Optimal Filtering of Jump Diffusions: Extracting Latent States from Asset Prices

Michael S. Johannes
Columbia University

Nicholas G. Polson
University of Chicago

Jonathan R. Stroud
George Washington University

This paper provides an optimal filtering methodology in discretely observed continuous-time jump-diffusion models. Although the filtering problem has received little attention, it is useful for estimating latent states, forecasting volatility and returns, computing model diagnostics such as likelihood ratios, and parameter estimation. Our approach combines time-discretization schemes with Monte Carlo methods. It is quite general, applying in nonlinear and multivariate jump-diffusion models and models with nonanalytic observation equations. We provide a detailed analysis of the filter's performance, and analyze four applications: disentangling jumps from stochastic volatility, forecasting volatility, comparing models via likelihood ratios, and filtering using option prices and returns. (*JEL* C11, C13, C15, C51, C52, G11, G12, G17)

This paper develops a filtering approach for learning about unobserved shocks and states from discretely observed prices generated by continuous-time jump-diffusion models. The optimal filtering problem, which has received little attention, is the natural companion to the parameter estimation problem as filters allow researchers to analyze and use their models for practical applications.[1] Despite its usefulness, there is currently no general method for solving the filtering problem in continuous-time models.

---

[1] There is now a long list of methods for estimating parameters in continuous-time models via the simulated method of moments (Duffie and Singleton 1993; Gallant and Long 1997; Gallant and Tauchen 2009), approximate maximum likelihood (Aït-Sahalia 2002, 2008; Aït-Sahalia and Kimmel 2005, 2007; Bates 2006), simulated maximum likelihood (Brandt and Santa-Clara 2002; Piazzesi 2005), GMM (Pan 2002), Markov chain Monte Carlo methods (Jones 1998; Eraker 2001; Elerian, Shephard, and Chib 2001; Chib, Pitt, and Shephard 2005), and nonparametric methods (Aït-Sahalia 1996a, 1996b).

Before discussing our filtering methodology, it is important to understand why optimal filters are so useful in practice. Filtering distributions provide estimates of latent variables that are useful for forecasting volatility or price distributions. They also allow researchers to identify or "disentangle" (Aït-Sahalia 2004) transient from persistent shocks. Forecasts are crucial for using continuous-time models for applications such as portfolio allocation or option pricing, which require volatility estimates. Optimal filters are also useful for model specification, as the construction of likelihood functions (and therefore likelihood ratio statistics) requires the filtering density. Existing specification tools such as computing residuals or well-chosen moments are rather *ad hoc*.[2] Conditional likelihoods and the associated likelihood ratio statistics are objective and universal metrics, as the likelihood embeds and properly weighs all of the information in the model, in contrast to certain well-chosen moments. Of course, optimal filters can and are used for likelihood based parameter estimation.[3]

On the theoretical side, we propose an optimal filtering scheme combining popular time-discretization schemes (e.g., the Euler scheme) with particle filtering methods (Gordon, Salmond, and Smith 1993; and Pitt and Shephard 1999). The approach is quite general, as it applies in general multivariate jump-diffusion models and even allows for nonanalytical observations that commonly arise in option pricing applications (see, for example, Christoffersen, Jacobs, and Mimouni 2007). Thus, our setup encompasses all of the types of models and data encountered in practice. The algorithm is easy to understand, simple to implement, and computationally fast, requiring only simulation from well-known distributions. Like all Monte Carlo procedures, our framework has the advantage that it is highly adaptable and can be tailored for specific applications.

On the empirical side, we use the filter to understand the structure of jumps and volatility in the context of the double-jump model of Duffie, Pan, and Singleton (2000). This model is popular for option pricing and portfolio allocation applications, due to its analytical tractability and its ability to generate flexible return distributions. Statistically, the model presents a significant filtering challenge due to the complicated shock structure, as price variation is driven by jumps in prices and persistent volatility, which itself is driven by a jump factor. Disentangling jumps and stochastic volatility, in particular, is important as prior research documents that jumps and diffusive components are compensated differently (see, for example, Pan 2002 or Broadie, Chernov, and Johannes 2007).

---

[2] Eraker, Johannes, and Polson (2003) and Li, Wells, and Yu (2007) analyze residual plots. Other papers, such as Andersen, Benzoni, and Lund (2001); Chernov et al. (2003); Johannes (2004); and Broadie, Chernov, and Johannes (2007) use unconditional moments. In affine models, Bates (2001) uses normalized factor transitions assuming volatilities are observed, and Bates (2006) uses a likelihood-based metric.

[3] Christoffersen, Jacobs, and Mimouni (2007) and Golightly and Wilkinson (2006, 2008) use variants of our particle filtering algorithms for parameter estimation in linear and nonlinear diffusion specifications.

We first perform simulations to assess the performance of our particle filtering algorithm. For realistic parameters, there is little discretization bias for high frequencies such as daily. The bias is significant for lower frequencies, but is mitigated by simulating a small number of data points between observations. Identifying jumps is possible with daily data, but it is more difficult to do so with less frequently sampled data. For example, we can identify about 60% of the jumps at a daily frequency, while the remaining 40% are too small to identify. At a weekly frequency, less than a third of the jumps are correctly identified. Since volatility aggregates, weekly variance is roughly five times as large as daily variance, and therefore only the largest jumps can be identified at the weekly frequency.

Next, we use the filter for four commonly encountered applications: disentangling jumps from stochastic volatility, forecasting realized volatility, comparing models using likelihood based metrics, and filtering using both option prices and underlying returns. The first is the relatively standard problem of disentangling different shocks from the observed price movements. Not surprisingly, volatility estimates can vary dramatically across models, especially during periods of market stress. Second, we consider volatility forecasting accuracy, which, surprisingly, has received little attention in parametric continuous-time stochastic volatility models. Despite these differences, the root mean squared error of volatility forecasts across models is virtually the same. This occurs because of the different shocks and mean-reversion speeds. While the data wants volatility to rapidly increase during periods of market stress, the large positive jumps also require reasonably fast mean-reversion. However, in normal times, models with jumps in volatility mean-revert too quickly. The net result is that the models look very similar in terms of volatility forecasts.

Third, we compute sequential likelihood ratios across a pure stochastic volatility (SV) model and models that incorporate jumps in prices (SVJ), and also jumps in volatility (SVCJ). Full-sample likelihood ratios provide a clear ranking of models: SVCJ and SVJ are overwhelmingly preferred to SV, and SVCJ is preferred to SVJ, in contrast to the volatility forecasts. The sequential likelihood ratios indicate that model differentiation occurs primarily during market stress periods, showing the importance of accurate jump modeling for overall model specification. Since the models we consider have simple jump specifications (normal distributions), it would be interesting to investigate the gains from using more flexible jump specifications such as those non-normal or skewed jump sizes or those generated by Levy processes. The result is particularly striking as a standard metric, the accuracy of volatility forecasts, is not able to differentiate the models, while likelihood ratios provide a clear ordering.

Finally, we use option prices and index returns to filter volatility using both sources of information. Our filter efficiently combines the information in both sources, even though option prices are nonanalytic functions of the current stock price and volatility state. We consider settings with and without factor risk premiums and quantify first how much of the volatility estimation risk

is reduced by introducing options. If options were observed without error, volatility could also be estimated without error, but this is not reasonable as even at-the-money options have large bid-ask spreads. We incorporate realistic bid-ask spreads into our filtering approach and quantify the estimation risk via the posterior standard deviation of the filtering distribution. Even with a 10% pricing error/bid-ask spread, estimation risk falls by as much as 40% with the addition of option prices. This shows how informative option prices are, even with large bid-ask spreads.

Our results also indicate that it is difficult to reconcile existing models with the information in both returns and option prices. In the models with jumps in prices and in volatility, volatility filtered using returns and options still tends to be higher than volatility filtered from just returns. Factor risk premiums reduce spot volatility, but the two sources are still not wholly consistent. This is consistent with either model misspecification in the volatility process or time-varying risk-premiums. The nature of the misspecification is consistent with a model where the long-run mean of volatility varies over time. This model has been suggested by Duffie, Pan, and Singleton (2000) but has not been analyzed. Time-varying risk-premiums are also able to reconcile the issues, as spot volatility is not required to move as much to explain option implied expectations.

1. Optimal Filtering of Jump-Diffusions

1.1 General jump-diffusion models
We assume that log-prices, $Y_t = \log(S_t)$, and underlying state variables, X_t, jointly solve:

$$dY_t = \mu^s(X_t)\, dt + \sigma^s(X_t)\, dW_t^s + d\left(\sum_{n=1}^{N_t^s} Z_n^s\right), \qquad (1)$$

$$dX_t = \mu^x(X_t)\, dt + \sigma^x(X_t)\, dW_t^x + d\left(\sum_{n=1}^{N_t^x} Z_n^x\right), \qquad (2)$$

where W_t^s and W_t^x are potentially correlated Brownian motions, N_t^s and N_t^x are point processes with predictable intensities $\lambda^s(X_{t-})$ and $\lambda^x(X_{t-})$, τ_n^s and τ_n^x are the jump times, and Z_n^s and Z_n^x are the jump sizes with $\mathcal{F}_{\tau_{n-}}$ conditional distributions $\Pi^s(X_{\tau_n-})$ and $\Pi^x(X_{\tau_n-})$. For example, S_t could be an equity index, X_t its stochastic variance, Z_n^s the price jumps, and Z_n^x the variance jump sizes. As our focus is on filtering, we suppress the dependence of prices on any parameters, except where explicitly needed.

As is common in financial applications, we assume exogenous state variables, in the sense that the characteristics in Equation (2) are independent of prices. We also make a number of other assumptions, for notational parsimony: we assume that S_t and X_t are univariate, although our approach applies equally to

multivariate models; we model directly log-prices and assume that the level of the asset price does not appear on the right-hand side of Equation (1); we assume that the jump intensities and jump size distributions are state independent, $\lambda^s(X_{t-}) = \lambda_s$ and $\lambda^x(X_{t-}) = \lambda_x$, respectively; and we assume the Brownian motions are uncorrelated. Each of these assumptions is easy to relax.

The model in Equations (1) and (2) covers nearly all of the cases of interest in finance, in particular, multivariate diffusions, multivariate jump-diffusions (Duffie, Pan, and Singleton 2000) and continuous-time regime-switching models, and diffusion models in which the drift and diffusion are driven by a Markov chain (see, for example, Dai, Singleton, and Yang 2007).[4] We do not consider infinitely active Levy processes, but our approach can be easily generalized to this class, as noted by Barndorff-Nielsen and Shephard (2004 and 2006b).

One novel feature of our approach is that we allow for the observation of "derivative" prices whose value depends on S_t and/or X_t. Here, we have in mind either bond prices or option prices. Here,

$$C(S_t, X_t) = E^{\mathbb{Q}}\big[e^{-\int_t^T r_s ds} f(S_T)|S_t, X_t\big] \tag{3}$$

is the price of a claim that pays $f(S_t)$ at time T and $\mathbb{Q}$ is a risk-neutral probability measure. We require only the minimal assumption that the C can be evaluated. Thus, we allow C to be nonanalytic, a case that arises in option pricing and nonlinear term structure applications. In this case, derivatives may be highly informative about state variables, and the goal of the filtering problem is to learn about the latent states from both Y_t and C_t, where C_t is the observed market price of the option.

Derivative prices also pose an interesting challenge for filtering. Since they are a deterministic function of the underlying prices and states, they are potentially fully revealing of states, introducing a stochastic singularity. To circumvent this problem, we follow the literature in assuming that derivative prices are observed with error. That is, $C_t = C(S_t, X_t) + \varepsilon_t$ or $C_t = C(S_t, X_t)e^{\varepsilon_t}$, where ε_t is normally distributed. The motivation for pricing errors could be a genuine concern with noisy price observations, such as those generated by the large bid-ask spreads (index option prices), interpolated prices (term structure models with interpolated yields), or model-induced pricing errors.

Given the general class of models, we now discuss the general filtering problem, particle filters, and our algorithms.

1.2 Optimal filters and their uses
The solution to the optimal filtering problem consists of a sequence of densities, $p(L_t|y^t)$, that are computed for each time period t, where $y^t = (y_1, \ldots, y_t)$ are

---

[4] The following provides a partial list of applications using jump-diffusion models with potentially latent variables. In portfolio settings, see Pan and Liu (2003); Liu, Longstaff, and Pan (2002); Das and Uppal (2004); and Aït-Sahalia, Cocho-Diaz, and Hurd (2006); in fixed income settings, see Andersen, Benzoni, and Lund (2004); Johannes (2004); Piazzesi (2005); and Dai, Singleton, and Yang (2007); in option pricing settings, see Bates (2001, 2006); Bakshi, Cao, and Chen (1997); Pan (2002); and Duffie, Pan, and Singleton (2000).

the observed data and L_t are any unobserved latent variables. In our setting, the data y_t consists of price or return observations, $Y_t - Y_{t-1}$, and derivatives, C_t, that are generated by discretely spaced observations of the continuous-time model. The latent variables consist of the Markov states, X_t, jump times or jump sizes. Before discussing our filtering approach, we briefly discuss the four main uses of optimal filters: disentangling shocks, forecasting, specification analysis, and parameter estimation.

First, a central question in models with multiple sources of risk is to identify the source of the shock. In the context of jump-diffusions, Aït-Sahalia (2004) refers to this as "disentangling" jumps from stochastic volatility, understanding whether a price change is due to a persistent shock (volatility is higher) or a transient shock (a jump occurred). This problem is crucial in many economic settings, as decision makers respond differently to persistent and transient shocks. The second, and maybe most common, use of optimal filters is for forecasting latent states or prices. Forecasts using $p(L_t|y^t)$ are optimal in the sense that each of its conditional moments has the lowest mean-squared error among all potential estimators. Unlike GARCH models, forecasting exercises are rarely done using parametric continuous-time models, most likely because of the difficulty in computing the filtering distribution.[5] The goal of this paper is to provide such filters so as to make these exercises feasible for continuous-time models that are commonly used in practice. Bates (2006) is one notable exception, although his approach has the limitations that it applies only in affine settings and is difficult to extend to multiple-dimensional settings.

Third, $p(L_t|y^t)$ is useful for specification diagnostics. Here, it is common in both classical and Bayesian settings to consider K non-nested models, $\{\mathcal{M}_i\}_{i=1}^{K}$. After estimating parameters (via simulated method of moments, simulated or approximate maximum likelihood, or Markov chain Monte Carlo (MCMC) methods), the goal is to assess the model fit. Existing approaches in models with latent variables tend to focus on various intuitive, but *ad hoc*, diagnostics such as analyzing residuals or the fit of various moments, as mentioned earlier. A preferable approach would be to compute likelihood ratio statistics, for both the full sample and for each data point, as the likelihood provides a natural metric. Likelihood ratios can be used for both nested and non-nested models, as a measure of goodness of fit.

