Comparison of MCMC algorithms in Stochastic Volatility Models

Background and Motivation

Maybe focus on the MCMC as opposed to SV upfront.

SV

Why model volatility. Quantifying risks, tail probabilities, worst case scenarios. These are all inputs in the pricing of assets and derivatives. (pg92 FE slides)

SV models is one way to model volatility. Unlike the classic Black Scholes option pricing formula, it assumes that the underlying variance of an asset's return is not constant. That is, the variance of the underlying asset return is treated as a latent random variable. This deals with some of the empirical short comings of the BS model such as the volatility smile and skew in the realised variance in financial assets.

SV models are typically expressed as non linear Gaussian state space models. This can make it difficult to estimate using classical methods. The likelihood is unavailable in closed form and there are at least as many variables as data points. There are however, a variety of Frequentist and Bayesian strategies for estimating SV models. Frequentist approaches include quasi likelihood estimation and GMM which rely on asymptotic sampling distributions. Bayesian approaches rely on computational Markov Chain Monte Carlo techniques to sample from the target joint posterior of these high dimensional state space models.

This research will focus on Bayesian estimation strategies for Stochastic Volatility models. Particular attention will be given to the computational tools and approaches used to sample from high dimensional posterior distributions. Early approaches for estimating these models relied on classic Gibbs and Metropolis Hastings algorithms to apply the MCMC routines. Since then, there have been many developments in the availability of new kinds of MCMC samplers.

Hamiltonian Monte Carlo is a . It has been popularised through its implementation in a variety of open source programming languages. It presents a new way to sample from increasingly complicated generative models .

However, as the developments of these tools enable the estimation of more complicated models (with relative ease and accessibility through the development of open source software), so do the risk of mistakes and the need for procedures diagnose problems or mistakes when implementing these methods. How do we know that these algorithms are doing what we expect them to be doing, and how can we effectively differentiate the strengths and weaknesses of different tools.

How do we validate the computation and implementation of our sampling strategies and algorithms in increasingly complex models? And how do these form part of the computational modelling workflow?

Research Goal

The objective of this research is to evaluate the performance of MCMC algorithms for estimating SV models through a simulation study. Simulations are useful since there is control over the data generating process. This will help us understand the behaviour of MCMC and evaluate its performance against known parameters. Particular attention will be given to the calibration of the MCMC algorithms in the stochastic volatility context. Further diagnostics to evaluate performance will be explored to determine the efficiency of the MCMC sampler.

The first step is to understand what "performance" of MCMC algorithms means and what exactly we are measuring. At a high level, the objective of MCMC is to sample from the target posterior distribution and to provide reliable inference. Convergence to the target posterior can be measured through efficiency measures such as rhat and effective sample size.

- Explcitly outline what is compared. KSC and HMC algorithms and estimation strategies.

RESEARCH GOAL: To compare the performance of different MCMC algorithms and estimation strategies when fitting stochastic volatility models. The study will use a discrete formulation is described

Literature Review

KSC

- KSC (1998) compare the performance of likelihood based inference and Bayesian inference at estimating the fit of these models.

- Specifically KSC propose a method of sampling using an offset mixture normal approximation with a reweighting procedure to sample from the joint posterior of the SV model. This is estimated using conjugate priors as well as MH.
- The use of simulation and sampling are used as part of model diagnostics and evaluation. This is compared with other estimation strategies (MLE) and volatility models such as autogregressive conditional heteroskedasticity (ARCH).

Stan HMC + NUTs

There have been many developments in MCMC algorithms since the introduction of classical methods such as GIBBS and MH. Hamiltoninn Monte Carlo (algorithm introduced when, widely available since 2010…?) is a relatively new im.

- Since then, there have been many developments in MCMC algorithms. In particular, HMC provides a flexible alternative to fitting and sampling from complicated high dimensional Bayesian models.
- It's key innovation is the use of gradients - enabling the flexible specification of any likelihood function.
- Constraints- can only sample from continuous
- HMC also provides new diagnostics for evaluating the effectiveness of the sampling procedure and how well the sampler is targeting the joint dist.
- How do we compare models? SBC. ESS in the tails, rhats.
- Given this context, my research looks to compare the efficacy of KSCs proposed MCMC sampling strategy with Hamiltonina Monte Carlo (NUTs).

