FinM 331/Stat 339 Financial Data Analysis,

Winter 2010

Floyd B. Hanson, Visiting Professor

Email: fhanson@uchicago.edu

Master of Science in Financial Mathematics Program
University of Chicago

Lecture 10

6:30-9:30 pm, 08 March 2010, Ryerson 251 in Chicago

7:30-10:30 pm, 08 March 2010 at UBS in Stamford

7:30-10:30 am, 09 March 2010 at Spring in Singapore

10. Bayesian Methods in Finance, Part II[a]:

• *10.1. Time-Varying Equity Premium Parameter:*

Time-variation of parameters is an intrinsic property of empirical financial and other economic systems. Many of the basic models, such as the Black-Scholes (BS1973) European option pricing model, assume constant coefficients, e.g., constant spot rates and volatility in Black-Scholes. Merton (M1973) simultaneously showed in his Black-Scholes justification paper how to include time-variable parameter in the option pricing formulas by replacing the parameter terms $r(T-t)$ and $\sigma^2(T-t)$ by

$$\int_t^T r(\tau)d\tau \;\; \text{and} \;\; \int_t^T \sigma^2(\tau)d\tau, \tag{10.1}$$

respectively.

[a] Again, for background to a good part of this lecture, see Michael Johannes and Nicholas Polson (JP2003/2006/2009), ***MCMC Methods for Continuous-Time Financial Econometrics,*** 95 pages, preprint for the **Handbook of Financial Econometrics**, edited by Yacine Ait-Sahalia and Lars Hansen.

For estimating the time-varying parameters, one can use ordinary methods of **linear regression or maximum likelihood estimation** with the simple idea of time-moving windows or intervals. However, there is a trade-off, the time-itervals have large enough to provide an adequate observation sample size of the underlying asset and be numerous or small enough to provide a reasonable portrayal of the time-dependence[a].

When **Bayesian estimation** is used with successive approximations as with **Gibbs Sampling or the Metropolis algorithm**, the sequential treatment of discretized-time parameters is not a good representation of the time-dependence overall and a cause of much higher computational complexity, compared the the simultaneous approximations in block form for all parameters and variables at each discrete time. Similar trade-offs occur in numerical analysis between the Gauss-Seidel successive approximation method and the Jacobi simultaneous approximation method for iterative solving of linear algebraic systems.

[a]F.B. Hanson and J.J. Westman (2003a, 2003b), *[CDC03] and [ACC03] publications,* on jump-diffusion time-varying parameter models, applied to optimal portfolio problems and stock-returns, respectively.

A **time-varying parameter version of the Black-Scholes linear diffusion model** of the underlying stock price is

$$dS(t)=S(t)\big((r(t)+\mu_s(t)+\sigma_s^2/2)dt+\sigma_s dW_s(t)\big)\,, \quad (10.2)$$

where $r(t)$ is the given time-varying spot interest rate, $\mu_s(t)$ is the proposed **equity risk premium**[a] for the stock and the diffusion coefficient $\sigma_s^2/2$ has been preadded for convenience in the log-return form of the equation and $dW_s(t)$ is the stock price diffusion process. The equity premium of Johannes and Polson (JP2003, JP2006) satisfies a **mean-reverting, constant parameter model**, resembling the Heston stochastic volatility model,

$$d\mu_s(t)=\kappa_\mu(\theta_\mu-\mu_s(t))\,dt+\sigma_\mu W_\mu(t), \quad (10.3)$$

where $\{\kappa_\mu, \theta_\mu, \sigma_\mu, \rho_{\mu,s}\}$ are the constant coefficients and

$$dW_\mu(t)=\rho_{\mu,s}dW_s(t)+\sqrt{1-\rho_{\mu,s}^2}\,dW_\perp(t), \quad (10.4)$$

resolved into independent components so $\mathrm{E}[dW_\mu dW_s(t)]=\rho_{\mu,s}dt$.

[a] The equity risk premium is used to statistically compensate for stocks that earn more than they would by the spot rate, as payment for increase risk. However, it is controversial, see ***Equity Premium Puzzle,*** and other sources.

Due to the constant coefficients in (10.3), the **method of integrating factors** of ODEs works independent of the stochastic term to eliminate the RHS linear term,

$$d\left(e^{\kappa_\mu t}\mu_s(t)\right)=e^{\kappa_\mu t}(\kappa_\mu\theta_\mu dt+\sigma_\mu dW_\mu(t))\,, \tag{10.5}$$

which can be checked by substitution, so it can be integrated in time steps Δt such that at $t_i=i\Delta t$,

$$\begin{aligned}\mu_{s,i}=\mu_{s,i-1}e^{-\kappa_\mu\Delta t}+\theta_\mu\left(1-e^{-\kappa_\mu\Delta t}\right)\\ +\sigma_\mu\int_{t_{i-1}}^{t_i}e^{-\kappa_\mu(t-\tau)}dW_\mu(\tau),\end{aligned} \tag{10.6}$$

for $i=1:n$ with $\mu_{s,0}$ given at $t=0$. Approximating the stochastic integral at the lower limit for sufficiently small Δt, but leaving the constant $e^{-\kappa_\mu\Delta t}$ intact, producing an **autoregressive time series model of order one, AR(1)**,

$$\mu_{s,i}=\alpha_\mu+\beta_\mu\mu_{s,i-1}+\widetilde{\sigma}_\mu Z_{\mu,i} \tag{10.7}$$

where $\alpha_\mu=\theta_\mu\left(1-e^{-\kappa_\mu\Delta t}\right)$, $\beta_\mu=e^{-\kappa_\mu\Delta t}$, $\widetilde{\sigma}_\mu=\sigma_\mu\sqrt{\Delta t}e^{-\kappa_\mu\Delta t}$ and $Z_{\mu,i}\stackrel{\text{dist}}{=}\mathcal{N}(0,1)$. Note that elements $[\mu_{s,i};\mu_{s,i-1}]$ are recursive.

Letting the current **observed excess log-return** element be, given the log-returns and the spot rate as well as an interest in estimating the equity premium,

$$X_i = \log(S_i/S_{i-1}) - r_{i-1}\Delta t, \tag{10.8}$$

$\widetilde{\sigma}_s = \sigma_s\sqrt{\Delta t}$ and $Z_{s,i} \stackrel{\text{dist}}{=} \mathcal{N}(0,1)$, then our **observation-transition system** of equations is

$$\begin{aligned} X_i &= \mu_{s,i} + \widetilde{\sigma}_s Z_{s,i} \\ \mu_{s,i} &= \alpha_\mu + \beta_\mu \mu_{s,i-1} + \widetilde{\sigma}_\mu Z_{\mu,i}, \end{aligned} \tag{10.9}$$

for $i = 1:n$. The $\widetilde{\sigma}_s Z_{s,i}$ is the **measurement error**.

In **summary of the problem**[a], the observation vector is $\vec{X} = [X_i]_{i=1}^n$, the latent variable vector is $\vec{Y} = \vec{\mu}_s = [\mu_{s,i}]_{i=1}^n$ and the parameter vector is

$$\vec{\Theta} = [\alpha_\mu; \beta_\mu; \widetilde{\sigma}_\mu; \widetilde{\sigma}_s; \rho_{\mu,s}]. \tag{10.10}$$

---

[a] Recall that where we have been using the variable pair $[\vec{X}; \vec{Y}]$ for the observation and transition distribution vectors, respectively, because we started using $\vec{X}$ for the data. However, Johannes and Polson (JP2006) use just the opposite nomenclature $[\vec{Y}; \vec{X}]$for the same vectors. In some fields it is standard to you $\vec{Y}$ for observations of a variable $\vec{X}$.

The prior distribution is of the form

$$f^{(\mathbf{prior})}_{\vec{\Theta},\vec{Y}}\left(\vec{\theta},\vec{\mu}_s\right) \tag{10.11}$$

and the posterior distribution is of the form

$$f^{(\mathbf{post})}_{\vec{\Theta},\vec{Y}|\vec{X}}\left(\vec{\theta},\vec{\mu}_s|\vec{x}\right). \tag{10.12}$$

Again, the Hammersley-Clifford theorem permits a decomposition of the posterior into simpler full conditional distributions,

$$f^{(\mathbf{post})}_{\vec{\Theta}|\vec{Y},\vec{X}}\left(\vec{\theta}|\vec{\mu}_s,\vec{x}\right) \tag{10.13}$$

and

$$f^{(\mathbf{post})}_{\vec{Y}|\vec{\Theta},\vec{X}}\left(\vec{\mu}_s|\vec{\theta},\vec{x}\right). \tag{10.14}$$

The full likelihood distribution intermediate between the prior and posterior is

$$f^{(\mathbf{post})}_{\vec{X}|\vec{\Theta},\vec{Y}}\left(\vec{x}|\vec{\theta},\vec{\mu}_s\right). \tag{10.15}$$

However, since these generate hybrids of normal distributions, which do not have random generators in **MATLAB**, it would be desirable to break these up even more. The first full conditional for $\vec{\Theta}$ can be replaced in the Gibbs sampler by

$$\begin{aligned} & f^{(\mathrm{post})}_{\vec{\Theta}_{1:2}|\vec{\Theta}_{3:5},\vec{Y},\vec{X}}(\alpha_\mu, \beta_\mu|\sigma_\mu, \sigma_s, \rho_{\mu,s}, \vec{\mu}_s, \vec{x}) \overset{\mathrm{dist}}{\sim} \mathcal{N}(\vec{m}_{1:2}, V_{1:2}); \\ & f^{(\mathrm{post})}_{\vec{\Theta}_{3,5}|\vec{\Theta}_{1:2,4},\vec{Y},\vec{X}}(\sigma_\mu, \rho_{\mu,s}|\alpha_\mu, \beta_\mu, \sigma_s, \vec{\mu}_s, \vec{x}) \overset{\mathrm{dist}}{\sim} \mathcal{IW}(\Psi_{3,5}, \mathrm{dof}_{3,5}); \\ & f^{(\mathrm{post})}_{\vec{\Theta}_4|\vec{\Theta}_{1:3,5},\vec{Y},\vec{X}}(\sigma_s|\alpha_\mu, \beta_\mu, \sigma_\mu, \rho_{\mu,s}, \vec{\mu}_s, \vec{x}) \overset{\mathrm{dist}}{\sim} \mathcal{IW}(\Psi_4, \mathrm{dof}_4); \end{aligned} \tag{10.16}$$

with similar forms for the priors without the conditioning on $[\vec{\mu}_s; \vec{x}]$.

The equity premium vector is an n-dimensional distribution $f^{(\text{post})}_{\vec{Y}|\vec{\Theta},\vec{X}}\left(\vec{\mu}_s|\vec{\theta},\vec{x}\right)$ with recursive complexity which is convenient for iterative simulations of the $\mu_{s,k}$ for $k = 1:n$, but not for determining the necessary Bayesian distribution.

Johannes and Polson (JP2003,JP2006) suggest a **forward filtering, backward sampling (FFBS) algorithm** that uses the famous **Kalman filter**[a] for the forward sweep. The Kalman filter algorithm is another of the top ten algorithms of the century along with the Metropolis Monte Carlo algorithm. The Kalman filter is for filtering estimates of the mean of a partially observed linear Gaussian process and the covariance of the error, but is extendable to other models. It has been used for finance applications a number of times.