Likelihood computation requires the optimal filtering distribution. The full-sample likelihood of model $\mathcal{M}_i$ based on y^T is

$$\mathcal{L}(y^T|\mathcal{M}_i) = \prod_{t=1}^{T} p(y_t|y^{t-1}, \mathcal{M}_i),$$

---

[5] Chernov et al. (2003) and Bates (2006) compute filtered estimates, but do not consider the accuracy of volatility forecasts across models.

where

$$p(y_t|y^{t-1}, \mathcal{M}_i) = \int p(y_t|L_t, \mathcal{M}_i)p(L_t|y^{t-1}, \mathcal{M}_i)dL_t$$

is the likelihood of data point y_t conditional on past data. This is identical to the Bayes factor commonly used in Bayesian statistics. Continuous-time models pose an additional problem over and above the standard filtering problem, as all of the densities needed to compute likelihoods are not known, since transition densities for states or observed variables are generally not known. Full-sample likelihoods have been used as a model diagnostic tool in a number of discrete-time models (see, for example, Kim, Shephard, and Chib 1998 or Chib, Nardari, and Shephard 2002).

Although not reported in the literature, one can also track *sequential likelihood ratios*. The likelihood ratio at time t, $\mathcal{LR}_{i,j}(t)$, is defined recursively as

$$\mathcal{LR}_{i,j}(t) = \frac{\mathcal{L}(y^t|\mathcal{M}_i)}{\mathcal{L}(y^t|\mathcal{M}_j)} = \frac{p(y_t|y^{t-1}, \mathcal{M}_i)}{p(y_t|y^{t-1}, \mathcal{M}_j)}\mathcal{LR}_{i,j}(t-1).$$

Unlike full-sample likelihoods, sequential likelihoods identify the exact time periods in which models fail, in a relative sense. In particular, they allow researchers to discriminate between abrupt failures and those that accumulate slowly, providing a deeper understanding of how the models fit the data, unlike full-sample likelihood ratios. Another interesting case is when $\mathcal{LR}_{i,j}(T) > 1$, but the differences are driven by a few influential observations. Ignoring these observations, the fit could be poor with, for example, $\mathcal{LR}_{i,j}(t) < 1$ for most of the sample. This insidious case would not be diagnosed using full-sample likelihoods.

Finally, particle filters are useful for parameter estimation. In discrete-time models, Pitt (2002) uses particle filters to estimate parameters (see Durham and Gallant 2002 and Durham 2003 for related simulated maximum likelihood estimation appqqroaches). Particle-based likelihoods are not smooth functions of the parameters, but it is easy to use optimizers for nondifferentiable objective functions or by smoothing out the discontinuities (Pitt 2002). Building on our methodology, Christoffersen, Jacobs, and Minouni (2005) estimate models using particle filters for option pricing applications, and Golightly and Wilkinson (2006) use time discretizations and particle filters for parameter estimation in diffusion models.

1.3 Particle filters: General issues

The essence of the filtering problem is computing or approximating $p(L_t|y^t)$, as it is generally a nonlinear and nonanalytical function of the observed data. Historically, the burdens of computing optimal filters were overwhelming, due to the "curse of dimensionality," since $p(L_t|y^t)$ is a function of the entire observed history and L_t could be multidimensional.

Particle filters approximate $p(L_t|y^t)$ via a discrete distribution consisting of N support points or particles, $L_t^{(i)}$, and probabilities, $\pi_t^{(i)}$, for $i = 1, \dots, N$. Together, the approximation is a random histogram, denoted $p^N(L_t|y^t)$. The support points change over time, differentiating this approach from deterministic approximations with fixed grid points. As N increases, $p^N(L_t|y^t)$ converges to $p(L_t|y^t)$ under mild regularity conditions. The key to particle filters is their adaptability, as they can be modified to handle the specific structure of a given model. In this regard, particle filters, like other Monte Carlo procedures (e.g., MCMC, simulated maximum-likelihood/method of moments, and Monte Carlo methods for numerical derivative pricing) rely on their flexibility to adapt to a given application.

Although there are many particle filters, there are two algorithms that we utilize: the sampling-importance resampling (SIR) algorithm and the auxiliary particle filtering (APF) algorithm. SIR is the classic particle filtering algorithm developed by Gordon, Salmond, and Smith (1993), but suffers from a well-known problem that the APF, developed by Pitt and Shephard (1999), alleviates.

The main alternative to particle filtering is deterministic numerical integration (e.g., quadrature), approximating the filtering density on a discrete grid, instead of using Monte Carlo methods as in the particle filter (see, for example, Kitagawa 1987). These filters require transition densities for state variables as well as likelihood functions, neither of which is generally available in continuous-time models.[6] Bates (2006) develops a novel approach using the analytical structure of affine models, combined with functional approximations and numerical integration. The analytical approximations must be tailored for the specific application at hand.[7]

Numerical integration schemes, while feasible and accurate in univariate settings, are generally computationally infeasible and inaccurate in higher dimensions, since the computing cost is exponential in the dimension of the state. On the other hand, standard Monte Carlo methods are independent of the state dimension, and because of this, particle filters typically are more efficient for higher-dimensional problems. This is why the filtering literature focuses on particle filters. The continuous-time nature of the model and complicated shock distributions further exacerbate the dimensionality problem, as intermediate sample paths need to be simulated, and we also filter jump times and sizes, in addition to persistent state variables.

1.3.1 SIR algorithm.
The SIR algorithm assumes only that the conditional likelihood, $p(y_t|L_t)$, can be evaluated and the latent states can be simulated from

---

[6] The analytical approximation approach of Aït-Sahalia (2006) could potentially be used for numerical integration in low-dimensional filtering problems, with extensions to jumps in Yu (2007).

[7] In Bates (2006), the approximating functions are chosen to match the first two conditional moments, which works well for latent state processes such as square-root specifications, which are close to conditionally normal. It is not clear how the approach would be applied to more general specifications, such as those with jumps in the state variables.

$p(L_{t+1}|L_t)$. Both $p(y_t|L_t)$ and $p(L_{t+1}|L_t)$ are specified by the model. In the filtering literature, these are most easily seen from a state-space representation, where $p(y_t|L_t)$ specifies the distribution of the observed variables, conditional on the latent variables, and $p(L_{t+1}|L_t)$ specifies how the states evolve over time. Given current particles $\{L_t^{(i)}\}_{i=1}^N$ from $p^N(L_t|y^t)$, the algorithm is two steps:

1. For $i = 1, \ldots, N$, simulate $L_{t+1}^{(i)} \sim p(L_{t+1}|L_t^{(i)})$.
2. For $i = 1, \ldots, N$, compute

$$w_{t+1}^{(i)} = p\big(y_{t+1}|L_{t+1}^{(i)}\big) \Big/ \sum_{\ell=1}^N p\big(y_{t+1}|L_{t+1}^{(\ell)}\big),$$

draw

$$z(i) \sim Mult\big(N; w_{t+1}^{(1)}, \ldots, w_{t+1}^{(N)}\big),$$

and set $L_{t+1}^{(i)} = L_{t+1}^{z(i)}$, where $Mult(N; \cdot)$ is the multinomial distribution.

The first step, called propagation, simulates new particles from the state transition using the old particles, $L_t^{(i)}$. The second step, called resampling, samples with replacement from the discrete distribution consisting of weights, $w_{t+1}^{(i)}$, and states, $L_{t+1}^{(i)}$. There are a number of ways to draw from a discrete distribution. We use independent multinomial draws, although other methods such as stratified sampling could be used. The probabilities after resampling are $\pi_t^{(i)} = 1/N$. Gordon, Salmond, and Smith (1993) show that the resulting particles are an approximate sample from $p^N(L_t|y^t)$. In importance sampling language, $p(L_{t+1}|L_t)$ is the importance density and the weights are the ratio of the target to importance density.

The SIR is very general, simple, and computationally fast, enabling applications with large N and/or T. As with all importance samplers, its performance depends on the importance distribution. In most applications, SIR performs well since next period's states are close to this period's states. In certain cases involving rare events or outliers, however, it has a well-known shortcoming that can typically easily be corrected. This occurs because Step 1 simulates states "blindly" from $p(L_{t+1}|L_t)$, only incorporating y_{t+1} in Step 2.

The problem can be clearly seen in the simple setting of filtering jump times and sizes in a time-discretization of Merton's jump model:

$$y_{t+1} = \mu + \sigma\varepsilon_{t+1} + J_{t+1}Z_{t+1},$$

where the jump times are specified via $Prob[J_{t+1} = 1] = \lambda$ and Z_{t+1} are the jump sizes.[8] Estimates typically imply that λ is small, around 1%, and jump sizes are large and negative. For example, assume that $Z_t \sim \mathcal{N}(-2.5, 4)$,

---

[8] We thank the referee for suggesting this example.

where the units are in percentages. Here, the filtering distribution is known for $L_{t+1} = (J_{t+1}, Z_{t+1})$ but we consider estimating it with the SIR algorithm to highlight the problems.

A problem known as sample impoverishment arises when SIR is confronted with a large negative observation like the crash of 1987 (-20%). In Step 1, since

$$p(L_{t+1}|L_t) = Prob(J_{t+1} = 1) = \lambda,$$

$J_{t+1} = 1$ will be simulated for only 1% of the particles, regardless of the size of the observation. For example, with 10,000 particles, only 100 jump sizes will be simulated from $Z_{t+1} \sim \mathcal{N}(-2.5, 4)$. In Step 2, since $p(y_{t+1}|J_{t+1}, Z_{t+1})$ has exponential tails, the weight associated with the most negative jump size, $p(y_{t+1}|J_{t+1} = 1, Z_{t+1}^{\min})$, will typically be much larger than the weights associated with all of the other particles.[9] Step 2 will repeatedly sample the same Z_{t+1} value, leading to what is known as "sample impoverishment," where the sample from the particle distribution consists of many repeated values. This problem is well-known and results in an inaccurate sample from $p^N(L_{t+1}|y^{t+1})$, unless N is implausibly large.

1.3.2 APF algorithm. Pitt and Shephard's (1999) algorithm corrects the sample impoverishment problem by reordering the algorithm, resampling first and propagating second, incorporating the new observation in both steps. Given current particles, the APF consists also of two steps:

1. For $i = 1, \ldots, N$, compute

$$w_t^{(i)} = p(y_{t+1}|L_t^{(i)}) \bigg/ \sum_{\ell=1}^N p(y_{t+1}|L_t^{(\ell)}),$$

   draw

$$z(i) \sim Mult(N; w_t^{(1)}, \ldots, w_t^{(N)}),$$

   and set $L_t^{(i)} = L_t^{z(i)}$ for $i = 1, \ldots N$.
2. For $i = 1, \ldots, N$, draw

$$L_{t+1}^{(i)} \sim p(L_{t+1}|L_t^{(i)}, y_{t+1}).$$

---

[9] To see this, draw 100 samples from $Z_t \sim \mathcal{N}(-2.5, 4)$ and compute

$$w_t = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(y_t - Z_t)^2}{\sigma^2}\right)$$

for $y_t = -20$ and $\sigma = 1$. The median ratio of largest to next largest weight is about 6000, indicating the probability that the smallest jump size is resampled for nearly every particle is very high.

This version of the APF also has weights $\pi_t^{(i)} = 1/N$. The APF has two novel features. By first resampling based on the predictive likelihood, $p(y_{t+1}|L_t)$, only the important or high-likelihood particles (those that could have generated next period's observation) move forward. This is important for persistent state variables. Second, the APF incorporates y_{t+1} into the propagation of L_{t+1} in Step 2, which is crucial for the outlier/rare event problem. This version of the APF provides a direct draw (or i.i.d. sample) from $p^N(L_{t+1}|y^{t+1})$ without using importance sampling, provided that $p(y_{t+1}|L_t)$ can be evaluated and that $p(L_{t+1}|L_t, y_{t+1})$ can be sampled.

To see how the APF corrects the deficiencies of the SIR algorithm, consider again Merton's model. Since the state variables are not persistent, the first step is not needed. The second step requires a draw from $p(Z_{t+1}, J_{t+1}|y_{t+1})$, which can be expressed as

$$p(Z_{t+1}, J_{t+1}|y_{t+1}) = p(Z_{t+1}|J_{t+1} = 1, y_{t+1})p(J_{t+1}|y_{t+1}),$$

since Z_{t+1} is not defined when $J_{t+1} = 0$. The APF algorithm first simulates jump times from $p(J_{t+1}|y_{t+1})$, which is a Bernoulli distribution that accounts for y_{t+1}. For a large observation (negative or positive), $p(J_{t+1} = 1|y_{t+1})$ is close to 1, implying that $J_{t+1} = 1$ will be simulated for nearly every particle, in stark contrast to SIR. Next, for every $J_{t+1} = 1$ particle, a jump size is simulated from $p(Z_{t+1}|J_{t+1} = 1, y_{t+1})$, which is quite different from $p(Z_{t+1})$, effectively tilting the jump size simulation toward values that could have generated y_{t+1}. Thus, there are more $J_{t+1} = 1$ and jump size particles simulated that could have generated the observed y_{t+1}. The APF clearly performs better, in the sense that it will not suffer from sample impoverishment. A similar modification of this example is key in our applications below with jumps in prices.

This idealized version of the APF described above assumes that $p(y_{t+1}|L_t)$ is easy to evaluate and that $p(L_{t+1}|L_t, y_{t+1})$ can be sampled. In most applications, this will not be feasible. In this case, approximations to these distributions are used and typically eliminate the problems that arise in the SIR algorithm. Pitt and Shephard (1999) refer to these approximations as various levels of adaption, with the algorithm outlined above being full or complete adaption. When approximations are used, an additional reweighting step is required at the end of the algorithm.[10]

---

[10] In the standard case when approximations are used, the first-stage resampling is done using an approximation, $(\widehat{y}_{t+1}|L_t^{(i)})$, latent variables are drawn from $\widehat{p}(L_{t+1}|L_t^{(i)}, y_{t+1})$, and the probabilities $\pi_{t+1}^{(i)}$ are defined via

$$\pi_{t+1}^{(i)} \propto \frac{p\left(L_{t+1}^{(i)}|L_t^{(i)}\right) p\left(y_{t+1}|L_{t+1}^{(i)}\right)}{w_t^{z(i)}\widehat{p}\left(L_{t+1}^{(i)}|L_t^{(i)}, y_{t+1}\right)}.$$

1.4 Adapting particle filters to continuous-time models

Solving the SDEs in Equations (1)–(2) at the frequency associated with the observed data, the observation and state equations are given by

$$Y_{t+1} = Y_t + \int_t^{t+1} \mu^s(X_v)dv + \int_t^{t+1} \sigma^s(X_v)dW_v + \sum_{n=N_t^s+1}^{N_{t+1}^s} Z_n^s, \quad (4)$$

$$X_{t+1} = X_t + \int_t^{t+1} \mu^x(X_v)dv + \int_t^{t+1} \sigma^x(X_v)dW_v^x + \sum_{n=N_t^x+1}^{N_{t+1}^x} Z_n^x. \quad (5)$$

The challenge presented by the continuous-time specification is to efficiently approximate the conditional distributions induced by the stochastic integrals and accumulated jumps. For now, we do not consider derivative prices, reintroducing them below.

