALSE,Hmax=1)
>   app.Naive <- MitISEM(KERNEL=GelmanMeng,mu0=mu0,control=control)
```

After obtaining the MitISEM approximation, the Student-t components of the obtained candidate can be plotted as follows:

```
>   mit <- App.GM$mit
>   x1 <- seq(-2,6,0.05)
>   x2 <- seq(-2,7,0.05)
>   H <- length(mit$p)
>   Mitcontour <- function(x1,x2,mit,log=FALSE){
+     dMit(cbind(x1,x2),mit=mit,log=log)
+   }
>   for (h in 1:H){
+     mit.h <- mapply(function(x)(as.matrix(x)[h,]),mit,SIMPLIFY=FALSE)
+     mit.h$mu = matrix(mit.h$mu,nrow=1)
+     mit.h$Sigma = matrix(mit.h$Sigma,nrow=1)
+     mit.h$p = 1
+     z <- outer(x1,x2,FUN=Mitcontour,mit=mit.h)
+     contour(x1,x2,z,col=h,lty=h,labels="",add=(h!=1),xlab="x1",ylab="x2",
+         main="MitISEM approximation")
+   }
>   legend("topright",paste("component ",1:H),lty=1:H,col=1:H)
```

We first set $A = 1, B = 0, C_1 = C_2 = 3$ in equation (12). This selection of parameters in (12) leads to a non-elliptical *banana shape* in the target kernel, as shown in the top-left panel in Figure 1. We compare the performance of the MitISEM approach with a 'naive' density. For both approximations we use $N = 10^4$ draws to form the mixture components. Figure 1 shows the target density kernel and approximations by the naive and MitISEM approximations, the computational time and CoV measures for both approximations. The naive Student-t density captures only one mode of the target distribution while the MitISEM approximation captures the *banana shape* in the target kernel, with 4 components. The accuracy measure, the coefficient of variation (CoV) of the importance weights, is substantially different for the two methods: the CoV is more than four times lower for the MitISEM candidate.



target kernel

composition of the MitISEM candidate

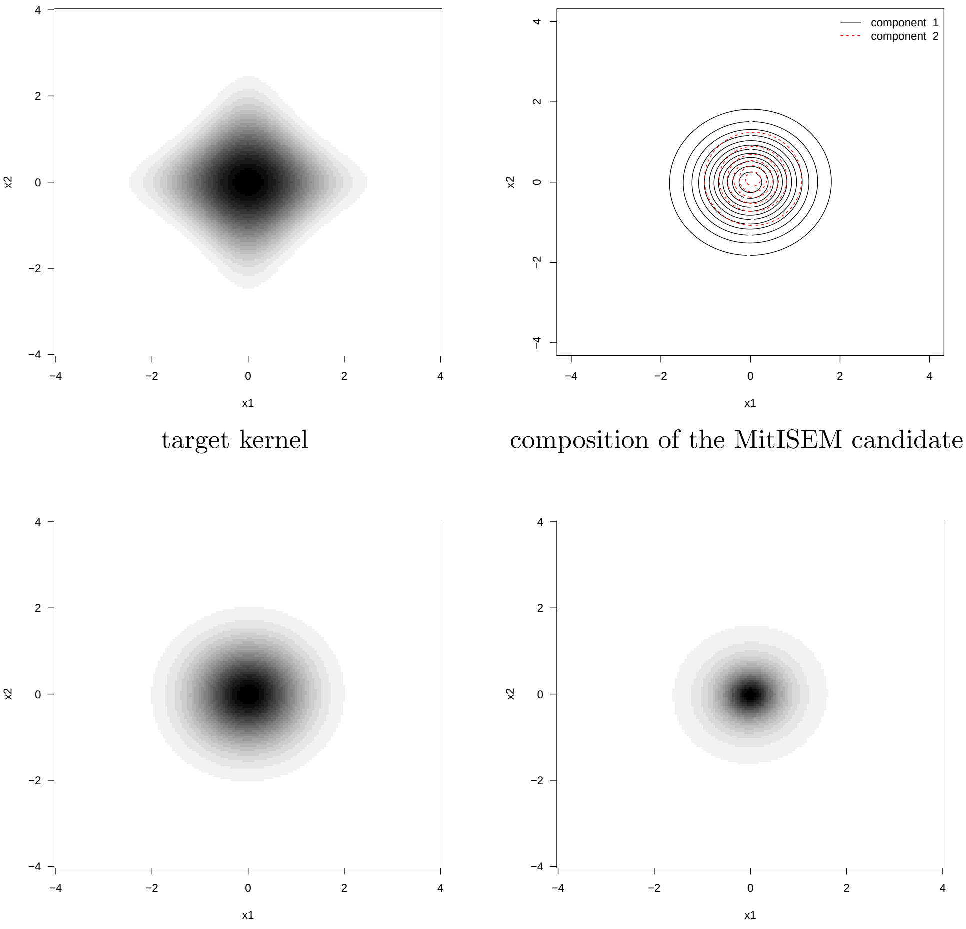
MitISEM candidate (4 components)
CoV: 0.34, time: 29.02 seconds

Naive candidate
CoV: 1.50, time: 0.53 seconds

Figure 1: Bimodal target density kernel, approximation by the naive Student-t density (achieved by Step 0 and Step 1 of the MitISEM algorithm), and optimal MitISEM candidate for the Gelman & Meng distribution with $A = 1, B = 0, C_1 = C_2 = 3$.

3.2. The Gelman-Meng distribution with diamond shape

In this section we set $A = 1, B = C_1 = C_2 = 0$ in the Gelman Meng kernel in equation (12), a choice that was analyzed in Hoogerheide, Kaashoek, and Van Dijk (2003). This choice of parameters leads to a *diamond shape* in the target density kernel as shown in the top-left graph in Figure 2.



target kernel

composition of the MitISEM candidate

MitISEM candidate (2 components)
CoV: 0.32, time: 5.87 seconds

naive candidate
CoV: 0.78, time: 0.50 seconds

Figure 2: Posterior kernel, approximation by the naive Student-t density and the optimal MitISEM candidate for the Gelman & Meng distribution with $A = 1, B = C_1 = C_2 = 0$.

We compare the precision of the naive and MitISEM approximations of the target density. Figure 2 presents these approximations of the target density. Similar to the previous case, the MitISEM approximation is more precise. This example shows that also in case of a target distribution that is relatively close to an elliptical distribution, the MitISEM method can be useful. Here the MitISEM algorithm stops quickly with adding Student-t components and still provides a substantially better approximation of the target density than the alternative approaches.

3.3. The posterior distribution of a GARCH(1,1) model

In this subsection the MitISEM approach is applied to the posterior density of a GARCH (1,1) model (Bollerslev 1986). The standard GARCH model and its extensions may adequately capture changing volatility patterns, but the likelihood function, hence the posterior density under an uninformative prior may have non-elliptical shapes (Zivot 2009).

