Journal of Statistical Software

July 2017, Volume 79, Issue 1. doi: 10.18637/jss.v079.i01

The R Package MitISEM: Efficient and Robust Simulation Procedures for Bayesian Inference

Nalan Baştürk
Maastricht University

Stefano Grassi
University of Rome "Tor Vergata"

Lennart Hoogerheide
Vrije Universiteit Amsterdam

Anne Opschoor
Vrije Universiteit Amsterdam

Herman K. van Dijk
Erasmus University Rotterdam

Abstract

This paper presents the R package **MitISEM** (*mixture of t by importance sampling weighted expectation maximization*) which provides an automatic and flexible two-stage method to approximate a non-elliptical target density kernel – typically a posterior density kernel – using an adaptive mixture of Student t densities as approximating density. In the first stage a mixture of Student t densities is fitted to the target using an expectation maximization algorithm where each step of the optimization procedure is weighted using importance sampling. In the second stage this mixture density is a candidate density for efficient and robust application of importance sampling or the Metropolis-Hastings (MH) method to estimate properties of the target distribution. The package enables Bayesian inference and prediction on model parameters and probabilities, in particular, for models where densities have multi-modal or other non-elliptical shapes like curved ridges. These shapes occur in research topics in several scientific fields. For instance, analysis of DNA data in bio-informatics, obtaining loans in the banking sector by heterogeneous groups in financial economics and analysis of education's effect on earned income in labor economics. The package **MitISEM** provides also an extended algorithm, 'sequential MitISEM', which substantially decreases computation time when the target density has to be approximated for increasing data samples. This occurs when the posterior or predictive density 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 MH using the candidate density obtained by MitISEM outperforms, in terms of numerical efficiency, MH using a simpler candidate, as well as the Gibbs sampler. The MitISEM approach is also used for Bayesian model comparison using predictive likelihoods.

Keywords: finite mixtures, Student t densities, importance sampling, MCMC, Metropolis-Hastings algorithm, expectation maximization, Bayesian inference, R software.

1. Introduction

There exist several classes of important statistical and econometric models where posterior and/or predictive distributions have unknown analytical properties and non-elliptical Bayesian highest posterior density (HPD) credible sets. For a theoretical background see, e.g., Berger (1985). As examples we name the class of so-called instrumental variable regression models with weak instruments where, for instance, the effect of years of education on income is modeled and measured. This is very 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 and it may also occur in epidemiological models with regional data patterns where very few observations of a disease occur. A detailed analysis of this literature is beyond the scope of the present paper. We refer to, e.g., Imbens and Angrist (1994) and Bos, Mahieu, and Van Dijk (2000), the references cited there and to several textbooks: Lancaster (2004); Geweke (2005); Rossi, Allemby, and McCulloch (2005) and Koop, Poirier, and Tobias (2007) for more background. In such studies an important technical issue is the development of efficient and robust procedures to generate (pseudo-)random draws from non-elliptical distributions in a numerically efficient way. Even if simulation from the conditional distributions is relatively easy, for example, using the well-known Gibbs sampler, multi-modality and/or high correlations between model parameters may cause this sampler to converge extremely slowly and yield erroneous results even with a relatively large sample of draws. We illustrate this in the present paper.

Two stage method and the approximation property

This paper presents the R (R Core Team 2017) package **MitISEM** (Baştürk, Hoogerheide, Opschoor, and Van Dijk 2017) which provides an automatic and flexible method to approximate a target posterior or predictive density by an adaptive mixture of Student t densities, also referred to in the following as *approximate*, *candidate* or *proposal* density.[1] The multivariate target density can be non-elliptical like being multi-modal, strongly correlated and/or having curved ridges in the surface. Only a kernel of the target density is required for the MitISEM method, which is typically a posterior density kernel. Our method consists of two stages. In the first stage a mixture of Student t candidate densities is fitted to the target using an expectation maximization (EM) algorithm where each step of the optimization procedure is weighted using importance sampling. Details are given in Section 2. In the second stage the obtained candidate density can be used in importance sampling or the independence chain Metropolis-Hastings method 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 mixtures of Student t densities falls within this framework.

[1] These three terms are interchangeably used in the literature and we also do that in the present paper.

Algorithmic steps

In the first stage the algorithm **MitISEM** iterates over importance weighted expectation maximization steps in order to efficiently construct a mixture of Student t densities that is an accurate approximation of the target density. Starting with a single Student t density, new mixture components are added in an iterative way until the required approximation is reached. At each iteration, parameters of the mixture components – consisting of mode, scale, degrees of freedom and mixing probability – are optimized such that the Kullback-Leibler divergence between target density and candidate density of Student t mixtures is minimized. The constructed mixture is used in the second stage for efficient and robust application of either importance sampling (IS) or the independence chain Metropolis-Hastings (MH) method.

Illustrations

We illustrate the MitISEM algorithm using a well-known statistical example distribution from Gelman and Meng (1991) that is characterized by a very non-elliptical, possibly bi-modal, joint distribution while the conditional distributions are normal. We also use the posterior distributions in two classes of canonical econometric models: the generalized autoregressive conditional heteroskedasticity (GARCH) model that is extensively used in financial econometrics and the instrumental variable (IV) regression model that is often used to study the effect of number of years of education on earned income. Both classes of models yield non-elliptical posterior and/or predictive distributions. 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 and model averaging. We also introduce an R program for an adapted MitISEM algorithm as in Hoogerheide *et al.* (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 idea of the MitISEM method and the 'sequential MitISEM' extension, and summarizes the steps of the expectation maximization algorithm that we make use of. Section 3 presents applications of the algorithm to several model structures and datasets. Section 4 concludes.

2. Searching for a mixture of Student t candidate densities

The mixture of Student t densities that is obtained by making use of importance sampling weighted expectation maximization is based on iteratively adding Student t densities to the mixture (Hoogerheide, Opschoor, and Van Dijk 2012). The algorithm provides an automatic and flexible method to construct a candidate density minimizing the Kullback-Leibler divergence (or cross-entropy distance, Kullback and Leibler 1951) between two densities: the so-called target density, typically a posterior density, and the approximate or candidate density. Each new Student t component in the candidate density covers the areas of the target density that are not well covered by the previous candidate density. The modes, scales, degrees of freedom and mixing probabilities are quickly optimized 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 defined as

$f(\theta|y)$ for data y and model parameters θ. Alternatively, $f(\theta)$ can be a density kernel $f(\theta|\alpha)$ for data θ and given fixed model parameters α. We concentrate on the former case, where $f(\theta)$ is a posterior density kernel, and simplify the notation by having removed the conditioning on data. Let $g(\theta)$ be a candidate 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 location parameters 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:

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

with $h = 1, \ldots, H$ and Σ_h is positive definite, $\eta_h \geq 0$ and $\sum_{h=1}^{H} \eta_h = 1$. We further restrict ν_h such that $\nu_h \geq 0.01$. Lower and upper bounds for the degrees of freedom parameter ν_h can be defined using an optional input to the function `MitISEM` (e.g., demanding $\nu_h > 2$).

MitISEM is a new approach that may be compared with the AdMit method (Hoogerheide, Kaashoek, and Van Dijk 2007b), implemented in Ardia, Hoogerheide, and Van Dijk (2009, 2017). 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, MitISEM minimizes the Kullback-Leibler divergence between target and candidate densities, while AdMit aims at minimizing the variance of the IS estimator, or the variance of the IS weights.

Second, in MitISEM all mixture parameters are optimized jointly by means of the relatively quick EM algorithm, while 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. That is, AdMit optimizes only mixture component weights. MitISEM implies a large reduction of the computing time in the approximation procedure, and is expected to lead to a better candidate in most applications. Therefore the MitISEM method may be considered to be a substitute for the AdMit method, rather than an accompanying method.

Third, as shown in Hoogerheide *et al.* (2012), AdMit requires the joint target density kernel, whereas MitISEM requires only candidate draws and importance weights. This implies that AdMit can not be applied partially to the marginal and conditional posterior densities 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, where one cannot directly generate random draws from in an easy manner, importance sampling is based on draws from a different density, the so-called *candidate* or *importance*

density $g(\theta)$, which is easy to simulate from and which is a reasonable approximation to $f(\theta)$. Given this indirect sampling, instead of direct sampling from $f(\theta)$, one needs a correction step. That is, the draws from the candidate density are *weighted* according to the importance sampling (IS) weights that are ratios of target over candidate density. 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 variance of the weights should be bounded (Geweke 1989). The finite sample accuracy of the estimator improves 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 where the candidate is much smaller than the target and this is a region that is covered too little by the candidate density.

Parallel Processing of Computations

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 method 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 Geweke and Durham (2011) for a very novel approach. We are currently exploring the idea to make use of paralleled computing in MitISEM. We comment on this possibility in Section 4.

2.2. Background on EM and our use of this algorithm

The EM algorithm (Dempster, Laird, and Rubin 1977) is a method 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.

We emphasize that during the first stage of the method (in which the candidate density is optimized) we do not have draws from the posterior but instead have draws from a previously chosen candidate and corresponding importance weights. As a consequence, we make use of an *importance weighted* EM algorithm. As shown in Hoogerheide *et al.* (2012), in the MitISEM approach, we minimize the estimated Kullback-Leibler divergence, which implies that we maximize the weighted average of 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.

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 density $IG(\nu_h/2, \nu_h/2)$. For a more extensive explanation of this mixture of Student t densities, see e.g., Peel and McLachlan (2000).

2.3. The IS weighted EM algorithm

We reemphasize that in the 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 make use of EM to find the optimal mixture of Student t densities for a given set of draws from a previous candidate (and their corresponding weights). We apply an *IS-weighted* EM algorithm to these candidate draws instead of a regular EM algorithm to posterior draws (obtained by applying the Metropolis-Hastings method to these candidate draws), since the former 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.

In Hoogerheide *et al.* (2012) the different steps of the IS-weighted EM algorithm in our case are derived. Here we summarize the steps for the mixture of Student t densities. Note that when one substitutes for the weight W^i the value $W^i = 1$ in Equations 8–10, then one is back in the case of a regular EM algorithm (without IS weighting). In our case the L-th expectation step for the mixture of Student t densities is specified 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 $\rho_h^i \equiv (\theta^i - \mu_h)^\top \Sigma_h^{-1} (\theta^i - \mu_h)$, $\psi(.)$ 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 Equation 4 to Equation 7, parameters of each mixture component are updated using the first order conditions of the expectation of the objective function in the maximization step:

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

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

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

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 densities and not the degrees of freedom, because there is no closed-form solution for the latter. We do optimize the degrees of freedom parameter ν_h during the EM procedure to obtain a better approximation of the target density. 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

Algorithm 1.