Our particle filtering algorithms use time-discretizations to simulate an additional $j = 0, \ldots, M-1$ state variables between those observations via the Euler scheme (other schemes can also be used)

$$y_{t+1} = \sum_{j=0}^{M-1} \mu_{t,j}^s + \sum_{j=0}^{M-1} \sigma_{t,j}^s \varepsilon_{t,j+1}^s + \sum_{j=0}^{M-1} Z_{t,j+1}^s J_{t,j+1}^s, \quad (6)$$

$$X_{t,j+1} = X_{t,j} + \mu_{t,j}^x + \sigma_{t,j}^x \varepsilon_{t,j+1}^x + Z_{t,j+1}^x J_{t,j+1}^x, \quad (7)$$

where $y_{t+1} = Y_{t+1} - Y_t$, $X_{t,j} = X_{t+j/M}$ are the intermediate simulated states

$$\mu_{t,j}^s = \mu^s(X_{t,j})M^{-1}, \sigma_{t,j}^s = \sigma^s(X_{t,j})M^{-1/2},$$
$$\mu_{t,j}^x = \mu^x(X_{t,j})M^{-1}, \sigma_{t,j}^x = \sigma^x(X_{t,j})M^{-1/2},$$

$\varepsilon_{t,j}^s$ and $\varepsilon_{t,j}^x$ are i.i.d. standard normal, and $J_{t,j}^s$ and $J_{t,j}^x$ are Bernoulli random variables with respective intensities $\lambda_s M^{-1}$ and $\lambda_x M^{-1}$. The jump size distribution is unaffected by time-discretization. The simulated (or augmented) states between observations are collected into a matrix of latent variables,

$$L_{t+1} = L_{t+1}^M = \left(X_{t+1}^M, Z_{t+1}^{s,M}, Z_{t+1}^{x,M}, J_{t+1}^{s,M}, J_{t+1}^{x,M}\right),$$

where, e.g., $X_{t+1}^M = (X_t, \ldots, X_{t,j}, \ldots, X_{t,M-1})$ or $J_{t+1}^{k,M} = (J_{t,1}^k, \ldots J_{t,j}^k,$ $\ldots, J_{t+1}^k)$ for $k = s, x$. Note that the states X_t are simulated up to one-discretization interval before the next observation. We define our particle filters over L_{t+1}. The distribution induced by the discretization, $p^M(y_{t+1}|L_{t+1})$, is Gaussian.

1.4.1 A generic SIR algorithm.
The following is a generic SIR algorithm, applicable to any jump-diffusion model. Given current particles,

2770

1. For $i = 1, \ldots, N$, simulate shocks from their i.i.d. distributions:[11]

$$\left(\left\{ \varepsilon_{t,j}^s, \varepsilon_{t,j}^x, J_{t,j}^s, J_{t,j}^x, Z_{t,j}^s, Z_{t,j}^x \right\}_{j=1}^M \right)^{(i)};$$

simulate states:

$$X_{t,j+1}^{(i)} = X_{t,j}^{(i)} + \left(\mu_{t,j}^x \right)^{(i)} + \left(\sigma_{t,j}^x \right)^{(i)} \varepsilon_{t,j+1}^{x,(i)} + Z_{t,j+1}^{x,(i)} J_{t,j+1}^{x,(i)};$$

and collect the new simulated prices and states into:

$$L_{t+1}^{(i)} = \left(X_{t+1}^M, Z_{t+1}^{s,M}, Z_{t+1}^{x,M}, J_{t+1}^{s,M}, J_{t+1}^{x,M} \right)^{(i)}.$$

2. Evaluate $p(y_{t+1}|L_{t+1}^{(i)})$, which is a normal density; compute

$$w_{t+1}^{(i)} = p\left(y_{t+1}|L_{t+1}^{(i)} \right) \Big/ \sum_{\ell=1}^N p\left(y_{t+1}|L_{t+1}^{(\ell)} \right);$$

draw

$$z(i) \sim Mult\left(N; w_{t+1}^{(1)}, \ldots, w_{t+1}^{(N)} \right);$$

and set $L_{t+1}^{(i)} = L_{t+1}^{z(i)}$.

This algorithm has three advantages. First, it is extremely easy to understand, modify, and code. Simulating shocks is straightforward and since $p(y_{t+1}|L_{t+1})$ is conditionally normally distributed, it is easy to calculate the weights. Second, like most SIR algorithms, this algorithm performs well in most settings. Third, incorporating option prices into the algorithm is straightforward, as the likelihood function, $p(y_{t+1}|L_{t+1})$, is now a bivariate density over $y_{t+1} = (Y_{t+1} - Y_t, C_{t+1})$. The problem with this algorithm, like the general SIR algorithm discussed above, is that it could suffer from sample impoverishment during periods with large movements driven by outliers or rare events.

1.4.2 APF modifications.   Unlike the SIR algorithm that uses a one-size-fits-all approach, the APF adapts to the structure of a given model under consideration. The first APF step requires an evaluation (or approximate evaluation) of $p(y_{t+1}|L_t)$, the predictive likelihood, which is given by

$$p\left(y_{t+1}|L_t \right) = \int p\left(y_{t+1}|L_{t+1} \right) p\left(L_{t+1}|L_t \right) dL_{t+1}.$$

This density is unknown for nearly all interesting models (outside of pure Gaussian specifications) and is costly to evaluate.[12]

---

[11] At this stage, it is clear how easy it is to incorporate correlated Brownian shocks.

[12] In some settings, analytical approximations could be used at this stage. The distribution $p(y_{t+1}|X_t)$ can be written as

$$p\left(y_{t+1}|X_t \right) = \int p\left(y_{t+1}, X_{t+1}|X_t \right) dX_{t+1}.$$

Our algorithm utilizes an Euler approximation to $p(y_{t+1}|L_{t+1})$:

$$p^M(y_{t+1}|L_{t+1}) = \mathcal{N}\left(\sum_{j=0}^{M-1}\mu_{t,j}^s + \sum_{j=0}^{M-1}Z_{t,j+1}^s J_{t,j+1}^s, \sum_{j=0}^{M-1}\left(\sigma_{t,j}^s\right)^2\right).$$

To approximate $p(y_{t+1}|L_t)$, intermediate states, jump times, and jump sizes must be integrated out. Integrating out the jump times and jump sizes is straight-forward, as jump times are Poisson (or conditionally Poisson) and jump sizes are typically assumed to be normally distributed or some other tractable distribution. In fact, with normally distributed jumps, one can exactly integrate out $\sum_{n=N_t+1}^{N_{t+1}} Z_n^s$ without discretization (we utilize this in our algorithms).

It is more difficult to deal with the persistent state variables, $X_{t,j}$, that appear in the drift and diffusion coefficients. We extend the generic APF algorithm of Pitt and Shephard (1999) that replaces the intermediate states, in this case $X_{t,j}$, by a likely value, such as the mean or mode. For example, we use mean approximations by calculating $\widehat{X}_{t,j} = E[X_{t,j}|X_t]$ (using the continuous-time specification) and defining $\widehat{\mu}_{t,j}^s = \mu^s(\widehat{X}_{t,j})M^{-1}$ and $\widehat{\sigma}_{t,j}^s = \sigma^s(\widehat{X}_{t,j})M^{-1/2}$, which implies that

$$\widehat{p}^M\left(y_{t+1}|\left\{Z_{t,j}^s, J_{t,j}^s\right\}_{j=1}^M, X_t\right) = \mathcal{N}\left(\sum_{j=0}^{M-1}\widehat{\mu}_{t,j}^s + \sum_{j=0}^{M-1}Z_{t,j+1}^s J_{t,j+1}^s, \sum_{j=0}^{M-1}\left(\widehat{\sigma}_{t,j}^s\right)^2\right).$$

The jump times and sizes can be analytically integrated out to deliver an analytical form for $\widehat{p}^M(y_{t+1}|X_t)$. This step is not exact, and an importance-sampling reweighting of particles is required at the end of the algorithm.

As an example of this step, consider a time-discretization of Bates' (1996) model:

$$y_{t+1} = \mu + \sqrt{V_t}\varepsilon_{t+1}^s + \sum_{j=N_t+1}^{N_{t+1}} Z_j^s$$

$$V_{t+1} = V_t + \kappa_v(\theta_v - V_t) + \sigma_v\sqrt{V_t}\varepsilon_{t+1}^v,$$

where $N_t \sim Poi(\lambda t)$ and $Z_j^s \sim \mathcal{N}(\mu_s, \sigma_s^2)$. In this model, due to the normally distributed jumps, there is no need to time-discretize the jump process. Using a mean approximation and integrating out the jump times and sizes,

$$\widehat{p}(y_{t+1}|L_t) = \sum_{k=0}^{\infty} \phi\left(y_{t+1}; \mu + k\mu_s, \widehat{V}_t + k\sigma_s^2\right)\frac{\lambda^k e^{-\lambda}}{k!},$$

For certain multivariate diffusion models (Aït-Sahalia 2006; Aït-Sahalia and Kimmel 2007) and even jump-diffusion models (Yu 2007), there are analytical approximations to $p(y_{t+1}, X_{t+1}|X_t)$ that are inexpensive to compute. Combined with a simple numerical integration scheme, this could provide a more accurate, but slightly more computationally intensive alternative to our scheme. We are currently analyzing these for filtering in pure diffusion settings.

where $\widehat{V}_t$ is a function of V_{t-1}. For most applications, the sum can be truncated with a low number of terms.[13] After resampling using $\widehat{p}(y_{t+1}|L_t)$, the volatility particles are tilted toward those that are likely to have generated y_{t+1}. For example, if $|y_{t+1}|$ is large, resampling results in more high V_t particles.

The second step in the APF algorithm simulates latent variables using the structure of common models. For example, models with jumps in prices typically assume that the jump size distribution is normally distributed. In this case, we can first simulate jump times marginalizing out jump sizes, then simulate jump sizes, and then simulate persistent states. Denoting the current persistent state as X_t, then

$$p\left(N_{t+1}^s - N_t^s = k | X_t, y_{t+1}\right) \propto p\left(y_{t+1}|\{\widehat{X}_{t,j}\}_{j=0}^{M-1}, N_{t+1}^s - N_t^s = k\right)$$

$$p\left(N_{t+1} - N_t = k\right) = \phi\left(y_{t+1}|\{\widehat{X}_{t,j}\}_{j=0}^{M-1}, N_{t+1}^s - N_t^s = k\right)\frac{\lambda^k e^{-\lambda}}{k!},$$

where ϕ is a normal density and $\widehat{X}_{t,j}$ is a known function of X_t, which implies that jump times can be easily sampled. Notice, as in the example of the APF in Merton's model, the new observation y_{t+1} is used when simulating jump times.

Next, it is also possible to generate a sample directly from

$$p\left(\left.\sum_{n=N_t+1}^{N_{t+1}} Z_n^s \right| N_{t+1} - N_t = k, \{\widehat{X}_{t,j}\}_{j=0}^{M-1}, y_{t+1}\right),$$

since it is just a normal distribution as we are conditioning on the number of jumps. Incorporating y_{t+1} is crucial in applications using stock index data, as there are both rare events and outliers (stock market crash of 1987), consistent with the intuition from Merton's example in the previous section.

In some models, the exact timing of the jumps matters. For example, in the models that we consider below, jumps in prices and states occur coincidentally, and therefore the dynamics of the state depends on exactly when the jump occurred. To characterize the times at which they jump, we can use a well-known property of Poisson process: conditional on a number of jumps over an interval, the times at which the jumps occurred are uniformly distributed over the interval.

Finally, the persistent state variables need to be drawn from:

$$p(X_{t+1}|X_t, y_{t+1}^*) \propto p(y_{t+1}^*|X_{t+1})p(X_{t+1}|X_t),$$

where $y_{t+1}^* = y_{t+1} - \sum_{n=N_t+1}^{N_{t+1}} Z_n^s$. This distribution is not available analytically, and approximations are required. A generic approach, used here, builds on the generic approach of Pitt and Shephard (1999) by simulating the persistent

---

[13] The algorithm at this stage can be contrasted with Chib, Nardari, and Shephard (2002), who do not integrate out the jump times and sizes and rather simulate jump times and sizes from the prior distribution, as in the SIR algorithm.

state variables from $p(X_{t+1}|X_t)$. This applies in every jump-diffusion model and works well in the models that we consider.

There are a number of alternative methods for incorporating the new observation at this stage. For example, one can combine an Euler discretization for the states, which implies that the time-discretized version of $p(X_{t+1}|X_t, y_{t+1}^*)$ is analytical, and then use a Taylor series approximation of the two densities to generate a normal updating step. We found little if any benefit from this scheme. Second, the analytical approximations of Aït-Sahalia could be used at this stage.[14]

1.4.3 Comparison to existing literature. Our approach is related to a number of different literatures, and in this subsection, we connect our work to this prior research. First, our approach complements the recent literature on "realized volatility" (see, for example, the forthcoming review by Andersen, Bollerslev, and Diebold). Initially, the focus was on high-frequency volatility estimators (Andersen et al. 2001; Aït-Sahalia, Mykland, and Zhang 2005), but more recently the emphasis has shifted to separate diffusive volatility from jumps in prices (see, for example, Barndorff-Nielson and Shephard 2004, 2006a, 2006b; Huang and Tauchen 2005; Aït-Sahalia and Jacod 2007; Andersen, Bollerslev, and Diebold 2007). Our approach is different along a number of dimensions. First, our approach is parametric while the realized volatility literature is nonparametric. In option pricing, term structure, and portfolio applications, parametric models are often used. Second, we neither require nor preclude the use of high-frequency data. Third, our approach can be used to filter any latent state variable, not just those that affect the second moment. Thus, for example, we can estimate both time-varying volatility and time-varying expected returns. Finally, our approach provides a method to coherently combine the information in different sources, from, for example, option and equity prices.

Bates (2006) develops a filter that works in affine models (either continuous or discrete time) with a single latent state variable. Bates uses deterministic, analytical approximations to filtering distributions and numerical integration, whereas our approach uses discrete particle approximations. The analytical approximations match moments, which works well when the state variables are approximately normally distributed, but implementation on more general models or over longer frequencies is unclear. Bates (2006) focuses on parameter estimation, but also analyzes filtered volatilities and jumps using simulations and index data. Our approach is more general than Bates' and easier to apply in higher dimensions, but does not utilize the special structure of affine models. Moreover, our approach applies nonlinear models and with derivative observations, which are nonlinear functions of the underlying stock prices.

---

[14] Since $p(X_{t+1}|X_t, y_{t+1}) \propto p(X_{t+1}, y_{t+1}|X_t)$ and the analytical approximations to $p(X_{t+1}, y_{t+1}|X_t)$ developed by Aït-Sahalia (2008) are generally bounded, it is possible to use rejection sampling to generates direct samples from $p(X_{t+1}|X_t, y_{t+1})$.

Finally, our approach builds on a large literature using particle filters in the context of discrete-time models in econometrics and statistics. Pitt and Shephard (1999); Chib, Nardari, and Shephard (2002); and Omori et al. (2007) use particle filters to construct likelihood functions in discrete time SV models. Pitt (2002), in concurrent work, uses particle filters for maximum-likelihood parameter estimation in diffusion-based stochastic volatility models. Durham and Gallant (2002) and Durham (2003) use an importance-sampling filtering algorithm resembling a particle filter for estimation of stochastic volatility models. Doucet, de Freitas, and Gordon (2001) provide a general overview of particle filters.

1.4.4 Convergence of particle filters.   Our approach relies on two approximations: discretizing the SDE and particle approximations. While there are convergence results for each of these components, it is not straightforward to combine the asymptotics.

Particle filtering algorithms share many properties with standard Monte Carlo estimates of expectations of smooth functions of the prices in continuous-time models. Here, there is a natural trade-off between reducing discretization bias (large M) and Monte Carlo error (large N). For smooth functions, there are standard results on the "efficient" asymptotic trade-off: using an Euler scheme, N should grow at a rate proportional to M^2 (Duffie and Glynn 1995). These results cannot be directly applied to our case because we are interested in pointwise convergence of filtering densities, not smooth functions of states.