For the applications of GARCH models we use the *S&P* 500 index percentage log-returns (100 times the change of the logarithm of the closing price) from January 2 1998 to December 26 2002. Figure 3 shows the returns data and their histogram.[1] These data are characterized by changing volatility patterns as well as fat tails. For this reason, several extensions of the standard GARCH models are proposed to capture such data patterns. We first illustrate the use of the MitISEM approach for the Bayesian estimation of the standard GARCH model for this data. An extended GARCH model, possibly leading to relatively more irregular posterior densities, is considered in the next subsection.

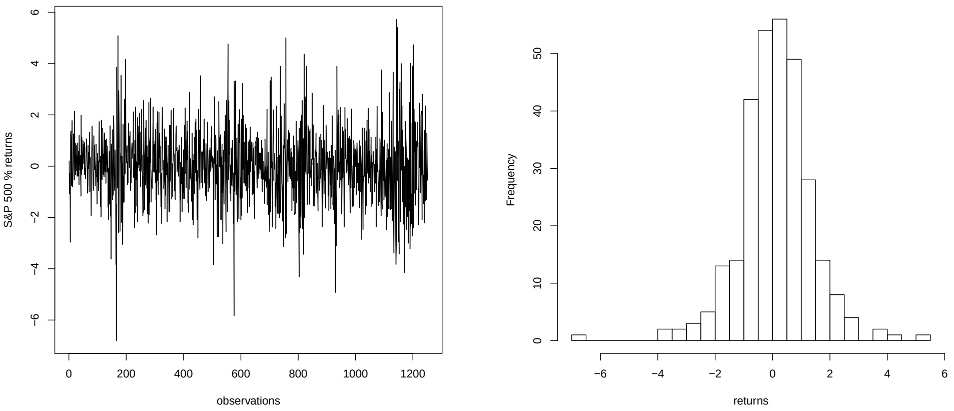


Figure 3: Daily log-returns of the S&P 500 index for the period from January 2 1998 to December 26 2002 .

The standard GARCH(1,1) model for a time series y_t $(t = 1, 2, \ldots, T)$ is given by

$$\begin{aligned} y_t &= \mu + \sqrt{h_t}\,\varepsilon_t, & (13)\\ h_t &= \omega + \alpha(y_{t-1} - \mu)^2 + \beta h_{t-1}, & (14)\\ \varepsilon_t &\sim N(0,1) \text{ i.i.d.} & (15)\end{aligned}$$

with h_t the conditional variance of y_t given the information set $I_{t-1} = \{y_{t-1}, y_{t-2}, y_{t-3}, \ldots\}$. In addition, h_0 is treated as a known constant, set as the sample variance of the time series y_t, which will consist of daily stock index (log) returns in our paper.

We restrict $\omega > 0, \alpha \geq 0$ and $\beta \geq 0$ to ensure positivity of h_t. We specify flat priors for the model parameters. Moreover, we truncate ω and μ such that these have proper (non-informative) priors. For the $k = 4$ dimensional parameter vector $\theta = (\mu, \omega, \alpha, \beta)$, we have a uniform prior on $[-1, 1] \times (0, 1] \times [0, 1) \times [0, 1)$ with $\alpha + \beta < 1$ which implies covariance stationarity.

The posterior density for the GARCH(1,1) model is implemented as follows:

[1] Standard and Poor's (S&P) 500 data can be obtained from several sources online, e.g. from http://finance.yahoo.com.

```
> prior.GARCH<-function(omega,beta,alpha,mu,log=TRUE){
+      c1 <- (omega>0 & omega <1 & beta>=0 & alpha>=0)
+      c2 <- (beta + alpha< 1)
+      c3 <- (mu>-1 & mu<1)
+      r1 <- c1 & c2 & c3
+      r2 <- rep.int(-Inf,length(omega))
+      r2[r1==TRUE] <- 1
+      if (!log)
+        r2 <- exp(r2)
+      cbind(r1,r2)
+ }
> post.GARCH <- function(theta,data,h1,log=TRUE){
+     if (is.vector(theta))
+       theta <- matrix(theta, nrow = 1)
+     omega <- theta[,1]
+     beta <- theta[,2]
+     alpha <- theta[,3]
+     mu <- theta[,4]
+     N <- nrow(theta)
+     pos <- 2:length(data)
+     prior <- prior.GARCH(omega=omega,beta=beta,alpha=alpha,mu=mu)
+     d <- rep.int(-Inf,N)
+     for (i in 1:N){
+        if (prior[i,1] == TRUE){
+            h <- c(h1, omega[i] + alpha[i] * (data[pos-1]-mu[i])^2)
+            for (j in pos)
+                h[j] <- h[j] + beta[i] * h[j-1]
+            tmp <- dnorm(data[pos],mu[i],sqrt(h[pos]),log=TRUE)
+            d[i] <- sum(tmp) + prior[i,2]
+        }
+       }
+     if (!log) d <- exp(d)
+     as.numeric(d)
+ }
```

The function prior.GARCH is coded outside the kernel function to render the program more readable and more flexible. The function prior.GARCH tests whether the constraints are fulfilled, and outputs a $(N \times 2)$ matrix whose first column indicates if the constraints are satisfied, and the second column returns the value of the prior at the corresponding point. Given the data vector/matrix and an initial point satisfying the prior parameter constraints, the MitISEM approximation is obtained. Posterior parameter draws can then be obtained using the Metropolis-Hastings or rejection sampling algorithm given the candidate constructed by MitISEM, or one can estimate posterior moments using Importance Sampling. In order to use the MitISEM candidate for Importance Sampling or the Metropolis Hastings algorithm, one can make use of the function AdMitIS or AdMitMH provided by the *R* package **AdMit**, since these functions just perform IS or MH using a given mixture of Student-t candidate:

```
> data(SP500)
> theta <- c(0.08, 0.86, 0.02, 0.03)
> names(theta) <- c("omega","beta","alpha","mu")
> h1 <- var(data);
> set.seed(1111)
> app.GARCH <- MitISEM(KERNEL=post.GARCH,mu0=theta,h1=h1,data=data)
> app.GARCH$summary

  H METHOD   TIME        CV
1 1   BFGS   1.91 0.8619167
2 1  IS-EM  63.13 0.4592493
3 2  IS-EM 126.33 0.4027914
4 3  IS-EM 137.03 0.3979564
> IS.GARCH <- AdMitIS(N = 10e4, KERNEL=post.GARCH,
+   mit=app.GARCH$mit,data=data,h1=h1)
> IS.GARCH
$ghat
[1] 0.08865471 0.84870010 0.10627791 0.03348575
$NSE
[1] 1.166434e-04 1.135065e-04 7.701554e-05 1.168803e-04
$RNE
[1] 0.6902620 0.7016991 0.7181217 0.8564122
```

The `summary` output of the function `MitISEM` is a data frame containing information on the adaptive fitting procedure: `H` is the number of Student-t components; `METHOD` indicates whether the IS-weighted EM algorithm has been used to optimize the candidate; `TIME` gives the computing time required for this optimization; `CV` gives the coefficient of variation of the importance sampling weights. The output of the function `AdMitIS` is a list. The first component is `ghat`, the importance sampling estimator $\hat{G} = \frac{\sum_{i=1}^{N} W^i G(\theta^i)}{\sum_{i=1}^{N} W^i}$ of the property of interest $E[G(\theta)]$, which is in our case the posterior mean of the parameters. The second component is `NSE`, a vector containing the numerical standard errors (i.e., the standard deviation of the estimates that can be expected if the simulations were to be repeated) of the components of `ghat`. The third component is `RNE`, a vector containing the relative numerical efficiencies of the components of `ghat` (i.e., the ratio between the estimated variance of a hypothetical estimator based on direct sampling and the importance sampling estimator's estimated variance with the same number of draws). `RNE` is an indicator of the efficiency of the chosen importance density; if target and importance densities coincide, `RNE` equals one, whereas a very poor importance density will have a `RNE` close to zero. Both `NSE` and `RNE` are estimated by the method given in Geweke (1989). For estimating $E[G(\theta)]$ the N candidate draws are approximately as 'valuable' as `RNE` × N independent draws from the target would be.