The MitISEM approach for obtaining an approximation to a target density:

(0) *Initialization*: Simulate draws $\theta^1, \ldots, \theta^N$ from a 'naive' candidate distribution with density g_{naive}, which is obtained as follows. First, we simulate candidate draws from a Student t distribution with density g_{mode}, where the mode is taken equal to the mode of the target density and scale matrix equal to minus the inverse Hessian of the log-target density (evaluated at the mode), and where the degrees of freedom are chosen by the user. Second, the mode and scale of g_{mode} are updated using the IS weighted EM algorithm. Note that g_{naive} is already a more advanced candidate than the commonly used g_{mode}; g_{mode} typically yields a substantially worse numerical efficiency than g_{naive}.

(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 with density $g_0 = g_{adaptive}$, 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 the distribution that corresponds with 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 of the candidate density's quality by inspecting the IS weights* and return to step (3) unless the algorithm has converged.

In step (0), we have specified a novel robustification by updating the initial proposal density using an IS weighted EM step compared to the MitISEM algorithm proposed in Hoogerheide *et al.* (2012). This step improves the algorithm when the initial mode and Hessian estimation in step (0) is poor. If these initial mode and Hessian estimates are obtained by grid-search algorithms, the estimates can be poor due to local maxima issues in the target density. In addition, first component of the candidate density can be user-specified, e.g., using another optimization algorithm as discussed in Ardia *et al.* (2009), and the accuracy of these optimization algorithms depend on the target density properties. The additional robustification step we define eliminates extreme dependence of results to user-specified values especially in case these user-specified values are not accurate.

Step (1) can be seen as an intermediate step which quickly tries to improve the initial candidate density g_0. If during the EM algorithm, a scale matrix Σ_h of a Student t component becomes (nearly) singular, then this h-th component is removed from the mixture. Also if during the EM algorithm, a weight η_h becomes very small, 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 *et al.* (2012), who use the candidate from MitISEM for importance sampling or the independence chain MH method. Zellner, Ando, Baştürk, Hoogerheide, and Van Dijk (2014), 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 of rejection sampling. 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 ν_{H+1} and η_{H+1} are at each iteration set at 1 and 0.10, respectively. That is, 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 target density kernel is 0 (i.e., draws outside the 'allowed' parameter region). 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 small, especially during 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.

Approximating the Gelman-Meng density using MitISEM

We illustrate the MitISEM approach using a non-elliptical, bivariate density function proposed by Gelman and Meng (1991). The target density kernel is:

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

where $(x_1, x_2)^\top$ 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 *et al.* 2017):

```
R> 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)
+  }
```

where an input `log` is added in this function for the following reason. We evaluate the IS weights in the following way. First, we compute the logarithm IS of the weight ($\log(W^i)$) as the logarithm of the target density kernel minus the logarithm of the candidate density. Second, we subtract the maximum of the $\log(W^i)$. Third, we take the exponent to obtain W^i. If we would directly evaluate the IS weight as the ratio of the target density kernel and candidate density, then there would often be numerical problems (in the sense of an 'underflow' or 'overflow', where all IS weights are stored as 0 or ∞). Note that rescaling the IS weights does not matter in Equations 8–11, as W^i always occurs in both a numerator and a denominator of a ratio. So rescaling the IS weights does not matter for the MitISEM algorithm. Obviously, the scale does matter for the evaluation of marginal and predictive likelihoods. In these procedures we do keep track of the scale when rescaling. On the other hand, we may want to plot the target density and the MitISEM approximation. In that case we want to plot the actual densities, not their logarithm. Therefore, the 'log' argument is optional.

MitISEM approximation to the target kernel is obtained using function `MitISEM` and an initial point, `mu0`, for the optimization in step (0), where with each additional mixture component,

the approximation to the target kernel becomes more accurate. This accuracy is measured by the standard deviation of IS weights in MitISEM steps. The MitISEM approximation and the evolution of the IS weights with each additional mixture component can be obtained as follows:

```
R> set.seed(1234)
R> mu0 <- c(3, 4)
R> app.MitISEM <- MitISEM(KERNEL = GelmanMeng, mu0 = mu0, control =
+    list(trace = TRUE, Hmax = 10))

 H METHOD TIME       CV IS weights std.dev.
 1   BFGS    0 2.795266        0.0002795266
 H METHOD TIME       CV IS weights std.dev.
 1   BFGS 0.00 2.795266        0.0002795266
 1  IS-EM 0.37 2.295592        0.0002295592
 H METHOD TIME        CV IS weights std.dev.
 2  IS-EM 6.66 0.4274236        4.274236e-05
 H METHOD TIME        CV IS weights std.dev.
 3  IS-EM 6.13 0.3567865        3.567865e-05
 H METHOD TIME        CV IS weights std.dev.
 4  IS-EM 3.74 0.3281487        3.281487e-05

R> mit.optcomp <- app.MitISEM$mit
R> Hvalues <- app.MitISEM$summary[,1]
R> ISwgt <- app.MitISEM$summary[,5] * 10^4
R> plot(Hvalues, ISwgt, type = "b", xlab = "# of mixture components",
+    ylab = "IS weights (x 10000)", xaxt = "n",
+    main = paste("IS weights from MitISEM approximation with ",
+    max(Hvalues), " comp." , sep = ""))
R> axis(1, at = sort(unique(Hvalues)), labels = sort(unique(Hvalues)))
```

Figure 1 shows the target density kernel for the Gelman-Meng function with a 'banana' shape and step-by-step approximations to this kernel using MitISEM approximations. The top panel in the figure shows that the target density has a 'banana' shape with two modes. The MitISEM approximation with a single Student t component and simulated data from this approximation are shown in the second panel of Figure 1. The obtained approximation with a single Student t candidate has a single mode, and a large fraction of simulated points are far from the high density region, or the banana shape, of the target density. The MitISEM candidate with 2 mixture components and simulated data from this approximation are shown in the third panel of Figure 1. Even with this relatively low number of mixture components, the contours of the MitISEM approximation are similar to the contours of the target density. Furthermore, simulated points are concentrated around the high density region of the target density. This concentration increases with the addition of the third and fourth mixture components, but the biggest gain is achieved by increasing the number of mixture components from one to two.

Gains from each additional mixture component in the MitISEM approximation are presented in detail in Figure 2, where the approximation accuracy is measured by the standard deviation

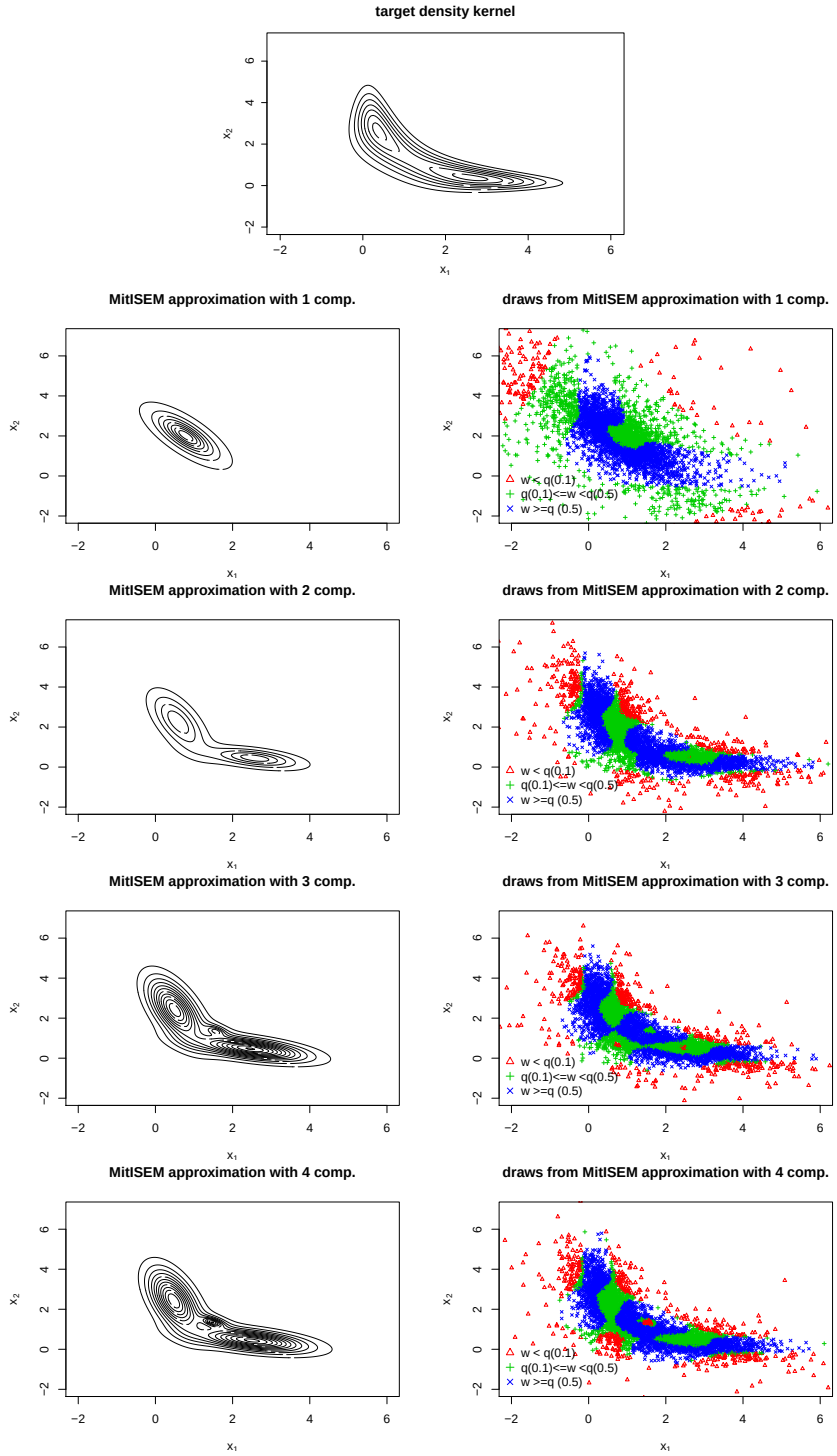


Figure 1: Evolution of the MitISEM candidate for the Gelman-Meng target density with a banana shape. The figure shows the target density kernel (top panel), the MitISEM approximation to the target density kernel, and draws from this approximation for the Gelman-Meng density with a banana shape in Section 3.1, for MitISEM approximation with one (second row) to four (fifth row) Student t components. MitISEM approximations are obtained using 10^5 draws, and 5×10^3 draws from the approximations are plotted. For draws from MitISEM approximations, w denotes the IS weights of draws where weights are normalized to have mean 1. $q(x)$ denotes the $x \times 100$ percentage quantile of IS weights from all draws.