Under mild regularity on the state transition and the likelihood, particle filters (SIR or APF) are consistent. If the true model is given by the time discretization at interval M, then the particle approximation, $p^{M,N}(L_t|y^t)$, converges to $p^M(L_t|y^t)$ in a pointwise sense as N increases (see, for example, Crisan and Doucet 2002 for a summary). This implies that the difficult step is proving pointwise convergence of the approximate transition density $p^M(L_t|L_{t-1})$ to the true transition density, $p(L_t|L_{t-1})$. For diffusions, there is a remarkable result in Bally and Talay (1996) proving the pointwise convergence of $p^M(X_t|X_{t-1})$ to $p(X_t|X_{t-1})$. Brandt and Santa-Clara (2002) use this result to prove convergence of simulated maximum-likelihood estimators (see also the appendix in Pedersen 1995). Del Moral and Jacod (2001) and Del Moral, Jacod, and Protter (2002) combine this result with standard particle convergence to show that a version of the particle filter is consistent provided that M increases slower than N (e.g., $M = \sqrt{N}$).[15] That N grows faster than M is intuitive and is similar to the results cited above.

If an analog to Bally and Talay (1996) existed for jump-diffusions, the arguments in Del Moral, Jacod, and Protter (2002) would apply to derive the limiting distribution of the particle filter in the case of jump-diffusions.

---

[15] The algorithms in Del Moral and Jacod (2001) and Del Moral, Jacod, and Protter (2002) do not perform any resampling nor can they adapt (as in Pitt and Shephard 1999), resulting in poor performance in practice.

However, to our knowledge, there is no analog to Bally and Talay (1996) for jump-diffusion processes. Proving the pointwise convergence of the densities for jump-diffusions is an open problem in stochastic analysis, although it has been conjectured that the Bally and Talay (1996) approach carries over to jump-diffusions by Hausenblas (2002). Due to this, we are careful to perform extensive simulations to guarantee that we obtain accurate estimates.

2. Empirical Results

This section provides empirical results using the Duffie, Pan, and Singleton (2000) model. Section 2.1 introduces the model, Section 2.2 provides simulation-based evidence for the SV and SVJ models, and Section 2.3 uses S&P 500 index returns for estimating/forecasting, disentangling jumps from stochastic volatility, model specification, and quantifying the informational content of option prices.

2.1 Stochastic volatility with jumps in prices and volatility
We assume that log-equity prices, Y_t, and its stochastic variance, V_t, jointly solve:

$$dY_t = (\mu - V_t/2)dt + \sqrt{V_t}dW_t^s + d\left(\sum_{j=1}^{N_t} Z_j^s\right),$$

$$dV_t = \kappa_v(\theta_v - V_t)dt + \sigma_v\sqrt{d}W_t^v + d\left(\sum_{j=1}^{N_t} Z_j^v\right),$$

where W_t^s and W_t^v are correlated (ρt) Brownian motions, μ is the diffusive equity risk premium, $Z_j^s = \mu_s + \rho_s Z_j^v + \sigma_s\varepsilon_j$, $\varepsilon_j \sim \mathcal{N}(0,1)$, $Z_j^v \sim \exp(\mu_v)$, and $N_t \sim Poi(\lambda t)$. "SV" denotes the special case with $\lambda = 0$, "SVJ" the case with $\mu_v = 0$, and "SVCJ" the general model. Throughout, we set $\rho_s = 0$, as this parameter is difficult to estimate, as argued by Eraker, Johannes, and Polson (2003) and Chernov et al. (2003). This also makes the filtering problem more difficult because there is a lower signal-to-noise ratio between jumps in volatility and returns.

This model provides analytically attractive option prices and portfolio rules, and has received significant attention as an empirical model of equity index returns and option prices (see Bates 2000; Pan 2002; Eraker, Johannes, and Polson 2003; Chernov et al. 2003; Eraker 2004; or Broadie, Chernov, and Johannes 2007). The model presents a challenge for filtering given the complicated factor structure: it contains an unobserved diffusive state (V_t), unobserved jumps in the observed state (Z_j^s), and unobserved jumps in the unobserved state (Z_j^v). Because of this complicated factor structure, the model provides a rigorous hurdle for filtering. For our simulation and real data results, we use objective measure

Table 1
Simulation results for the SV model using the parameters $\theta_v = 0.9$, $\kappa_v = 0.02$, $\sigma_v = 0.15$, **and** $\rho = 0$

	Monthly				Weekly				Daily			
M	R_1	R_2	M_1	M_2	R_1	R_2	M_1	M_2	R_1	R_2	M_1	M_2
1	2.76	7.28	2.47	5.86	0.93	5.44	0.73	4.15	0.25	4.14	0.19	3.07
2	1.08	6.84	0.95	5.27	0.30	5.40	0.20	4.05	0.20	4.15	0.14	3.06
5	0.41	6.76	0.33	5.08	0.23	5.40	0.14	4.04	0.21	4.15	0.14	3.06
10	0.22	6.76	0.18	5.04	0.20	5.40	0.13	4.04	0.19	4.14	0.13	3.06
25	0.17	6.76	0.13	5.02	0.19	5.40	0.13	4.04	0.19	4.15	0.13	3.06

R_1 (R_2) and M_1 (M_2) are RMSE and MAE errors between the filtered mean for a given M and the true filtered mean (or the true volatilities). The numbers are multiplied by 10 and throughout we use $N = 10,000$.

parameters from Eraker, Johannes, and Polson (2003), but in simulations we do study the performance for alternative parameter values.

The details of the filtering algorithms are given in the Appendix, specializing the general approach of the previous section.

2.2 Simulation-based results
2.2.1 SV model.
We simulate 100 sample paths using the Euler scheme at a frequency 1/100 of a day, and returns are then sampled at various frequencies (daily, weekly, monthly). Using these returns, the filter is run for $M = 1, 2, 5, 10, 25$, and 100. The algorithm is computationally fast; for example, running the filter for 500 observations with $N = 10,000$ and $M = 10$ took about three minutes of CPU time on a Xeon 1.8 GHz processor in Linux with code written in C. Since it is impossible to compute the true filter, we regard $M = 100$ as the true filter, in the sense that it is devoid of discretization error.

Two sets of statistics summarize the filter's performance. The first set is the root mean squared error (RMSE) and mean absolute error (MAE) comparing the approximate filtered means for a given M, $E^M[V_t|y^t]$, and the true filter, $E[V_t|y^t]$. $E[V_t|y^t]$ is the minimum variance estimator of V_t. We also report the RMSE and MAE between the filtered means and the true simulated volatilities. These results are generally of less interest in terms of the algorithm's performance (as the optimal filtering distribution is our target), but are important for practical applications.

Tables 1 and 2 summarize the performance for two parameter configurations, for daily, weekly, and monthly data, and for various values of M. Table 1 reports a "base case" parameterization, $\mu = 0.5$, $\theta_v = 0.9$, $\kappa_v = 0.02$, $\sigma_v = 0.15$, and $\rho = 0$ (the units are in daily variances). For example, $\theta_v = 0.9$ corresponds to about 15% annualized ($\sqrt{\theta_v \cdot 252}$)· ρ is constrained to zero to reduce the signal to noise ratio, making it more difficult to estimate volatility. Table 2 reports filtering results for different parameters that generate high and volatile variance paths ($\kappa_v = 0.08$, $\theta_v = 1.8$, and $\sigma_v = 0.30$).

Results indicate that for a fixed sampling interval (daily, weekly, or monthly), increasing M reduces the discretization bias, improving the accuracy of the filter. For monthly data, the RMSE and MAE are roughly halved by increasing

Table 2
Simulation results for the SV model using the parameters $\kappa_v = 0.08$, $\theta_v = 1.8$, **and** $\sigma_v = 0.30$

	Monthly				Weekly				Daily			
M	R_1	R_2	M_1	M_2	R_1	R_2	M_1	M_2	R_1	R_2	M_1	M_2
1	6.44	12.39	6.28	10.47	2.17	10.03	1.82	7.95	0.81	8.41	0.62	6.46
2	3.02	11.01	2.89	8.97	0.52	9.85	0.43	7.70	0.32	8.37	0.24	6.41
5	0.80	10.62	0.72	8.36	0.26	9.84	0.21	7.67	0.28	8.38	0.21	6.41
10	0.30	10.60	0.25	8.27	0.24	9.85	0.19	7.67	0.29	8.39	0.22	6.41
25	0.22	10.60	0.17	8.25	0.24	9.85	0.18	7.67	0.28	8.37	0.21	6.40

R_1 (R_2) and M_1 (M_2) are RMSE and MAE errors between the filtered mean for a given M and the true filtered mean (or the true volatilities). The numbers are multiplied by 10 and throughout we use $N = 10{,}000$.

M from 1 to 2, from 2 to 5, and from 5 to 10. $p(V_{t+1}|V_t)$ is noncentral χ^2 for all sampling intervals, but the degree of non-normality is generally greater for longer horizons, peaking at frequencies close to the monthly range as noted in Das and Sundaram (1999). While data augmentation always improves the performance, the effects are modest for daily sampling intervals. Here, the RMSE falls from 0.25 for $M = 1$ to 0.20 for $M = 2$, indicating that discretization bias is not a first-order effect for daily data. This is consistent with prior research showing that the effect of discretization bias on parameter estimates is swamped by sampling variation, at least for the daily sampling frequencies and common specifications (see, for example, Pritsker 1998).

The accuracy is sensitive to parameter values, in large part because the degree of non-normality in $p(V_{t+1}|V_t)$ varies with the parameters, especially κ_v and σ_v. When these parameters are small, there is little bias introduced when using the $M = 1$ Euler scheme. Table 2 considers a set of parameters with higher average volatility ($\theta_v = 1.8$, annualized), faster mean-reversion ($\kappa_v = 0.08$), and higher volatility of volatility ($\sigma_v = 0.30$), consistent with volatility processes for individual stocks (Bakshi and Cao 2004). Table 2 indicates that, as expected, there are larger gains to data augmentation and the gains are greater for less frequent sampling. The results show that even a modest amount of data augmentation ($M = 2$) with daily data drastically reduces the errors.

Figures 1 and 2 provide a graphical depiction of the filter's performance for daily (Figure 1) and monthly (Figure 2) sampling frequencies. The top panels display simulated returns; the middle panels display the true volatilities (dots) and posterior means for $M = 1, 2, 5$, and 25. The bottom panel displays the difference between the true filtered posterior mean ($M = 100$) and the filtered posterior means for each M. It is clear that errors at both frequencies and for all degrees of data augmentation are greater when volatility is higher. Filtered estimates underestimate volatility when volatility is high and overestimate volatility when it is low. This is clearly the pattern in Figures 1 and 2 and occurs because filtered estimates are smoothed expectations based on past returns and volatility is mean-reverting.

Economically, there is no difference across the filtered volatility for daily sampling and various degrees of data augmentation. In fact, it is not possible to
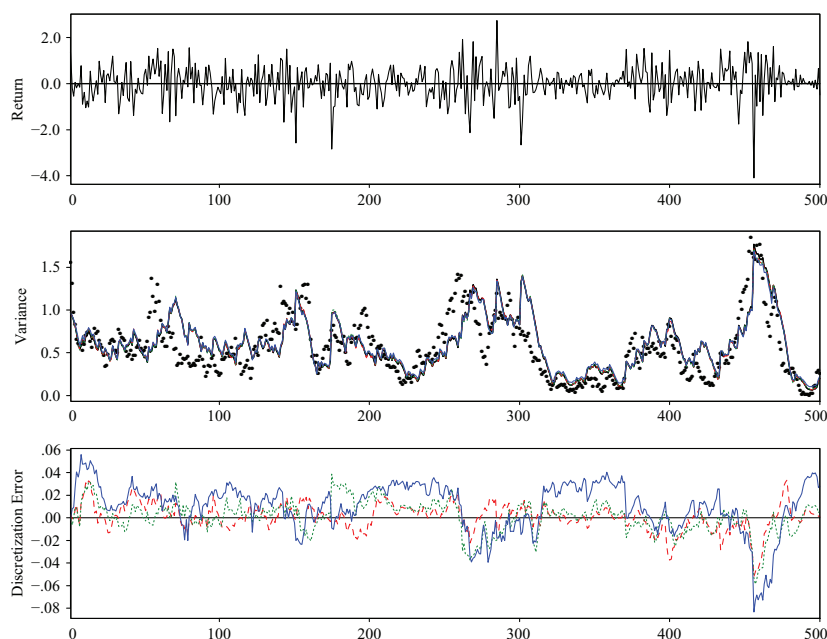
Figure 1
A representative simulated sample path of daily returns (top panel), posterior means of the filtering distribution of volatility for $M = 1, 2,$ and 5 (the true volatilities are the dots), and discretization error for the difference between the posterior means and true mean for $M = 1, 2,$ and 5.

differentiate the estimates in the middle panel as the largest differences are on the order of 0.05 when V_t is over 1.5. With monthly sampling, data augmentation is economically important as large errors for $M = 1$ are drastically reduced via data augmentation. For example, around data points 375, the true variance is about four (30% annualized volatility) and the filtered estimate for $M = 1$ is significantly less than the filtered estimates for $M > 1$. This shows how important data augmentation is with infrequently sampled data.

Figure 3 displays a smoothed version of $p^M(V_t|y^t)$ for daily return sampling and various values of M. $p^1(V_t|y^t)$ is close to normal as it utilizes a normal approximation to $p(V_{t+1}|V_t)$. For higher values of M, $p^M(V_t|y^t)$ is increasingly non-normal and positively skewed for higher values of M, which occurs because $p(V_{t+1}|V_t)$ is non-normal and skewed and any approximation errors are small. Although this figure shows an average volatility day, the effect is stronger during extremely high- or low-volatility periods, when mean-reversion is stronger. There is a similar effect on the filtered distribution of returns as the kurtosis increases with data augmentation as the approximations to the conditional distributions are more accurate.