[a] See the classic text of B.D.O Anderson and J.B. Moore (AM1979), Optimal Filtering, Dover Publications Edition 2005 and inexpensive; Kalman Filter section, pp. 105-115. For a more relevant source here, see R.S. Tsay (T2005), Analysis of Financial Time Series, pp. 490-496. For this last lecture we do not have time to go into the details of this, but two of my PhD students did their theses on optimal filtering.

The conditional mean of the state is needed for its estimate,

$$\mu_{k|k-1} \equiv \mathrm{E}[\mu_{s,k}|\vec{X}_{k-1}], \tag{10.17}$$

where the conditional past observations are $\vec{X}_{k-1} \equiv [X_i]_{i=1}^{k-1}$. Note that the same conditional mean of the observations is the same,

$$y_{k|k-1} \equiv \mathrm{E}[y_k|\vec{X}_{k-1}] = \widehat{\mu}_k. \tag{10.18}$$

Next let the variance of the state conditioned on past observations be

$$\Sigma_{k|k-1} \equiv \mathrm{Var}[\mu_{s,k}|\vec{X}_{k-1}], \tag{10.19}$$

and let the corresponding variance of the error $e_{s,t} \equiv y_k - y_{k|k-1}$ of the observation estimate be

$$V_k \equiv \mathrm{Var}[e_{s,t}|\vec{X}_{k-1}] = \Sigma_{k|k-1} + \widetilde{\sigma}_s^2. \tag{10.20}$$

Some other properties of the one-step forecast error $e_{s,t}$ are zero-mean, $\mathrm{E}[e_{s,t}] = 0$, and no correlation with the previous observation, $\mathrm{Cov}[e_{s,t}, y_{k-1}] = 0$.

For actual calculations of the state, two recursions for the state forcast estimate and its error variance are needed and after much analysis[a]

$$\begin{aligned} \mu_{k|k+1} &= \mu_{k|k-1} + K_k e_{s,t}; \\ \Sigma_{k|k+1} &= (1-K_k)\Sigma_{k|k-1} + \widetilde{\sigma}^2_{\mu}; \end{aligned} \tag{10.21}$$

where $K_k \equiv \Sigma_{k|k-1}/V_k$ is the **Kalman gain**. There are many option for the initial condition, a simple one could be $\mu_{1|0} = 0 = \Sigma_{1,0}$. Using the result of these formulas can yield the forward estimate of moments of $\vec{\mu}_s$, i.e., of $f^{(\mathrm{kf})}_{\vec{Y}|\vec{\Theta},\vec{X}}\left(\vec{\mu}_s|\vec{\theta},\vec{x}\right)$.

An intermediate step, between forward and backward sweeps, is the estimated sample the last state,

$$\widehat{\mu}_n \overset{\mathrm{dist}}{\sim} f^{(\mathrm{kf})}_{Y_n|\vec{Y}_{n-1},\vec{\Theta},\vec{X}}\left(\mu_{s,k}|\vec{y}_{n-1},\vec{\theta},\vec{x}\right), \tag{10.22}$$

as the start for the backward step.

[a] See R.S. Tsay (T2005), Analysis of Financial Time Series, pp. 490-496.

The backward step uses a decomposition of the posterior joint expected returns premium $\vec{\mu}_s$,

$$f^{(\mathrm{post})}_{\vec{Y}|\vec{\Theta},\vec{X}}\left(\vec{\mu}_s|\vec{\theta},\vec{x}\right) \propto f\left(\mu_{s,n}|\vec{\theta},\vec{x}\right)\prod_{i=0}^{n-1} f\left(\mu_{s,i}|\widehat{\mu}_{s,i+1},\vec{\theta},\vec{x}\right). \quad (10.23)$$

Reinterpreting the state transition equation (10.7) given the forward estimate for $\widehat{\mu}_{s,i+1}$ and needing the backward estimate of $\mu_{s,i}$ for state i, using

$$Z_{\mu,i+1} = \left(\alpha_\mu + \beta_\mu \mu_{s,i} - \widehat{\mu}_{s,i+1}\right)/\widetilde{\sigma}_\mu \quad (10.24)$$

to formulate a normal distribution.

$\star$ Johannes and Polson (JP2006) give several other time-varying parameter models, including Merton's (M1974) default model which is like a Black-Scholes (BS1973) option model with time-varying parameters.

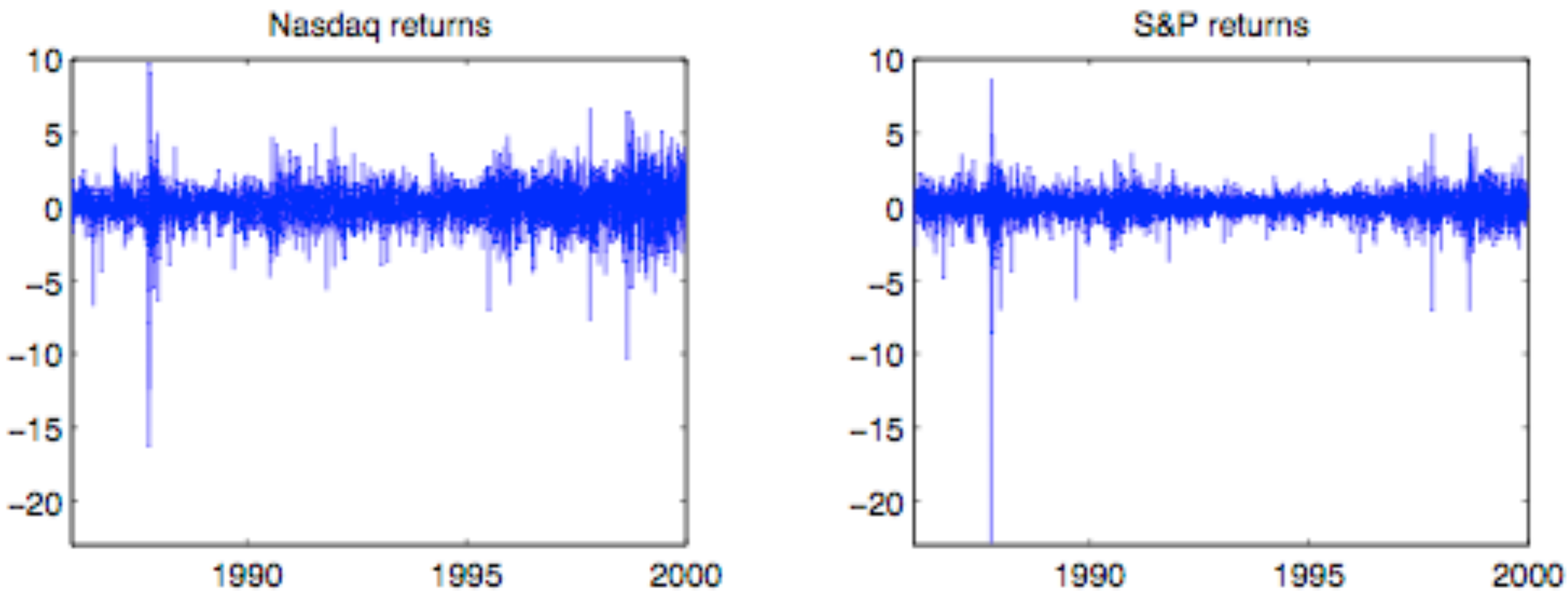


Figure 3: Estimated jump sizes in returns for the Nasdaq and S&P 500 and actual returns over the same period.

Figure 10.1: Johannes and Polson (JP2006), Fig. 3B, p. 49, time-varying equity premium problem showing trajectories of 1987-2000 S&P 500 and Nasdaq 100 indices.

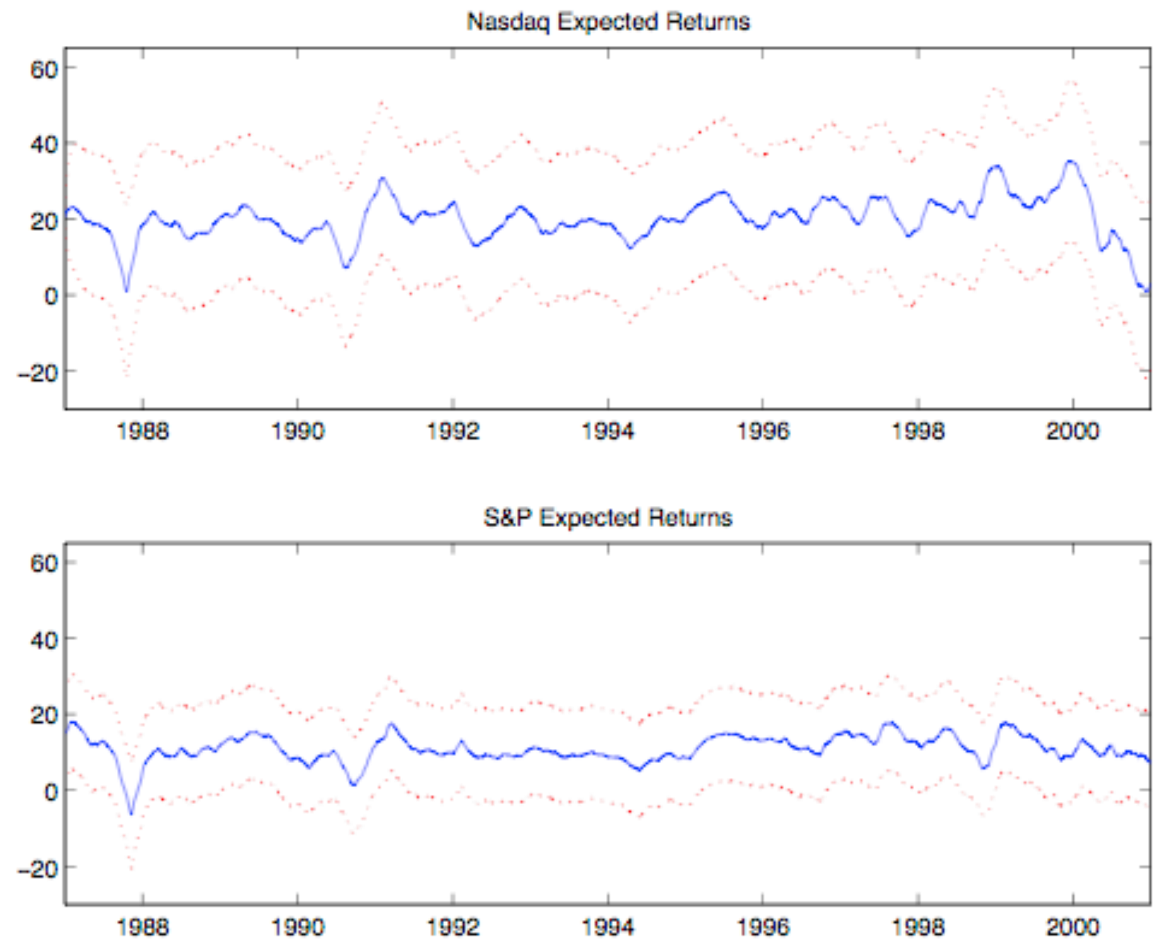


Figure 4: Smoothed expected return paths (with confidence bands) for the S&P 500 and Nasdaq 100 from 1987-2001.