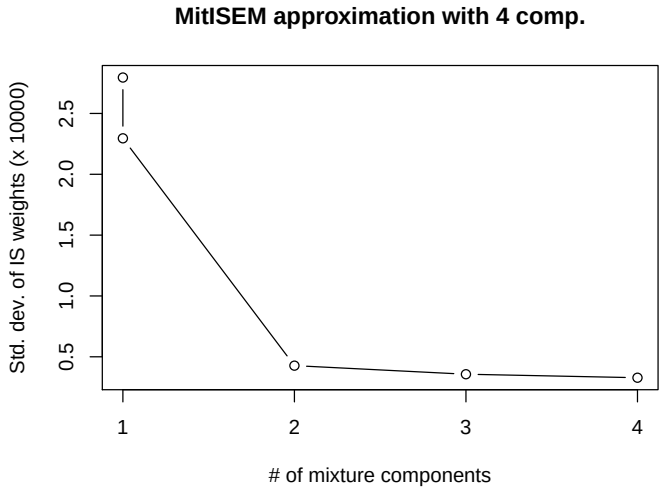


Figure 2: IS weights for candidate distributions obtained after 1, 2, 3 and 4 iterations of the MitISEM algorithm. The figure shows standard deviations of IS weights for the step-by-step MitISEM approximation in Figure 1. Note that there are two candidate densities consisting of a single Student t density: g_{mode} (around one of the two modes of the target density) and g_{naive} (obtained by the IS-EM algorithm), which lead to the highest and second highest standard deviation.

of IS weights from the MitISEM approximation, for each step of the MitISEM algorithm.[2] The MitISEM algorithm suggests 4 mixture components for the approximation. Note that a smaller standard deviation in IS weights implies that the MitISEM candidate and the target kernel are closer to each other. In terms of these IS weights, the largest gain is obtained with the addition of the second Student t component as was already concluded above. The third and the fourth components of the MitISEM approximation provide relatively smaller gains compared to the second component.

2.5. Sequential MitISEM

This subsection presents the algorithm 'sequential MitISEM', proposed by Hoogerheide *et al.* (2012) which applies MitISEM in a sequential manner, so that the candidate density approximating the target density, typically used for posterior and predictive 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 fact that the posterior for $y_{1:T+1} = \{y_1, \ldots, y_{T+1}\}$ is often 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 density if the previous candidate is a good approximation to the posterior for the updated data. Even if the 'old' candidate is not a good approximation to the posterior for the updated data, it may suffice to perform an update using the IS-weighted EM algorithm, keeping the

[2] R scripts for the replication of Figure 1 and Figure 2 are provided in the replication materials of the paper.

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 in the MitISEM method 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 $y_{1:T}$ (the data set at 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 too small, i.e., if the probability of one component is close to zero. The default tolerance for the required mixing probability is 0, and a mixture component is removed from the MitISEM approximation if it has a 0 probability. Hence the number of Student t components is not necessarily monotonically increasing over time. This criterion for the removal of a mixture component can be altered by the user through an optional input to the function `MitISEM`. 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 three domains

In the following subsections, we make use of the MitISEM and the sequential MitISEM methods in order to deal with distributions that have non-elliptical density contours in three domains of applications:

(i) *Approximating a specific class of well-known, non-elliptical densities* in Section 3.1. Here, we continue to analyze the conditionally normal density of Gelman and Meng (1991), which can have non-elliptical, and even distinctly bi-modal, shapes in the joint density depending on specific values of the density function parameters. Note that this is not a posterior density.

(ii) *Approximating posterior densities* of a class of models popular in financial econometrics in Section 3.2. We consider a standard GARCH model and a mixture GARCH model (for S&P 500 data), which are classes of models extensively used in financial practice. We further consider an instrumental variables (IV) model and compare the approximation performance using MitISEM candidate with the griddy Gibbs sampler in Ritter and Tanner (1992).

(iii) *Approximating model probabilities* using the concept of predictive likelihoods in Section 3.3. We consider a mixture GARCH model and an IV model. The latter one using income-education data. Both GARCH and IV models yield non-elliptical distributions for posterior and predictive densities.

For cases (ii) and (iii) 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.

For all cases, 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 density (due to the low degrees of freedom parameter 1) reduce the probability that relevant parts of the target density 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 multi-modal target densities. In Section 3.2, we also compare the MitISEM method with the AdMit method implemented in Ardia *et al.* (2017), which also aims to construct a candidate density as a mixture of Student t densities, in terms of approximation accuracy and computing time.

3.1. Approximating densities: The Gelman-Meng function

We continue the use of the MitISEM algorithm to approximate the Gelman-Meng density presented in Section 2.4. Here we compare the MitISEM approximation's speed and accuracy with a 'naive' approximation of the Gelman-Meng density. Next, we show for the case of a 'distinctly' bi-modal Gelman-Meng density, the failure of a simple Gibbs sampler to yield accurate results even when the sample is relatively large.

Gelman-Meng density with a banana shape

For the Gelman-Meng target density, we set $A = 1, B = 0, C_1 = C_2 = 3$ in Equation 12, where this parameter setting leads to a non-elliptical *banana shape* in the target kernel, as shown in the top-left panel in Figure 3. We compare the performance of the MitISEM approach with a 'naive' density. We note that the MitISEM approximation to the target density is obtained as before, using the target density function in Section 2.4 and function `MitISEM`:

```
R> set.seed(1111)
R> mu0 <- c(3, 4)
R> App.GM <- MitISEM(KERNEL = GelmanMeng, mu0 = mu0)
```

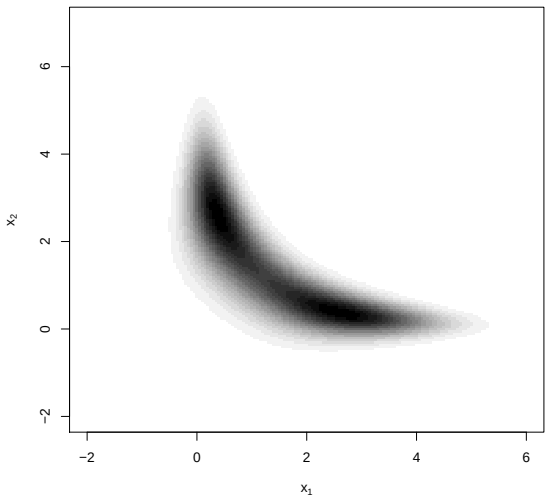
The output of the function `MitISEM` is a list. The first component is `CV`, a vector containing the coefficient of variation at each step of the adaptive fitting procedure. The second component is `mit`, a list consisting of the modes (`mu`), scale matrices (`Sigma`), degrees of freedom parameters (`df`) and mixing probabilities (`p`) of the mixture of Student t densities 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, where only the mode and scale are optimized in the MitISEM algorithm:

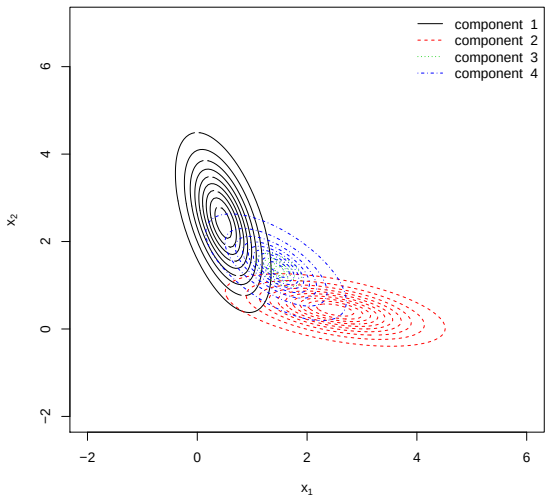
```
R> control <- list(optim.df = FALSE, Hmax = 1)
R> app.Naive <- MitISEM(KERNEL = GelmanMeng, mu0 = mu0, control = control)
```

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

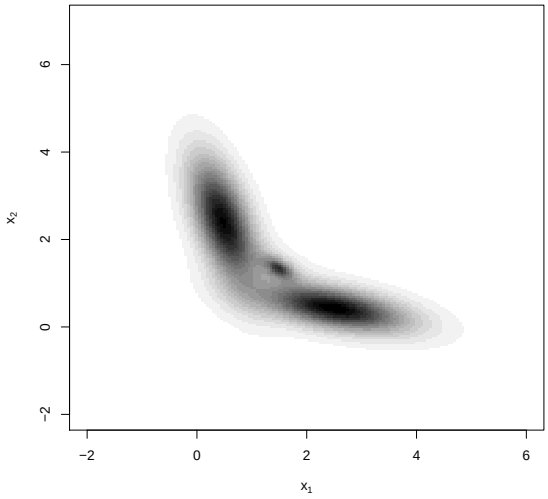
```
R> mit <- App.GM$mit
R> x1 <- seq(-2, 6, 0.05)
R> x2 <- seq(-2, 7, 0.05)
R> H <- length(mit$p)
R> Mitcontour <- function(x1, x2, mit, log = FALSE) {
+    dmvgt(cbind(x1, x2), mit = mit, log = log)
+  }
R> for (h in 1:H) {
+    mit.h <- mapply(function(x)(as.matrix(x)[h,]), mit, SIMPLIFY = FALSE)
+    mit.h$mu <- matrix(mit.h$mu, nrow = 1)
+    mit.h$Sigma <- matrix(mit.h$Sigma, nrow = 1)
+    it.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 = expression(x[1]), ylab = expression(x[2]),
+      main = "MitISEM approximation")
+ }
R> legend("topright", paste("component", 1:H), lty = 1:H, col = 1:H,
+    bty = "n")
```



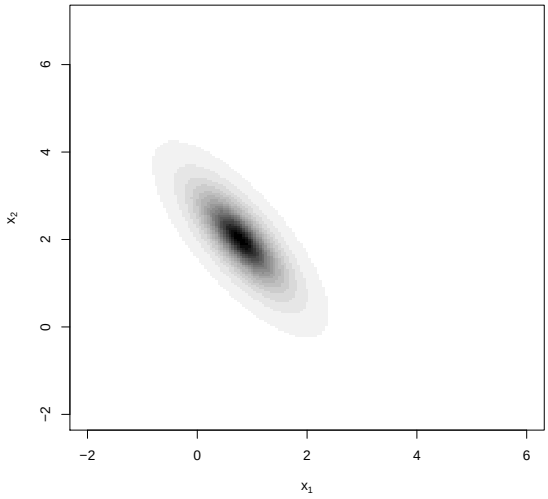
Target kernel



Composition of the MitISEM candidate



MitISEM candidate (4 components)
CoV: 0.33, time: 17.07 seconds



Naive candidate
CoV: 2.28, time: 1.20 seconds

Figure 3: Banana-shaped 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 density with $A = 1, B = 0, C_1 = C_2 = 3$.