2.2.2 Incorporating jumps: the SVJ model. This section considers adding price jumps to the SV model. "Volatility" is now generated by two components:
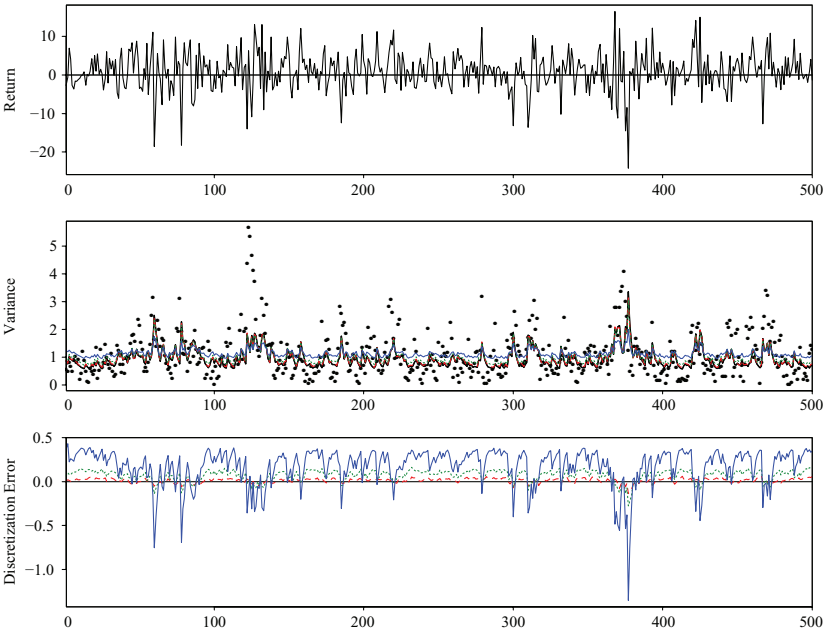
Figure 2
A representative simulated sample path of monthly returns (top panel), posterior means of the filtered volatility distribution for $M = 1, 2$, and 5 (the true volatilities are the dots), and discretization error for the difference between the posterior means and the true mean for $M = 1, 2$, and 5.
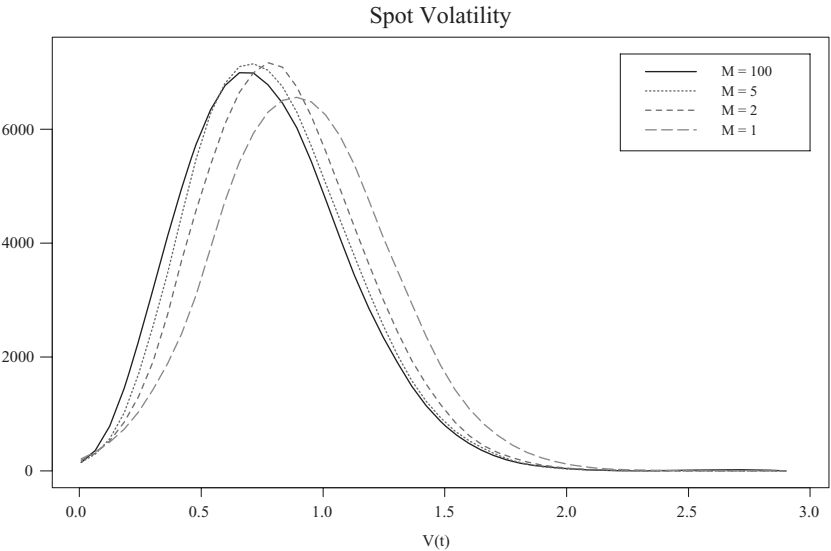

Figure 3
Filtering densities for the stochastic volatility model as a function of the degree of data augmentation (M). The densities are defined over spot variance, V_t, and are smoothed using kernels.

Table 3
Simulation results for the SVJ model

M	Weekly				Daily			
	R_1	R_2	M_1	M_2	R_1	R_2	M_1	M_2
				Variances				
1	0.20	3.65	0.16	2.82	0.20	2.81	0.15	2.14
2	0.16	3.64	0.12	2.82	0.20	2.81	0.14	2.14
5	0.15	3.64	0.12	2.82	0.20	2.81	0.14	2.14
				Jump sizes				
1	0.16	5.88	0.05	1.32	0.05	1.20	0.01	0.11
2	0.14	5.87	0.05	1.31	0.04	1.20	0.01	0.11
5	0.13	5.87	0.05	1.31	0.04	1.20	0.01	0.11
				Jump times				
	Hit_1		Hit_2		Hit_1		Hit_2	
1	98.6		29.5		98.6		60.2	
2	99.1		29.5		98.6		60.2	
5	99.1		29.5		100		61.1	

The summary statistics for variance estimation are the same as in the previous tables. Hit_1 reports the percentage of jumps (as identified by the true filter) that are identified by the approximate filter. Hit_2 identifies the percentage of actual jumps that are identified by the approximate filter. The numbers are multiplied by 10 and throughout we use $N = 10,000$.

a slow-moving stochastic volatility component and the infrequent, but large, jumps, complicating filtering because large returns could be generated by either high volatility or jumps. The SVJ model parameters are again based on Eraker, Johannes, and Polson (2003): $\mu = 0.05$, $\theta_v = 0.82$, $\kappa_v = 0.02$, $\sigma_v = 0.10$, $\mu_s = -2.5$, $\sigma_s = 4.0$, and $\lambda_s = 0.006$. Here, volatility is "smoother," as σ_v is significantly lower than in the SV model.

Table 3 summarizes the results for daily and weekly frequencies (jump estimation results were so poor at the monthly frequency, they are not reported). For the variances and jump sizes, the first two panels provide the RMSE and MSE distances between the filtered posterior means for a given M and either the true filter or the true state, as in the previous section. The differences between $M = 5$ and $M = 25$ are quite small, so we do not report the results for $M = 25$. For volatility estimation, the results are qualitatively similar to those in the previous section.

The results indicate that there is little benefit to data augmentation at either the daily or weekly frequency. For volatility, this is not surprising, because the parameters used in the SVJ model imply that volatility moves more slowly than in the SV model, reducing the benefits of data-augmentation. Similarly, there is little increase in the precision of the jump size estimate vis-à-vis the true jump sizes as augmentation increases. Since jumps are rare (about $0.006 \cdot 252 = 1.6$ per year), it is unlikely that more than one jump would occur over a daily or weekly frequency, leading to small gains from data augmentation. Jump sizes are more precisely estimated using daily versus weekly data, which is also not surprising.

We use two statistics to quantify the types of errors that could be made estimating jump times. We assume a jump occurred if the filtered jump probability is greater than 50%. The first statistic, Hit_1, quantifies how data augmentation affects jump time estimates and is defined by the percentage of jump times identified by the approximate filter that are also identified by the true ($M = 100$) filter. For example, $Hit_1 = 90$ for $M = 2$ indicates that of the total jump times identified by the true filter, 90% were also identified by the $M = 2$ approximate filter, quantifying discretization bias. The second statistic, Hit_2, gives the percentage of true simulated jumps that were also identified by the approximate filter.

It is very difficult to identify jump times at a weekly frequency. A comparison of the distance measures to the truth for daily and weekly frequencies shows that RMSE (MSE) for weekly data is about five (12) times as large. Hit_2 indicates that with daily data, about 60% of the jump times are correctly identified. This should not be surprising. Since $Z_j \sim \mathcal{N}(2.5, 4^2)$, just over 40% of the jump sizes are between $\pm 2.5\%$. Since daily diffusive volatility is about 0.9% ($\sqrt{0.8}$), more than 40% of the jumps are not "significant" as they are less than a three-standard-deviation diffusive shock. In addition, volatility changes over time, and thus it is not surprising that even with daily data, only about 60% of the jumps are identified.[16]

The ability to identify jumps falls for longer sampling intervals. For weekly data, only 30% of the true jump times are correctly identified. Via the time-aggregation, total diffusive return volatility over a week is roughly $\sqrt{5} \approx 2.25$ times larger than daily volatility. If daily volatility is about 1%, a 3% move is a statistical outlier. For weekly data, an outlier is a move greater than 7%, which is more likely to be generated by high volatility.

Thus, even if parameters are known, it is very difficult to identify jumps using weekly data. Although most papers analyzing models with jumps use daily data (even many of the earliest ones, for example, Beckers 1981 or Jarrow and Rosenfeld 1984), there are papers that use less frequently sampled data. For example, Andersen, Benzoni, and Lund (2004) use weekly data to estimate a term structure model with jumps while Das and Uppal (2004) use monthly data to estimate a jump-diffusion model on equity returns. The results here indicate that it is very difficult to identify jump components at these frequencies.

Figure 4 provides a graphical view of the filtering results for a randomly selected simulation of 2000 days for the SVJ model with filtered estimates using $M = 10$ and $N = 10,000$. The top panel displays the simulated returns; the second panel displays the true volatility path (dots), the (2.5, 50, 97.5) quantiles of $p^M(V_t|y^t)$; the third panel displays the true jump times (dots) and estimated jump probabilities; and the bottom panel displays the true jump sizes

---

[16] This raises a subtle specification issue. The normally distributed jump specification implies that many of the jumps are going to be too small to identify even with daily data. This implies that either higher-frequency data is needed, or alternatively, a model specification that constrains all jump movements to be large.
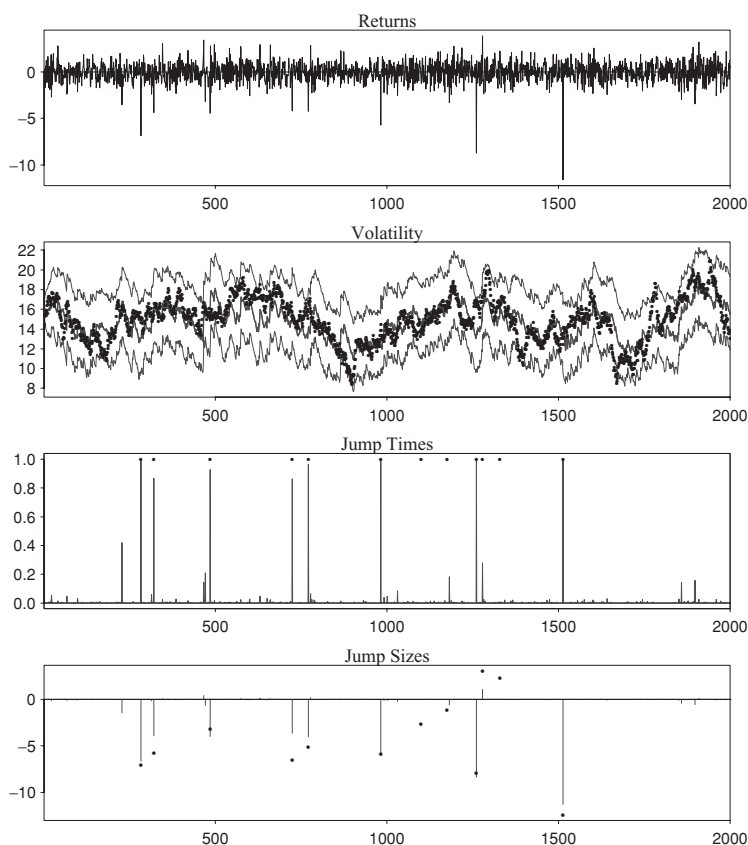
Figure 4
Performance of the filtering algorithm using simulated daily data for the SVJ model. The top panel displays returns, the second panel displays the simulated volatilities (dots) and the (10, 50, 90) quantiles of the filtering distribution, the third panel displays filtered jump probabilities (true jumps displayed as dots), and the fourth panel displays the filtered jump size means (true jump sizes are dots). Prices were simulated from an Euler scheme with 100 time steps per day and the algorithm was run for $M = 10$ and $N = 10,000$.

(dots) and the filtered jump size means. The confidence bands on the volatility state are typically about 2% on each side of the posterior mean. This estimation risk has always been a major concern, given the important role that volatility estimates play in finance applications.[17]

Our algorithm identifies large jumps, but misses many of the smaller jumps that are less than about 2% in absolute value. The missed jumps are too small to be identified as the algorithm cannot separate them from normal, day-to-day movements. These results are important for practical applications, as the results

---

[17] In fact, Merton (1980) argues that the "most important direction is to develop accurate variance estimation models which take into account the errors in variance estimates" (p. 355).

show that the algorithm is effective in identifying important (i.e., reasonably large sized jumps) movements.

We have also performed extensive simulations using the general SVCJ model. The results are similar to those for the SVJ model and are not reported. At daily frequencies, it is possible to identify both price and volatility jumps as their occurrence is coincidental. In effect, price jumps signal that volatility has increased. If jump occurrence is not coincidental, as is the case in one of the double-jump models in Duffie, Pan, and Singleton (2000), then it is possible to estimate price jumps, but estimates of volatility jumps are not reliable as the signal-to-noise ratio is too small.

2.3 Applications with S&P 500 returns

We now consider filtering jumps and stochastic volatility from index returns and index option prices for the three models. We consider four separate uses of the optimal filter: disentangling jumps from stochastic volatility, forecasting volatility, sequential model testing, and analyzing the informational content of returns regarding volatility vis-à-vis option prices.

2.3.1 Disentangling stochastic volatility and jumps and forecasting volatility. We next turn to the problem of disentangling stochastic volatility from jumps and forecasting volatility. To do this, we again use the parameters from Eraker, Johannes, and Polson (2003) and consider filtering over the time period from 1980 to 2003. Unlike the results in the previous sections, we incorporate the leverage effect ($\rho = -0.40$ for SV, -0.47 for SVJ, and -0.48 for SVCJ). We use $N = 50,000$ and $M = 10$.

Figure 5 displays filtered volatility estimates ($\sqrt{V_t}$) for the SV, SVJ, and SVCJ models. Average filtered volatility is highest for the SV model, as all the return volatility is attributed to V_t, in contrast to the SVJ and SVCJ, that attribute about 10–15% of total return volatility to price jumps. The dynamics of volatility differ substantively, at least at certain times, across models. In periods with no jumps, volatility estimates are similar, but estimates diverge rather dramatically during periods of market stress, such as 1987, 1997, and 1998 and because each model responds differently to shocks, as the long-run volatility mean and κ_v are different across models.

To see the differences along alternative dimensions, Figure 6 displays the predictive volatility distributions, $p(\sqrt{V_{t+1}}|y^t)$, and predictive return distributions, $p(y_{t+1}|y^t)$, for a day just prior to the crash (Wednesday, October 14, 1987), just after the crash of 1987 (Wednesday, October 21, 1987), and two weeks after the crash (Wednesday, November 2, 1987). Prior to the crash, the predictive densities were already different, but in an interesting way. The SV density is shifted to the right of the other two. This occurs because of relatively large moves in the months prior to the crash that were in part explained by jump components, resulting in a less elevated predictive volatility for the SVJ and SVCJ models. The very heavy right tail for the SVCJ model
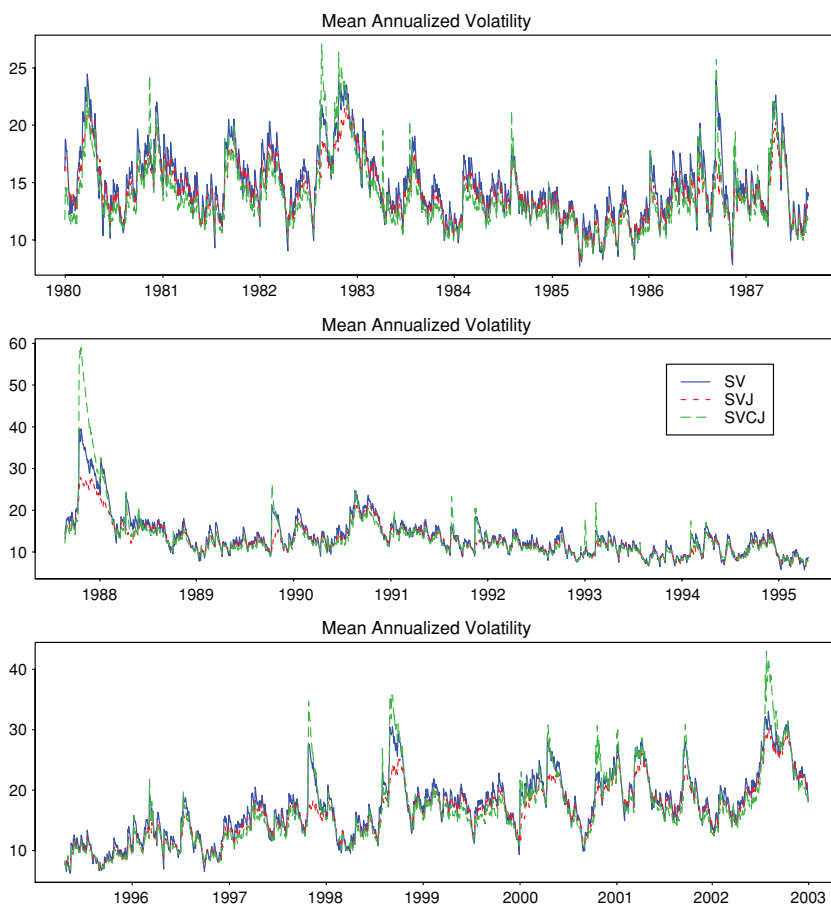
Figure 5
Filtered volatility states. This figure provides filtered volatilities for the SV model (blue, solid line), the SVCJ model (green, lightly shaded line), and the SVJ model (red, shaded line) over various time periods using daily S&P 500 returns.

generated by potential jumps in variance is also clear. Despite the high volatility, the predictive return distributions were similar, although the SVJ and SVCJ model return distributions are noticeably more peaked, indicative of additional kurtosis relative to the SV model. It is difficult to see the tail differences.