Simulation based calibration

- 2 papers

3) Methodology

The aim of this research is to use Simulation Based Calibration (SBC) to validate the inference of posterior samples generated by different MCMC algorithms in the context of Stochastic Volatility models. This will help identify any bias in computation and inconsistencies in model implementation and estimation strategies. Comparison of the SBC results between different computational approaches may help identify any strengths or shortcomings as well as any errors in model specification.

3.1) Stochastic Volatility Model

The univariate stochastic volatility model described by Kim, Shepherd and Chib (1998), now denoted as KSC, models the variance as a random variable following some latent stochastic process. y_t is the mean corrected returns of some asset for equally spaced intervals t. β is a constant scaling factor which is also defined as $exp(\mu/2)$ representing instantaneous volatility. h_t is log volatility, where h_1 is a draw from a stationary distribution and the state equation h_{t+1} follows a stationary process governed by the autoregressive parameter ϕ such that $|\phi| < 1$. This autoregressive parameter represents the persistence or "stickiness" of log volatility and the dispersion parameter σ_η is its variance. ϵ_t and η_t are standard normal white noise shocks.

$$\begin{aligned}
y_t =& \beta exp(h_t/2)\epsilon_t \\
h_{t+1} =& \mu + \phi(h_t - \mu) + \sigma_\eta \eta_t \\
h_1 \sim& normal\left(\mu, \frac{\sigma_\eta^2}{1-\phi^2}\right) \\
\\
\epsilon_t \sim& normal(0, 1) \\
\delta_t \sim& normal(0, 1)
\end{aligned}$$

Setting $\beta = 1$, the model can be expressed more succinctly as:

$$\begin{aligned}
y_t \sim& normal(0, exp(h_t/2)) \\
h_1 \sim& normal\left(\mu, \frac{\sigma_\eta^2}{1-\phi^2}\right) \\
h_{t+1} \sim& normal(\mu + \phi(h_t - \mu), \sigma_\eta^2), t \neq 1
\end{aligned}$$

With priors for the static parameters as defined in KSC:

$$\begin{aligned}
\mu \sim& normal(0, 10^2) \\
\sigma_\eta^2 \sim& IG(5/2, (0.01 \times 5)/2) \\
\phi^* \sim& beta(20, 1.5) \\
\phi =& 2\phi^* - 1
\end{aligned}$$

The prior on ϕ is a "stretched" beta distribution. This is a beta distribution (as defined on the parameter ϕ^*) which has been transformed to have support (-1, 1).

3.2) Estimation Strategy

KSC

KSC's model uses a simulation smoother[1] (de Jong and Shephard, 1995) to estimate from the latent states. This requires the state space model to be linear and conditionally Gaussian.

Since the model is not linear, a transformation is applied by squaring and taking the log of y_t.

$$\begin{aligned} y_t^* &= log(y_t^2) \\ &= log((\epsilon_t exp(h_t/2))^2) \\ &= log(exp(h_t)) + log(\epsilon_t^2) \\ &= h_t + log(\epsilon_t^2) \\ &= h_t + z_t \end{aligned}$$

Where $z_t = log(\epsilon_t^2)$ follows a log chi-squared distribution with mean -1.2704 and varaince 4.93. However, it is not simple to sample from this parameterisation of the model. KSC use a gaussian mixture model to **approximate** the first 4 moments of the log chi squared distribution. This is defined by:

$$f(z_t) = \sum_{i=1}^{K} q_i f_N(z_i | m_i - 1.2704, \nu_i^2)$$

Where K is the mixture of normal densities f_N, component probabilities q_i, mean $m_i - 1.2704$ and variance ν_i^2. These parameters were seleted using moment matching where they found 7 normal densities with varying mean and variance parameters best approximated the log chi squared moments. These parameters and weights can be found in Appendix A.

The model can be sampled via the simulation smoother since the model is now linear and conditionally gaussian. The static parameters μ and σ^2 are sampled directly from their conjugate posterior distributions where as ϕ is sampled via a metropolis hastings accept/reject procedure. The details can be be around in Appendix B.

[1] In this research the exact software to apply the simulation smoother is unavailable, so a more recent simulation smoother is used which is based on the software written by the same author.

Hamiltonian Monte Carlo (No-U-Turn Sampler)

Stan's default Hamiltonian Monte Carlo algorithm, the No-U-Turn Sampler, allows for direct sampling of the spcified stochastic volatility model. The HMC algorithm and more broadly the Stan programming language allows for sampling of the generative model and can flexibly handle complicated likelihood functions.