Figure 10.2: Johannes and Polson (JP2006), Fig. 4, p. 52, time-varying equity premium problem results for 1987-2000 S&P 500 and Nasdaq 100 indices.

• *10.2. Log-Stochastic Volatility (Log-SV or Log-SVD) Models:*

Stochastic volatility or variance models are nicely treatable by MCMC methods, even though they lead to non-Gaussian process.

Consider the **stochastic system of log-return and log-volatility**[a],

$$d\log(S(t)) = \mu_s(t)dt + \sqrt{V(t)}dW_s(t); \tag{10.25}$$

$$d\log(V(t)) = \kappa_v(\theta_v - \log(V(t)))dt + \sigma_v dW_v(t); \tag{10.26}$$

where the $[\kappa_v; \theta_v; \sigma_v]$ are constant and the Wiener processes are assumed independent so $\rho_{s,v} \equiv 0$. (In the cited paper correlations are included.)

---

[a] E. Jacquier, N.G. Polson and P.E. Rossi (JPR2004), Bayesian Analysis of Stochastic Volatility Models with Fat-Tails and Correlated Errors, Journal of Econometrics, vol. 122(1), pp. 185-212; E. Jacquier, N.G. Polson and P.E. Rossi (JPR2002), Bayesian Analysis of Stochastic Volatility Models, Journal of Business and Economic Statistics, vol. 20(1), pp. 69-87.

Let $\mu_s(t)=0$ to focus on the stochastic properties, $X_i=\log(S_i/S_{i-1})/\sqrt{\Delta t}$ for $i=1:n$, $Z_{j,i}\stackrel{\text{dist}}{=}\mathcal{N}(0,1)$ for $j=s:v$, $\alpha_v=\kappa_v\theta_v\Delta t$, $\beta_v=1-\kappa_v\Delta t$ and $\widetilde{\sigma}_v=\sigma_v\sqrt{\Delta t}$, so the time-discretized system that will be solved is

$$X_i = \sqrt{V_{i-1}}Z_{s,i}; \tag{10.27}$$

$$\log(V_i)=\alpha_v+\beta_v\log(V_{i-1})+\widetilde{\sigma}_v Z_{v,i}. \tag{10.28}$$

The parameter vector is $\vec{\Theta}=[\alpha_v;\beta_v;\widetilde{\sigma}_v]$ and latent or hidden state vector is $\vec{Y}=\vec{V}=[V_i]_{i=1}^n$. According to Hammersley-Clifford, the full posterior $f^{(\text{post})}_{\vec{\Theta},\vec{Y}|\vec{X}}(\vec{\theta},\vec{v}|\vec{x})$ can decomposed into more convenient, full conditionals

$$f^{(\text{post})}_{\vec{\Theta}_{1:2}|\vec{\Theta}_3,\vec{Y}\vec{X}}(\alpha_v,\beta_v|\widetilde{\sigma}^2_v,\vec{v},\vec{x}); \tag{10.29}$$

$$f^{(\text{post})}_{\vec{\Theta}_3|\vec{\Theta}_{1:2},\vec{Y}\vec{X}}(\widetilde{\sigma}^2_v|\alpha_v,\beta_v,\vec{v},\vec{x}); \tag{10.30}$$

$$f^{(\text{post})}_{\vec{Y}|\vec{\Theta},\vec{X}}(\vec{v}|\alpha_v,\beta_v,\widetilde{\sigma}^2_v,\vec{x}). \tag{10.31}$$

This choice of full conditional makes for standard choices of conjugate priors, least for the volatility parameter priors,

$$f^{(\text{prior})}_{\vec{\Theta}_{1:2}}(\alpha_v, \beta_v) \overset{\text{dist}}{\sim} \mathcal{N}(\vec{m}_{1:2}, V_{1:2}); \tag{10.32}$$

$$f^{(\text{prior})}_{\vec{\Theta}_3}(\widetilde{\sigma}^2_v) \overset{\text{dist}}{\sim} \mathcal{IG}(\Psi_3, \text{dof}_3). \tag{10.33}$$

Forcing conjugacy leads to the full conditional parameter posteriors according to the Bayes rules,

$$\begin{aligned} f^{(\text{post})}(\alpha_v, \beta_v | \widetilde{\sigma}^2_v, \vec{v}, \vec{x}) &\propto \textstyle\prod_{i=1}^n f(v_i | v_{i-1}, \alpha_v, \beta_v, \widetilde{\sigma}^2_v) f(\alpha_v, \beta_v) \\ &\overset{\text{dist}}{\sim} \mathcal{N}(\vec{m}^{post}_{1:2}, V^{(\text{post})}_{1:2}); \end{aligned} \tag{10.34}$$

$$\begin{aligned} f^{(\text{post})}(\widetilde{\sigma}^2_v | \alpha_v, \beta_v, \vec{v}, \vec{x}) &\propto \textstyle\prod_{i=1}^n f(v_i | v_{i-1}, \alpha_v, \beta_v, \widetilde{\sigma}^2_v) f(\widetilde{\sigma}^2_v) \\ &\overset{\text{dist}}{\sim} \mathcal{IG}(\Psi^{(\text{post})}_3, \text{dof}^{(\text{post})}_3); \end{aligned} \tag{10.35}$$

where the conditional two-step volatility likelihood follows from the volatility normal error,

$$Z_{v,i} = (\log(V_i) - \beta_v \log(V_{i-1}) - \alpha_v) / \widetilde{\sigma}_v. \tag{10.36}$$

So, the volatility likelihood for the parameter Bayesian rule is

$$\begin{aligned} f(v_i|v_{i-1},\alpha_v,\beta_v,\widetilde{\sigma}_v^2) \propto \widetilde{\sigma}_v^{-1} \exp\Big(-0.5\widetilde{\sigma}_v^{-2}(\log(V_i) \\ -\beta_v \log(V_{i-1})-\alpha_v)^2\Big). \end{aligned} \tag{10.37}$$

The posterior for the latent state vector $\vec{V}$ is quite complex because of the connectivity of nearest neighbor is tight and the volatility full joint posterior is, using the simple p notation for the probability,

$$p(\vec{V}|\vec{\Theta},\vec{X}) \propto p(\vec{X}|\vec{\Theta},\vec{V}) \times p(\vec{V}|\vec{\Theta}), \tag{10.38}$$

but the conditional independence and Markov property requiring only nearest neighbor property (NNbor),

$$\begin{aligned} p(\vec{X}|\vec{\Theta},\vec{V}) \propto \textstyle\prod_{i=1}^n p(X_i|V_i,\vec{\Theta}); \\ p(\vec{V}|\vec{\Theta}) \propto \textstyle\prod_{i=1}^n p(V_i|V_{i-1},\vec{\Theta}). \end{aligned} \tag{10.39}$$

Combining the two products based on (NNbor),

$$p(\vec{V}|\vec{\Theta},\vec{X}) \propto \prod_{i=1}^{n} p(X_i|V_i,\vec{\Theta}) \times p(V_i|V_{i-1},\vec{\Theta}). \tag{10.40}$$

By Hammersley-Clifford and (NNbor, both backward and forward), the **volatility complete conditional** instead is

$$p(V_i|V_{j\neq i}, \vec{X}) = p(V_i|V_{i-1}, V_{i+1}, \vec{\Theta}, \vec{X}), \tag{10.41}$$

but this can be further reduced by several applications of Bayes rule ***(Bayesian IQ test?)*** and more (NNbor) properties for $\vec{X}$[a] and V_i,

$$\begin{aligned} p(V_i|V_{i-1}, V_{i+1}, \vec{\Theta}, \vec{X}) &\propto p(V_i, V_{i-1}, V_{i+1}|\vec{\Theta}, \vec{X}) \\ &\propto p(X_{i+1}|V_i, \vec{\Theta}) \times p(V_i, V_{i-1}, V_{i+1}|\vec{\Theta}) \\ &\propto p(X_{i+1}|V_i, \vec{\Theta}) \times p(V_i|V_{i-1}, \vec{\Theta}) \\ &\quad \times p(V_{i+1}|V_i, \vec{\Theta}), \end{aligned} \tag{10.42}$$

where only (NNbor) terms have been retained and $p(V_i, V_{i-1}, V_{i+1}|\vec{\Theta})$ has been split up into two terms reflecting the binary dependence in (10.36) for $Z_{v,i}, Z_{v,i+1}$.

[a] JP2006 introduce term inconsistency relating X_i to V_i, when in (10.27) X_i depends on V_{i-1}. In the published JPR2004, they just define $X_i = \sqrt{V_i} Z_{s,i}$ which **fixes** the problem.

Substituting from (10.36) and $Z_{s,i+1} = X_{i+1}/\sqrt{V_i}$ into the volatility posterior likelihood,

$$\begin{aligned} p(V_i|V_{i-1}, V_{i+1}, \vec{\Theta}, \vec{X}) &\propto V_i^{-0.5} \exp(-0.5 Z_{s,i+1}^2) \frac{d\log(\log(V_i))}{dV_i} \\ &\quad \cdot \exp(-0.5 Z_{v,i}^2) \exp(-0.5 Z_{v,i+1}^2) \\ &\propto V_i^{-1.5} \exp(-0.5 (V_i^{-1} X_{i+1}^2)) \\ &\quad \cdot \exp(-0.5 \sigma_v^{-2} (\log(V_i) - \mu_{v,i\pm1})^2), \end{aligned} \tag{10.43}$$

where we use the proper X_{i+1}^2 instead of X_i^2 in JP2006, explicitly marked the the use of the Jacobian to change the density with respect to $\log(V_i)$ to that of V_i, our main interest, and have combined two volatility exponentials by the completing the square technique as in JPR2004, giving the corrected,

$$\mu_{v,i\pm1} \equiv (\beta_v \log(V_{i+1} V_{i-1}) + \alpha_v (1 - \beta_v)) / (1 + \beta_v^2). \tag{10.44}$$

Neither exponential in (10.43) is a normal density in the variable V_i, but are normal in some other functional form, so it will be necessary to use the **Metropolis-Hastings acceptance-rejection algorithm** with acceptance-rejection for

$$\pi(V_i) \equiv p(V_i | V_{i-1}, V_{i+1}, \vec{\Theta}, \vec{X}). \qquad (10.45)$$

JPR1994 use the independence version of Metropolis-Hastings with a **gamma proposal density** $q(V_i)$ as $\pi(V_i)$ (10.43) is is a inverse gamma density, ignoring the non-log terms.