The MitISEM approximation of the posterior density consists of 3 Student-t components. The low CoV values and the high RNE values show that the MitISEM candidate approximates the posterior density accurately.

3.4. The posterior distribution of a mixture GARCH(1,1) model

In this subsection the MitISEM approach is applied to the non-elliptical posterior distribution in the two-component Gaussian Mixture GARCH (1,1) model of Ausín and Galeano (2007). For the Bayesian estimation of this model, Ausín and Galeano (2007) propose a Griddy-Gibbs sampler (Ritter and Tanner 1992), since the recursive structure of the likelihood in GARCH-type models implies that a regular Gibbs sampling approach is not feasible.

The Griddy-Gibbs sampler is known to be very slow. As an alternative we use Importance Sampling with a candidate density resulting from the MitISEM algorithm, and compare the performance of the MitISEM candidate density with the naive Student-t candidate density and a candidate obtained from the AdMit method.

The two-component Gaussian mixture GARCH(1,1) model for the returns y_t $(t = 1, 2, \ldots, T)$ is given by

$$
\begin{aligned}
y_t &= \mu + \sqrt{h_t}\,\varepsilon_t, && (16)\\
h_t &= \omega + \alpha(y_{t-1} - \mu)^2 + \beta h_{t-1}, && (17)\\
\varepsilon_t &\sim \begin{cases} N(0,\sigma^2) & \text{with probability } \rho,\\ N(0,\sigma^2/\lambda) & \text{with probability } 1-\rho, \end{cases} && (18)
\end{aligned}
$$

with h_t the conditional variance of y_t given the information set $I_{t-1} = \{y_{t-1}, y_{t-2}, y_{t-3}, \ldots\}$. In addition, $0 < \lambda < 1$, and $\sigma^2 \equiv 1/(\rho + (1-\rho)/\lambda)$ so that $\text{var}(\varepsilon_t) = 1$; h_0 is treated as a known constant, set as the sample variance of the return series. We restrict $\omega > 0, \alpha \geq 0$ and $\beta \geq 0$ to ensure positivity of h_t. We follow Ausín and Galeano (2007) by imposing the prior restriction $0.5 < \rho < 1$, so that it is ensured that the state with smaller variance has larger probability than the state with larger variance. The mixture distribution in (18) has fatter tails than a Gaussian distribution. We follow Ausín and Galeano (2007) also in specifying flat priors for the model parameters. Moreover, we truncate ω and μ such that these have proper (non-informative) priors. For the $k = 6$ dimensional parameter vector $\theta = (\rho, \lambda, \mu, \omega, \alpha, \beta)$, we have a uniform prior on $(0.5, 1] \times (0, 1) \times [-1, 1] \times (0, 1] \times [0, 1) \times [0, 1)$ with $\alpha + \beta < 1$ which implies covariance stationarity.

The posterior density for the Gaussian mixture GARCH(1,1) model is implemented as follows:

```
>   prior.mGARCH<-function(omega, lambda, beta, alpha, rho, mu, log=TRUE){
+     c1 <- (omega>0 & omega<1 & beta>=0 & alpha>=0)
+     c2 <- (beta + alpha< 1)
+     c3 <- (lambda>=0 & lambda<=1)
+     c4 <- (rho>0.5 & rho<1)
+     c5 <- (mu>-1 & mu<1)
+     r1 <- c1 & c2 & c3 & c4 & c5
+
+     r2 <- rep.int(-Inf,length(omega))
+     tmp <- log(1/2) # uniform prior
+     r2[r1==TRUE] <- tmp
+
+     if (!log)
+     r2 <- exp(r2)
+     cbind(r1,r2)
+   }
```

```
>   post.mGARCH <- function(theta, data, h1, log = TRUE){
+     if (is.vector(theta))
+       theta <- matrix(theta, nrow = 1)
+     omega <- theta[,1]
+     lambda <- theta[,2]
+     beta <- theta[,3]
+     alpha <- theta[,4]
+     rho <- theta[,5]
+     mu <- theta[,6]
+     N <- nrow(theta)
+     pos <- 2:length(data) # # observation index (removing 1st)
+     prior <- prior.mGARCH(omega=omega,lambda=lambda,beta=beta,alpha=alpha,
+                     rho=rho,mu=mu)
+     d <- rep.int(-Inf,N)
+     for (i in 1:N){
+       if (prior[i,1] == TRUE){
+         h <- c(h1, omega[i] + alpha[i] * (data[pos-1]-mu[i])^2)
+         for (j in pos){
+           h[j] <- h[j] + beta[i] * h[j-1]
+         }
+       sigma <- 1 / (rho[i] + ((1-rho[i]) / lambda[i]))
+       tmp1 <- dnorm(data[pos],mu[i],sqrt(h[pos]*sigma),log=TRUE)
+       tmp2 <- dnorm(data[pos],mu[i],sqrt(h[pos]*sigma/lambda[i]),log=TRUE)
+
+       tmp <- log(rho[i] * exp(tmp1) + (1-rho[i]) * exp(tmp2))
+       d[i] <- sum(tmp) + prior[i,2]
+       }
+     }
+     if (!log)
+       d <- exp(d)
+     as.numeric(d)
+   }
```

Given the data vector/matrix *data* the MitISEM approximation is calculated starting from an initial point satisfying the prior parameter constraints. Posterior parameter draws (or appropriately weighted candidate draws) are then obtained using the Metropolis-Hastings algorithm (or Importance Sampling) given the candidate constructed by MitISEM.