HMC will use the same priors are specified in the KSC model, however, the HMC algorithm does not require the priors to be conjugate (nor does the algorithm sample from conjugate or conditionally posterior distributions).

Simulation Based Calibration (SBC)

SBC is conducted by comparing the distribution of rank statistics to the discrete uniform distribution which arises from a correctly calibrated algorithm. The procedure starts by generating datasets implied by the prior distribution and data generating process and calculating rank statistics from the resulting posterior samples. Starting with the parameters θ and data y, draw a sample from the prior distribution:

$$\theta^{sim} \sim \pi(\theta)$$

Generate a dataset given by the prior draws and data generating process p.

$$y^{sim} \sim p(y|\theta^{sim})$$

Then take draws from the posterior distribution generated by a MCMC algorithm or estimation strategy (i.e estimate the model on the simulated data using either KSC's strategy or HMC):

$$\{\theta_1, \dots, \theta_L\} \sim p(\theta|y^{sim})$$

A key result from this procedure is that the posterior sample $\{\theta_1, \dots, \theta_L\}$ will share the same distribution as the prior samples θ^{sim}. This is implied by the following expression:

$$\begin{aligned}\pi(\theta) &= \int \pi(\theta|y^{sim})\pi(y^{sim}|\theta^{sim})\pi(\theta^{sim})dy^{sim}d\theta^{sim} \\ &= \int \pi(\theta|y^{sim})\pi(y^{sim},\theta^{sim})dy^{sim}d\theta^{sim}\end{aligned}$$

That is, the posterior averaged over the joint distribution follows the same distribution as the prior. The procedure of generating posterior samples implicitly performs this integral. The expression on the right hand side is proportional to the prior density and the simulation if done correctly should produce samples from the prior distribution. Therefore, any deviation of the posterior samples from the prior distribution suggests an error - either in computation, code or overall analysis.

Since the distribution of the posterior samples follows the prior distribution, the rank statistic for a given parameter and simultion follows a discrete uniform distribution[2].

$$r = rank(\{\theta_1, \dots, \theta_L\}, \theta^{sim}) = \sum_{l=1}^{L} 1[\theta_l < \theta^{sim}]$$

This completes one iteration of SBC. Multiple iterations are run and the rank statistics are calculated for each parameter. The resulting rank statistics are compared to the discrete uniform distribution to determine if any problematic features or errors exist.

If the computation is well calibrated and the samples follow a discrete uniform distribution, then the posterior (credible) intervals will have sufficient coverage. That is, for any percentage interval selected (for example 90%) then there is 90% chance that any θ^{sim} falls within this interval. In other words, a Bayesian analysis and computation is well calibrated if 90% of constructed intervals contain the “true” parameter 90% of the time.

Results

SBC for the HMC algorithm and the KSC Gaussian approxmiation are conducted with simulated dataset sizes of 1000 observations and 1000 simulation interations. The preliminary results focus on a subset of parameters since there are parameters for each latent state (i.e one for each data observation) plus the static parameters.

Figure 1 shows the the distribution of rank statistics for the HMC algorithm, with the horizontal black line at 50 representing the discrete uniform distribution. Phi and sigma_sqd look relatively uniform. However mu and h.995 have some lumpiness which may suggest a lack of calibration.

One of the drackbacks of SBC is that any deviation from uniform from a calibrated analysis maybe due to finite samples. Increasing the number of iterations from 1000 to 5000 shows more consistent uniform behaviour from all parameters (Figure 2).

[2] Proof of this result in Talts et al.