The predictive distributions for October 21 are markedly different. SV and SVJ model predictive volatility is much lower, as their Brownian shocks do not allow volatility to increase rapidly. The SVJ model's filtered volatility is lower than the SV model, because it attributes much of the crash to a price jump. To be precise, $E[\sum_{n=N_t+1}^{N_{t+1}} Z_n^s | y^{t+1}]$ is -20% for the SVJ model and -12% for the SVCJ model. Thus, the filtering results indicate that almost half of the crash's movement is due to elevated volatility. The predictive return distributions have
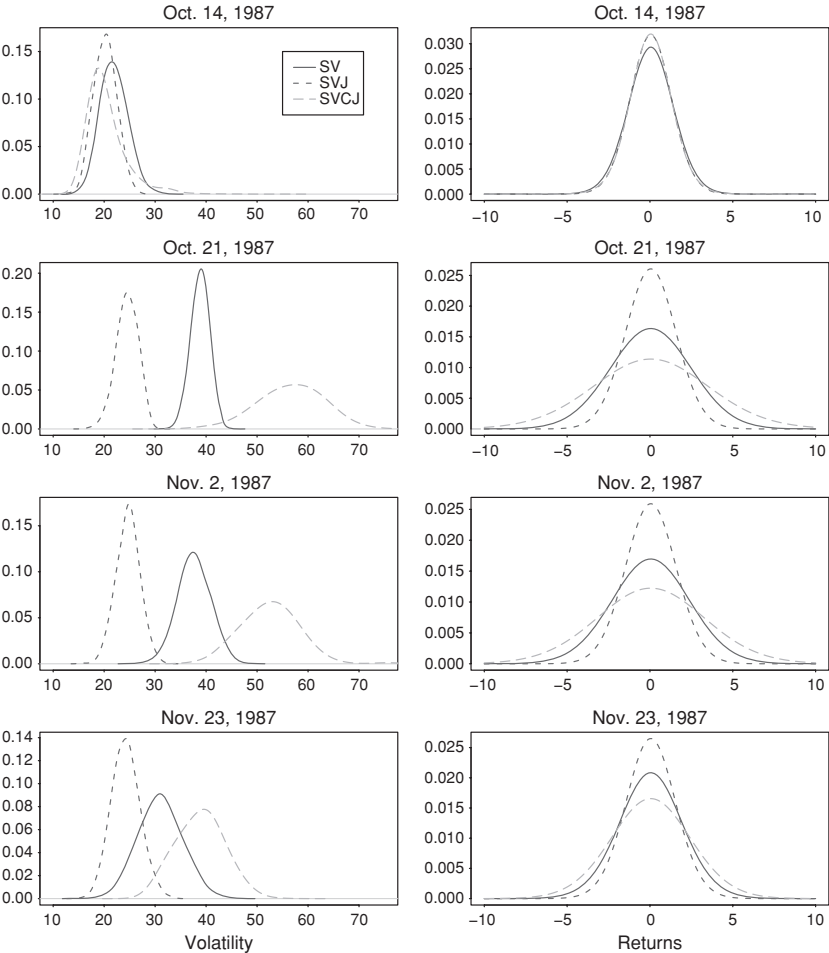
Figure 6
This figure provides predictive volatility and return distributions around the crash of 1987. The left-hand panels provide predictive volatility densities $p(\sqrt{V_{t+1}} \mid y^t)$ and the right-hand panels provide predictive return distributions $p(y_{t+1} \mid y^t)$ for each model, for three days around the crash (October 19, 1987).

a very different tail behavior, with SVCJ having much more tail mass than the SV and SVJ due to high volatility. Thus, the SVCJ model is more able to capture the +8% move that was realized on October 21. The differences linger for a considerable amount of time, as the bottom panels indicate. More than two weeks after the crash, SVCJ volatility is much higher than that for SV and SVJ. Even though SVCJ has a faster speed of mean-reversion (κ_v) than SV or SVJ, volatility has not returned to average values very quickly.

Although predictive volatility and return estimates across models are different, there is no formal indication as to which estimates are more accurate. To

Table 4
Variance forecasting results

	Daily		Weekly		Monthly	
	Bias	RMSE	Bias	RMSE	Bias	RMSE
SV	0.12	7.72	0.56	20.34	1.84	52.54
SVJ	0.09	7.74	0.42	20.48	1.52	51.75
SVCJ	0.09	7.69	0.38	20.28	1.26	51.67

Forecasting results over three time horizons for each model.

address this issue, we compute forecasts of total variance over various horizons (daily, weekly, and monthly), and compare the estimates to those subsequently realized. In each model, expected future variance is given by

$$
E_t \left[\int_t^{t+\tau} V_s ds \right] + var_t \left[\sum_{n=N_t^s+1}^{N_{t+\tau}^s} Z_n^s \right],
$$

which is known analytically. Realized τ-day variance is $\sum_{j=0}^{\tau-1} \log(S_{t+j+1}/S_{t+j})^2$.

Table 4 reports the bias and RMSE for daily, weekly, and monthly total variance forecasts for each model. To our knowledge, these are the first formal forecasting exercises done for continuous-time stochastic volatility models. To get a sense of the scale of the RMSE's, consider a movement like the crash of 1987, which was -22.5 (returns are multiplied by 100). If forecasted daily volatility was 3, then the absolute variance forecast error is $|22.5^2 - 3^2|$. Results indicate that the SVCJ model performs better for forecasting, as it has the lowest bias and RMSE for each time horizon. However, the differences are rather small, and not likely statistically or economically significant. Daily variance is about 1, so the differences in bias at a daily frequency are certainly insignificant, although the bias differences are a bit more substantial at the monthly frequency. The differences in RMSE are similarly quite modest, not likely economically or statistically significant.

2.3.2 Sequential likelihood ratios and model choice. We next consider the problem of sequentially monitoring model fit via likelihood ratios. Figure 7 displays index returns and sequential likelihood ratios, $\mathcal{LR}_{SV,SVCJ}(t)$ and $\mathcal{LR}_{SVJ,SVCJ}(t)$, over the full sample, and also during the subsample of the months surrounding the crash of 1987. We do not include $\mathcal{LR}_{SV,SVJ}(t)$, since it is the ratio of the two likelihood ratios that are reported.

Full-sample log-likelihood ratios indicate that the SVCJ model generates a much higher likelihood than the SV or SVJ models. More interesting are the dynamics. Focusing on $\mathcal{LR}_{SV,SVCJ}(t)$ first, it is interesting to note that for a good portion of the early 1980s, the SV model provided a better fit than the SVCJ model. This occurs because κ_v is smaller in the SV model, allowing volatility to remain high or low for longer periods, and there were few large
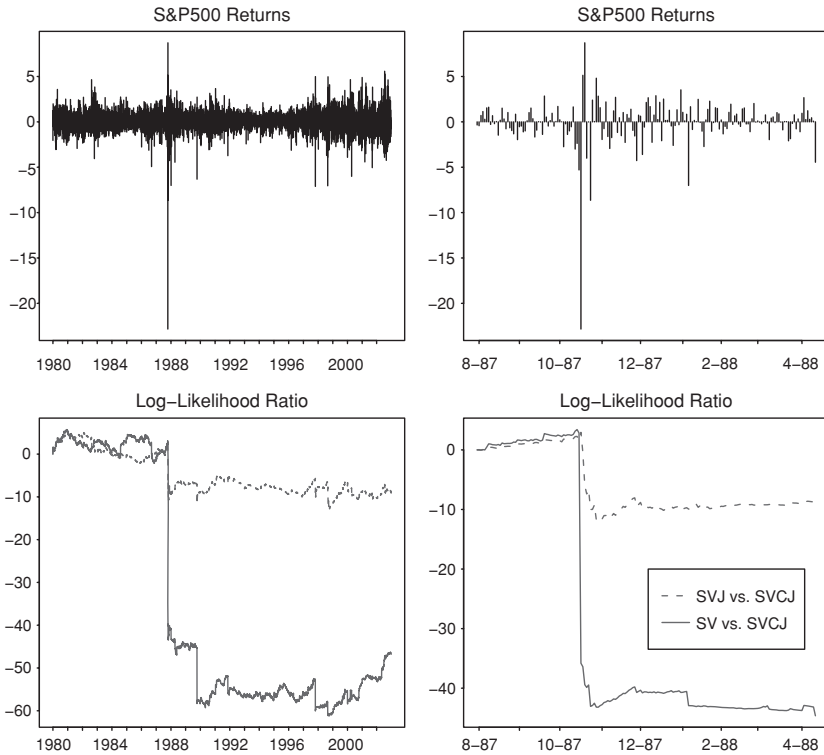
Figure 7
Sequential likelihood ratios. This figure provides returns and sequential likelihood ratios for the full sample (left-hand panels) and for the months around the crash of 1987 (right-hand panels).

movements during this period (the largest absolute returns over this period were about 4%). The period around the crash of 1987 quickly differentiates the models, as indicated by the right panels of Figure 7. The difference between SV and SVCJ occurs mostly during the crash of 1987, as the SV model cannot generate a large movement when predicted volatility was only 2% or 3%. The SV model also had difficulties with a number of other large movements in 1989, 1991, 1997, and 1998. Since 1999, it is interesting to note that the SV model has significantly outperformed the SVCJ model, which again occurs due to the slower speed of mean-reversion in the SV model.

The SVJ and SVCJ model fits were roughly similar prior to the crash. There is a modest difference on the crash data, but the difference was clearer in the days after as the SVJ model has a very difficult time handling these large movements. The fit of the SVCJ model was modestly better throughout most of the 1990s, with larger improvements occurring during Fall 1997 and 1998. Overall, there is little difference between the SVJ and SVCJ models over the last 15 years.

What can we learn from these results? First, model fits are mainly driven by large returns and periods of market stress, showing the importance of accurate models of these periods. In particular, it would be interesting to assess how different jump size distributions (e.g., non-normal) or time-varying jump intensities (e.g., Bates 2000) would impact the likelihood ratios. In terms of non-jump periods, it seems reasonable that time-variation in the long-run mean or speed of mean reversion of V_t would also impact the results. The improved fit of the SV model compared to the SVCJ model since 1999 suggests that relatively small differences in volatility parameters can generate substantial improvements in fits, at least over certain time periods.

2.3.3 The informational content of returns and option prices.   Most approaches for estimating volatility and jumps rely exclusively on returns, ignoring the information embedded in option prices. In principle, options are a rich source of information regarding volatility, which explains the common use of Black-Scholes implied volatility as a volatility proxy for practitioners. In contrast to this is the common academic approach of only using returns to estimate volatility.

In this section, we consider volatility estimates using returns and, additionally, option prices. Our goal is to quantify the informativeness of options regarding volatility and also to examine whether the informational content of the two sources is consistent, as a good model would indicate. For a given model, option prices are given by

$$C(S_t, V_t, \theta^{\mathbb{P}}, \theta^{\mathbb{Q}}) = E_t^{\mathbb{Q}}\left[\exp\left(-\int_t^T r_s ds\right)\max(S_T - K, 0)\right],$$

where the expectation is taken under the risk-neutral dynamics

$$dY_t = (r_t - V_t/2)dt + \sqrt{V_t}dW_t^s(\mathbb{Q}) + d\left(\sum_{j=1}^{N_t^{\mathbb{Q}}} Z_j^s(\mathbb{Q})\right)$$

$$- \lambda^{\mathbb{Q}}\exp\left(\mu_z^{\mathbb{Q}} + 0.5(\sigma_z^{\mathbb{Q}})^2\right)dt,$$

$$dV_t = \kappa_v^{\mathbb{Q}}(\theta_v^{\mathbb{Q}} - V_t)dt + \sigma_v\sqrt{V_t}dW_t^v(\mathbb{Q}) + d\left(\sum_{j=1}^{N_t^{\mathbb{Q}}} Z_j^v(\mathbb{Q})\right),$$

where $N_t^{\mathbb{Q}} \sim Poi(\lambda^{\mathbb{Q}}t)$, $Z_j^s(\mathbb{Q}) \sim \mathcal{N}(\mu_z^{\mathbb{Q}}, (\sigma_z^{\mathbb{Q}})^2)$, and $Z_j^v(\mathbb{Q}) \sim \exp(\mu_v^{\mathbb{Q}})$.

Options are highly informative about volatility, but the information content is dependent on the model specification and $\mathbb{P}$- and $\mathbb{Q}$-measure parameter estimates. Misspecification or poorly estimated risk-premiums could result in directionally biased estimates of V_t using option prices. This informativeness of index options is mitigated by the fact that these options contain relatively large bid-ask spreads, as noted by George and Longstaff (1993) or Bakshi, Cao,

and Chen (1997). This implies that while option prices are informative, they may be quite noisy as a measure of volatility.

To capture these error sources (misspecification, parameter estimates, and bid-ask spreads), we assume that option prices are observed with errors,

$$\log(C_t) = \log(C(S_t, V_t, \theta^{\mathbb{P}}, \theta^{\mathbb{Q}})) + \varepsilon_t,$$

and $\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is the pricing error. We assume $\sigma_\varepsilon = 0.10$, consistent with observed at-the-money index option prices. In general, the higher the bid-ask spread, the less informative option prices are about volatility, which implies our pricing error is conservative and will not bias our findings. Broadie, Chernov, and Johannes (2007) provide estimates of diffusive and jump risk premiums that we use for this section.

We focus on two issues: informativeness and consistency. The first is the general informativeness of the two data sources regarding volatility. We filter volatilities using only returns and then using returns and options jointly and compare the average standard deviations of $p(V_t|y^t)$ and $p(V_t|y^t, C^t)$. We expect the posterior standard deviations to be smaller incorporating options. Second, we compare the filtering distributions with and without options to analyze the extent to which the information in returns is consistent with the information in option prices. The existing literature typically compares Black-Scholes implied volatility to future realized volatility in order to understand the predictive content of implied volatility. This literature is firmly nonparametric in the sense that specific models or risk-premium estimates are not considered. Here, we are interested in understanding the informational content of the two sources in the context of a model that has been shown to fit both returns and options well.

We use S&P 500 index returns and options on the S&P 500 index futures contract from 1987 to 2003. We use an at-the-money call option contract with maturity greater than one week, but less than five weeks. The options are American and thus it is important to account for the possibility of early exercise. We use the American adjustment feature in Broadie, Chernov, and Johannes (2007).