Given the MitISEM candidate, one can again obtain Importance Sampling results using the `AdMitIS` function provided by the *R* package **AdMit**:

```
> data(SP500)
> mu0 <- c(0.08, 0.37, 0.86, 0.03, 0.82, 0.03)
> names(mu0) <- c("omega","lambda","beta","alpha","p","mu")
> h1 = var(data); # initial variance
> app.mGARCH <- MitISEM(KERNEL=post.mGARCH,mu0=mu0,h1=h1,data=data)
> app.mGARCH <- MitISEM(KERNEL=post.mGARCH,mu0=mu0,h1=h1)
> app.mGARCH$summary
```

```
  H METHOD   TIME        CV
1 1   BFGS   4.22 1.5817680
2 1  IS-EM  72.58 1.1987712
3 2  IS-EM 148.02 1.0519608
4 3  IS-EM 141.15 0.9878679
> IS.mGARCH <- AdMitIS(N = 10e4, KERNEL=post.mGARCH, mit=app.mGARCH$mit,
+     data=data,h1=h1)
> print(IS.mGARCH,2)
$ghat
[1] 0.079 0.369 0.862 0.098 0.787 0.029
$NSE
[1] 0.00054 0.00170 0.00050 0.00033 0.00231 0.00050
$RNE
[1] 0.32 0.25 0.35 0.39 0.29 0.46
```

Whereas MitISEM yields a mixture of 3 Student-t components, AdMit yields a mixture of 5 Student-t components. That AdMit requires more components reflects MitISEM's superiority, since it jointly optimizes all Student-t components. The conditional posterior density kernel of parameters (ρ, λ) given the posterior means of the other parameters and the approximations by three methods are shown in Figure 4. The MitISEM density is clearly the best approximation of the posterior. Table 1 and Table 2 show that for this example, both naive and MitISEM candidates outperform the AdMit approximation in terms of the importance weights' CoV, and in terms of the NSEs of the estimated posterior means. There are two reasons for the better performance of the naive candidate compared with the AdMit candidate. First, the IS-weighted EM algorithm implies that the naive candidate's single Student-t distribution is specified in an optimal way. Second, the novel robustification introduced in this paper, discarding candidate draws outside the 'allowed range' from the number of candidate draws during the construction of a new candidate, ensures that enough relevant, 'allowed' candidate draws are obtained for the construction of the naive candidate. In particular for target densities with several parameter restrictions, such as the posterior in the mixture GARCH model, this robustification is important. Further, the additional Student-t components of the MitISEM candidate imply that it has a higher accuracy than the naive candidate. Here we make two additional remarks on the comparison between the naive, AdMit and MitISEM method. First, if we require simulation results with a certain very high precision, then MitISEM would obviously require much fewer draws than the alternative methods, so that the total computing time (for both the construction and the subsequent use of the candidate density) would be shorter for MitISEM. Second, the higher quality of the MitISEM approximation of the target density implies that there is less risk that a relevant part of the target distribution is 'missed', for example in case of a multimodal target distribution, which would possibly cause substantially biased results for the other methods.

3.5. Predictive likelihood for a mixture GARCH(1,1) model

In this subsection we show the candidate density obtained by the MitISEM method can be used to accurately calculate a model's predictive likelihood. The calculation of model probabilities can be based on the models' marginal likelihoods or the predictive likelihoods, where the former are problematic under non-informative priors on parameters that only occur

Table 1: Summary of naive, AdMit and MitISEM candidates for the mixture GARCH(1,1) model for S&P 500 data. The table reports number of Student-t components (# t), optimization method, time (in seconds) and CoV for all compared algorithms. Candidates are constructed using by 10^4 draws.

algorithm	# t components	time (seconds)	CoV
AdMit	5	419.48	1.99
naive	1	72.82	1.22
MitISEM	3	365.97	0.99

Table 2: Posterior means of parameters in mixture GARCH(1,1) model and Numerical Standard Errors (NSE) of the IS estimates using the naive, AdMit and MitISEM candidates for the S&P 500 data. Candidate approximations and posterior results are based on 10^4 and 10^3 draws, respectively.

	Posterior mean			NSE ×100		
	AdMit	naive	MitISEM	AdMit	naive	MitISEM
ω	0.08	0.08	0.08	0.07	0.06	0.05
λ	0.37	0.37	0.37	0.17	0.20	0.17
β	0.86	0.86	0.86	0.07	0.06	0.05
α	0.10	0.10	0.10	0.06	0.04	0.03
ρ	0.79	0.79	0.79	0.30	0.28	0.23
μ	0.03	0.03	0.03	0.07	0.07	0.05

in one of the models, in the sense that the 'smaller' model may be favored even if the 'larger' model is the true Data Generating Process (DGP) (Bartlett 1957). The **MitISEM** package provides functions to calculate the marginal or predictive likelihood of a model given its posterior density kernel and a candidate density obtained by the MitISEM method. The reason is that the computation of marginal or predictive likelihoods is an important ingredient of many Bayesian analyses.