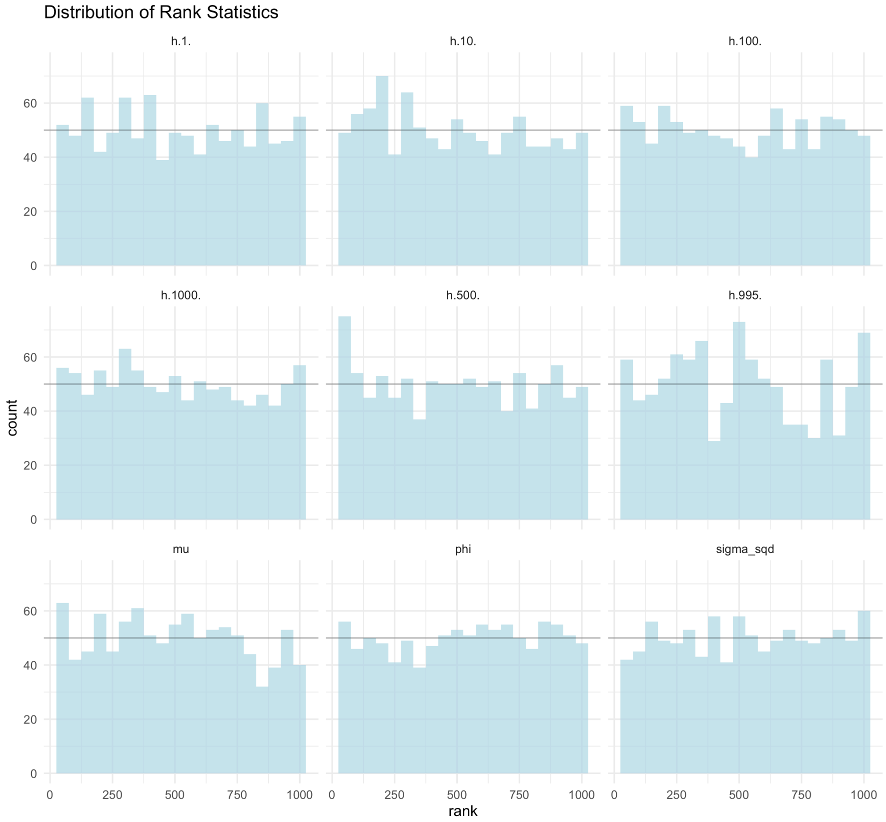

Figure 1: Distribution of parameter ranks for HMC algorithm

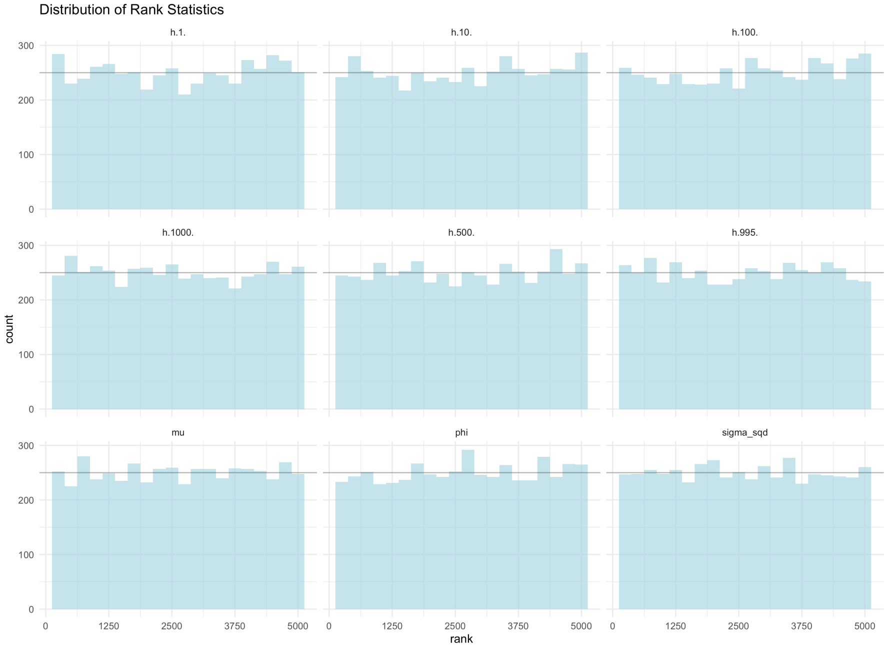

Figure 2: Distribution of parameter ranks for HMC algorithm (5000 iterations)

Conclusion and future research considerations

Plan

The KSC model also also apply a Metropois Hastings correction for the fact that the sampling strategy approximates the error distribution.

Extensions to the model table:

```
+ Stan (HMC), KSC (GMA), KSC (GMA + MH correction)
+ CP, NCP
```

Parameterisation

Additional considerations

- Out of sample prediction on real dataset

References

Appendix A: Mixture Gaussian weights

Weights

Appendix B: Conjugate posterior distributions