The results for each model are in Table 4, with a graphical summary for the SVCJ model in Figures 8 and 9. Regarding informativeness, Table 5 indicates that the posterior standard deviation falls drastically in each model when options are added. The decrease is greatest when incorporating risk-premiums, as the posterior standard deviation is roughly 40% lower in the SVJ and SVCJ models. This decrease in estimation uncertainty occurs even though we assumed option pricing errors, justifying the common perception that options are highly informative about volatility. Moreover, if parameters were estimated, options might be even more informative about volatility, as options aid in estimating risk-neutral and structural parameters.
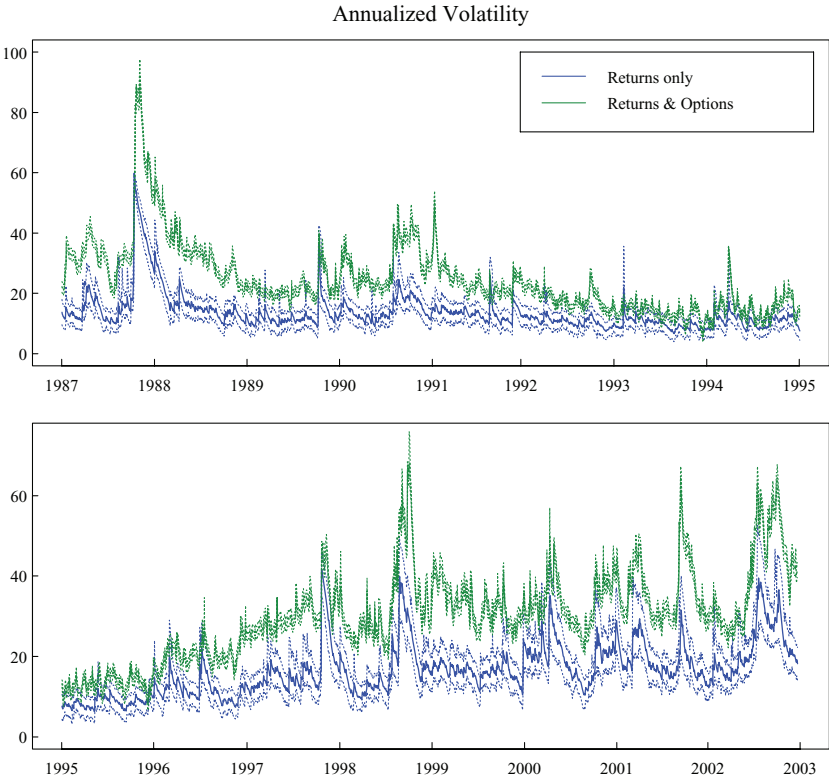
Annualized Volatility

Figure 8
Filtered volatility using only returns data and both returns and options data. For each we include the posterior means and a (5, 95) confidence band. The darker lines are using only returns (generally the higher of the two sets of lines) and the lighter lines are using both returns and options. We assume no risk-premiums.

Table 5
Filtering results using returns and returns and option prices for the SVJ and SVCJ models

Model	Risk-premiums	SD $(V_t \mid y^t)$	SD $(V_t \mid y^t, C^t)$	Prob $(\mathbb{P}95 > \mathbb{Q}05)$
SV	No	0.354	0.235	0.061
SVJ	No	0.298	0.228	0.144
	Yes	0.298	0.175	0.560
SVCJ	No	0.384	0.256	0.117
	Yes	0.384	0.233	0.274

Regarding consistency, the results are less encouraging. First, assuming no risk-premiums, we see from Table 5 that the upper 95% tail of the filtering distribution using only returns is typically lower than the lower 5% tail of the filtering distribution using options and returns. This is most easily seen in Figure 8 where the green lines (which represent the 5th, 50th, and 95th percentiles of the filtering distribution using options and returns) are generally
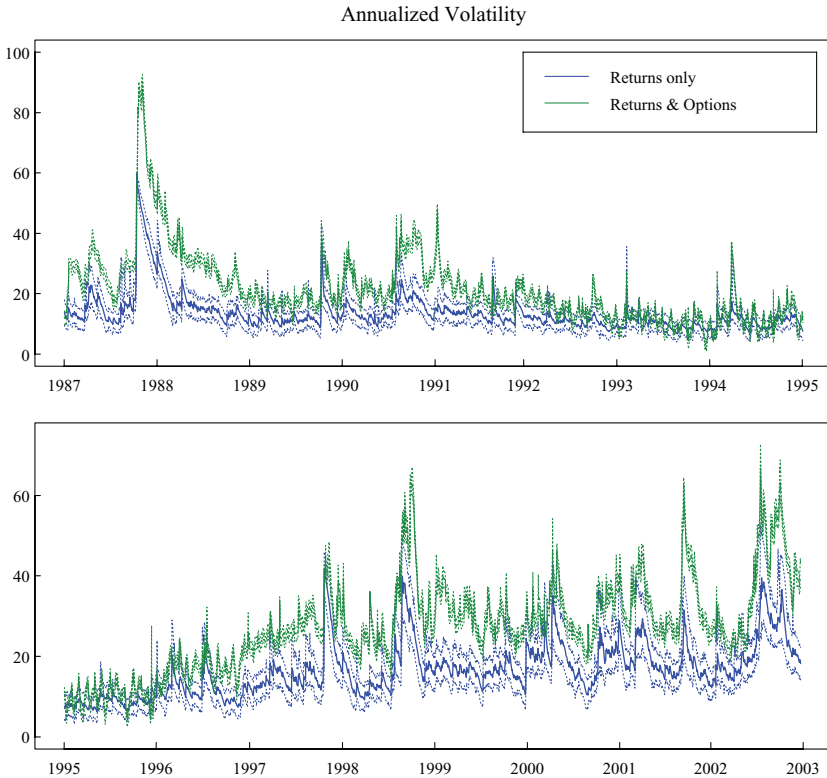
Annualized Volatility

Figure 9
Filtered volatility using only returns data and both returns and options data. For each we include the posterior means and a (5, 95) confidence band. The darker lines are using only returns (generally the higher of the two sets of lines) and the lighter lines are using both returns and options. We assume the risk-premiums estimated in Broadie, Chernov, and Johannes (2005).

higher than those obtained only from returns. While the two series tend to move together, the levels are somewhat different.

A few examples provide the necessary intuition. In early 1990, after Iraq's invasion of Kuwait, estimates of V_t including options were higher than those based on returns. Clearly, the information embedded in options anticipated larger moves than were subsequently observed. Next, note that the levels of volatility from the two sources were broadly consistent in the low-volatility period from 1993 to 1996. From 1997 onward, the estimates based on options are substantially higher than those based only on returns.

Figure 9 incorporates jump risk premiums based on the estimates in Broadie, Chernov, and Johannes (2007). The jump risk premiums reduce μ_s (from about -4% to -6%) and increase σ_s (from about 3% to 7%) and μ_v (from 1.5 to 7), and have the net result of increasing risk-neutral expected variance, thereby decreasing option-based estimates of V_t. The two data sources are now

generally consistent from 1993 to 1996; however, the option-based information still implies a higher volatility for the pre-1993 period and the post-1997 period, indicating that the two sources are not generally consistent.

The main explanations for the differences are model misspecification, under either $\mathbb{P}$- or $\mathbb{Q}$-measure. Note that the actual volatility (whether under $\mathbb{P}$ (returns) or $\mathbb{Q}$ (options)) is generally high for long periods of time (1987–1992 and 1997–2003) and low for long periods of time (1992–1996). In the model specified above, the long-run mean of V_t is constant (θ_v in the SV and SVJ models, and $\theta_v + \mu_v\lambda/\kappa_v$ in the SVCJ models). It is unlikely that V_t would remain above or below its long-run mean for an extended period of time. Duffie, Pan, and Singleton (2000) introduced a model with a time-varying long-run diffusive variance mean, which could capture this feature. We find this to be a plausible explanation. Second, time-varying risk-neutral parameters could explain the result and be driven by time-varying risk aversion or business cycle risk. The periods with the largest differences (1991 and the late 1990s/early 2000s) were also periods with substantial macroeconomic uncertainty (e.g., recession). It would be interesting to analyze a model that generates substantial time-varying jump risk premiums, to see if these models could account for these observations. Finally, as mentioned above, joint estimation of the parameters using both options and returns data would likely result in more accurate parameter estimates and could alleviate some of the differences, although it is not likely that all of them would be eliminated.

3. Conclusions

This paper develops particle filtering algorithms for filtering and sequential parameter learning. The methods developed apply generically in multivariate jump-diffusion models. The algorithm performs well in simulations, and we also apply the methodology to filter volatility, jumps in returns, and jumps in volatility from S&P 500 index and index option returns.

We find that while volatility forecasts are often quite different across models, the different forecasts are roughly of the same accuracy, although the more general models do offer some improvement. In terms of model diagnostics, we compute likelihood ratios and find that they are highly informative about model specification, even though volatility forecasts do not provide a firm ranking of models. We also analyze the information embedded in options and returns, quantifying the informational content of options vis-à-vis returns and analyzing the consistency of options and returns data. We find that if there is a priced jump risk, returns and options often are consistent, but they are also often not consistent. We discuss potential explanations for this inconsistency.

In future work, we are working on four methodological and empirical extensions. First, we are working on incorporating Aït-Sahalia's (2008) analytical approximations for resampling and state propagation in the APF algorithm for pure diffusion models. Second, we are working on sequentially learning

about state variables and parameters in continuous-time. Johannes, Polson, and Stroud (2007) find that particle filtering methods work well in certain discrete-time models. Third, another interesting extension is to apply particle filtering to continuous-time models with small jumps or non-normal jump distributions, such as those driven by Levy processes (see, for example, Barndorff-Nielson and Shephard 2006b). Finally, an interesting extension would be to compare the accuracy of volatility and jump estimates obtained from our filtering methodology to those obtained from high-frequency data.

Appendix: Detailed Algorithms

This appendix provides the full algorithms that we used to analyze simulated and S&P 500 index return data for the SV and SVJ models. Here, $y_{t+1} = Y_{t+1} - Y_t$ denotes the return, $\overline{V}_{t+1}$ denotes the integrated volatility over the interval $[t, t+1)$, $\phi(\cdot|m, v)$ denotes the normal density with mean m and variance v, $Poi(\cdot|\lambda)$ denotes the Poisson mass function with rate λ, and $Mult(N; \pi_1, \ldots, \pi_N)$ denotes the multinomial distribution with N outcomes with probabilities $\pi_1, \ldots, \pi_N$.

A.1 SV model

1. *First-stage resampling.* Given the initial weighted particle set, $\{V_t^{(i)}, \pi_t^{(i)}\}_{i=1}^N$, compute the expected integrated volatility $\widehat{V}_{t+1}^{(i)} = E[\overline{V}_{t+1}|V_t^{(i)}]$, and compute the first-stage weights for each particle $i = 1, \ldots, N$:

$$w_{t+1}^{(i)} \propto \phi(y_{t+1} \,|\, \mu_y, \widehat{V}_{t+1}^{(i)}).$$

   Resample the initial particles by drawing the indices $z(i) \sim Mult(N; w_{t+1}^{(1)}, \ldots, w_{t+1}^{(N)})$ and defining the resampled volatilities $V_t^{(i)} = V_t^{z(i)}$. At this stage, we note that (a) we need only to retain the initial state and not the whole path (these will be simulated forward from this initial state in the next step) and (b) $E[\overline{V}_{t+1}|V_t^{(i)}]$ are available in closed form for all affine models.

2. *Generating volatilities.* Simulate volatility paths using the Euler approximation to the transition density:

$$V_{t,j}^{(i)} = V_{t,j-1}^{(i)} + \kappa_v(\theta_v - V_{t,j-1}^{(i)})M^{-1} + \sigma_v\sqrt{V_{t,j-1}^{(i)}}\,\varepsilon_{t,j}^{(i)},$$

   for $j = 0, \ldots, M-1$, where $\varepsilon_{t,j}^{(i)}$ are iid $\mathcal{N}(0, M^{-1})$. Collect the simulated volatilities into a vector $V_{t+1}^{(i)} = (V_{t,0}^{(i)}, \ldots, V_{t,M-1}^{(i)})$, and compute the volatility integrals:

$$\overline{W}_{t+1}^{(i)} = \sum_{j=0}^{M-1} \sqrt{V_{t,j-1}^{(i)}}\,\varepsilon_{t,j}^{(i)}, \ \overline{V}_{t+1}^{(i)} = \frac{1}{M}\sum_{j=0}^{M-1} V_{t,j-1}^{(i)}.$$

3. *Second-stage reweighting.* Evaluate the second-stage (importance sampling) weights:

$$\pi_{t+1}^{(i)} \propto \pi_t^{(i)}\phi(y_{t+1}|\mu_y + \rho\overline{W}_{t+1}^{(i)}, (1-\rho^2)\overline{V}_{t+1}^{(i)})/w_{t+1}^{z(i)}.$$

This provides a weighted particle set $\{V_{t+1}^{(i)}, \pi_{t+1}^{(i)}\}_{i=1}^N$, which represents the filtering distribution at time $t+1$. At this stage, we may resample the particles from the discrete distribution to obtain an unweighted sample from the filtering distribution.

A.2 SVJ model

1. *First-stage resampling.* Given initial particles $\{V_t^{(i)}, \pi_t^{(i)}\}_{i=1}^N$, compute the expected integrated volatility $\widehat{V}_{t+1}^{(i)} = E[\overline{V}_{t+1}|V_t^{(i)}]$, and evaluate the first-stage weights:

$$w_{t+1}^{(i)} \propto \sum_{k=0}^{\infty} \phi\big(y_{t+1}|\mu_y + k\mu_s, \widehat{V}_{t+1}^{(i)} + k\sigma_s^2\big) \frac{\lambda^k e^{-\lambda}}{k!}.$$

   Resample the particles by drawing the indices $z(i) \sim Mult(N; w_{t+1}^{(1)}, \ldots, w_{t+1}^{(N)})$ and defining the resampled volatilities as $(V_t^{(i)}, \widehat{V}_{t+1}^{(i)}) = (V_t^{z(i)}, \widehat{V}_{t+1}^{z(i)})$.