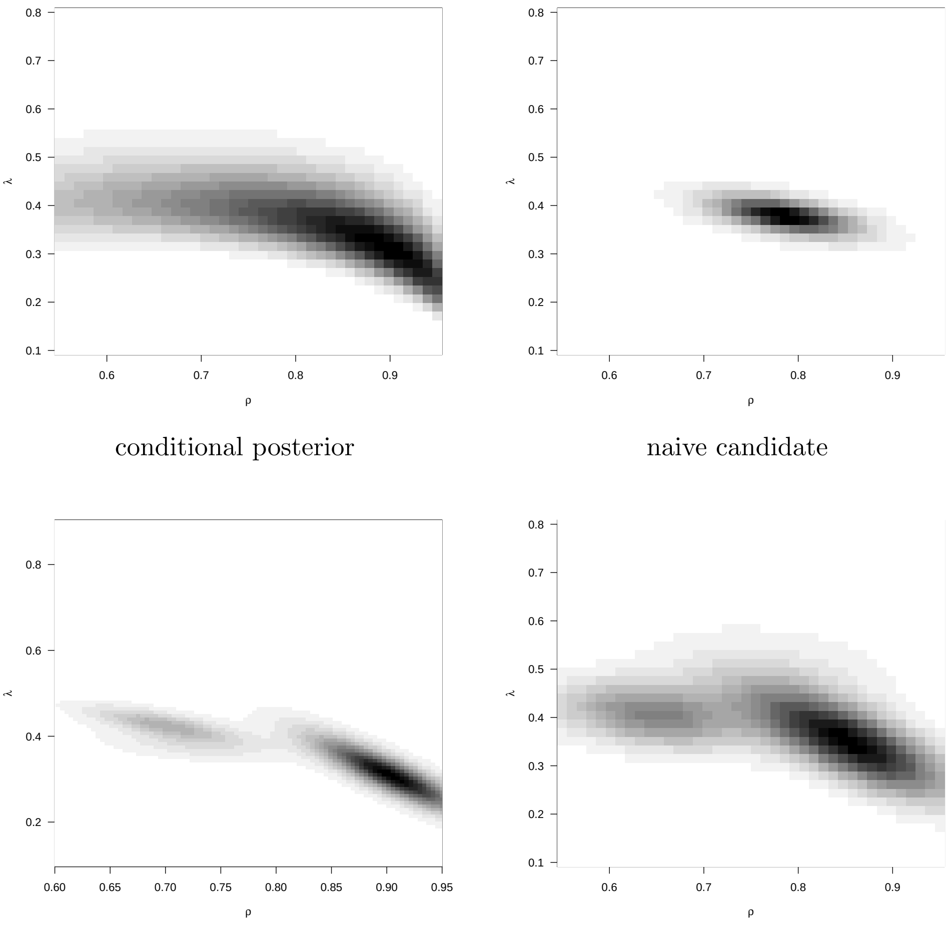
The predictive likelihood of a model M_1 is obtained by splitting the data $y = (y_1, \dots y_T)$ into $y^* = (y_1, \dots y_m)$ and $\tilde{y} = (y_{m+1}, \dots y_T)$ (Gelfand and Dey 1994; Eklund and Karlsson 2007):

$$p(\tilde{y}|y^*, M_1) = \int p(\tilde{y}|\theta_1, y^*, M_1) p(\theta_1|y^*, M_1) d\theta_1, \tag{19}$$

which is the marginal likelihood if we consider $\tilde{y}$ as 'the data' and $p(\theta_1|y^*, M_1)$, the exact posterior density after observing y^*, as the prior. Using Bayes' rule for this exact posterior density $p(\theta_1|y^*, M_1)$ and substituting into (19) yields

$$p(\tilde{y}|y^*, M_1) = \frac{\int p(y|\theta_1, M_1) p(\theta_1|M_1) d\theta_1}{\int p(y^*|\theta_1, M_1) p(\theta_1|, M_1) d\theta_1}. \tag{20}$$

Hence this predictive likelihood is the ratio of the marginal likelihood for all observations over the marginal likelihood for the first part of the data.



conditional posterior

naive candidate

AdMit candidate
5 components

MitISEM candidate
3 components

Figure 4: Conditional posterior density kernel of (ρ, λ) given posterior means of the other parameters $(\omega, \beta, \alpha, \mu)$ in the mixture GARCH(1,1) model together with the naive, AdMit and MitISEM approximations

As an illustration, we apply the model in (16)–(18) to S&P 500 data and perform the simulation-based computation of the predictive likelihoods. We also compare the performance of the MitISEM candidate with a ‘naive’ candidate. The first half of the observations are regarded as the ‘training sample’ $y^* = (y_1, \ldots y_m)$. The calculation of the predictive likelihood for this example is implemented as:

```
> data(SP500)
> y.ss <- y[1:626] # subsample of data
> h1 = var(y); # initial variance
> KERNEL = match.fun(post.mGARCH) # posterior density
> N <- 1e3 # # draws for IS
> mu0 <- c(0.08, 0.37, 0.86, 0.03, 0.82, 0.03)   # initial parameters
> names(theta) <- c("omega","lambda","beta","alpha","p","mu")
> # full sample approximation
> mit.fs <- MitISEM(KERNEL=KERNEL,mu0=mu0,y=y,h1=h1)$mit
> # subsample approximation
> mit.ss <- MitISEM(KERNEL=KERNEL,mu0=mu0,y=y.ss,h1=h1)$mit
> # predictive likelihood
> PL.mGARCH <- PredLik(N=N,mit.fs=mit.fs$mit,mit.ss=mit.ss$mit,KERNEL=KERNEL,
+        y.fs=y,y.ss=y.ss,h1=h1)
```

In order to calculate the accuracy of the estimates, we replicate the predictive likelihood calculation 50 times. Table 3 shows simulation results where the average predictive likelihoods and Numerical Standard Errors are calculated from 50 replications. The candidates for all cases are calculated using 10^4 draws, and the estimated predictive likelihood values are based on 10^3 draws, where the latter was done to decrease the computing time of the 50 repetitions.

Table 3: Approximation and predictive likelihood for the mixture of GARCH model. Candidate approximations and posterior results are based on 10^4 and 10^3 draws, respectively. Mean and Numerical Standard Error (NSE) for each estimate is based on 50 replications.

# t components		Predictive likelihood	
full sample	training sample	mean $\times 10^{470}$	NSE $\times 10^{472}$
3	3	1.68	1.25

The MitISEM candidate consists of three mixture components for both the training sample and the full sample, indicating highly non-elliptical posterior shapes for both datasets. Despite these irregularities in the posterior densities, the small NSE reported in Table 3 shows that, even with the relatively small number of posterior draws, calculated predictive likelihoods for this model are quite accurate given the MitISEM approximation to the posterior density.

3.6. Computing a sequence of predictive likelihoods using Sequential MitISEM

We next apply the Sequential MitISEM algorithm to the two-component mixture GARCH model with the S&P 500 data. Sequential MitISEM is used to efficiently construct a series of candidates that approximate posteriors for increasing data sets, where the candidate can be used for estimation of posterior moments, marginal likelihoods or predictive likelihoods. In this example we consider the latter. We use the first half of the observations as the training sample y^* (for the marginal likelihood in the denominator of the predictive likelihood (20)). At each time $t = 1203, \ldots, 1217$, the predictive likelihood is computed while the training sample y^* is kept fixed. Such a sequence of updated predictive likelihoods could be used in an application of Bayesian Model Averaging (BMA), combining forecasts from several models at each time $t = 1203, \ldots, 1217$ by weighting these with the estimated model probabilities. Once the posterior kernel is specified, the sequential MitISEM approximations can be obtained as follows:

```
>   data(SP500)
>   mu0 <- c(0.08, 0.37, 0.86, 0.03, 0.82, 0.03) # initial parameters
>   names(mu0) <- c("omega","lambda","beta","alpha","p","mu")
>   h1 = var(data) # initial variance
>   # MitISEM approximation to the initial sample
>   data.ss <- data[1:floor(length(data)/2)]
>   MitISEMapp.subsample <- MitISEM(KERNEL=post.mGARCH,mu0=mu0,,h1=h1,data=data.ss)
>   # Sequential MitISEM applied to updated samples
>   control.seq <- list(T0 = 1203,tau=1:20)
>   app.mGARCH.SeqMitISEM <- SeqMitISEM(data,KERNEL=post.mGARCH,mu0,
+           control.seq=control.seq,h1=h1)
```

Table 4 presents the number of mixture components, CoV values and estimated predictive likelihoods for each sequential algorithm, and provides more details about the results of the Sequential MitISEM algorithm. Note that for the calculation of predictive likelihoods in increased data samples, an ‘ad hoc’ MitISEM procedure would be applied 14 times, while the Sequential MitISEM ‘extends’ the candidate density only once, for the sample size of 1213. In the remaining time periods, the candidate is simply ‘reused’, with minimal computational time.