For both approximations we use $N = 10^4$ draws to form the mixture components. Figure 3 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 density 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 six times lower for the MitISEM candidate.

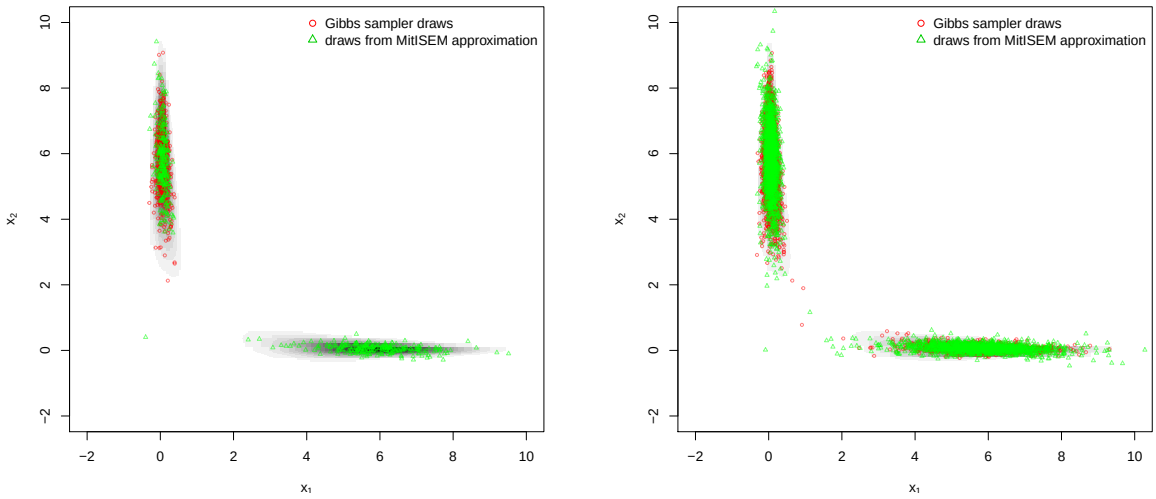


Figure 4: Comparison of simulations using the MitISEM approximation and the Gibbs sampler for Gelman-Meng density with $A = 1, B = 0, C_1 = C_2 = 6$. The left panel shows 500 draws from both samplers, the right panel shows 10000 draws from both samplers.

Gelman-Meng density with a distinctly bi-modal shape

In this subsection, we simulate draws from a Gelman-Meng distribution using the Metropolis-Hastings algorithm with the MitISEM candidate, and compare these simulations with the simulated draws from the Gibbs sampler. We specifically show that the simulated points using the Gibbs sampler fail to cover the whole domain of the Gelman-Meng density. For this comparison, we consider a Gelman-Meng density in Equation 12 with two distinct modes, with parameters $A = 1, B = 0, C_1 = C_2 = 6$.

Figure 4 shows simulated points from this density using the MitISEM approximation and using the Gibbs sampler, based on 500 draws (left panel) and 10000 draws (right panel). The MitISEM approximation to the Gelman-Meng density is obtained using only 1000 draws. Shaded areas in the figure correspond to the high-density regions of the Gelman-Meng function. The left panel of Figure 4 shows that using the MitISEM candidate and a relatively small number of simulations, we obtain points from both modes of the density. In the left panel of Figure 4, the Gibbs sampler fails to 'cover' the second mode. Note that the MitISEM approximation is obtained using merely 1000 draws for the approximation. The MitISEM approximation would become even more accurate if a larger number of draws would be used for density approximation. In this case, the outperformance of the MitISEM method compared to the Gibbs sampler would be even more pronounced. From this comparison, we conclude that simulated data points and inference based on these simulations are erroneous using the Gibbs sampler even for a large number of draws from the density.

3.2. Approximating posterior densities: GARCH and IV models

In this subsection the MitISEM approach is applied to the posterior density of two GARCH models and an instrumental variable (IV) model using data from Card (1995).

The standard GARCH model and its extensions may adequately capture changing volatility patterns, but the likelihood function, hence the posterior density under an uninformative

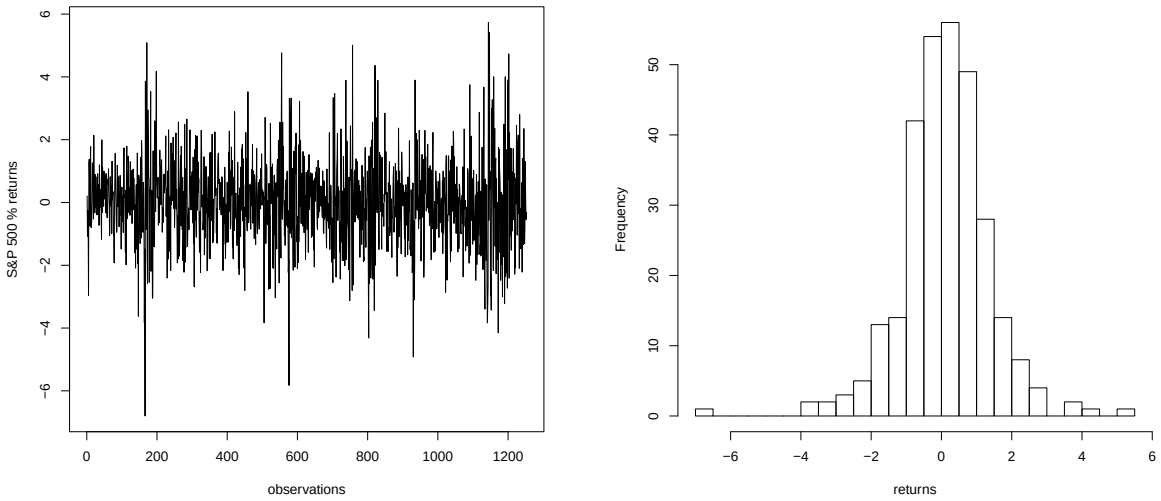


Figure 5: Daily log-returns of the S&P 500 index for the period from 1998-01-02 to 2002-12-26.

prior may have non-elliptical shapes (Zivot 2009). For the applications of GARCH models we use the S&P 500 index percentage returns (100 times the change of the closing price) from 1998-01-02 to 2002-12-26. Figure 5 shows the returns data and their histogram. 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.

Approximating posterior densities: A standard GARCH(1,1) model

We first illustrate the use of the MitISEM approach for the Bayesian estimation of the standard GARCH model (Bollerslev 1986) for the *S&P* 500 data. An extended two-component Gaussian Mixture GARCH (1,1) model (Ausín and Galeano 2007), which possibly leads to more irregular posterior densities, is considered afterwards.

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

$$y_t = \mu + \sqrt{h_t}\,\varepsilon_t, \tag{13}$$

$$h_t = \omega + \alpha(y_{t-1} - \mu)^2 + \beta h_{t-1}, \tag{14}$$

$$\varepsilon_t \sim N(0,1) \text{ i.i.d.} \tag{15}$$

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 this example.

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 $\theta = (\omega, \beta, \alpha, \mu)$, 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:

```
R> 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] <- 0
+    if (!log) r2 <- exp(r2)
+    cbind(r1, r2)
+  }
R> 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 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 candidate that is a mixture of Student t distributions. Specifically, the mixture of Student t distribution obtained from the MitISEM candidate is used as an input, `mit`, for functions `AdMitIS` or `AdMitMH` for posterior inference. An R code to obtain posterior parameters of the GARCH model is provided below, where we use the R package **tseries** (Trapletti and Hornik 2017) to extract S&P 500 data.

```
R> library("tseries")
```

```
R> library("AdMit")
R> prices <- as.vector(get.hist.quote("^GSPC", quote = "AdjClose",
+    start = "1998-01-02", end = "2002-12-26"))
R> data <- 100 * (prices[-1] - prices[-length(prices)]) /
+    (prices[-length(prices)])
R> plot(data, xlab = "observation", ylab = "S&P500 % returns")
R> hist(data, xlab = "returns")
R> theta <- c(0.08, 0.86, 0.02, 0.03)
R> names(theta) <- c("omega", "beta", "alpha", "mu")
R> h1 <- var(data)
R> set.seed(1111)
R> app.GARCH <- MitISEM(KERNEL = post.GARCH, mu0 = theta, h1 = h1,
+    data = data, control = list(trace = TRUE))

1 1   BFGS  0.76 1.1352420        1.135242e-04
2 1  IS-EM 28.70 0.7567956        7.567956e-05
3 2  IS-EM 57.91 0.4105256        4.105256e-05
4 3  IS-EM 56.76 0.3864224        3.864224e-05

R> IS.GARCH <- AdMitIS(N = 10e4, KERNEL = post.GARCH,
+    mit = app.GARCH$mit, data = data, h1 = h1)
R> print(IS.GARCH)