2. *Generating the number of jumps.* Generate the number of jumps $J_{t+1}^{(i)} = N_{t+1}^{s(i)} - N_t^{s(i)}$ from the discrete distribution $\{p_k^{(i)}\}_{k=0}^{\infty}$, with probabilities defined by

$$p_k^{(i)} \propto \phi\big(y_{t+1} \,|\, \mu_y + k\mu_s, \widehat{V}_{t+1}^{(i)} + k\sigma_s^2\big) \frac{\lambda^k e^{-\lambda}}{k!}.$$

3. *Generating jump sizes.* Generate the total jump size $Z_{t+1}^{(i)} = \sum_{n=N_t^{s(i)}+1}^{N_{t+1}^{s(i)}} Z_n^s$ from a normal distribution $Z_{t+1}^{(i)} \sim \mathcal{N}(\mu_Z^{(i)}, \sigma_Z^{(i)})$, with mean and variance given by

$$\mu_Z^{(i)} = J_{t+1}^{(i)}\mu_s + J_{t+1}^{(i)}\sigma_s^2/\widehat{V}_{t+1}^{(i)}\big(y_{t+1} - \mu_y - J_{t+1}^{(i)}\mu_s\big),$$
$$\sigma_Z^{(i)} = J_{t+1}^{(i)}\sigma_s^2\big(1 - J_{t+1}^{(i)}\sigma_s^2/\widehat{V}_{t+1}^{(i)}\big).$$

4. *Generating volatilities.* Simulate volatility paths from the transition density using the Euler approximation for $j = 0, \ldots, M-1$,

$$V_{t,j}^{(i)} = V_{t,j-1}^{(i)} + \kappa_v\big(\theta_v - V_{t,j-1}^{(i)}\big)M^{-1} + \sigma_v\sqrt{V_{t,j-1}^{(i)}}\,\varepsilon_{t,j}^{(i)}, \quad \varepsilon_{t,j}^{(i)} \sim \mathcal{N}(0, M^{-1}).$$

   Collect the simulated volatilities into a vector $V_{t+1}^{(i)} = (V_{t,0}^{(i)}, \ldots, V_{t,M-1}^{(i)})$, and compute the volatility integrals:

$$\overline{W}_{t+1}^{(i)} = \sum_{j=0}^{M-1} \sqrt{V_{t,j-1}^{(i)}}\,\varepsilon_{t,j}^{(i)}, \quad \overline{V}_{t+1}^{(i)} = \frac{1}{M}\sum_{j=0}^{M-1} V_{t,j-1}^{(i)}.$$

5. *Second-stage reweighting.* Compute the second-stage weights:

$$\pi_{t+1}^{(i)} \propto \pi_t^{(i)} \frac{Poi\big(J_{t+1}^{(i)}|\lambda\big)}{p_{J_{t+1}^{(i)}}^{(i)}} \frac{\phi\big(Z_{t+1}^{(i)}|\mu_s, \sigma_s^2\big)}{\phi\big(Z_{t+1}^{(i)}|\mu_Z^{(i)}, \sigma_Z^{(i)}\big)} \frac{\phi\big(y_{t+1}|\mu_y + \rho\overline{W}_{t+1}^{(i)} + Z_{t+1}^{(i)}, (1-\rho^2)\overline{V}_{t+1}^{(i)}\big)}{w_{t+1}^{z(i)}},$$

   where $Poi(\cdot)$ is the Poisson probability mass function, and $p_k^{(i)}$ are the probabilities defined in Step 2.

   This results in the discrete distribution: $\{(V_{t+1}^{(i)}, J_{t+1}^{(i)}, Z_{t+1}^{(i)}), \pi_{t+1}^{(i)}\}_{i=1}^N$, which can be used for Monte Carlo approximation.

Note that in practice the Poisson approximation in Steps 1 and 2 is truncated at a low number. In fact, for our applications with rare jumps (λ small), we found no difference between this and using a Bernoulli approximation (assuming that k is either 0 or 1). The SVCJ model is handled similarly with two minor modifications. First, in Step 1, the formula for $E[\overline{V}_{t+1}|V_t^{(i)}]$ changes, and naturally incorporates jumps in volatility. Second, in Step 4, the variance jump sizes are simulated to generate volatility paths.

References

Aït-Sahalia, Y. 1996a. Nonparametric Pricing of Interest Rate Dependent Securities. *Econometrica* 64:527–60.

Aït-Sahalia, Y. 1996b. Testing Continuous Time Models of the Spot Interest Rate. *Review of Financial Studies* 9:385–426.

Aït-Sahalia, Y. 2002. Maximum-Likelihood Estimation of Discretely-Sampled Diffusions: A Closed-Form Approximation Approach. *Econometrica* 70:223–62.

Aït-Sahalia, Y. 2004. Disentangling Diffusion from Jumps. *Journal of Financial Economics* 74:487–528.

Aït-Sahalia, Y. 2008. Closed-Form Likelihood Expansions for Multivariate Diffusions. *Annals of Statistics* 36:906–37.

Aït-Sahalia, Y., J. Cocho-Diaz, and T. Hurd. 2006. Portfolio Choice with a Large Number of Assets: Jumps and Diversification. Working Paper, Princeton University.

Aït-Sahalia, Y., and J. Jacod. 2007. Testing for Jumps in a Discretely Observed Process. Working Paper, Princeton University.

Aït-Sahalia, Y., and R. Kimmel. 2005. Estimating Affine Multifactor Term Structure Models Using Closed-Form Likelihood Expansions. Working Paper, Princeton University.

Aït-Sahalia, Y., and R. Kimmel. 2007. Maximum Likelihood Estimation of Stochastic Volatility Models. *Journal of Financial Economics* 83:413–52.

Aït-Sahalia, Y., P. Mykland, and L. Zhang. 2005. A Tale of Two Time Scales: Determining Integrated Volatility with Noisy High-Frequency Data. *Journal of the American Statistical Association* 100:1394–411.

Andersen, T., L. Benzoni, and J. Lund. 2001. Towards an Empirical Foundation for Continuous-Time Equity Return Models. *Journal of Finance* 57:1239–84.

Andersen, T., L. Benzoni, and J. Lund. 2004. Stochastic Volatility, Mean Drift, and Jumps in the Short-Term Interest Rates. Working Paper, Northwestern University.

Andersen, T., T. Bollerslev, and F. Diebold. 2009. Parametric and Nonparametric Volatility Measurement, in L. P. Hansen and Y. Aït-Sahalia (eds.), *Handbook of Financial Econometrics.* Amsterdam: North-Holland.

Andersen, T., T. Bollerslev, and F. Diebold. 2007. Roughing It Up: Including Jump Components in the Measurement, Modeling and Forecasting of Return Volatility. Working Paper, Northwestern University.

Andersen, T., T. Bollerslev, F. Diebold, and P. Labys. 2001. The Distribution of Realized Exchange Rate Volatility. *Journal of the American Statistical Association* 96:42–55.

Bakshi, G., and C. Cao. 2004. Risk-Neutral Kurtosis, Jumps, and Option Pricing: Evidence from Most Actively Traded Firms. Working Paper, University of Maryland.

Bakshi, G., C. Cao, and Z. Chen. 1997. Empirical Performance of Alternative Option Pricing Models. *Journal of Finance* 52:2003–49.

Bates, D. 1996. Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options. *Review of Financial Studies* 9:69–107

Bally, V., and D. Talay. 1996. The Law of the Euler Scheme for Stochastic Differential Equations (II): Convergence Rate of the Density. *Monte Carlo Methods and Applications* 2:93–128.

Barndorff-Nielson, O., and N. Shephard. 2004. Power and Bipower Variation with Stochastic Volatility and Jumps. *Journal of Financial Econometrics* 2:1–48.

Barndorff-Nielson, O., and N. Shephard. 2006a. Econometrics of Testing for Jumps in Financial Economics Using Bipower Variation. *Journal of Financial Econometrics* 4:1–30.

Barndorff-Nielson, O., and N. Shephard. 2006b. Impact of Jumps on Returns and Realised Variances: Econometric Analysis of Time-Deformed Lévy Processes. Working Paper, Oxford University.

Bates, D. 2001. Post-'87 Crash Fears in S&P 500 Futures Options. *Journal of Econometrics* 94:181–238.

Bates, D. 2006. Maximum Likelihood Estimation of Latent Affine Processes. *Review of Financial Studies* 19:909–65.

Beckers, S. 1981. A Note on Estimating the Parameters of the Diffusion-Jump Model of Stock Returns. *Journal of Financial and Quantitative Analysis* 16:127–40.

Brandt, M., and P. Santa-Clara. 2002. Simulated Likelihood Estimation of Diffusions with an Application to Exchange Rate Dynamics in Incomplete Markets. *Journal of Financial Economics* 63:161–210.

Broadie, M., M. Chernov, and M. Johannes. 2007. Model Specification and Risk Premiums: Evidence from Futures Options. *Journal of Finance* 62:1453–90.

Chernov, M., E. Ghysels, A. Gallant, and G. Tauchen. 2003. Alternative Models for Stock Price Dynamics. *Journal of Econometrics* 116:225–57.

Chib, S., F. Nardari, and N. Shephard. 2002. Markov Chain Monte Carlo Methods for Stochastic Volatility Models. *Journal of Econometrics* 108:281–316.

Chib, S., M. Pitt, and N. Shephard. 2005. Likelihood Based Inference for Diffusion Driven Models. Working Paper, Oxford University.

Christoffersen, P., K. Jacobs, and K. Mimouni. 2007. Models for S&P 500 Dynamics: Evidence from Realized Volatility, Daily Returns, and Option Prices. Working paper, McGill University.

Crisan, D., and A. Doucet. 2002. A Survey of Theoretical Results on Particle Filtering for Practitioners. *IEEE Transactions on Signal Processing* 50:736–46.

Dai, Q., K. Singleton, and W. Yang. 2007. Regime Shifts in a Dynamic Term Structure Model of U.S. Treasury Bond Yields. *Review of Financial Studies* 20:1669–1706.

Das, S., and R. Sundaram. 1999. Of Smiles and Smirks: A Term Structure Perspective. *Journal of Financial and Quantitative Analysis* 34:211–40.

Das, S., and R. Uppal. 2004. International Portfolio Choice with Systemic Risk. *Journal of Finance* 59:2809–34.

Del Moral, P., and J. Jacod. 2001. Interacting Particle Filtering with Discrete Observations, in A. Doucet, J. F. G. de Freitas, and N. J. Gordon (eds.), *Sequential Monte Carlo Methods in Practice,* 43–77. New York: Springer.

Del Moral, P., J. Jacod, and P. Protter. 2002. The Monte Carlo Method for Filtering with Discrete-Time Observations. *Probability Theory and Related Fields* 120:346–68.

Doucet, A., N. de Freitas, and N. Gordon. 2001. *Sequential Monte Carlo Methods in Practice*. New York: Springer.

Duffie, D., and P. Glynn. 1995. Efficient Monte Carlo Estimation of Security Prices. *Annals of Applied Probability* 5:897–905.

Duffie, D., J. Pan, and K. Singleton. 2000. Transform Analysis and Asset Pricing for Affine Jump-Diffusions. *Econometrica* 68:1343–76.

Duffie, D., and K. Singleton. 1993. Simulated Moments Estimation of Markov Models of Asset Prices. *Econometrica* 61:929–52.

Durham, G., and R. Gallant. 2002. Numerical Techniques for Maximum Likelihood Estimation of Continuous-Time Diffusion Processes. *Journal of Business and Economic Statistics* 20:297–316.

Durham, G. 2003. Likelihood-Based Specification Analysis of Continuous-Time Models of the Short-Term Interest Rate. *Journal of Financial Economics* 70:463–87.

Elerian, O., S. Chib, and N. Shephard. 2001. Likelihood Inference for Discretely Observed Non-Linear Diffusions. *Econometrica* 69:959–93.

Eraker, B. 2001. MCMC Analysis of Diffusion Models with Applications to Finance. *Journal of Business and Economic Statistics* 19(2):177–91.

Eraker, B. 2004. Do Equity Prices and Volatility Jump? Reconciling Evidence from Spot and Option Prices. *Journal of Finance* 59:1367–403.

Eraker, B., M. Johannes, and N. Polson. 2003. The Impact of Jumps in Volatility and Returns. *Journal of Finance* 59:227–60.

Gallant, R., and J. Long. 1997. Estimating Stochastic Differential Equations Efficiently by Minimum Chi-Squared. *Biometrika* 84:125–41.

Gallant, R., and G. Tauchen. 2009. Simulated Score Methods and Indirect Inference for Continuous-Time Models, in L. P. Hansen and Y. Aït-Sahalia (eds.), *Handbook of Financial Econometrics.* Amsterdam: North-Holland.

George, T., and F. Longstaff. 1993. Bid-Ask Spreads and Trading Activity in the S&P 100 Index Options Market. *Journal of Financial and Quantitative Analysis* 28:381–97.

Golightly, A., and D. Wilkinson. 2006. Bayesian Sequential Inference for Nonlinear Multivariate Diffusions. *Statistics and Computing* 16:323–38.

Golightly, A., and D. Wilkinson. 2008. Bayesian Inference for Nonlinear Multivariate Diffusion Models Observed with Error. *Computational Statistics and Data Analysis* 52:1674–1693.

Gordon, N., D. Salmond, and A. Smith. 1993. Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings* F-140:107–13.

Hausenblas, E. 2002. Error Analysis for Approximation of Stochastic Differential Equations Driven by Poisson Random Measures. *SIAM Journal of Numerical Analysis* 40:87–113.

Huang, X., and G. Tauchen. 2005. The Relative Contribution of Jumps to Total Price Variance. *Journal of Financial Econometrics* 3:456-99.

Jarrow, R., and E. Rosenfeld. 1984. Jump Risks and the Intertemporal Capital Asset Pricing Model. *Journal of Business* 7:337–51.

Johannes, M. 2004. The Statistical and Economic Role of Jumps in Interest Rates. *Journal of Finance* 59:227–60.

Johannes, M. S., N. G. Polson, and J. R. Stroud. 2007. Sequential Parameter Estimation in Jump-Diffusion Models with Stochastic Volatility. Working Paper, Columbia University.

Jones, C. 1998. Bayesian Estimation of Continuous-Time Finance Models. Working Paper, University of Southern California.

Kim, S., N. Shephard, and S. Chib. 1998. Stochastic Volatility: Likelihood Inference and Comparison with ARCH Models. *Review of Economic Studies* 65:361–93.

Kitagawa, G. 1987. Non-Gaussian State Space Modeling of Nonstationary Time Series. *Journal of the American Statistical Association* 82:1032–51.

Li, H., M. Wells, and C. Yu. 2008. A Bayesian Analysis of Return Dynamics with Lévy Jumps. *Review of Financial Studies*. 21:2345–2378.

Liu, J., F. Longstaff, and J. Pan. 2002. Dynamic Asset Allocation with Event Risk. *Journal of Finance* 58:231–59.

Merton, R. 1980. On Estimating the Expected Return on the Market: An Exploratory Investigation. *Journal of Financial Economics* 8:323–61.

Omori, Y., S. Chib, N. Shephard, and K. Nakajima. 2007. Stochastic Volatility with Leverage: Fast Likelihood Inference. *Journal of Econometrics*. 140:425–449.

Pan, J. 2002. The Jump-Risk Premia Implicit in Options: Evidence from an Integrated Time-Series Study. *Journal of Financial Economics* 63:3–50.

Pan, J., and J. Liu. 2003. Dynamic Derivative Strategies. *Journal of Financial Economics* 69:401–30.

Pedersen, A. 1995. A New Approach to Maximum Likelihood Estimation for Stochastic Differential Equations Based on Discrete Observations. *Scandinavian Journal of Statistics* 22:55–71.

Piazzesi, M. 2005. Bond Yields and the Federal Reserve. *Journal of Political Economy*. 113:311–344.

Pitt, M. K. 2002. Smooth Particle Filters for Likelihood Evaluation and Maximization. Working Paper, University of Warwick.

Pitt, M. K., and N. Shephard. 1999. Filtering via Simulation: Auxiliary Particle Filter. *Journal of the American Statistical Association* 94:590–99.

Pritsker, M. 1998. Nonparametric Density Estimation and Tests of Continuous-Time Interest Rate Models. *The Review of Financial Studies* 11:449–87.

Yu, J. 2007. Closed-Form Likelihood Approximation and Estimation of Jump-Diffusions with an Application to the Realignment Risk of the Chinese Yuan. *Journal of Econometrics* 141:1245–80.