Table 4: Predictive Likelihoods for the mixture GARCH model using Sequential MitISEM. Candidate approximations and posterior results are based on 10^4 and 10^3 draws, respectively.

#observations	# t components	CoV	Predictive Likelihood
1203	2	1.00	0.35×10^{433}
1204	2	0.94	0.38×10^{434}
1205	2	0.97	0.61×10^{435}
1206	2	1.01	0.80×10^{436}
1207	2	0.95	1.31×10^{437}
1208	2	0.97	0.27×10^{438}
1209	2	1.01	0.37×10^{439}
1210	2	1.07	0.51×10^{440}
1211	2	0.97	0.94×10^{441}
1212	2	1.04	0.20×10^{442}
1213	4	0.89	0.42×10^{443}
1214	4	0.94	1.00×10^{444}
1215	4	1.01	1.41×10^{445}
1216	4	0.97	0.32×10^{446}
1217	4	1.00	0.80×10^{447}

Sequential MitISEM steps	
# reused	13
# adapted	0
# adapted and extended	1

3.7. Model Probabilities from predictive likelihoods for an IV model

In this section we apply the MitISEM algorithm to an Instrumental Variables (IV) model that describes the effect of education on income. Our IV model with one explanatory endogenous variable and p instruments is defined by (Bowden and Turkington 1990):

$$y = x\beta + \varepsilon, \tag{21}$$

$$x = z\Pi + v, \tag{22}$$

where y is the $N \times 1$ vector of the dependent variable income, x is the $N \times 1$ vector of the endogenous explanatory variable, education, z is the $N \times p$ matrix of instruments and all variables are demeaned i.e. y, x and z do not include a constant term. The residuals are assumed to come from a normal distribution: $(\epsilon', \ v')' \sim NID(0, \Sigma \otimes I)$.

The posteriors resulting from the IV model in (21)-(22) are non-standard due to the possible 'endogeneity' of the variable x. The endogeneity problem in the model simply arises from the correlation between the structural errors: if the covariance matrix Σ is diagonal, the IV model simplifies to a simple regression model, with elliptical posterior densities (Zellner 1971). Therefore the instruments are only necessary if the correlation between the structural errors, $\rho \equiv \mathrm{corr}\,(\epsilon_i, \nu_i)$ is different from zero. The effect of latent abilities (leading to both a higher education and a higher income given a certain education level) may cause a positive correlation ρ, whereas measurement errors in observed education may cause a negative ρ. Under conventional flat priors, the posterior density for the parameters for the IV model is non-standard (Drèze 1976, 1977; Kleibergen and Van Dijk 1998). For an exactly identified model with a single instrument, the posterior density resulting from this model is improper. We specify a Jeffreys prior (see e.g. Hoogerheide, Kleibergen, and Van Dijk (2007a)), which leads to a proper posterior density. The posterior density of the model in (21)-(22) under the Jeffreys prior can be implemented as:

```
> KERNEL<-function(theta,data,scale=0,log=TRUE){
+   if(is.vector(theta))
+     theta = matrix(theta,nrow=1)
+
+   y <- data[,1] # dependent variable
+   x <- data[,2] # endogenous variable
+   z <- data[,3] # instrument
+
+   # (log) Jeffreys prior p(beta,Pi,Sigma) = |Pi|*(|Sigma|^(-2))
+   # for an IV model with a single instrument
+   Jeff.prior<-function(beta, Pi,Sigma11,rho, Sigma22){
+     # check covariance constraints
+     c1 <- (Sigma11 > 0)
+     c2 <- (Sigma22 > 0)
+     c3 <- (rho > -1)
+     c4 <- (rho <  1)
+
+     r1 <- c1 & c2 & c3 &c4
+     r2 <- rep.int(-Inf,length(Sigma11))
+     r2[r1==TRUE] <- log(abs(Pi[r1==TRUE])) - 2* log(Sigma11[r1==TRUE] *
+                    Sigma22[r1==TRUE] * (1- rho[r1==TRUE]^2))
+     return(cbind(r1,r2))
+   }
+   if (is.vector(theta))
+     theta <- matrix(theta, nrow = 1)
+
+   logprior <- Jeff.prior(theta[,1],theta[,2],theta[,3],theta[,4],theta[,5])
+   rcov <- logprior[,1]==TRUE # covariance restriction
+
+   # determinant and exponent factors in likelihood for each parameter set
+   fn_aux <- function(theta_aug,y,x,z){
+     # covariance matrix
+     tmp <- matrix(c(theta_aug[3],theta_aug[4],theta_aug[4],theta_aug[5]),2,2)
+     # log-determinant
+     detfac <- log(det(tmp))
+     # (log) exponent
+     beta <- theta_aug[1]
+     Pi <- theta_aug[2]
+     res <- cbind(y - x * beta,x - z * Pi)
+     SigmaInv <- solve(tmp)
+     S <- SigmaInv %*% crossprod(res)
+     expfac <- -0.5*sum(diag(S))
+     (c(detfac,expfac))
+   }
+   # covariance matrix from correlations
+   theta_aug = theta[rcov,]
+   if(is.vector(theta_aug))
+     theta_aug = matrix(theta_aug,nrow=1)
+   Sigma12 <- theta[rcov,4] * sqrt(theta[rcov,3] * theta[rcov,5])
+   theta_aug[,4] = Sigma12
+
+   # log posterior
+   T <- length(y)
+   d <- rep.int(-Inf,nrow(theta))
+   if(any(rcov)){
+     tmp_1 = t(apply(theta_aug,1,FUN=fn_aux,y=y,x=x,z=z))
+     d[rcov] = - (T/2)* tmp_1[,1] + tmp_1[,2] + logprior[rcov,2]
+   }
+
+   if (!log)
+     d <- exp(d)
+   as.numeric(d)
+ }
```

For the application we consider data of Card (1995) on income and education. In these data, income levels are measured by hourly wage (in natural logarithms), education level is 1 if the individual attended college and 0 otherwise. College proximity, which takes the value 1 if there is a nearby college and 0 otherwise, is the proposed instrument for the education level of individuals. The data further consist of other covariates such as gender, experience and area of residence, for 1030 men in 1976.[2] For the analysis of the IV model, we first demean the income, education and college proximity data, and transform these into residuals after regression on the exogenous covariates in the dataset, which is equivalent to integrating out the corresponding coefficients under a flat prior.