$ghat
[1] 0.08884915 0.84851733 0.10637618 0.03354568
$NSE
[1] 1.158643e-04 1.133241e-04 7.695703e-05 1.187978e-04
$RNE
[1] 0.6974687 0.7058110 0.7292880 0.8363404
```

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 (where the BFGS method has been used to compute the mode of the target density); `TIME` gives the computing time required for this optimization; `CV` gives the coefficient of variation of the importance sampling weights; `std.dev.` gives the standard deviation of the IS 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` $\times N$ 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.

Approximating posterior densities: A mixture GARCH(1,1) model

In this subsection the MitISEM approach is applied to the non-elliptical posterior density 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

$$y_t = \mu + \sqrt{h_t}\,\varepsilon_t, \tag{16}$$

$$h_t = \omega + \alpha(y_{t-1} - \mu)^2 + \beta h_{t-1}, \tag{17}$$

$$\varepsilon_t \sim \begin{cases} N(0, \sigma^2) & \text{with probability } \rho, \\ N(0, \sigma^2/\lambda) & \text{with probability } 1 - \rho, \end{cases} \tag{18}$$

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 $\theta = (\omega, \lambda, \beta, \alpha, \rho, \mu)$, we have a uniform prior on $(0, 1] \times (0, 1) \times [0, 1) \times [0, 1) \times (0.5, 1] \times [-1, 1]$ with $\alpha + \beta < 1$ which implies covariance stationarity.

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

```
R> 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(2)
+    r2[r1 == TRUE] <- tmp
+    if (!log) r2 <- exp(r2)
+    cbind(r1, r2)
+  }
R> 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)
+    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**, where the MitISEM candidate is used as an input to `AdMitIS`.

```
R> prices <- as.vector(get.hist.quote("^GSPC", quote = "AdjClose",
+    start = "1998-01-02", end = "2002-12-26"))
```

```
R> data <- 100 * (prices[-1] - prices[-length(prices)]) /
+    (prices[-length(prices)])
R> mu0 <- c(0.08, 0.37, 0.86, 0.03, 0.82, 0.03)
R> names(mu0) <- c("omega", "lambda", "beta", "alpha", "rho", "mu")
R> h1 <- var(data)
R> set.seed(1234)
R> app.mGARCH <- MitISEM(KERNEL = post.mGARCH, mu0 = mu0, h1 = h1,
+    data = data)
R> app.mGARCH$summary

  H METHOD  TIME        CV IS weights std.dev.
1 1   BFGS  2.03 2.1170292        2.117029e-04
2 1  IS-EM 25.97 1.5332795        1.533280e-04
3 2  IS-EM 56.01 1.0081923        1.008192e-04
4 3  IS-EM 60.32 0.8610536        8.610536e-05
5 4  IS-EM 58.75 0.7905704        7.905704e-05

R> IS.mGARCH <- AdMitIS(N = 10e4, KERNEL = post.mGARCH,
+    mit = app.mGARCH$mit, data = data, h1 = h1)
R> print(IS.mGARCH, 2)

$ghat
[1] 0.079 0.369 0.862 0.099 0.788 0.029
$NSE
[1] 1.4e-04 3.7e-04 1.3e-04 9.2e-05 5.8e-04 1.4e-04
$RNE
[1] 0.47 0.51 0.50 0.51 0.46 0.61
```

MitISEM method and the AdMit method (for which no output is shown above, since our main focus is on the novel **MitISEM** package) yield an approximation that is a mixture of 7 and 4 Student t components. The conditional posterior density kernel of parameters (ρ, λ) given that the other four parameters are equal to their (estimated) posterior means and the approximations by three methods are shown in Figure 6. 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 density 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. First, if we require simulation results with a certain very high precision, then

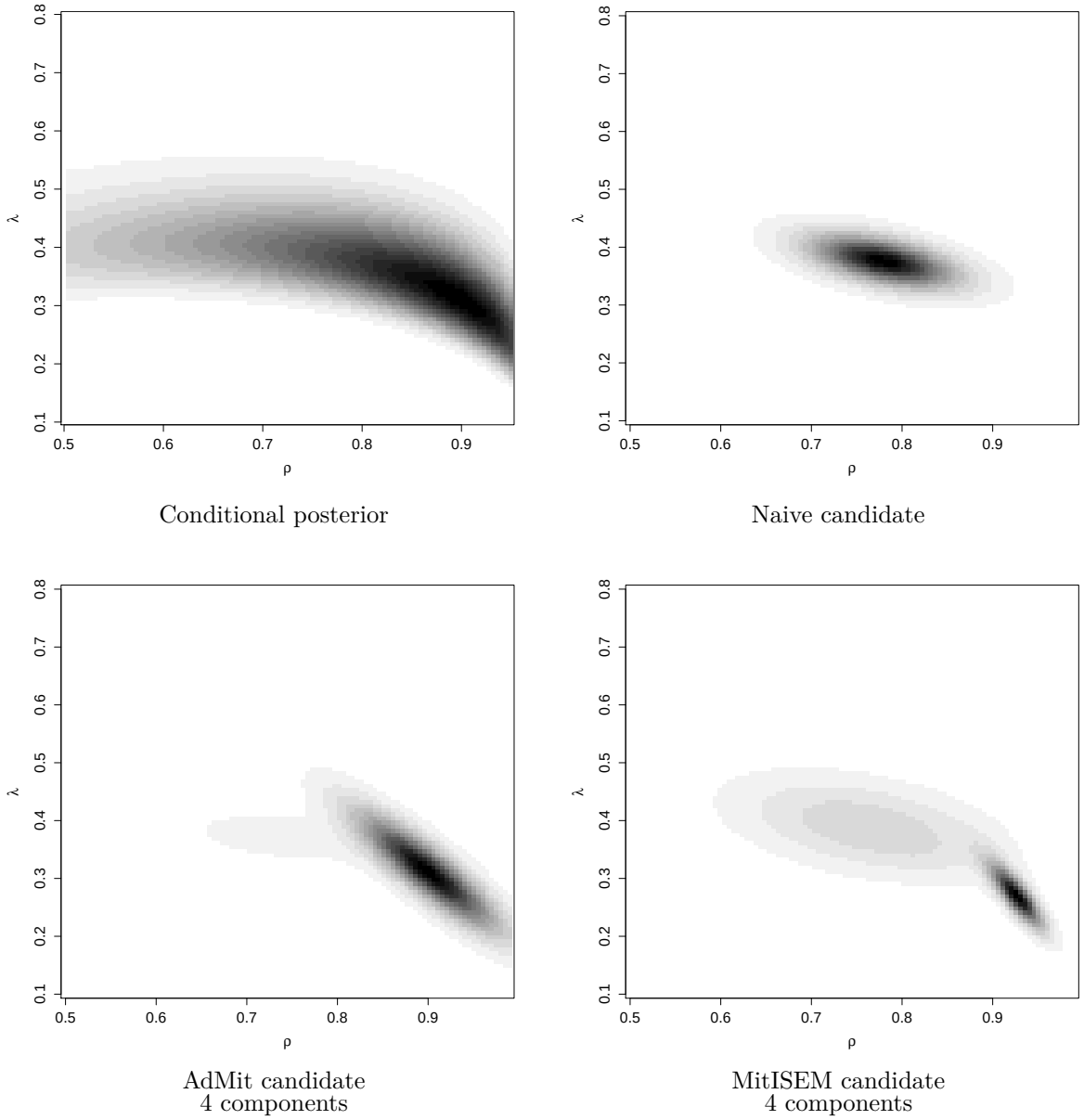


Figure 6: Conditional posterior density kernel of (ρ, λ) given posterior means of the other parameters $(\omega, \beta, \alpha, \mu)$ in the mixture GARCH(1,1) model together with the naive, AdMit and MitISEM approximations.

MitISEM would obviously require much fewer draws than the 'naive' and AdMit approximations, 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 compared to 'naive' and AdMit approximations implies that there is less risk that a relevant part of the target density is 'missed', for example in case of a multi-modal target density, which would possibly cause substantially biased results for the other methods.

Algorithm	# t Components	Time (seconds)	CoV
AdMit	4	144.30	2.61
Naive	1	19.40	2.47
MitISEM	7	358.57	0.72

Table 1: Summary of naive, AdMit and MitISEM candidates for the mixture GARCH(1,1) model for S&P 500 data. The table reports the algorithm for obtaining the candidate distribution, the number of Student t components (# t), time (in seconds) and CoV (coefficient of variation of the IS weights) for all compared algorithms. Candidates are constructed using 10^4 draws.

	Posterior mean			NSE $\times 100$		
	AdMit	Naive	MitISEM	AdMit	Naive	MitISEM
ω	0.08	0.08	0.08	0.09	0.06	0.07
λ	0.38	0.37	0.37	0.25	0.24	0.12
β	0.86	0.86	0.86	0.09	0.06	0.06
α	0.10	0.10	0.10	0.06	0.04	0.03
ρ	0.78	0.78	0.78	0.43	0.32	0.21
μ	0.03	0.03	0.03	0.10	0.07	0.04

Table 2: Estimated 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.

Approximating posterior densities: An IV model

In this subsection we apply the MitISEM algorithm to an instrumental variables (IV) regression model. We first make use of a set of simulated data and report the accuracy of posterior inference from the Metropolis-Hastings and importance sampling algorithms based on a MitISEM approximation to the posterior density and compare the results with those obtained using the griddy Gibbs sampler of Ritter and Tanner (1992). The griddy Gibbs algorithm that we specify uses the inverse CDF technique to obtain posterior draws for each parameter.[3] Second, we use empirical data from Card (1995) and apply the MitISEM algorithm to an IV model that describes the effect of years of education on earned income.

The IV model with one explanatory endogenous variable and p instruments is defined by Bowden and Turkington (1990):

$$y = x\beta + \varepsilon, \tag{19}$$
$$x = z\Pi + v, \tag{20}$$

where the scalar β and the $p \times 1$ vector Π are model parameters, y is the $N \times 1$ vector of observations on the dependent variable income, x is the $N \times 1$ vector of observations on the endogenous explanatory variable, education, z is the $N \times p$ matrix of observations on the instruments. All variables are demeaned, i.e., both model equations do not include a constant term. The disturbances are assumed to come from a normal distribution:

[3] A replication routine for this simulation study is provided in replication materials of the paper.

$(\varepsilon^\top,\ v^\top)^\top \sim NID(0, \Sigma \otimes I)$, where $\Sigma = \begin{pmatrix} \sigma_{11}^2 & \rho\sigma_{11}\sigma_{22} \\ \rho\sigma_{11}\sigma_{22} & \sigma_{22}^2 \end{pmatrix}$ is a positive definite and symmetric 2×2 matrix, I denotes the $N \times N$ identity matrix and $\otimes$ denotes the Kronecker product operator.

'Endogeneity' of the variable x arises from possible correlation between the disturbances, given as $\rho \equiv \text{cor}(\varepsilon_i, v_i)$ for $i = 1, \ldots, N$. 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. We note that in case 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 disturbances is different from zero.

Under conventional flat priors, it can be shown that 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. For more details on the derivation we refer to Zellner *et al.* (2014). We specify a Jeffreys prior which leads to a proper posterior density, see e.g., Hoogerheide, Kleibergen, and Van Dijk (2007a) for the derivation. The posterior density of the model in Equations 19–20 under the Jeffreys prior can be implemented as follows:

```
R> Jeff.prior <- function(beta, Pi, Sigma11, rho, Sigma22) {
+    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))
+  }
R> post.IV <- function(theta, data, log = TRUE) {
+    if (is.vector(theta)) theta <- matrix(theta, nrow = 1)
+    y <- data[,1]
+    x <- data[,2]
+    z <- data[,3]
+    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)
+    fn_aux <- function(theta_aug, y, x, z) {
+      tmp <- matrix(c(theta_aug[3], theta_aug[4], theta_aug[4],
+        theta_aug[5]), 2, 2)
+      detfac <- log(det(tmp))
+      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))
+     }
+     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
+     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)
+   }
```

As mentioned, we first apply the MitISEM method to artificial data for the case of an IV model. We simulate 300 observations from the IV model in Equations 19 and 20, with 'true' parameter values $(\beta, \pi, \sigma_{11}^2, \rho, \sigma_{22}^2) = (0.73, 0.06, 0.21, -0.43, 0.17)$. These values correspond to the posterior means of the parameters in the real data application, using Card (1995). Posterior results from MH and IS methods using the MitISEM approximation to the posterior density are obtained using the functions `AdMitMH` and `AdMitIS` in the R package **AdMit**. Specifically, the MitISEM candidate is used as an input to `AdMitMH` and `AdMitIS` functions. These two functions perform sampling from the MitISEM candidate and posterior inference using the MitISEM candidate. The MitISEM approximation to the posterior density is based on 10000 draws, leading to a 3-component mixture of Student t densities, i.e., the posterior density is highly non-elliptical. For the Metropolis-Hastings algorithm we use 10000 burn-in draws and 10000 posterior draws. Importance sampling results are based on 10000 draws. The alternative method, the griddy Gibbs sampler, is based on 10000 posterior draws, and 10000 burn-in draws. For each parameter draw using the griddy Gibbs sampler, 200 equi-distant grid points are taken on the parameter space $\beta \in [-2, 2]$, $\pi \in [-2, 2]$, $(\sigma_{11}^2, \sigma_{22}^2) \in (0, 2]^2$ and $\rho \in (-1, 1)$.

Estimated posterior means and standard deviations for parameters using the three sampling algorithms are shown in Table 3.[4] The griddy Gibbs sampling results are very different from the MH and IS results using the MitISEM candidate. In particular, posterior draws of the correlation coefficient ρ are concentrated around 0.06 with a relatively small standard deviation compared to results from MH and IS. This indicates that the griddy Gibbs sampler fails to cover the whole domain of the posterior density and it is seen that the MitISEM approximation substantially improves the simulation inference for this example. We also computed two convergence test results for griddy Gibbs draws as well as for MH draws (that are based on a MitISEM approximation to the posterior). Results are obtained using the R package **MCMCpack** (Martin, Quinn, and Park 2011, 2017). The p values are based on a

[4]We emphasize that Bayesian posterior analysis, using a Jeffreys prior, does not necessarily yield posterior means that are equal to the so-called 'true' parameter values. The difference may be due to a flat or skew posterior and/or a relatively small sample.

		β	π	σ_{11}^2	ρ	σ_{22}^2	Time
True values		0.73	0.06	0.21	-0.43	0.17	
Posterior means and standard deviations							
Griddy Gibbs	Mean	0.13	0.05	0.15	0.10	0.16	2728
	Std. dev.	0.09	0.02	0.01	0.10	0.01	
MH	Mean	0.76	0.06	0.22	-0.44	0.17	64
	Std. dev.	0.33	0.02	0.07	0.22	0.01	
IS	Mean	0.76	0.06	0.21	-0.44	0.16	62
	Std. dev.	0.30	0.02	0.06	0.20	0.01	

Table 3: Estimated posterior means and standard deviations of parameters for simulated IV data based on griddy Gibbs sampler and MH and IS algorithms using MitISEM approximation. MitISEM candidate is based on 10000 draws. MH and griddy Gibbs results are based on 10000 posterior and 10000 burn-in draws. IS results are based on 10000 draws. For the griddy Gibbs sampler we use 200 grid points for each parameter. The last column of the table reports the elapsed time for each posterior sampler in minutes. For IS and MH algorithms we report total time, including the time required to construct the MitISEM candidate.

convergence criterion presented in Geweke (1992), where the null hypothesis is the equality of posterior means from the first and last parts of the Markov Chain. At the 5% level, the null hypothesis is not rejected for griddy Gibbs draws as well as for the MH results although the posterior mean estimates are numerically different. We additionally report results for the Heidelberger and Welch (1981) test where the null hypothesis is that the samples of posterior draws come from a stationary distribution. According to this test, β and ρ draws from the griddy Gibbs sampler do not come from a stationary distribution while all draws from MH 'pass' the test. These test results show that these tests are rather sensitive to the exact specification of the test and we recommend the use of multiple tests for convergence assessment.

The relatively poor performance of the griddy Gibbs sampler is clearly shown in Figure 7 and 8, where we present draws from (β, π, ρ) and the traceplot of all parameters from the MH algorithm and the griddy Gibbs sampler. The left panel of Figure 7 shows that the griddy Gibbs sampler leads to many draws of (β, π) that are far from the central part of the posterior while draws from the MH algorithm using the MitISEM approximation are relatively more concentrated in that center. Similarly, on the right panel of Figure 7, relatively more (β, ρ) draws from the griddy Gibbs sampler are far from the important region of the posterior and lie at the bottom right area of the figure compared to MH draws. A reason for this poor performance is the much higher serial correlation between griddy Gibbs draws compared to MH draws. This is shown in Figure 8. Griddy Gibbs draws are strongly serially correlated, particularly for parameters (β, ρ), while MH draws using the MitISEM candidate have very few consecutive draws with same parameter values. We conclude that the griddy Gibbs sampler is much less accurate given the same number of draws compared to IS and MH algorithms using the MitISEM approximation as the candidate density.[5]

We next apply the MitISEM algorithm to approximate the posterior density of the IV model

[5]When the number of draws is increased to 10000 burn-in and 20000 posterior draws, the slight difference in MH and IS parameter estimates disappear, while posterior results for the griddy Gibbs sampler remain similar to those in Table 3.

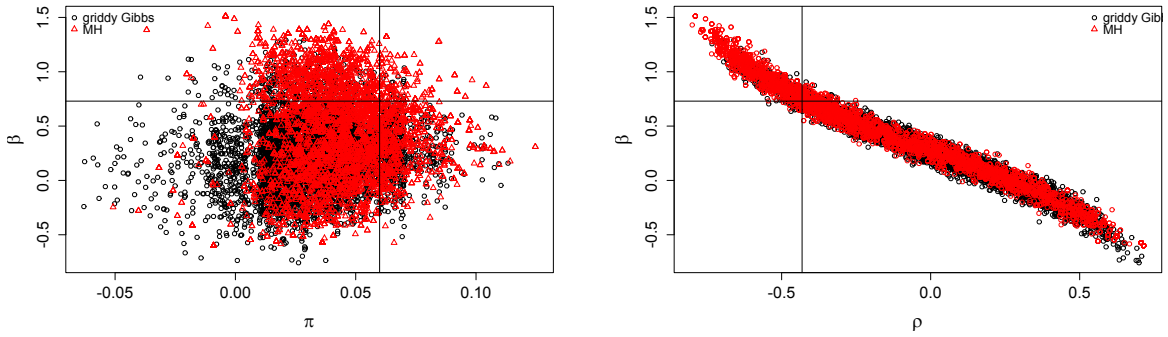


Figure 7: Parameter (β, π) and (β, ρ) draws for simulated IV data based on the griddy Gibbs sampler and the MH algorithm using MitISEM approximation. Horizontal and vertical lines are the true parameter values. Posterior results are obtained as in Table 3.

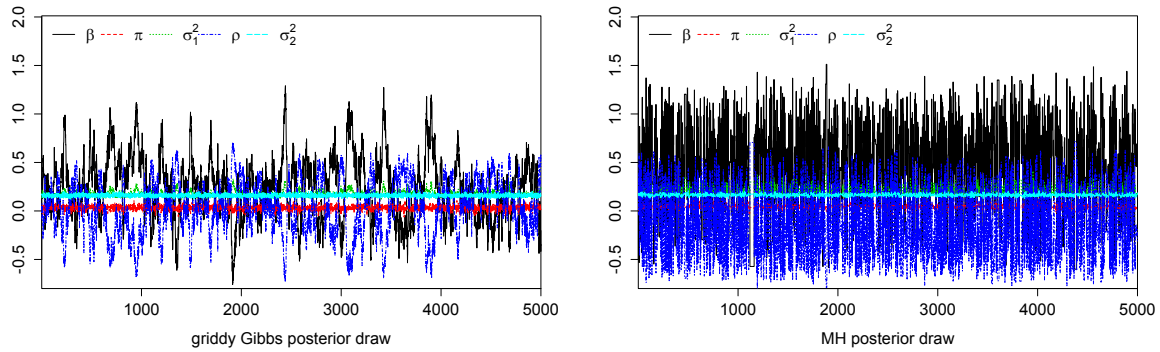


Figure 8: Posterior draws of all parameters for simulated IV data based on the griddy Gibbs sampler and the MH algorithm using MitISEM approximation. Posterior results are obtained as in Table 3.

in Equations 19 and 20 for the Card (1995) data on income and education, and compare the results with those obtained from the griddy Gibbs sampler. 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.[6] 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.

Similar to the simulation study, the griddy Gibbs sampling results are different from the Metropolis-Hastings and importance sampling results using the MitISEM candidate, as seen

[6]Data can be obtained from http://davidcard.berkeley.edu/data_sets/proximity.zip. The data includes several additional covariates compared to the standard model outlined in this section. Relevant variables need to be extracted using the authors' codebook in the website. The R code to prepare the Card (1995) data for the example IV model is supplied in the replication materials.

		β	π	σ_{11}^2	ρ	σ_{22}^2	Time
Griddy Gibbs	Mean	0.11	0.05	0.15	0.12	0.16	23.7
	Std. dev.	0.09	0.02	0.01	0.10	0.01	
MH	Mean	0.73	0.07	0.21	-0.43	0.17	1.1
	Std. dev.	0.28	0.02	0.05	0.20	$0.00^{(*)}$	
IS	Mean	0.73	0.06	0.21	-0.43	0.17	1.1
	Std. dev.	0.28	0.02	0.05	0.20	$0.00^{(*)}$	

Table 4: Estimated posterior means and standard deviations of parameters for the IV model for Card (1995) data obtained using the griddy Gibbs sampler, Metropolis-Hastings algorithm based on the MitISEM candidate and importance sampling based on the MitISEM candidate. MitISEM candidate is based on 10^4 draws. MH and griddy Gibbs results are based on 5000 posterior and 5000 burn-in draws. IS results are based on 10000 draws. For the griddy Gibbs sampler we use 200 grid points for each parameter. The last column of the table reports the elapsed time for each posterior sampler in minutes. For IS and MH algorithms we report total time, including the time required to construct the MitISEM candidate. For values indicated by $(*)$, the standard deviation is 0.0044 at the 4-digit level.

in Table 4. Here, posterior draws from the correlation coefficient ρ are concentrated around 0.12 with again a relatively small standard deviation. This small standard deviation confirms that the griddy Gibbs sampler fails to cover the whole domain of the posterior density and that the MitISEM approximation substantially improves the simulation inference for these data.

3.3. Approximating model probabilities using predictive likelihoods

In this subsection we show how 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 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):

$$p(\tilde{y}|y^*, M_1) = \int p(\tilde{y}|\theta_1, y^*, M_1) p(\theta_1|y^*, M_1) d\theta_1, \tag{21}$$

which is the marginal likelihood if we consider $\tilde{y}$ as 'the data' and $p(\theta_1|y^*, M_1)$, the exact posterior density, i.e., density including the normalizing constant, and the posterior kernel, after observing y^*, as the prior. Using Bayes' rule for this exact posterior density $p(\theta_1|y^*, M_1)$ and substituting into Equation 21 yields

$$p(\tilde{y}|y^*, M_1) = \frac{\int p(y|\theta_1, M_1) p(\theta_1|M_1) d\theta_1}{\int p(y^*|\theta_1, M_1) p(\theta_1|M_1) d\theta_1}, \tag{22}$$

where $p(\theta_1|M_1)$ is the prior on model parameters θ_1 for model M_1. 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. The integral in Equation 22 can be evaluated by numerical integration using θ_1 draws obtained from the Metropolis-Hastings algorithm in package **AdMit** using the *MitISEM* candidate density.

Model probability for M_1 is then calculated using the posterior density in Equation 22

$$p(M_1|y) = p(M_1|\tilde{y}, y^*) = \frac{p(M_1|y^*)p(\tilde{y}|y^*, M_1)}{p(\tilde{y}|y^*)} \propto p(M_1|y^*)p(\tilde{y}|y^*, M_1), \tag{23}$$

where $(p(\tilde{y}|y^*))^{-1}$, which is independent of the model M_1, is the normalizing constant and $p(M_1|y^*)$ is the prior model probability conditional on prior data points y^*. In practice, this prior model probability is often defined independent of prior data points and ensures equal prior model weights.

Approximating model probabilities using predictive likelihood: Mixture GARCH(1,1)

As a first 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' $y^* = (y_1, \ldots y_m)$. Predictive likelihood calculation using the MitISEM approximation is implemented as:

```
R> source("PostmGARCH.R")
R> prices <- as.vector(get.hist.quote("^GSPC", quote = "AdjClose",
+    start = "1998-01-02", end = "2002-12-26"))
R> data <- 100 * (prices[-1] - prices[-length(prices)]) /
+    (prices[-length(prices)])
R> data.ss <- data[1:626]
R> h1 <- var(data)
R> mu0 <- c(0.08, 0.37, 0.86, 0.03, 0.82, 0.03)
R> names(mu0) <- c("omega", "lambda", "beta", "alpha", "p", "mu")
R> set.seed(1234)
R> mit.ss <- MitISEM(KERNEL = post.mGARCH, mu0 = mu0, data = data.ss,
+    h1 = h1, control = list(trace = TRUE))$mit
R> mit.fs <- MitISEM(KERNEL = post.mGARCH, mu0 = mu0, data = data,
+    h1 = h1, control = list(trace = TRUE))$mit
R> PL.mGARCH <- PredLik(N, mit.fs, mit.ss, post.mGARCH, data, data.ss,
+    h1 = h1)
```

where the posterior density function for the mixture of GARCH(1,1) is defined as in Section 3.2, `y.ss` is the training sample of 626 observations for predictive likelihood calculations, the initial variance `h1` of the GARCH model is defined as the sample variance of the data, initial parameters are defined as `mu0` and predictive likelihood calculation is based on `N` draws.

In order to calculate the accuracy of the estimates, we replicate the predictive likelihood calculation 50 times. Table 5 shows simulation results where the average predictive likelihoods and Numerical Standard Errors are calculated from 50 replications. The candidates for all

# t components		Predictive likelihood	
Training sample	Full sample	Mean	NSE
4	3	1.68×10^{-470}	1.28×10^{-472}

Table 5: 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 are based on 50 replications.

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. The MitISEM candidate consists of four and three mixture components for the training sample and the full sample, respectively, indicating highly non-elliptical posterior shapes for both datasets. Despite these irregularities in the posterior densities, the small NSE reported in Table 5 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.

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 in Equation 22). At each time $t = 1204, \ldots, 1252$, 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 $t = 1204, \ldots, 1252$ by weighting these with the estimated model probabilities. Once the posterior kernel is specified, the sequential MitISEM approximations can be obtained as follows:

```
R> prices <- as.vector(get.hist.quote("^GSPC", quote = "AdjClose",
+    start = "1998-01-02", end = "2002-12-26"))
R> data  <- 100 * (prices[-1] - prices[-length(prices)]) /
+    (prices[-length(prices)])
R> mu0 <- c(0.08, 0.37, 0.86, 0.03, 0.82, 0.03)
R> names(mu0) <- c("omega", "lambda", "beta", "alpha", "rho", "mu")
R> h1 = var(data)
R> data.ss <- data[1:floor(length(data) / 2)]
R> MitISEMapp.subsample <- MitISEM(KERNEL = post.mGARCH, mu0 = mu0, h1 = h1,
+    data = data.ss)
R> control.seq <- list(T0 = 1203, tau = 1:20)
R> app.mGARCH.SeqMitISEM <- SeqMitISEM(data, KERNEL = post.mGARCH, mu0 = mu0,
+    control.seq = control.seq, h1 = h1)
```

Table 6 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

# Observations	#t	CoV	Predictive likelihood
1204	3	0.98	3.76×10^{-435}
1205	3	1.01	0.64×10^{-435}
1206	3	0.95	0.83×10^{-436}
1207	3	1.04	1.35×10^{-437}
1208	6	0.99	2.36×10^{-438}
1209	6^*	0.72	3.57×10^{-439}
1210	6	0.72	0.50×10^{-439}
1211	6	0.72	1.04×10^{-440}
1212	6	0.73	2.09×10^{-441}
1213	6	0.85	3.93×10^{-442}
1214	6	0.76	0.97×10^{-442}
1215	6	0.76	1.27×10^{-443}
1216	6	0.75	2.97×10^{-444}
1217	6	0.81	0.75×10^{-444}
1218	6	0.82	1.81×10^{-445}
1219	6	0.72	1.17×10^{-446}
1220	6	0.77	2.59×10^{-447}
1221	6	0.73	2.22×10^{-448}
1222	6	0.74	0.52×10^{-448}
1223	6	0.74	1.43×10^{-449}
Sequential MitISEM steps			
# reused		18	
# adapted		1	
# adapted and extended		1	

Table 6: 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. #t denotes the number of Student t components in MitISEM approximation. (*) indicates that the candidate density from the previous approximation is adapted.

sequential MitISEM algorithm. Note that for the calculation of predictive likelihoods in increased data samples, an 'ad hoc' MitISEM procedure would be applied 20 times, while the sequential MitISEM 'adopts' the candidate density only once for sample size 1208 and 'extends' the candidate density only once for sample size 1209. In the remaining time periods, the candidate is simply 'reused', with minimal computational time. Table 6 shows that the MitISEM approximation using a subsample of the data is 'reused' 18 times, indication approximately 18 times speed gains compared to the standard MitISEM approximation for each sample. Similarly, one MitISEM candidate is only 'adjusted', again providing speed gains compared to employing the whole MitISEM algorithm for each subsample. Note that the CoV values remain low, so that the huge gains in computing time do not lead to a bad quality of the candidate distribution.

Approximating model probabilities using predictive likelihood: IV model

As a third application of predictive likelihood approximations using the MitISEM algorithm, we consider the IV model in Equation 19 and Equation 20 with the Jeffreys prior for the Card

data on income and education described in Section 3.1. We define two models, one treating education as an endogenous explanatory variable (i.e., the IV model in Equation 19 and Equation 20, and the second model treating education as an exogenous explanatory variable (i.e., the simple linear regression model). The linear regression model is a nested model compared to the IV model in Equation 19 and Equation 20 with the parameter restriction $\rho = 0$ and posterior inference for the effect of education on income, β, in this case is based only on Equation 19. 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 null model is a restricted version of the alternative model and the prior densities satisfy the condition that the prior for the restricted model equals the corresponding conditional prior (conditional upon satisfying the restriction) in the unrestricted model. Under that condition the model probabilities can be simplified to:

$$\frac{p(M_0 \mid y)}{p(M_1 \mid y)} = \frac{p(\tilde{y} \mid y^*, M_0)}{p(\tilde{y} \mid y^*, M_1)} \frac{p(M_0)}{p(M_1)} = \frac{p(\rho = 0 \mid \tilde{y}, y^*, M_1)}{p(\rho = 0 \mid y^*, M_1)} \times \frac{p(M_0)}{p(M_1)}, \tag{24}$$

hence the model probabilities can be calculated from the unrestricted 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**. Then we calculate the predictive likelihoods using Equation 24, 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**:

```
R> load("dataIV.Rdata")
R> data.fs <- dataIV
R> set.seed(1234)
R> mu0 <- c(0.8, 0.1, 0.5, 0, 0.5)
R> pc.train = 0.05
R> N <- nrow(data.fs)
R> M <- round(N * pc.train)
R> data.ss <- data.fs[sample(1:N, M),]
R> mit.fs <- MitISEM(KERNEL = post.IV, mu0 = mu0, df0 = 30, data = data.fs,
+    control = list(tol.pr = 0.02))
R> mit.ss <- MitISEM(KERNEL = post.IV, mu0 = mu0, df0 = 30, data = data.ss,
+    control = list(tol.pr = 0.02))
R> post.fs <- AdMitMH(N = N, post.IV, mit = mit.fs$mit, data = data.fs)
R> post.ss <- AdMitMH(N = N, post.IV, mit = mit.ss$mit, data = data.ss)
R> ind.post = (N / 5 + 1):N
R> rho.fs <- post.fs$draws[ind.post,4]
R> rho.ss <- post.ss$draws[ind.post,4]
R> Pred.Lik <- density(rho.fs, from = 0, to = 0)$y[1] / density(rho.ss,
+    from = 0, to = 0)$y[1]
```

	MitISEM candidate		Naive candidate	
	CoV	# t components	CoV	
Full Sample	1.33	2.9	14.4	
Training Sample	2.61	2.8	13.4	
	MitISEM candidate		Naive candidate	
	Mean	NSE	Mean	NSE
$p(M_0\|y)$	0.65	0.10	0.65	0.10
$p(M_1\|y)$	0.35	0.10	0.35	0.10

Table 7: Model probabilities ($p(M_0|y)$ and $p(M_1|y)$, for models without/with endogeneity) based on predictive likelihood Equation 24 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.

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 the training data are different each time. We specify equal prior probabilities $p(M_0) = p(M_1) = \frac{1}{2}$. Table 7 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 close to three 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 of the resulting estimated posterior model probabilities, when the candidate density is obtained from the MitISEM method. Note that 0.09 may seem only slightly smaller than 0.10, 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 candidate draws, the relative improvement of quality provided by the MitISEM candidate is still considerable. In examples with a fixed training sample, the relative outperformance of the MitISEM method is much stronger.

4. Concluding remarks

We presented the R package **MitISEM** which provides an automatic algorithm for the approximation of a possibly non-elliptical target density using an adaptive mixture of Student t densities as approximating or candidate density. The obtained approximation can, in particular, be used for 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 densities or model probabilities for increasing data samples, where the posterior density is updated using new observations. For Bayesian estimation, the package provides an efficient method to calculate marginal and predictive likelihoods, given a user-supplied kernel of a posterior density.

We illustrated the approximation properties of the MitISEM algorithm using two different,

distinctly non-elliptical, Gelman-Meng (Gelman and Meng 1991) densities. After that, we made use of posterior densities of two canonical econometric models: a mixture GARCH model for S&P 500 data and an IV model for the Card (1995) data. The posterior densities of these models are also characterized by non-elliptical shapes in which case Bayesian inference of model parameters and model probabilities, using importance sampling and Metropolis-Hastings algorithms, requires a flexible and appropriate candidate density. We illustrated the use of the MitISEM method for forming such a flexible candidate density, and showed 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.

Acknowledgments

This paper should not be reported as representing the views of Norges Bank. The views expressed are those of the authors and do not necessarily reflect those of Norges Bank. During this project, Nalan Baştürk and Herman K. van Dijk were partially supported by the Netherlands Organisation for Scientific Research (NWO) Secondment Grant, number 400-07-703. Lennart Hoogerheide was partially 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 (2009). "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. doi:10.18637/jss.v029.i03.

Ardia D, Hoogerheide LF, Van Dijk HK (2017). ***AdMit**: Adaptive Mixture of Student-t Distributions in R*. R package version 2.1.3, URL https://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. doi:10.1016/j.csda.2006.01.006.

Bartlett MS (1957). "A Comment on DV Lindley's Statistical Paradox." *Biometrika*, **44**(3–4), 533. doi:10.1093/biomet/44.3-4.533.

Baştürk N, Hoogerheide LF, Opschoor A, Van Dijk HK (2017). ***MitISEM**: Mixture of Student-t Distributions Using Importance Sampling and Expectation Maximization in R*. R package version 1.2, URL https://CRAN.R-project.org/package=MitISEM.

Berger JO (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag. doi:10.1007/978-1-4757-4286-2.

Bollerslev T (1986). "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics*, **31**(3), 307–327. doi:10.1016/0304-4076(86)90063-1.

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. doi:10.1002/jae.577.

Bowden RJ, Turkington DA (1990). *Instrumental Variables*. Cambridge University 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. doi:10.1007/s11222-008-9059-x.

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 B*, 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. doi:10.1214/aoms/1177693507.

Drèze JH (1976). "Bayesian Limited Information Analysis of the Simultaneous Equations Model." *Econometrica*, **44**(5), 1045–1075. doi:10.2307/1911544.

Drèze JH (1977). "Bayesian Regression Analysis Using Poly-t Densities." *Journal of Econometrics*, **6**(3), 329–354. doi:10.1016/0304-4076(77)90004-5.

Eklund J, Karlsson S (2007). "Forecast Combination and Model Averaging Using Predictive Measures." *Econometric Reviews*, **26**(2–4), 329–363. doi:10.1080/07474930701220550.

Gelfand AE, Dey DK (1994). "Bayesian Model Choice: Asymptotics and Exact Calculations." *Journal of the Royal Statistical Society B*, **56**(3), 501–514. URL http://www.jstor.org/stable/2346123.

Gelman A, Meng XL (1991). "A Note on Bivariate Distributions That Are Conditionally Normal." *The American Statistician*, **45**(2), 125–126. doi:10.2307/2684374.

Geweke J (1989). "Bayesian Inference in Econometric Models Using Monte Carlo Integration." *Econometrica*, **57**, 1317–1339. doi:10.2307/1913710.

Geweke J (1992). "Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments." In JM Bernado, JO Berger, AP Dawid, AFM Smith (eds.), *Bayesian Statistics*, volume 4, pp. 169–193. Oxford University Press, New York.

Geweke J (2005). *Contemporary Bayesian Econometrics and Statistics*. John Wiley & Sons. doi:10.1002/0471744735.

Geweke J, Durham G (2011). "Massively Parallel Sequential Monte Carlo for Bayesian Inference." Manuscript, URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1964731.

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

Heidelberger P, Welch PD (1981). "A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations." *Communications of the ACM*, **24**(4), 233–245. doi:10.1145/358598.358630.

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. doi:10.1016/j.jeconom.2006.05.015.

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*, **171**(1), 101–120. doi:10.1016/j.jeconom.2012.06.011.

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. doi:10.1016/j.jeconom.2006.06.009.

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.

Imbens GW, Angrist JD (1994). "Identification and Estimation of Local Average Treatment Effects." *Econometrica*, **62**(2), 467–475. doi:10.2307/2951620.

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.

Koop G, Poirier DJ, Tobias JL (2007). *Bayesian Econometric Methods*, volume 7. Cambridge University Press.

Kullback S, Leibler RA (1951). "On Information and Sufficiency." *The Annals of Mathematical Statistics*, **22**(1), 79–86. doi:10.1214/aoms/1177729694.

Lancaster T (2004). *An Introduction to Modern Bayesian Econometrics*. Blackwell Oxford.

Martin AD, Quinn KM, Park JH (2011). "**MCMCpack**: Markov Chain Monte Carlo in R." *Journal of Statistical Software*, **42**(9), 22. doi:10.18637/jss.v042.i09.

Martin AD, Quinn KM, Park JH (2017). ***MCMCpack***: *Markov Chain Monte Carlo (MCMC) Package*. R package version 1.3-9, URL https://CRAN.R-project.org/package=MCMCpack.

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

McLachlan GJ, Peel D (2000). *Finite Mixture Models*. John Wiley & Sons. doi:10.1002/0471721182.

Peel D, McLachlan GJ (2000). "Robust Mixture Modelling Using the t Distribution." *Statistics and Computing*, **10**(4), 339–348. doi:10.1023/a:1008981510081.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

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. doi:10.2307/2290225.

Rossi PE, Allemby GM, McCulloch R (2005). *Bayesian Statistics and Marketing*. John Wiley & Sons, West Sussex. doi:10.1002/0470863692.

Trapletti A, Hornik K (2017). ***tseries**: Time Series Analysis and Computational Finance*. R package version 0.10-33., URL https://CRAN.R-project.org/package=tseries.

Van Dijk HK (1984). "Posterior Analysis of Econometric Models Using Monte Carlo Integration." Erasmus University Press.

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. doi:10.2307/2348500.

Zeevi AJ, Meir R (1997). "Density Estimation through Convex Combinations of Densities: Approximation and Estimation Bounds." *Neural Networks*, **10**(1), 99–109. doi:10.1016/S0893-6080(96)00037-8.

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

Zellner A, Ando T, Baştürk N, Hoogerheide L, Van Dijk HK (2014). "Bayesian Analysis of Instrumental Variable Models: Acceptance-Rejection within Direct Monte Carlo." *Econometric Reviews*, **33**(1–4), 3–35. doi:10.1080/07474938.2013.807094.

Zivot E (2009). "Practical Issues in the Analysis of Univariate GARCH Models." In TG Andersen, RA Davis, JP Krei, T Mikosch (eds.), *Handbook of Financial Time Series*, pp. 113–155. Springer-Verlag, New York.

Affiliation:

Nalan Baştürk
Maastricht University
Maastricht, The Netherlands
E-mail: n.basturk@maastrichtuniversity.nl

Stefano Grassi
University of Rome "Tor Vergata"
Rome, Italy

Lennart Hoogerheide
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
and
Tinbergen Institute
Amsterdam, The Netherlands

Anne Opschoor
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
and
Tinbergen Institute
Amsterdam, The Netherlands

Herman K. van Dijk
Erasmus University Rotterdam
Rotterdam, The Netherlands
and
Norges Bank
Oslo, Norway
and
Tinbergen Institute
Amsterdam, The Netherlands

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
July 2017, Volume 79, Issue 1 *Submitted:* 2012-11-08
doi:10.18637/jss.v079.i01 *Accepted:* 2016-01-18