Let the acceptance probability be

$$\alpha(V_i^{(g)}, V_i^{(g+1)}) = \min\left[\frac{\pi\left(V_i^{(g+1)}\right) / q\left(V_i^{(g+1)}\right)}{\pi\left(V_i^{(g)}\right) / q\left(V_i^{(g)}\right)}, 1\right]. \qquad (10.46)$$

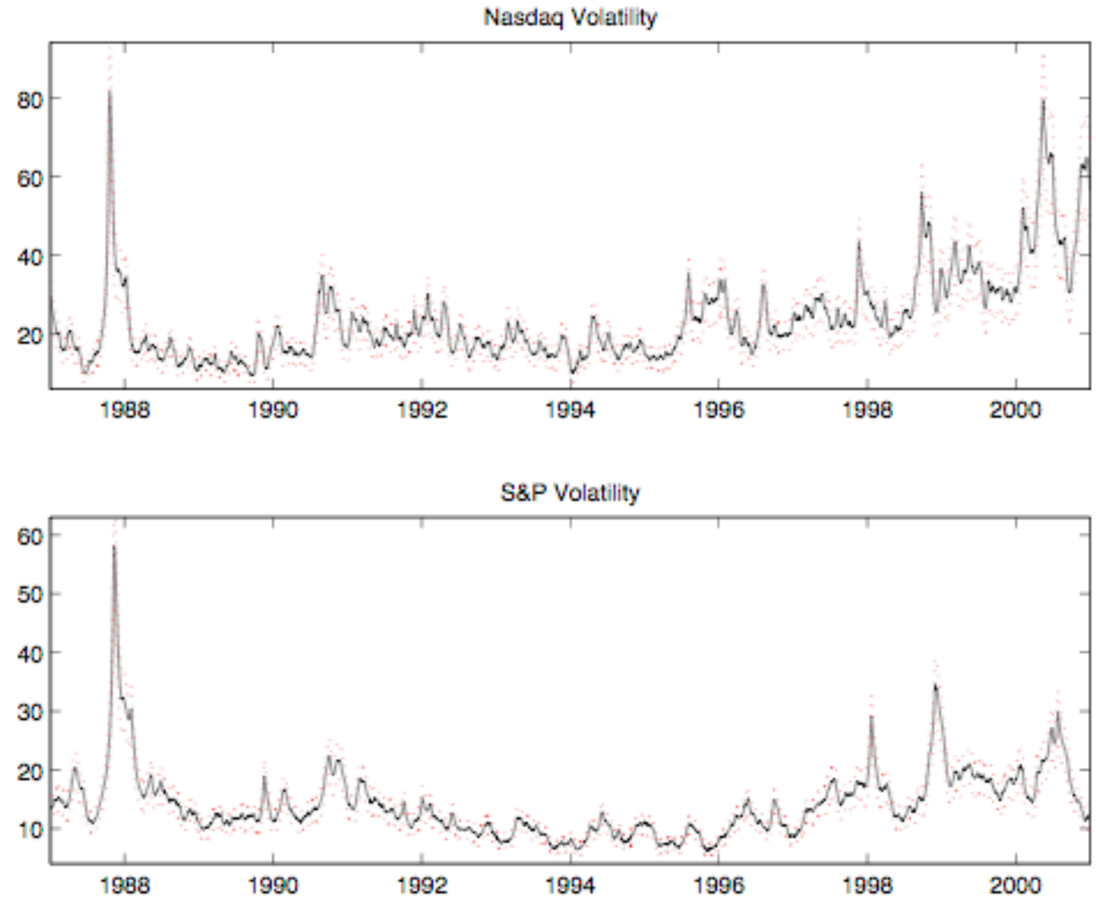


Figure 5: Smoothed volatility paths (with posterior confidence bands) for the S&P 500 and Nasdaq 100 from 1987-2001.

Figure 10.3: Johannes and Polson (JP2006), Fig. 5, p. 59, smoothed volatility paths $\mathbf{E}[V_i|\vec{X}]$ with 95% credible intervals (light red dashed lines) for 1987-2000 S&P 500 and Nasdaq 100 indices.

• *10.3. Correlated Volatility (Correlated-SVD) Leverage Effect Models:*

Fischer Black is credited for showing that the correlation between equity return prices and volatility were important[a] and so is called the **Black volatility leverage effect**. Thus, assume that the equity price and volatility are correlated,

$$\mathrm{Corr}[dW_s(t), dW_v(t)] \equiv \mathrm{Cov}[dW_s(t), dW_v(t)]/dt = \rho_{s,v}. \tag{10.47}$$

Empirical evidence implies that $\rho_{s,v} < 0$, higher volatility is related to lower stock prices and lower volatility leverage prices higher.

The complicates the method of the prior subsection, but JPR2004 handle this by proposing the correlations revised system model,

$$X_i = \sqrt{V_{i-1}} Z_{s,i}; \tag{10.48}$$

$$\log(V_i) = \alpha_v + \beta_v \log(V_{i-1}) + \widetilde{\sigma}_v \left(\rho_{s,v} Z_{s,i} + \sqrt{1-\rho^2_{s,v}} Z_{v,i}\right), \tag{10.49}$$

where now $\mathrm{Corr}[Z_{s,i}, Z_{v,i}] = 0$ and $[Z_{s,i}; Z_{v,i}] \overset{\mathrm{dist}}{=} \mathcal{N}(\vec{0}_2, I_2)$.

[a] Fischer Black, (1976), Studies in Stock Price Volatility Changes, In American Statistical Association, Proceedings of the Business and Economic Statistics Section, pp. 177–181.

The added parameter $\rho_{s,v}$ suggest added two new parameters, $\phi_v = \widetilde{\sigma}_v \rho_{s,v}$ and $\omega_v = \widetilde{\sigma}_v^2 (1 - \rho_{s,v}^2)$, so the expanded parameter set is

$$\vec{\Theta} = [\alpha_v; \beta_v; \phi_v; \omega_v]. \tag{10.50}$$

The prior conjugate distributions for the parameters are

$$[\alpha_v; \beta_v]^{(\text{prior})} \overset{\text{dist}}{\sim} \mathcal{N}(\vec{m}_{1:2}, V_{1:2}); \tag{10.51}$$

$$\phi_v^{(\text{prior})} \overset{\text{dist}}{\sim} \mathcal{N}(m_3, V_3); \tag{10.52}$$

$$\omega_v^{(\text{prior})} \overset{\text{dist}}{\sim} \mathcal{IG}(\Psi_4, \text{dof}_4). \tag{10.53}$$

Conjugacy leads to the posterior full conditional parameter posteriors according to the Bayes rules,

$$f^{(\text{post})}(\alpha_v, \beta_v | \phi_v, \omega_v, \vec{v}, \vec{x}) \overset{\text{dist}}{\sim} \mathcal{N}\left(\vec{m}_{1:2}^{(\text{post})}, V_{1:2}^{(\text{post})}\right); \tag{10.54}$$

$$f^{(\text{post})}(\phi_v | \alpha_v, \beta_v, \phi_v, \omega_v, \vec{v}, \vec{x}) \overset{\text{dist}}{\sim} \mathcal{N}\left(m_3^{(\text{post})}, V_3^{(\text{post})}\right); \tag{10.55}$$

$$f^{(\text{post})}(\omega_v | \alpha_v, \beta_v, \phi_v, \vec{v}, \vec{x}) \overset{\text{dist}}{\sim} \mathcal{IG}\left(\Psi_4^{(\text{post})}, \text{dof}_4^{(\text{post})}\right). \tag{10.56}$$

These need to be converted by Bayes rules for likelihood dependence.

For the volatility latent variable $\vec{V}$, the nearest neighborhood connections need the Metropolis algorithm as in the previous case for

$$f^{(\mathrm{post})}(V_i|V_{i-1}, V_{i+1}, \vec{\Theta}, \vec{X}). \tag{10.57}$$

See (JPR2004) and the previous subsection for further details.

• *10.4. Heston's Square Root Volatility (Heston SVD) Model:*

The equity price, Heston[a] volatility system used by JP2006 is

$$dS(t) = S(t)\Big((\mu_s(t)+(\eta_v+0.5)V(t))dt+\sqrt{V(t)}dW_s(t)\Big); \quad (10.58)$$

$$dV(t) = \kappa_v(\theta_v - V(t))dt+\sigma_v\sqrt{V(t)}dW_v(t); \quad (10.59)$$

where presumably the term $(\eta_v+0.5)V(t)$ is an added volatility risk term plus a preemptive insertion of the diffusion coefficient so that it does appear in the equation for the log-return. It is assumed that the two Brownian motion term have constant correlation,

$$\mathrm{Corr}[dW_s(t), dW_v(t)] = \rho_{s,v}. \quad (10.60)$$

[a] S. L. Heston (1993), A Closedform Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, Reviews of Financial Studies, vol. 6, pp. 327–343.

The discretized system[a] is then

$$X_i = \widetilde{\eta}_v V_{i-1} + \sqrt{V_{i-1}} Z_{s,i}; \tag{10.61}$$

$$V_i = \alpha_v + \beta_v V_{i-1} + \widetilde{\sigma}_v \sqrt{V_{i-1}} Z_{v,i}, \tag{10.62}$$

where the excess log-return per year over the given spot rate is

$$X_i = \log(S_i/S_{i-1})/\sqrt{\Delta t} - r_{i-1}\sqrt{\Delta t}, \tag{10.63}$$

$\alpha_v = \kappa_v \theta_v \Delta t$, $\beta_v = 1 - \kappa_v \Delta t$, $\widetilde{\sigma}_v = \sigma_v \sqrt{\Delta t}$, $\widetilde{\eta}_v = \eta_v \sqrt{\Delta t}$, and $Z_{v,i} = \rho_{s,v} Z_{s,i} + \sqrt{1 - \rho_{s,v}^2}\,\widetilde{Z}_{v,i}$.

---

[a]Note only integration without transformation has been used to get (10.62) and it must be realized that the many models of finance are **singular diffusions**, so that care must taken in transforming them. Geometric Brownian motion is a singular diffusion, but linear coefficient mean that the singularities are exactly removable by the log-transformation. However, Heston's model is to complex for that and many transformations can spoil the diffusion property. See F.B. Hanson's recent paper on these issues ***Stochastic Calculus of Heston's Stochastic-Volatility Model, MTNS2010.***

For the parameters,

$$\vec{\Theta} = [\widetilde{\eta}_v; \alpha_v; \beta_v; \widetilde{\sigma}_v^2; \rho_{s,v}], \tag{10.64}$$

the **prior conjugate distributions** are

$$\widetilde{\eta}_v^{(\text{prior})} \overset{\text{dist}}{\sim} \mathcal{N}(m_1, V_1); \tag{10.65}$$

$$[\alpha_v; \beta_v]^{(\text{prior})} \overset{\text{dist}}{\sim} \mathcal{N}(\vec{m}_{2:3}, V_{2:3}); \tag{10.66}$$

$$\left[\widetilde{\sigma}_v^2\right]^{(\text{prior})} \overset{\text{dist}}{\sim} \mathcal{IG}(\Psi_4, \text{dof}_4); \tag{10.67}$$

$$\rho_{s,v}^{(\text{prior})} \overset{\text{dist}}{\sim} \mathcal{U}(-1, 1); \tag{10.68}$$

where $\mathcal{U}(a, b)$ denotes a uniform distribution on $[a, b]$ **and noting that the correlation coefficient is bounded** $-1 \leq \rho_{s,v} \leq 1$.

By Hammersley-Clifford, the **full posterior conditionals** are given in the Metropolis-Hastings MCMC algorithm, omitting the Bayes rule steps,

$$p(\widetilde{\eta}_v|\alpha_v,\beta_v,\widetilde{\sigma}_v^2,\rho_{s,v},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{N}(m_1^{post},V_1^{(\text{post})}); \tag{10.69}$$

$$p(\alpha_v,\beta_v|\widetilde{\sigma}_v^2,\rho_{s,v},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{N}(\vec{m}_{2:3}^{post},V_{2:3}^{(\text{post})}); \tag{10.70}$$

$$p(\widetilde{\sigma}_v^2|\alpha_v,\beta_v,\rho_{s,v},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{IG}(\Psi_4^{post},\text{dof}_4^{(\text{post})}); \tag{10.71}$$

$$p(\rho_{s,v}|\alpha_v,\beta_v,\widetilde{\sigma}_v^2,\vec{V},\vec{X}) \text{ by Metropolis algorithm;} \tag{10.72}$$

$$p(V_i|V_{i-1},V_{i+1},\vec{\Theta},\vec{X}) \text{ by Metropolis algorithm.} \tag{10.73}$$

After Eraker, Johannes and Polson (EJP2003)[a], a suggested volatility scheme for the nearest neighbors is

$$[V_i|V_{i-1}]^{(\text{post})} \stackrel{\text{dist}}{\sim} \mathcal{N}(\alpha_v+\beta_v V_{i-1},V_{i-1}\widetilde{\sigma}_v^2); \tag{10.74}$$

$$[V_{i+1}|V_i]^{(\text{post})} \stackrel{\text{dist}}{\sim} \mathcal{N}(\alpha_v+\beta_v V_i,V_i\widetilde{\sigma}_v^2), \tag{10.75}$$

---

[a] B. Eraker, M. Johannes and N. Polson (EJP2003), The Impact of Jumps in Equity Index Volatility and Returns, Journal of Finance, vol. 58, pp. 1269–1300.

• *10.4. Stochastic-Volatility, Contemporaneous Jump-Diffusion (SVCJ or SVCJD) in Both Return and Volatility Models:*

The model of Duffie, Pan and Singleton (DPS2000)[a] is a system of a two jump-diffusion for both equity prices and stochastic volatility (variance),

$$dS(t) = S(t)\Big((\mu_s(t)+\eta_v V(t))dt+\sqrt{V(t)}dW_s(t) \qquad (10.76)$$

$$+\left(e^{Q_s}-1\right) dP(t)\Big);$$

$$dV(t)= \kappa_v(\theta_v - V(t))dt+\sigma_v\sqrt{V(t)}dW_v(t)+Q_v dP(t); \qquad (10.77)$$

where again the pure Brownian motions are correlated as in (10.60) with coefficient $\rho_{s,v}$, $dP(t)$ is the **common or contemporaneous jump** counter with jump rate λ, with the jth jump time τ_j or jump counter J_j.

---

[a] D. Duffie, K. Singleton and J. Pan (DPS2000), Transform Analysis and Asset Pricing for Affine Jump–Diffusions, Econometrica, vol. 68, pp. 1343–1376. See also Eraker, Johannes and Polson (EJP2003) who compare SVCJD with independent equity-volatility jumps SVIJD model, as well as the SVD and SVJD models, which they call misspecified models; the (EJP2003) gives many further details for this section that are not in (JP2003/2006/2009).

However, the variance jump sizes are exponentially distributed $Q_{v,j} \stackrel{\text{dist}}{\sim} \mathcal{E}(\mu_v)$ and the equity price jump sizes are distributed conditionally normal stochastic volatility (variance),

$$[Q_{s,j}|Q_{v,j}] \stackrel{\text{dist}}{\sim} \mathcal{N}(\mu_s + \rho_s Q_{v,j}, \sigma_s^2). \tag{10.78}$$

The given spot rate is again $r(t)$.

The discretized double jump-diffusion model is, assuming a $0-1$ jump law for sufficiently small $\Lambda = \lambda \Delta t$,

$$X_i = \mu_x + \widetilde{\eta}_v V_{i-1} + \sqrt{V_{i-1}} Z_{s,i} + Q_{s,i} J_i; \tag{10.79}$$

$$V_i = \alpha_v + \beta_v V_{i-1} + \widetilde{\sigma}_v \sqrt{V_{i-1}} Z_{v,i} + Q_{v,i} J_i, \tag{10.80}$$

where the excess log-return per year over the given spot rate is

$$X_i = \log(S_i/S_{i-1})/\sqrt{\Delta t} - r_{i-1}\sqrt{\Delta t}, \tag{10.81}$$

$\alpha_v = \kappa_v \theta_v \Delta t$, $\beta_v = 1 - \kappa_v \Delta t$, $\widetilde{\sigma}_v = \sigma_v \sqrt{\Delta t}$, μ_x is another state-less risk (fudge?) factor, $\widetilde{\eta}_v = (\eta_v - 0.5)\sqrt{\Delta t}$, $\widetilde{\rho}_{s,v} = \rho_{s,v}/\sqrt{\Delta t}$ $\widetilde{\mu}_s = \mu_s/\sqrt{\Delta t}$, $\widetilde{\sigma}_s = \sigma_s/\sqrt{\Delta t}$, and $J_i = \Delta P(t - \Delta t)$.

Again, $Z_{v,i} = \rho_{s,v} Z_{s,i} + \sqrt{1-\rho_{s,v}^2}\,\widetilde{Z}_{v,i}$. The parameter vector is

$$\vec{\Theta} = [\mu_x, \widetilde{\eta}_v; \alpha_v; \beta_v; \widetilde{\sigma}_v^2; \rho_{s,v}; \Lambda; \mu_v; \widetilde{\mu}_s, \widetilde{\rho}_s, \widetilde{\sigma}_s]. \tag{10.82}$$

The **prior conjugate distributions** are

$$[\mu_x; \widetilde{\eta}_v]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{N}(\vec{m}_{1:2}, V_{1:2}); \tag{10.83}$$

$$[\alpha_v; \beta_v]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{N}(\vec{m}_{3:4}, V_{3:4}); \tag{10.84}$$

$$[\widetilde{\mu}_s; \widetilde{\rho}_s]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{N}(\vec{m}_{9:10}, V_{9:10}); \tag{10.85}$$

$$[\widetilde{\sigma}_v^2]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{IG}(\Psi_5, \mathrm{dof}_5); \tag{10.86}$$

$$[\widetilde{\sigma}_s^2]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{IG}(\Psi_{11}, \mathrm{dof}_{11}); \tag{10.87}$$

$$[\Lambda]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{B}eta(a_7, b_7); \tag{10.88}$$

$$[\mu_v]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{G}(a_8, b_8); \tag{10.89}$$

$$[\rho_{s,v}]^{(\mathrm{prior})} \overset{\mathrm{dist}}{\sim} \mathcal{U}(-1, 1); \tag{10.90}$$

where is assumed that the normal priors are independent.

Again, by Hammersley-Clifford, the **parameter full posterior conditionals** are given in the Metropolis-Hastings MCMC algorithm,

$$p(\mu_x;\widetilde{\eta}_v|\vec{\Theta}_{i\neq 1:2},J,\vec{Q},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{N}(\vec{m}_{1:2}^{(\text{post})},V_{1:2}^{(\text{post})}); \quad (10.91)$$

$$p(\alpha_v,\beta_v|\vec{\Theta}_{i\neq 3:4},J,\vec{Q},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{N}(\vec{m}_{3:4}^{(\text{post})},V_{3:4}^{(\text{post})}); \quad (10.92)$$

$$p(\widetilde{\sigma}_v^2|\vec{\Theta}_{i\neq 5},J,\vec{Q},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{IG}(\Psi_5^{(\text{post})},\text{dof}_5^{(\text{post})}); \quad (10.93)$$

$$p(\Lambda|\vec{\Theta}_{i\neq 7},J,\vec{Q},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{B}eta(a_7^{(\text{post})},b_7^{(\text{post})}); \quad (10.94)$$

$$p(\widetilde{\mu}_s;\widetilde{\rho}_s|\vec{\Theta}_{i\neq 9:10},J,\vec{Q},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{N}(\vec{m}_{9:10}^{(\text{post})},V_{9:10}^{(\text{post})}); \quad (10.95)$$

$$p(\widetilde{\sigma}_s^2|\vec{\Theta}_{i\neq 11},J,\vec{Q},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{IG}(\Psi_{11}^{(\text{post})},\text{dof}_{11}^{(\text{post})}); \quad (10.96)$$

$$p(\mu_v|\vec{\Theta}_{i\neq 8},J,\vec{Q},\vec{V},\vec{X}) \stackrel{\text{dist}}{\sim} \mathcal{G}(a_8^{(\text{post})},b_8^{(\text{post})}); \quad (10.97)$$

$$p(\rho_{s,v}|\alpha_v,\beta_v,\widetilde{\sigma}_v^2,\vec{V},\vec{X}) \text{ by Metropolis algorithm}; \quad (10.98)$$

the last needs Metropolis due to lack of conjugacy of the uniform prior with the likelihood distribution. Again. the Bayes step is omitted.

By Hammersley-Clifford, the **latent state full posterior conditionals** are given in the Metropolis-Hastings MCMC algorithm,

$$p(Q_{v,i}|\vec{\Theta},J_i=1,Q_{s,i},V_i,V_{i-1}) \overset{\text{dist}}{\sim} \mathcal{TN} \overset{\text{dist}}{=} \mathcal{N}\times 1_{Q_{v,i}>0}; \quad (10.99)$$

$$p(Q_{s,i}|\vec{\Theta},J_i=1,Q_{v,i},V_i,V_{i-1},X_i) \overset{\text{dist}}{\sim} \mathcal{N}; \quad (10.100)$$

$$p(J_i=1|\vec{\Theta},Q_{s,i},Q_{v,i},V_i,V_{i-1},X_i) \overset{\text{dist}}{\sim} \mathcal{B}er; \quad (10.101)$$

$$p(J_i=0|\vec{\Theta},Q_{s,i},Q_{v,i},V_i,V_{i-1},X_i) \overset{\text{dist}}{\sim} \mathcal{B}er; \quad (10.102)$$

$$p(V_i|V_{i-1},V_{i+1},\vec{\Theta},\vec{X})\ \text{by Metropolis.} \quad (10.103)$$

The $\mathcal{TN}$ denotes the **truncated-normal distribution**, here used to enforce the positivity of volatility jumps, $Q_{v,i}>0$. The $\mathcal{B}er$ denotes the **Bernoulli distribution**, such that the posterior probability of a jump (success) is $\Lambda/(1+\Lambda)$ (avoid poor approximation Λ) for the assumed $0-1$ jump law.

While the standard Bayesian estimation procedure suggests using uninformative priors, (EJP2003) recommend using some informative priors for consistency with the $\mathbf{0-1}$ jump law assumption that **jumps are rare**, but also that **jump being large is their distinguishing feature** from the continuous diffusion changes, so $\boldsymbol{\Lambda}$ should be chosen low and that the equity mean jump $\boldsymbol{\mu_s}$ and possibly $\boldsymbol{\sigma_s}$ be chosen large to focus on large jumps.

The (EJP3003) authors give the following **advantages of the MCMC estimation methods in general**:

1. MCMC efficiently estimates latent (hidden) state variables such as volatility, jump counters and jump sizes.
2. MCMC handles estimation of risk factors.
3. MCMC offers superior sampling procedures through decomposing variables and parameters into **manageable modules**.
4. MCMC is computationally efficient with built in simulation checks of accuracy.

Table III

S&P 500 Parameter Estimates

Parameter estimates for the S&P 500 index data, January 2, 1980, to December 31, 1999. The models and parameterizations are given in Section I and the estimates correspond to percentage changes in the index value. For each parameter, we report the mean of the posterior deviation and the standard deviation of the posterior in parentheses.

	SV	SVJ	SVCJ	SVIJ
μ	0.0444 (0.0110)	0.0496 (0.0109)	0.0554 (0.0112)	0.0506 (0.0111)
θ	0.9052 (0.1077)	0.8136 (0.1244)	0.5376 (0.0539)	0.5585 (0.0811)
κ	0.0231 (0.0068)	0.0128 (0.0039)	0.0260 (0.0041)	0.0250 (0.0057)
σ_v	0.1434 (0.0128)	0.0954 (0.0104)	0.0790 (0.0074)	0.0896 (0.0115)
μ_y		− 2.5862 (1.3034)	− 1.7533 (1.5566)	− 3.0851 (3.2485)
ρ_J			− 0.6008 (0.9918)	
σ_y		4.0720 (1.7210)	2.8864 (0.5679)	2.9890 (0.7486)
μ_v			1.4832 (0.3404)	1.7980 (0.5737)
ρ	− 0.3974 (0.0516)	− 0.4668 (0.0579)	− 0.4838 (0.0623)	− 0.5040 (0.0661)
λ_y		0.0060 (0.0021)	0.0066 (0.0020)	0.0046 (0.0020)
λ_v				0.0055 (0.0032)

Figure 10.4: Eraker, Johannes and Polson (EJP2003), Table III, p. 1280, for 1980-1999 S&P 500 index parameter estimates comparing SVD, SVJD, SVCJD (model of this section) and SVIJD models. Recall the two $\boldsymbol{\lambda_k}$ should be $\boldsymbol{\Lambda_k}$ for $\boldsymbol{k = y : v}$.

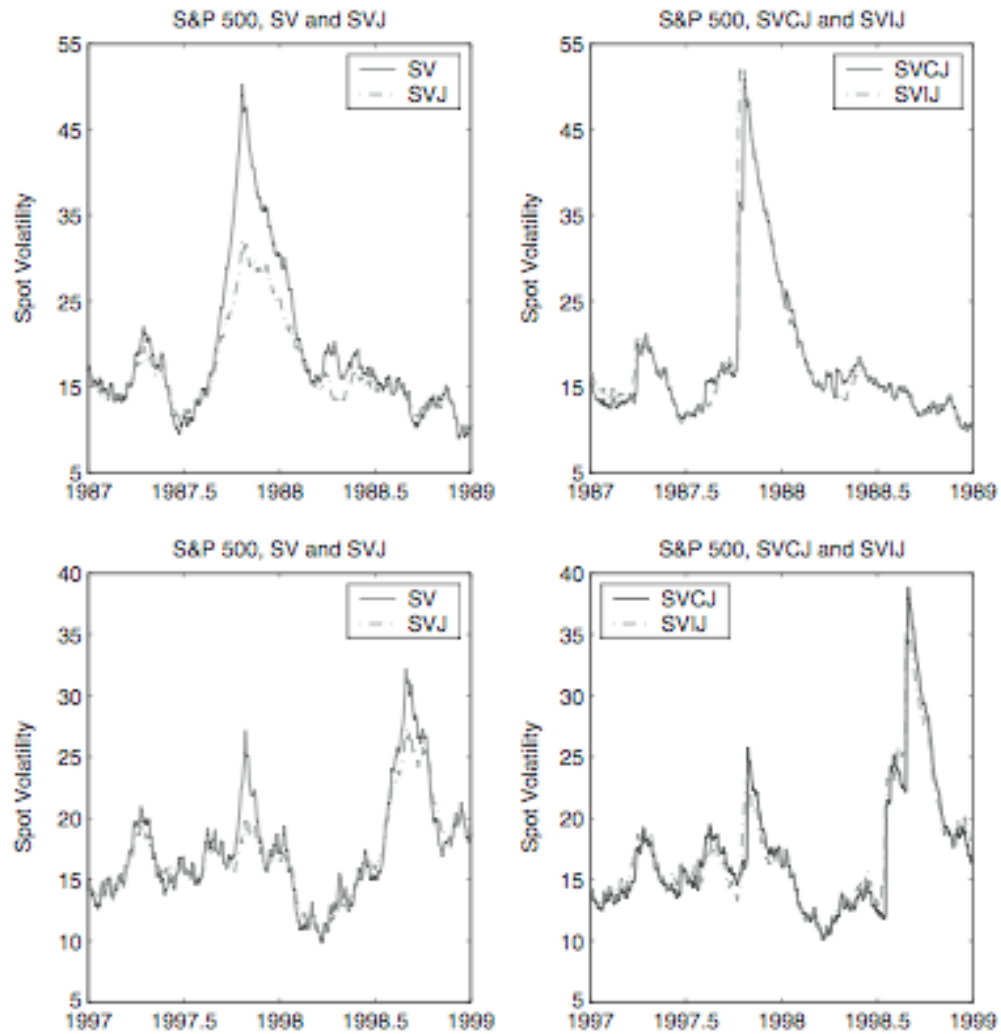


Figure 1. Estimated volatility paths for the S&P 500 index for the four models.

Figure 10.5: Eraker, Johannes and Polson (EJP2003), Figure 1, p. 1282, 1980's top and 1990's bottom, for the S&P 500 index estimated volatility paths comparing SVD, SVJD, SVCJD (model of this section) and SVIJD models.

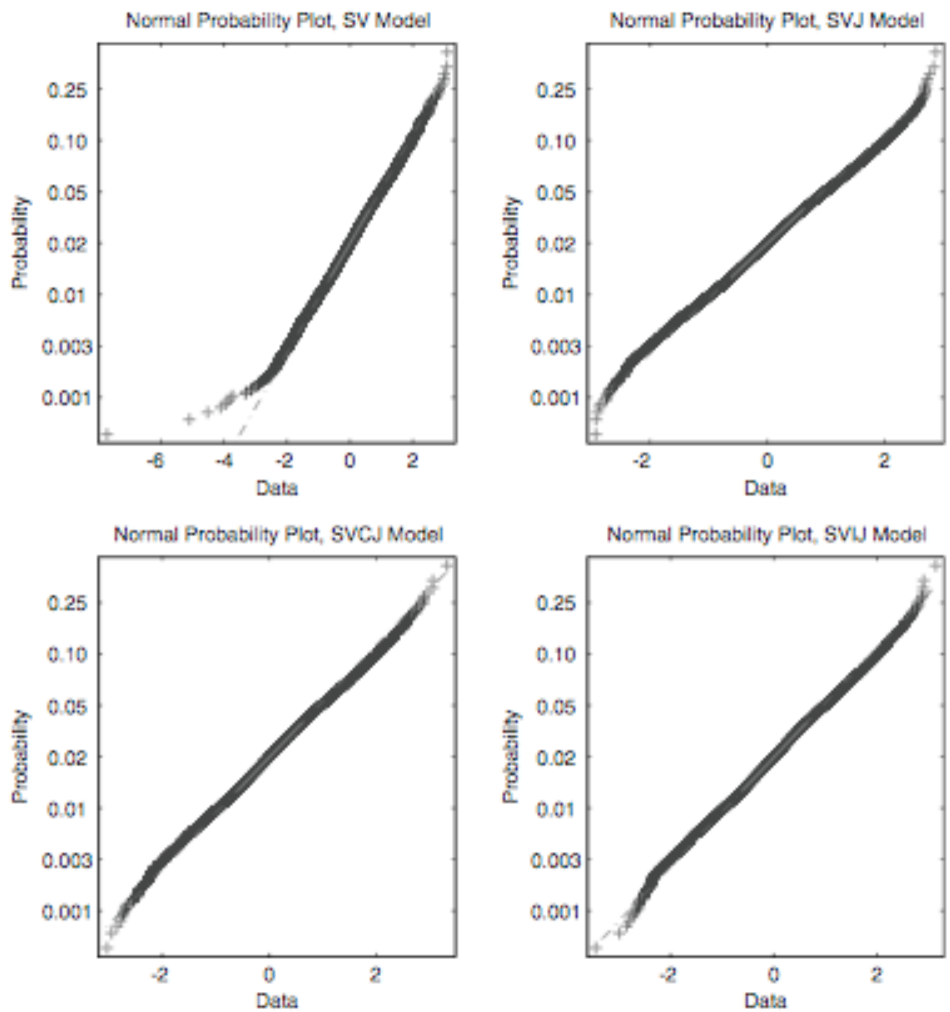


Figure 2. QQ or normality plot of the residuals for each of the models.

Figure 10.6: Eraker, Johannes and Polson (EJP2003), Figure 2, p. 1283, for 1980-1999 S&P 500 index QQ plot test comparing SVD, SVJD, SVCJD (model of this section) and SVIJD model residuals against the data.

Table VI
Log-Bayes Factors for the S&P 500 and Nasdaq 100

	SV vs. SVJ	SV vs. SVCJ	SV vs. SVIJ	SVJ vs. SVIJ	SVCJ vs. SVIJ
S&P 500	47.62	49.32	59.45	33.83	10.12
Nasdaq 100	27.16	28.95	56.03	33.19	27.08

Figure 10.7: Eraker, Johannes and Polson (EJP2003), Table VI, p. 1283, for 1980-1999 S&P 500 index comparing the Log-Bayes factors for the model. **Caution**: from the text commentary on both QQ plot and Bayes factors, SV and SVJ are **misspecified** models, hence using the reciprocal of (JP2006) Bayes factor for **SV1 vs. SV2** with arguments in the reversed order $\mathbf{LogBF(SV2, SV1)} = \log(P2) - \log(P1) > 0$ means **P2>P1** or **SV2≫SV1**, i.e., **SV2** is better. From the comments in (JP2003), looks like Bayes factor used is $BF(SV2, SV1) = p(X|SV2)/p(X|SV1)$, the likelihood ratio, so that $p(SV2|X)/p(SV1|X) = BF(SV2, SV1)p(SV2)/p(SV1)$.

mhsample

Purpose Metropolis-Hastings sample

Syntax

```
smpl = mhsample(start,nsamples,'pdf',pdf,'proppdf',proppdf,
    'proprnd',proprnd)
smpl = mhsample(...,'symmetric',sym)
smpl = mhsample(...,'burnin',K)
smpl = mhsample(...,'thin',m)
smpl = mhsample(...,'nchain',n)
[smpl,accept] = mhsample(...)
```

Description `smpl =`
`mhsample(start,nsamples,'pdf',pdf,'proppdf',proppdf,'proprnd',proprnd)`
draws `nsamples` random samples from a target stationary distribution `pdf` using the Metropolis-Hastings algorithm.

Figure 10.8: MATLAB **Statistics Toolbox (STB)** **`mhsample`** Metropolis Hastings function, p. 1a. **See the STB Guide.**

pdf using the Metropolis-Hastings algorithm.

`start` is a row vector containing the start value of the Markov Chain, `nsamples` is an integer specifying the number of samples to be generated, and `pdf`, `proppdf`, and `proprnd` are function handles created using `@`. `proppdf` defines the proposal distribution density, and `proprnd` defines the random number generator for the proposal distribution. `pdf` and `proprnd` take one argument as an input with the same type and size as `start`. `proppdf` takes two arguments as inputs with the same type and size as `start`.

`smpl` is a column vector or matrix containing the samples. If the log density function is preferred, `'pdf'` and `'proppdf'` can be replaced with `'logpdf'` and `'logproppdf'`. The density functions used in Metropolis-Hastings algorithm are not necessarily normalized.

The proposal distribution $q(x,y)$ gives the probability density for choosing x as the next point when y is the current point. It is sometimes written as $q(x|y)$.

If the `proppdf` or `logproppdf` satisfies $q(x,y) = q(y,x)$, that is, the proposal distribution is symmetric, `mhsample` implements Random Walk Metropolis-Hastings sampling. If the `proppdf` or `logproppdf` satisfies $q(x,y) = q(x)$, that is, the proposal distribution is independent of current values, `mhsample` implements Independent Metropolis-Hastings sampling.

Figure 10.9: MATLAB STB **`mhsample`** Metropolis Hastings function, p. 1b.
The advantage of **`'logpdf'`** is that is cuts down growth of the likelihood.

`smpl = mhsample(...,'symmetric',sym)` draws `nsamples` random samples from a target stationary distribution `pdf` using the Metropolis-Hastings algorithm. `sym` is a logical value that indicates whether the proposal distribution is symmetric. The default value is false, which corresponds to the asymmetric proposal distribution. If `sym` is true, for example, the proposal distribution is symmetric, `proppdf` and `logproppdf` are optional.

`smpl = mhsample(...,'burnin',K)` generates a Markov chain with values between the starting point and the kth point omitted in the generated sequence. Values beyond the kth point are kept. `k` is a nonnegative integer with default value of `0`.

`smpl = mhsample(...,'thin',m)` generates a Markov chain with `m-1` out of `m` values omitted in the generated sequence. `m` is a positive integer with default value of `1`.

`smpl = mhsample(...,'nchain',n)` generates `n` Markov chains using the Metropolis-Hastings algorithm. `n` is a positive integer with a default value of 1. `smpl` is a matrix containing the samples. The last dimension contains the indices for individual chains.

Figure 10.10: MATLAB STB **`mhsample`** Metropolis Hastings function, p. 2a.

[smpl,accept] = mhsample(...) also returns accept, the acceptance rate of the proposed distribution. accept is a scalar if a single chain is generated and is a vector if multiple chains are generated.

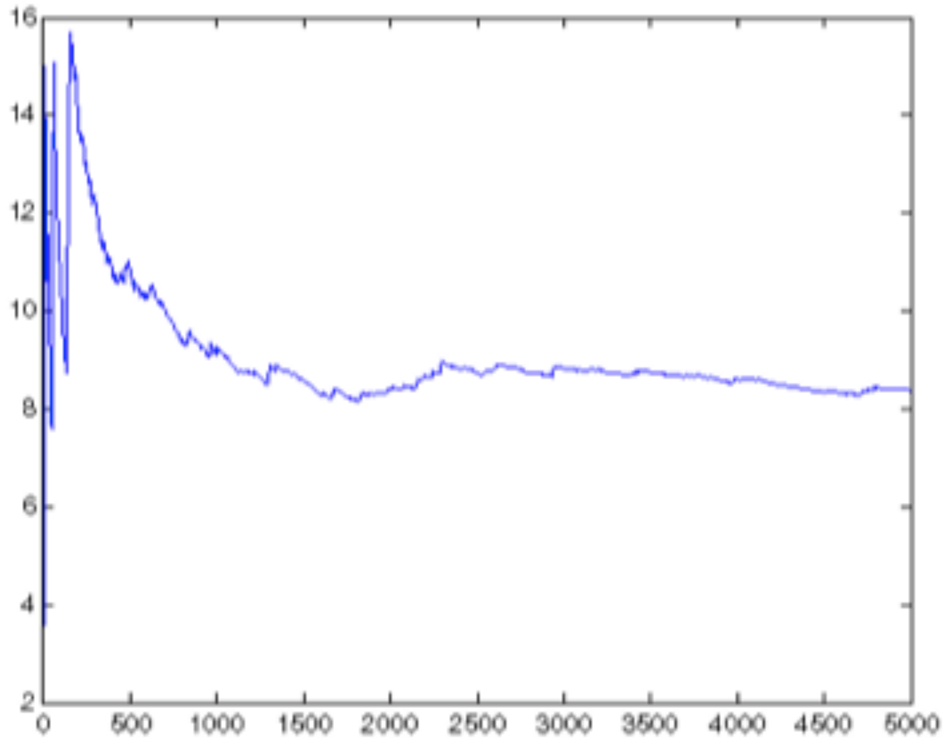
Examples

Estimate the second order moment of a Gamma distribution using the Independent Metropolis-Hastings sampling.

```
alpha = 2.43;
beta = 1;
pdf = @(x)gampdf(x,alpha,beta); %target distribution
proppdf = @(x,y)gampdf(x,floor(alpha),floor(alpha)/alpha);
proprnd = @(x)sum(...
              exprnd(floor(alpha)/alpha,floor(alpha),1));
nsamples = 5000;
smpl = mhsample(1,nsamples,'pdf',pdf,'proprnd',proprnd,...
                'proppdf',proppdf);
xxhat = cumsum(smpl.^2)./(1:nsamples)';
```

Figure 10.11: MATLAB STB **mhsample** Metropolis Hastings function, p. 2b.

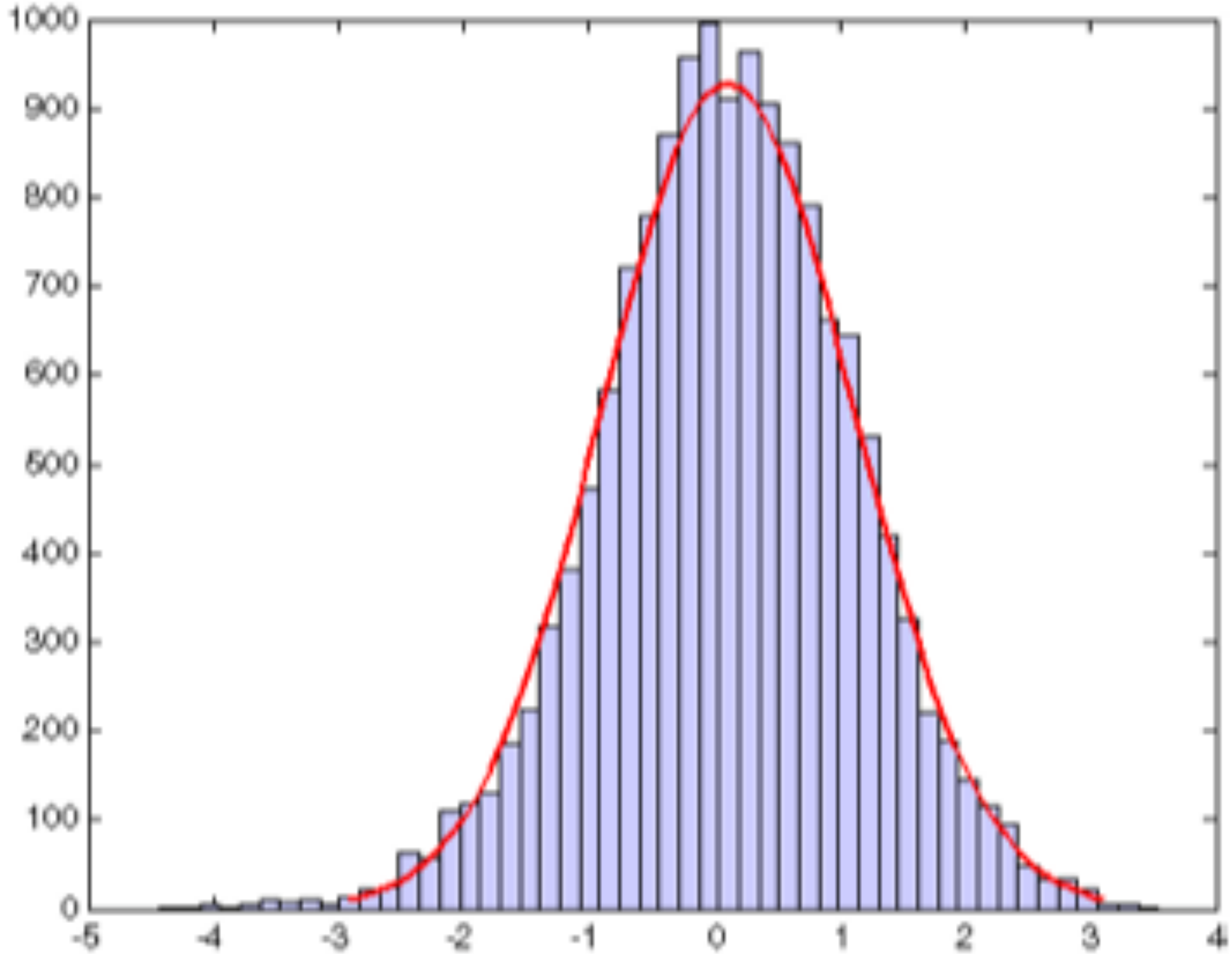
```
plot(1:nsamples,xxhat)
```



Generate random samples from N(0,1) using the Random Walk Metropolis-Hastings sampling.

```
delta = .5;
pdf = @(x) normpdf(x);
proppdf = @(x,y) unifpdf(y-x,-delta,delta);
proprnd = @(x) x + rand*2*delta - delta;
nsamples = 15000;
x = mhsample(1,nsamples,'pdf',pdf,'proprnd',proprnd,'symmetric',1);
histfit(x,50)
h = get(gca,'Children');
set(h(2),'FaceColor',[.8 .8 1])
```

Figure 10.12: MATLAB STB **mhsample** Metropolis Hastings function, p. 3.



See Also `slicesample`, `rand`

Figure 10.13: MATLAB STB **`mhsample`** Metropolis Hastings function, p. 4.

slicesample

Purpose Slice sampler

Syntax

```
rnd = slicesample(initial,nsamples,'pdf',pdf)
rnd = slicesample(...,'width',w)
rnd = slicesample(...,'burnin',k)
rnd = slicesample(...,'thin',m)
[rnd,neval] = slicesample(...)
```

Description `rnd = slicesample(initial,nsamples,'pdf',pdf)` generates `nsamples` random samples from a target distribution whose density function is defined by `pdf` using the slice sampling method. `initial` is a row vector or scalar containing the initial value of the random sample sequences. `initial` must be within the domain of the target distribution. `nsamples` is the number of samples to be generated. `pdf` is a function handle created using `@`. `pdf` accepts only one argument that must be the same type and size as `initial`. It defines a function that is proportional to the target density function. If the log density function is preferred, `'pdf'` can be replaced with `'logpdf'`. The log density function is not necessarily normalized.

Figure 10.14: MATLAB STB **`sliceesample`** Slice Sampling function, p. 1a. Slice sampling is simpler to use than Metropolis Hastings.

`rnd = slicesample(...,'width',w)` performs slice sampling for the target distribution with a typical width `w`. `w` is a scalar or vector. If it is a scalar, all dimensions are assumed to have the same typical widths. If it is a vector, each element of the vector is the typical width of the marginal target distribution in that dimension. The default value of `w` is `10`.

`rnd = slicesample(...,'burnin',k)` generates random samples with values between the starting point and the `k`th point omitted in the generated sequence. Values beyond the `k`th point are kept. `k` is a nonnegative integer with default value of `0`.

`rnd = slicesample(...,'thin',m)` generates random samples with `m-1` out of `m` values omitted in the generated sequence. `m` is a positive integer with default value of `1`.

`[rnd,neval] = slicesample(...)` also returns `neval`, the averaged number of function evaluations that occurred in the slice sampling. `neval` is a scalar.

Figure 10.15: MATLAB STB **`slicesample`** Slice Sampling function, p. 1b.

Example

Generate random samples from a distribution with a user-defined pdf.

First, define the function that is proportional to the pdf for a multi-modal distribution.

```
f = @(x) exp( -x.^2/2).*(1+(sin(3*x)).^2).* ...
    (1+(cos(5*x).^2));
```

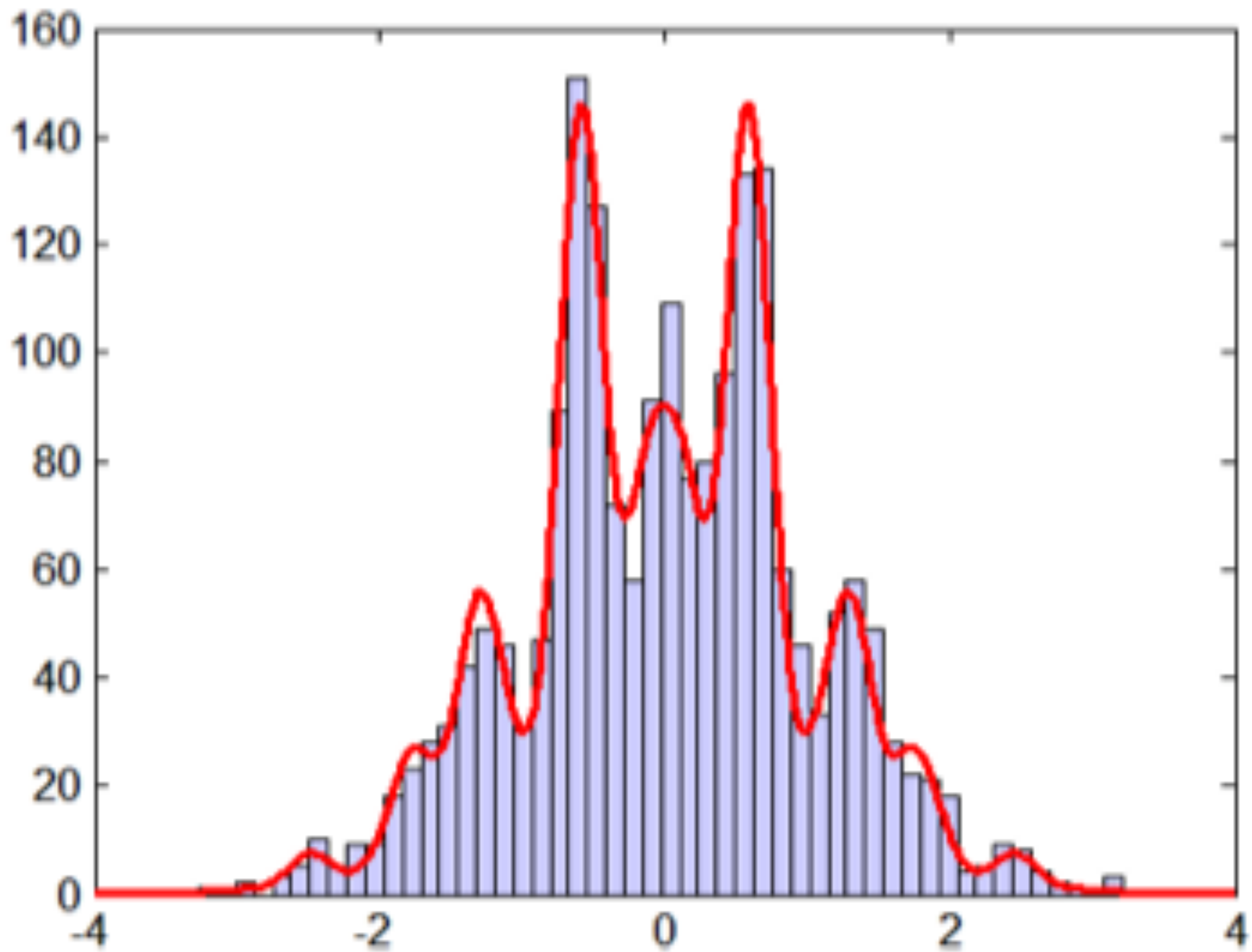
Next, use the `slicesample` function to generate the random samples for the function defined above.

```
x = slicesample(1,2000,'pdf',f,'thin',5,'burnin',1000);
```

Now, plot a histogram of the random samples generated.

```
hist(x,50)
set(get(gca,'child'),'facecolor',[0.8 .8 1]);
hold on
xd = get(gca,'XLim'); % Gets the xdata of the bins
binwidth = (xd(2)-xd(1)); % Finds the width of each bin
% Use linspace to normalize the histogram
y = 5.6398*binwidth*f(linspace(xd(1),xd(2),1000));
plot(linspace(xd(1),xd(2),1000),y,'r','LineWidth',2)
```

Figure 10.16: MATLAB STB **`mhsample`** Slice Sampling function, p. 2. For a MATLAB demo. code use **`help bayesdemo`**, for a 11 page ***Bayesian Analysis of a Logistic Regression*** document with a link at the top to open or run the code.



See Also rand, mhsample, randsample

Figure 10.17: MATLAB STB **slicesample** Slice Sampling function, p. 3.

Markov Chain Samplers

Introduction

The methods discussed in "Common Generation Methods" on page 5-133 may be inadequate when sampling distributions are difficult to represent in computations. Such distributions arise, for example, in Bayesian data analysis and in the large combinatorial problems of Markov chain Monte Carlo (MCMC) simulations. An alternative is to construct a Markov chain with a stationary distribution equal to the target sampling distribution, using the states of the chain to generate random numbers after an initial burn-in period in which the state distribution converges to the target.

Figure 10.18: MATLAB STB Guide Markov Chain Samplers, p. 1a. Some brief explanations of **mhsample** and **slicesample** methods.

Metropolis-Hastings Sampler

The Metropolis-Hastings algorithm draws samples from a distribution that is only known up to a constant. Random numbers are generated from a distribution with a probability density function that is equal to or proportional to a proposal function.

The following steps are used to generate random numbers:

1. Assume a initial value $x(t)$.

2. Draw a sample, $y(t)$, from a proposal distribution $q(y \mid x(t))$.

3. Accept $y(t)$ as the next sample $x(t+1)$ with probability $r(x(t),y(t))$, and keep $x(t)$ as the next sample $x(t+1)$ with probability $1-r(x(t),y(t))$, where

$$r(x,y) = min\left\{\frac{f(y)}{f(x)}\frac{q(x \mid y)}{q(y \mid x)}, 1\right\}$$

4. Increment $t \to t+1$, and repeat steps 2 and 3 until the desired number of samples are obtained.

Figure 10.19: MATLAB STB Guide Markov Chain Samplers, p. 1b. Some brief explanations of `mhsample` and `slicesample` methods.

You can generate random numbers using the Metropolis-Hastings method with the `mhsample` function. To produce quality samples efficiently with Metropolis-Hastings algorithm, it is crucial to select a good proposal distribution. If it is difficult to find an efficient proposal distribution, you can use the slice sampling algorithm without explicitly specifying a proposal distribution.

Slice Sampler

In instances where it is difficult to find an efficient Metropolis-Hastings proposal distribution, there are a few algorithms, such as the slice sampling algorithm, that do not require an explicit specification for the proposal distribution. The slice sampling algorithm draws samples from the region under the density function using a sequence of vertical and horizontal steps. First, it selects a height at random between 0 and the density function $f(x)$. Then, it selects a new x value at random by sampling from the horizontal "slice" of the density above the selected height. A similar slice sampling algorithm is used for a multivariate distribution.

Figure 10.20: MATLAB STB Guide Markov Chain Samplers, p. 2a. Some brief explanations of **`mhsample`** and **`slicesample`** methods.

If a function $f(x)$ proportional to the density function is given, the following steps are used to generate random numbers:

1 Assume a initial value $x(t)$ within the domain of $f(x)$.

2 Draw a real value y uniformly from $(0, f(x(t)))$, thereby defining a horizontal "slice" as $S = \{x: y < f(x)\}$.

3 Find an interval $I = (L, R)$ around $x(t)$ that contains all, or much of the "slice" S.

4 Draw the new point $x(t+1)$ within this interval.

5 Increment $t \rightarrow t+1$ and repeat steps 2 through 4 until the desired number of samples are obtained.

Slice sampling can generate random numbers from a distribution with an arbitrary form of the density function, provided that an efficient numerical procedure is available to find the interval $I = (L, R)$, which is the "slice" of the density.

Figure 10.21: MATLAB STB Guide Markov Chain Samplers, p. 2b. Some brief explanations. Generate random numbers using the slice sampling method with the **`slicesample`** function. **`y=f(x(t))*rand; x(t+1)=L+(R-L)*rand;`**

★ **Notes on Slice, Metropolis-Hastings and Sampling in General:**

- Emphasize the **posterior full conditional distributions** to avoid handling multivariate, joint distributions due to excessive analytical complexity and large scale computational demands.
- The **basic statistical properties of the posterior** parameter and other distributions can be obtained from the final sampling using **MATLAB** **`mean`**, **`std`** and other functions; the sample mean is the usual estimate for parameters.
- **Latent or hidden state trajectories** can be obtained by plotting a sample sample from the final iteration posterior distribution of the final iteration.
- A good reference on both samplers is C.P. Robert and G. Casella (2004), Monte Carlo Statistical Methods, pp. 267-320 for the Metropolis-Hastings sampler and pp. 321-336f or the slice sampler, with the original published paper by R. Neal (2003), Slice Sampling, Ann. Stat., vol. 31, pp. 705-767, including discussion.

- The size of the full likelihood distribution can become quite large due to the product of the partial likelihoods of each of the data, but for **`slicesample`** or **`mhsample`** functions the **`'logpdf'`** option in place of the **`'pdf'`** default can be used to input the log-densities to reduce growth. The **`mhsample`** also has a **`'logproppdf'`** instead of **`'proppdf'`** for the proposed PDF option.

- For a **`MATLAB slicesample`** **demonstration code** use

$$\texttt{help bayesdemo} \tag{10.104}$$

  or use the **`Help`** window, to obtain an 11 page ***Bayesian Analysis of a Logistic Regression*** document with a link at the top to open or run the code. The code and document are a Bayesian example applied to experimental data on car performance using a **logistic or logit model**. The short document and also the code contain much useful tutorial information on Bayesian inference, slice sampling and the analysis of sampler output.

* Summary of Lecture 10:

1. Bayesian Estimation Estimation in Finance, Part II
2. Bayesian Time-Varying Equity Risk Premium for BS Model
3. Bayesian Log-Stochastic Volatility,Log-SVD
4. Bayesian Correlated-SVD Leverage Effect
5. Bayesian Heston Square Root SVD
6. Bayesian Contemporaneous Jumps, SVCJD
7. Eraker, Johannes & Polson (2003) SVD, SVJD, SVCJD & SVIJD Comparisons
8. MATLAB Metropolis Hastings `mhsample` Function
9. MATLAB Slice Sampler `slicesample` Function
10. MATLAB Brief Explanations of mhsample and slicesample Methods.