We define two models, one treating education as an endogenous explanatory variable (i.e. the IV model), and the second model treating education as an exogenous explanatory variable (i.e. the simple linear regression model). For a comparison of these two models under uninformative priors, we use the predictive likelihoods. The predictive probability of the null model (which assumes exogeneity) can be calculated using the Savage-Dickey Density Ratio (SDDR). Dickey (1971) shows that the Bayes factor can be calculated using a single model if the alternative models are nested and the prior densities satisfy the condition that the prior for the restricted model equals the corresponding conditional prior in the unrestricted model. Under that condition the model probabilities can be simplified to:

$$\frac{p(M_0 \mid y)}{p(M_1 \mid y)} \quad = \quad \frac{p(\tilde{y} \mid y^*, M_0)}{p(\tilde{y} \mid y^*, M_1)} \frac{p(M_0)}{p(M_1)} = \frac{p(\rho = 0 \mid \tilde{y}, y^*, M_1)}{p(\rho = 0 \mid y^*, M_1)} \times \frac{p\,(M_0)}{p\,(M_1)}, \tag{23}$$

hence the model probabilities can be calculated from the general model only, using draws from the marginal posterior density of the endogeneity parameter ρ conditioning on the training sample and the full sample to compute $p(\rho = 0 \mid \tilde{y}, y^*, M_1)$ and $p(\rho = 0 \mid y^*, M_1)$ using kernel density estimates. We get these parameter draws from the Metropolis Hastings sampler, using the MitISEM candidate and the `AdMitMH` function from the *R* package **AdMit** (Ardia *et al.* 2009b). Then we calculate the predictive likelihoods using (23), a random training sample consisting of 5% of the original data points, and using the 'naive' and the MitISEM approximations to the posterior density. The implementation of this predictive likelihood approach is straightforward using **MitISEM**:

```
> require(AdMit)
> data(Card)
> # random training sample
> pc.train = 0.05 # training sample percentage
> N <- nrow(data.fs)
> M <- round(N*pc.train)
> data.ss <- data.fs[sample(1:N,M),]
>
> # full sample MitISEM approximation
> mit.fs <- MitISEM(KERNEL=KERNEL,mu0=mu0,df0=30,data=data.fs,
>       control=list(tol.pr=0.02))
> # training sample MitISEM approximation
> mit.ss <- MitISEM(KERNEL=KERNEL,mu0=mu0,df0=30,data=data.ss,
>       control=list(tol.pr=0.02))
>
> # Metropolis Hastings using the MitISEM candidate
```

[2] The dataset can be obtained from `http://davidcard.berkeley.edu/data_sets.html`.

```
> post.fs <- AdMitMH(N=N,KERNEL,mit=mit.fs$mit,y=y,x=x,z=z)
> post.ss <- AdMitMH(N=N,KERNEL,mit=mit.ss$mit,y=y.ss,x=x.ss,z=z.ss)
>
> # Predictive likelihood calculation
> ind.post = (N/5+1):N # remove the burn-in period
> rho.fs <- post.fs$draws[ind.post,4]
> rho.ss <- post.ss$draws[ind.post,4]
> Pred.Lik <- density(rho.fs,from=0,to=0)$y[1] / density(rho.ss,from=0,to=0)$y[1]
```

We repeat the whole predictive likelihood estimation 20 times, with 20 different random seeds, so that also the random selection of the training sample and the approximation of the posterior for all data are different each time. We specify equal prior probabilities $p\left(M_0\right) = p\left(M_1\right) = \frac{1}{2}$. Table 5 presents the details of the MitISEM and naive density approximations to the posterior, together with the predictive likelihoods. First, the average number of Student-t components is typically rather high in training samples and the full sample, indicating non-elliptical posterior shapes for this model. Hence a flexible candidate density, such as the MitISEM candidate, is motivated. Second, the obtained predictive likelihoods are more accurate, as indicated by relatively smaller NSE values, when the candidate density is obtained from the MitISEM method. Note that 0.14 may seem only slightly smaller than 0.16, but since in this case most of the variation is caused by the random selection of the training sample rather than the finiteness of the number of posterior draws, the relative improvement of quality provided by the MitISEM candidate is still considerable.

Table 5: Model probabilities ($p(M_0|y)$ and $p(M_1|y)$, for models without/with endogeneity) based on predictive likelihood (23) using ‘naive’ and MitISEM approximations. ‘# t components’ denotes the average number of Student-t components in the MitISEM candidate over the 20 repetitions. Mean, NSE and # t are also based on these 20 repetitions. The candidate and posterior results at each repetition are based on 10^4 draws, respectively. For the Metropolis Hastings method, we use a burn-in sample size of 2000.

	MitISEM candidate		naive candidate	
	CoV	# t components	CoV	
full sample	2.65	3.15	9.85	
training sample	1.46	3.80	6.76	
	MitISEM candidate		naive candidate	
	mean	NSE	mean	NSE
$p(M_0\mid y)$	0.37	0.16	0.38	0.14
$p(M_1\mid y)$	0.63	0.16	0.62	0.14

4. Concluding remarks

We presented the *R* package **MitISEM** which provides an automatic algorithm for the approximation of a possibly non-elliptical target density. In particular the obtained approximation can be used for the Bayesian analysis of models with non-elliptical posterior shapes, and for Bayesian model comparison based on marginal or predictive likelihoods. The package also provides the 'Sequential MitISEM' algorithm, which decreases the computational time substantially if the candidate density is used to assess posterior distributions or model probabilities for increasing data samples, when the posterior distribution is updated using new observations. For the Bayesian estimation, the package provides an efficient method to calculate marginal and predictive likelihoods, given a user-supplied posterior density kernel of the model parameters.

We illustrated the MitISEM algorithm using several canonical statistical and econometric models: two different Gelman-Meng (Gelman and Meng 1991) distributions, a mixture GARCH model for the S&P 500 data and an IV model for the Card (1995) data. The Gelman-Meng distribution is a standard example of a distribution with possibly non-elliptical shapes. The posterior densities of the IV and (mixture) GARCH models are also characterized by non-elliptical shapes, in which case Bayesian inference of the model parameters and model probabilities using Importance Sampling and Metropolis Hastings algorithms require a flexible and appropriate candidate density. We illustrated the use of the MitISEM method for forming such a flexible candidate density, and show that the obtained candidate can be used for efficient estimations of model parameters as well as predictive likelihoods. Finally, we showed that the 'Sequential MitISEM' algorithm provides computational gains in subsequent estimation of the predictive likelihoods. In future research we will explore the possibility of parallelized computation for the different steps of the MitISEM method, so that one can utilize graphical cards or multi-core computer systems to substantially speed up the calculations.

Acknowledgements

Nalan Baştürk and Herman K. van Dijk are supported by the Netherlands Organisation for Scientific Research (NWO) Secondment Grant, number 400-07-703. Lennart Hoogerheide is supported by the NWO Veni Grant, number 451-09-028. We thank Tommi Tervonen and David Ardia for helpful comments.

References

Ardia D, Hoogerheide LF, Van Dijk HK (2009a). "Adaptive Mixture of Student-t Distributions as a Flexible Candidate Distribution for Efficient Simulation: The R Package AdMit." *Journal of Statistical Software*, **29**(3), 1–32.

Ardia D, Hoogerheide LF, Van Dijk HK (2009b). *The AdMit Package: Adaptive Mixture of Student-t Distributions*. Version 1-01.03.1, URL http://CRAN.R-project.org/package=AdMit.

Ausín MC, Galeano P (2007). "Bayesian Estimation of the Gaussian Mixture GARCH Model." *Computational Statistics & Data Analysis*, **51**(5), 2636–2652.

Bartlett MS (1957). "A Comment on DV Lindley's Statistical Paradox." *Biometrika*, **44**(3–4), 533.

Berger JO (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer.

Bollerslev T (1986). "Generalized autoregressive conditional heteroskedasticity." *Journal of Econometrics*, **31**(3), 307–327.

Bos CS, Mahieu RJ, Van Dijk HK (2000). "Daily exchange rate behaviour and hedging of currency risk." *Journal of Applied Econometrics*, **15**(6), 671–696.

Bowden RJ, Turkington DA (1990). *Instrumental Variables*. Cambridge Univ Press.

Cappé O, Douc R, Guillin A, Marin JM, Robert CP (2008). "Adaptive importance sampling in general mixture classes." *Statistics and Computing*, **18**(4), 447–459.

Card D (1995). "Using Geographic Variation in College Proximity to Estimate the Return to Schooling." In LN Christofides, EK Grant, R Swidinsky (eds.), *Aspects of labour market behaviour: essays in honour of John Vanderkamp*, chapter 7. University of Toronto Press, Toronto.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38.

Dickey JM (1971). "The weighted likelihood ratio, linear hypotheses on normal location parameters." *The Annals of Mathematical Statistics*, **42**(1), 204–223.

Drèze JH (1976). "Bayesian Limited Information Analysis of the Simultaneous Equations Model." *Econometrica: Journal of the Econometric Society*, **44**(5), 1045–1075.

Drèze JH (1977). "Bayesian Regression Analysis Using Poly-t Densities." *Journal of Econometrics*, **6**(3), 329–354.

Durham G, Geweke J (2011). "Massively Parallel Sequential Monte Carlo for Bayesian Inference." Manuscript, URL http://www.censoc.uts.edu.au/pdfs/geweke_papers/gp_working_9.pdf.

Eklund J, Karlsson S (2007). "Forecast Combination and Model Averaging Using Predictive Measures." *Econometric Reviews*, **26**(2–4), 329–363.

Gelfand AE, Dey DK (1994). "Bayesian Model Choice: Asymptotics and Exact Calculations." *Journal of the Royal Statistical Society Series B*, **56**(3), 501–514.

Gelman A, Meng XL (1991). "A Note on Bivariate Distributions that are Conditionally Normal." *The American Statistician*, **45**(2), 125–126.

Geweke J (1989). "Bayesian Inference in Econometric Models Using Monte Carlo Integration." *Econometrica: Journal of the Econometric Society*, **57**, 1317–1339.

Hammersley JM, Handscomb DC (1975). *Monte Carlo Methods*. Taylor & Francis.

Hoogerheide L, Kleibergen F, Van Dijk HK (2007a). "Natural Conjugate Priors for the Instrumental Variables Regression Model Applied to the Angrist-Krueger Data." *Journal of Econometrics*, **138**(1), 63–103.

Hoogerheide L, Opschoor A, Van Dijk HK (2012). "A Class of Adaptive Importance Sampling Weighted EM Algorithms for Efficient and Robust Posterior and Predictive Simulation." *Journal of Econometrics*. In press, URL http://www.sciencedirect.com/science/article/pii/S0304407612001583.

Hoogerheide LF, Kaashoek JF, Van Dijk HK (2003). "Neural network approximations to posterior densities: An analytical approach." In *Proceedings of the Section on Bayesian Statistical Science*. American Statistical Association.

Hoogerheide LF, Kaashoek JF, Van Dijk HK (2007b). "On the Shape of Posterior Densities and Credible Sets in Instrumental Variable Regression Models with Reduced Rank: An Application of Flexible Sampling Methods Using Neural Networks." *Journal of Econometrics*, **139**(1), 154–180.

Hop JP, Van Dijk HK (1992). "SISAM and MIXIN: Two Algorithms for the Computation of Posterior Moments and Densities Using Monte Carlo Integration." *Computer Science in Economics & Management (Computational Economics)*, **5**(3), 183–220. Reprinted in Bulletin of the International Statistical Institute, Cairo, vol. LIV, book 3, 29 pages.

Imbens GW, Angrist JD (1994). "Identification and Estimation of Local Average Treatment Effects." *Econometrica*, **62**(2), pp. 467–475. ISSN 00129682.

Kleibergen F, Van Dijk HK (1998). "Bayesian simultaneous equations analysis using reduced rank structures." *Econometric Theory*, **14**(06), 701–743.

Kloek T, Van Dijk HK (1978). "Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo." *Econometrica*, **46**, 1–20.

Kullback S, Leibler RA (1951). "On Information and Sufficiency." *The Annals of Mathematical Statistics*, **22**(1), 79–86.

McLachlan GJ, Krishnan T (2008). *The EM Algorithm and Extensions*. John Wiley and Sons.

McLachlan GJ, Peel D (2000). *Finite Mixture Models*. John Wiley and Sons.

Peel D, McLachlan GJ (2000). "Robust Mixture Modelling Using the t Distribution." *Statistics and Computing*, **10**(4), 339–348.

Ritter C, Tanner MA (1992). "Facilitating the Gibbs Sampler: The Gibbs Stopper and the Griddy-Gibbs Sampler." *Journal of the American Statistical Association*, **87**, 861–868.

Van Dijk HK (1984). "Posterior Analysis of Econometric Models using Monte Carlo Integration." Erasmus University Press, 207 pages.

Van Dijk HK, Hop JP, Louter AS (1987). "An algorithm for the computation of posterior moments and densities using simple importance sampling." *The Statistician*, **36**, 83–90.

Zeevi AJ, Meir R (1997). "Density Estimation through Convex Combinations of Densities: Approximation and Estimation Bounds." *Neural Networks*, **10**(1), 99–109.

Zellner A (1971). *An Introduction to Bayesian Inference in Econometrics*. Wiley, New York.

Zellner A, Ando T, Baştürk N, Hoogerheide L, Van Dijk HK (2012). "Bayesian Analysis of Instrumental Variable Models: the potential of Direct Monte Carlo." Unpublished manuscript.

Zivot E (2009). "Practical issues in the analysis of univariate GARCH models." In T Andersen, R Davis, JP Krei, T Mikosch (eds.), *Handbook of Financial Time Series*, pp. 113–155. Springer Verlag, New York.

Affiliation: