Local Stochastic Volatility Model: Model Validation

MATH-GA.2753.001: HW7

Author: Oksana Kitaychik

Reviewers: Ken Abbot, Irena Khrebtova

Table of Contents

1 Executive Summary

In this study, I examine the validity of the local stochastic volatility model (LSVM), specifically, its implementation as suggested by Julien Guyon and Pierre Henry-Labordere (2014).[1] This model is a modern extension (and a practical combination) of both local volatility and stochastic volatility models. In this paper, it is implemented via a so-called 'particle method' which builds on the chaos propagation theory and McKean non-linear stochastic differential equations (SDEs).[2]

First I examine a theoretical foundation for the LSVM model (including a discussion of McKean non-linear SDEs and the particle method). Next, I briefly discuss possible implementations for the LSVM model, and within the framework of some of such implementations, the results of the testing of the model with a range of the inputs and parameters as well as the comparison between the model performance with the performance of a simpler Black-Scholes (BS) type model and a 'pure; stochastic volatility model (SVM). I also analyze the model strengths, as well as its weaknesses, and recommend further testing and validation steps.

Finally, based on the model validation results, I conclude that the LSVM is likely to be superior in capturing the actual market data, including so-called 'volatility smile' (i.e., the dependence of the derivatives' prices on the volatility at different strike levels),[3] ubiquitous in many financial asset derivative prices as compared to simpler models such as BS and SVM. However, the LSVM is fairly difficult to implement with sufficient precision in non-industry settings given a relatively lengthy computational time and effort. Also, the choice and calibration of model parameters is a somewhat subjective process, and may present a number of implementation difficulties (such as inconsistency of parameters depending on the market conditions etc). A further study and testing of the LSVM is recommended to ensure a complete and thorough model validation.

[1] See Guyon, Julien & Henry-Labordere (2014), *Nonlinear Option Pricing*, Chapter 11. I also discussed the model and its implementation with one of the authors, Professor Julien Guyon, when I took his class "Computational Methods for Finance" (Courant, Fall 2016).
[2] See, for example, the description of theory and implementation in Guyon, Julien & Henry-Labordere (2014), *Nonlinear Option Pricing*, Chapter 10.
[3] More specifically, a 'volatility smile' is the difference in implied volatility (volatility implied by the observed derivative (option) prices) between out-of-the-money options, at-the-money options and in-the-money options.

2 Overview

In accordance with the OCC 2011-12 specifications and the SR 11-7 standard/guidelines, this paper lays out the model (LSVM) specifications, model objectives, core assumptions, logic and implementation. It also tests the LSVM against two other simpler (benchmark) models, BS model and SVM.

In Section 3, “Technical Specifications / Model Description”, I explain the theory behind the LSVM. In Section 4, “Review of Assumptions”, I analyze input, processing and output of the LSVM. In Section 5, “Sensitivity Analysis”, I perform sensitivity tests of the LSVM key inputs and parameters (including some of the simulated stress test scenarios). In Section 6, “Comparison with the Benchmark Models”, I compare the performance of the LSVM with the performance of the two benchmark models, i.e., the BS model and a ‘pure’ SVM. In Section 7, I describe the key LSVM’s relative strengths and weaknesses (as compared to the benchmarks and a number of other commonly used volatility modeling techniques). Finally, in Section 8, I conclude and offer recommendations regarding the LSVM further use and validation.

Appendix 1 lists the sources I relied on in this study. Appendix 2 references the attached Python code I used in this project.

3 Technical Specification / Model Description

This section first introduces key foundational principles behind the LSVM and its implementation, specifically, the McKean SDEs and the particle method.[4] Next, I describe the LSVM technical specification as presented, for example, in Guyon, J. & Henry-Labordere, P. (2014), *Nonlinear Option Pricing*, Chapter 11. Finally, I discuss some of the key current and potential implementation areas for the LSVM.

3.1 McKean Non-Linear SDEs

McKean non-linear SDEs were first introduced by Henry McKean in 1966. Unlike other types of SDEs, these SDEs are concerned with stochastic processes where the drift and volatility depend not only on the values of the stochastic process over time but also *on the probability distribution of the stochastic process*. For example, for an n-dimensional process X_t with probability distribution P_t:

$$dX_t = \mu(t, X_t, P_t)\, dt + \sigma(t, X_t, P_t)\, dW_t \quad \text{with} \quad X_0 \in R^n \qquad (1)$$

where W_t is a standard d-dimensional Brownian motion; t is time; μ (.) is a drift term, and σ (.) is the volatility (diffusion) term of the X_t process.[5]

3.2 Particle Method and Chaos Propagation Theory

The particle method has been widely used, for example, in statistical physics, and more recently, it was introduced in the areas of mathematical finance. In this study, it is used to simulate the McKean non-linear SDEs described above. Specifically, we replace the probability law P_t of the stochastic process X_t by its approximation $P_t^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_t^{i,N}}$ where $X_t^{i,N}$ (with $1 \le i \le N$) are solutions to the system of the N standard R^n SDEs of the type:

$$dX_t^{i,N} = \mu(t, X_t^{i,N}, P_t^N)\, dt + \sigma(t, X_t^{i,N}, P_t^N)\, dW_t^i \quad \text{with} \quad P_0 = \text{probability law of } (X_0^{i,N}) \qquad (2)$$

Thus, in the particle method of simulating the paths of X_t, the paths of $X^{i,N}$ *interact* with each other. In order to simulate such process, not only we need to know the position of each path $X_t^{i,N}$ at time t (as would be sufficient in a classic Monte Carlo simulation), we also need to know the position of other paths $X_t^{j,N}$ ($j \neq i$) at each time t. In the common terminology of the particle method, the stochastic process paths are often referred to as "particles".

The particle method results in an empirical approximation of the distributions convergent to the 'true' distributions if the so-called *chaos propagation property* holds. As shown, for example, in Guyon & Henry-Labordere (2014), the propagation of chaos holds if at time t=0, the $X_0^{i,N}$ are independent particles, then as $N \to \infty$, for any fixed time t>0, the $X_0^{i,N}$ are asymptotically independent and their empirical measure P_t^N converges in distribution toward the true measure P_t.

[4] The description in these sub-sections are largely based on the presentation of McKean non-linear SDEs in Chapter 10 of Guyon, J. & Henry-Labordere, P. (2014), *Nonlinear Option Pricing*.

[5] "d." indicates a standard difference operator.

3.3 LSVM Specification

In this paper, I will consider the following LSVM specification:[6]

$$df_t = a_t f_t \sigma(t,f_t)\, dW_t \qquad (3)$$

where f_t is the forward; a_t is the stochastic volatility process (which can be in turn modeled by any stochastic volatility model (SVM) such as SABR, Heston, Stein-Stein, etc); and $\sigma(t,f_t)$ is the local volatility function that can allow us to calibrate to a full surface of implied volatilities. The latter function is also known as the "leverage" function and will be occasionally referred to as $l(t,f_t)$ in this paper (and in the attached Appendix 2 code). Note that when an observed market volatility smile happens to be matching to the smile modeled by the pure SVM (i.e., a_t component), the leverage function $l(t,f_t)$ will be 1 (or close to 1).

As shown, for example, in Guyon & Henry-Labordere (2014), the LSVM is exactly calibrated to market smiles iff:

$$[\sigma_{Dup}(t,f_t)]^2 = l(t,f_t)^2\, E^Q[a_t^2 \mid f_t = f] \qquad (4)$$

where $E^Q[.]$ is the expectation operator under the risk-neutral measure Q, and $[\sigma_{Dup}(t,f_t)$ is the Dupire local volatility that is being inferred from the vanilla (i.e., implied by BS) market smile, i.e.:

$$[\sigma_{Dup}(t,f_t)]^2 = \frac{\partial_t\ C(t,f)}{0.5 f^2 \partial_f^2\, C(t,f)} \qquad (5)$$

and C(t,f) is the market price (at time 0) of a call option with strike *f* and maturity t written on the forward of maturity T.

Let us substitute an expression for the leverage function from (4) into the equation (3) above:

$$df_t = a_t f_t \frac{\sigma_{Dup}(t,f_t)}{\sqrt{E^Q[a_t^2|f_t]}} dW_t \qquad (6)$$

Thus, the leverage function depends on the *joint probability distribution function* (PDF) p(t,f,a) of the stochastic process (f_t, a_t), i.e., this is an example of McKean SDEs introduces above, and which can be solved via the particle method as will be demonstrated further. Specifically, note that the volatility of f_t depends on the joint distribution of (f_t, a_t) through the conditional expectation $E^Q[a_t^2 \mid f_t]$. In summary, we can write the expression for the leverage function as:

$$l(t, f, p) = \sigma_{Dup}(t,f) \sqrt{\frac{\int p(t,f,a)da}{\int a^2 p(t,f,a)da}} \qquad (7)$$

In practical implementations, as further discussed in Section 4.2 and 5.1 below, the $\int(.)$ portion of the equation (7) is approximated via fitting a non-parametric regression, i.e., a kernel:

[6] We assume deterministic interest rates and no dividends in this section. This model can be extended stochastic interest rates and dividends/repos, see Guyon & Henry-Labordere (2014).

$$l_N(t, f) = \sigma_{Dup}(t,f) \sqrt{\frac{\sum_{i=1}^{N} \delta_{t,N}(f_t^{i.N}-f)}{\sum_{i=1}^{N} \left(a_t^{i.N}\right)^2 \delta_{t,N}(f_t^{i.N}-f)}} \tag{8}$$

where N is the number of interacting stochastic process paths ("particles") (i=1,2,3….N); δ(.) is a regularizing kernel.

As discussed in Guyon and Henry-Labordere (2014), it is natural (and common) to assume the following form of the kernel:

$$\delta_{t,N}(x) = \frac{1}{h_{t,N}} K\left(\frac{x}{h_{t,N}}\right) \tag{9}$$

where K is a fixed, symmetric kernel with a bandwidth $h_{t,N}$ that tends to zero as N grows to infinity.

As suggested by Guyon and Henry-Labordere (2014), we can take the following form of the bandwidth:

$$h_{t,N} = \kappa\, f_0\, \sigma_{market,t} \sqrt{\max(t,\ t_{\min})}\ N^{-0.2} \tag{10}$$

where σ_{market} is some observed market implied volatility surface at maturity t, f_0 is the initial forward value, t_{min} is the minimum time-step (market instrument maturity) used (e.g., t_{min}=0.25 years[7]), and κ is the bandwidth parameter ("kappa").

The bandwidth factor $N^{-0.2}$ in equation (10) above comes from the minimization of the asymptotic mean integrated squared error of the non-parametric (e.g., Nadaraya-Watson) estimator, which is the sum of two terms: bias and variance. The smaller the bandwidth, the smaller the bias, but the larger the variance. The critical bandwidth that minimizes the sum of bias and variance decreases at the rate of $N^{-0.2}$ for large N.[8]

Note that from equations (4) and (8) it follows that the expected value of squared stochastic volatility process ("stochastic variance") given each forward value is:

$$E^Q[a_t^2 \mid f_t = f] = 1 / \frac{\sum_{i=1}^{N} \delta_{t,N}(f_t^{i.N}-f)}{\sum_{i=1}^{N} \left(a_t^{i.N}\right)^2 \delta_{t,N}(f_t^{i.N}-f)} \tag{11}$$

And the leverage function is:

$$l_N(t, f) = \frac{\sigma_{Dup}(t,f_t)}{\sqrt{E^Q\,[a_t^2 | f_t = f]}} \tag{12}$$

LSVM can be seen as both an extension (and combination of) both a local volatility model (LVM) (or Dupire LVM) and a SVM. Note that a standard SVM can handle only a *finite* number of parameters (e.g., volatility-of-volatility (VoV), volatility mean reversion etc), and, thus, by definition, cannot be calibrated to the whole implied volatility surface. On the other hand, standard LVMs (which can calibrate to the

[7] This is also the value used in this study.
[8] See discussion in Guyon & Henry-Labordere (2014), p. 281.

observed volatility surface) cannot consistently model the observed properties of volatility such as mean reversion and VoV. The LSVM overcomes both the weaknesses of SVMs and LVMs, and, thus, at least theoretically offers a superior option pricing and risk management tool.

3.4 Common Practical Uses of LSVM

Stochastic volatility models (SVMs) and LSVM have been and can be used for pricing and hedging a number of derivatives and especially derivatives that bear such risks as volatility-of-volatility risk, spot/volatility correlation risk, forward smile risk, etc. Such models found a wide application in the field of interest rate modeling and pricing; see relevant discussion in Andersen & Piterbarg (2010), *Interest Rate Modeling*, and Brigo & Mercurio (2006), *Interest Rate Models – Theory and Practice*, for example.[9] Unlike standard SVMs, the LSVM combines the theoretical appeal of SVMs with the ability to calibrate precisely to the observed market volatility-smile dependent data (i.e., the feature that could have been previously found only in local volatility models (LVM)).

In this study, I demonstrate the use of the LSVM for pricing both vanilla options and a forward-starting call option.

4 Review of Assumptions

Below I analyze three aspects of the model: input (parameters), processing (implementation) and output.

4.1 Parameters / Inputs

The first component of the model is the information input component. It delivers assumptions and data to the model.

The key inputs and parameters (assumptions) for the LSVM are:

- Time to maturity T. For simplicity and without loss of generality, I assume T=1 in this paper.
- Specification of the forward process.[10] I assume a commonly encountered log-normal forward process under risk-neutral measure of S_t: $dS_t = a_t\, l(t, S_t)\, S_t\, dW_t^{(1)}$
- Initial forward (underlying asset) value of S_0. Without loss of generality, let us assume S_0=100.
- Initial volatility process value of a_o. Without loss of generality, I assume it to be 30%.
- VoV or γ. **This is an important input and I test sensitivity of the model to a number of VoVs** such as 0%, 25%, 50% and 75%.
- Specification of the volatility process a_t. As commonly encountered in practical implementations, I assume that the volatility process is also log-normal, and has no drift under the risk-neutral measure. Specifically, $da_t = a_t\, \gamma\, dW_t^{(2)}$
- Market implied volatility surface. I assume such surface to be flat for the ease of stress testing; such surface is encountered in some practical implementations.[11] However, note that all the key

[9] See also discussion in Joshi (2003; Chapters 16 and 18) and Wilmott for other practical implementations of stochastic volatility models.

[10] Subscripts (1) and (2) in this section refer to two different Brownian motions $W_t^{(1)}$ and $W_t^{(2)}$.

sensitivity testing and model validation results discussed in this paper will apply to more complex (and realistic) cases of non-flat market implied volatility surface.

- Correlation between two Brownian motions underlying the forward process St (i.e., $W_t^{(1)}$) and its volatility process at (i.e., $W_t^{(1)}$): ρ = - 50%. **This is another important input parameter**, and I test **sensitivity of the model to a number of correlation parameters** from -100% to 100% (e.g., in increments of 25%).
- As the market implied volatility surface is assumed to be flat at 30%, that means that we also have $\sigma_{Dup}(t,f_t)$ = 30%.
- Forward values of the stochastic (forward / underlying asset) process.
- Strike values (e.g., for a vanilla call option). I have tried a range of strikes: from 60 to 150 in increments of 10.
- Parameters (assumptions) to implement a dependence of the forward (stochastic process) volatility (through its leverage function) on the joint probability distribution of the forward/stochastic process itself and its volatility (i.e., see equation (7) above). These parameters may be also considered a processing component of the model, and thus, they are discussed in further detail in the next section.

In Section 5, I test the model sensitivity to all important inputs.

4.2 Processing / Implementation

The second component of the model is the processing component. It transforms inputs into estimates.

There are a few key processing (implementation steps) for the LSVM:

- As mentioned above, the **kernel function** approximates the joint pdf of the stochastic (forward) process and its volatility. The components of the kernel are:
    - o Choice of a kernel to implement a pdf. In this study, I test the sensitivity of the model to two types of a kernel: Gaussian and quartic.
    - o The form of the kernel bandwidth $h_{t,N}$. It was selected according to the literature (Guyon & Henry-Labordere (2014)) as discussed in Section 3 and shown in equation (10).
    - o The pre-factor coefficient of the bandwidth κ (see equation (10)) which determines the size of the bandwidth. The choice of κ is important to the effectiveness of the local volatility function (leverage) calibration to the observed market data. In Section 5, I test the model to a number of κ values, and choose the κ value that results in the best fit of the model to the observed market data (given all other parameters and processing techniques I chose).
- Kernel is then used to estimate the expected stochastic variance given a forward value, i.e., $E^Q[a_t^2 \mid f_t= f]$ (see equation (11)) via a non-parametric kernel (Nadaraya – Watson) regression. See my Python code in Appendix 2. There are two ways to estimate such expected stochastic variance:

[11] See, for example, discussion in Hull (2008).

  - a *precise* estimation, i.e., an estimation of the expected variance for every simulated forward value via the non-parametric kernel regression (option precise = 1 in the Python function "lev_function" I developed, see Appendix 2).
  - an approximation based on creating a *grid* of forward values of a certain size (e.g., such as dividing the range of all possible simulated forward values in 2000 equal size 'bins', as set by option precise=0 in the Python function "lev_function", see Appendix 2). Then the non-parametric kernel regression is effectively used to estimate the expected variance for the grid of the forward values.[12] This saves some computational time.
  - In Section 5 below, I discuss the advantages and disadvantages of each of these two estimation methods ('*precise*' vs '*grid*' method) and the sensitivity of the model results to each of these two methods.
- In this study, for the purposes of an option valuation, a *linear interpolation method* was used to map the calibrated leverage functions to the simulated stochastic process (forward) values. I tested the sensitivity of the model to this interpolation assumption, and settled on the (simpler) linear method, as for example, a cubic interpolation methods resulted in a greater estimation cost (time) yet did not appear to add much in terms of the precision.
- For each of the option strike levels, we can obtain a leverage function and thus the option price using the LSVM. In this study, a *cubic interpolation method* was used to interpolate between a finite set of the strikes I used for the model validation purposes. See Figure 1 below. As this is just an auxiliary technique to present the results, I did not test the model sensitivity to the choice of this interpolation scheme (linear vs cubic etc). Moreover, the choice of the cubic interpolation for such or similar purposes was suggested by the relevant academic/industry literature.[13]
- For the simulation of the stochastic forward and volatility values, one has to use a certain **discretization scheme**. As described in Section 5, I tested the model precision using a (regular) Euler discretization and a log-Euler discretization; I chose a log-Euler discretization scheme as it produces more precise estimates.[14]
- The reliability of a simulation technique also generally depends on the choice of a time step (**discretization interval**). I chose the time step of 0.1 years (less than 3 calendar days). (I tested whether the model's precision improved substantially with even smaller sensitivity but did not find it to be the case.)
- Finally, the precision of the simulation depends on the **number of stochastic paths** chosen for the simulation. For the local volatility (leverage) function calibration, as discussed in Section 5, I tried a number of different simulation paths, and ended up choosing 7,000 simulation paths as the balance between the computational time and precision (see "lev_function" in the attached Python code; "N1" input).

[12] To improve precision of this method, a linear interpolation may be used to interpolate forward values between each of the grid points, as it was implemented in my Python code; see Appendix 2.
[13] See, for example, Guyon & Henry-Labordere (2014); Andersen & Piterbarg (2010).
[14] This also follows from the theory of discretization. See, for example, Andersen & Piterbarg (2010; Chapter 3), and Glasserman (2004; Chapter 6).

- For the actual option pricing, I used 100,000 simulation paths (see "call_LVM" function in the attached Python code; "N2" input). I tested whether the model's precision improved substantially with even a larger number of simulation paths, but generally did not find any marked difference (particularly after taking into account the resulting decrease in computational speed).

In Section 5, I test the model sensitivity to all key model processing components.

4.3 Output

The third component of the model is the output, or reporting, component. It translates the model estimates into useful business information.

In this study, one of the key outputs is the forward-starting call spread price.[15]

In addition, I considered:

- Vanilla call option prices (particularly, to study the performance of the model in terms of capturing the market data; see Section 5.2);
- Ability of the model to capture volatility skew (i.e. leverage function at different forward ('strike") levels; see, for example, Section 5.2).

[15] In practice (and time permitting), one would also need to obtain all key risk management metrics (such as option spread sensitivities to the change in the underlying (delta), change in delta (gamma), volatility (vega), etc (all "Greeks")).

5 Sensitivity Analysis

5.1 Model Implementation

In the attached Appendix 2, I describe the implementation of the LSVM step by step, and show all the underlying key functions and computational steps (implemented via my Python code). I also summarize the LSVM implementation below.

The LSVM in this study has been implemented as described in Section 3 above.

More specifically, I assume the log-normal distribution of the underlying asset (forward) process $X_t = S_t$:[16]

$$dS_t = a_t\, l(t,S_t)\, S_t\, dW_t^{(1)} \tag{13}$$

And also the log-normal distribution of the stochastic volatility process (similar, for example, to a simple SABR specification):

$$da_t = a_t\, \gamma\, dW_t^{(2)} \tag{14}$$

The term γ is VoV, as also introduced in Section 4.1 above.

The correlation between two underlying Brownian motions $W_t^{(1)}$ and $W_t^{(2)}$ is denoted by ρ:

$$d \langle W_t^{(1)}, W_t^{(2)} \rangle_t = \rho\, dt \tag{15}$$

See Section 4.1 (and Appendix 2) for the parameter values.

The first step in the model is the calibration of a local volatility (leverage) function $l(t,S_t)$ so that this model matches the market prices of vanilla options. As discussed above, I assume that all the vanilla option prices in the market are such that they match those of a Black-Scholes model, i.e., that the market implied volatility surface is flat at $\sigma_{Market} \equiv 30\%$. In that case, we also have $\sigma_{Dup}(t,S_t) \equiv 30\%$.

Next, I check that the resulting model is indeed calibrated to the market implied volatilities $\sigma_{Market} \equiv 30\%$. To this end, I compute estimates of the (vanilla) call prices (maturity T=1) in the calibrated LSVM for strikes equal to 60, 70, 80, 90, 100, 110, 120, 130, 140, and 150, and invert the Black-Scholes formula to get the corresponding estimation of the implied volatilities $\sigma(T,K)$. To estimate the call prices in the calibrated model, I will use the calibrated leverage function $l(.)$, and, as mentioned in Section 4.2 above, N_2 of 100,000 simulation paths. The sensitivity of the LSVM to the option 'moneyness' (strike levels) is described in Section 5.2 below.

I also conduct a number of other sensitivity checks as part of the model validation process, as introduced in Sections 4.1 and 4.2 above, and further described in Section 5.2 below. In addition, as described in Section 6, I compare the performance of the model to one of the benchmarks, a 'pure' SVM.

[16] This is a common assumption, for example, for stock markets. See Hull (2008).

Finally, I use the developed calibrated LSVM to price a forward-starting call spread. This call spread has the following payoff:

$$Call\ Spread\ Payoff = \left(\frac{S_{T_2}}{S_{T_1}} - K_1\right) - \left(\frac{S_{T_2}}{S_{T_1}} - K_2\right) \quad (16)$$

where $T_1 = T - \frac{1}{12}$; $T_2 = T = 1$ year; $K_1 = 95\%$; $K_2 = 105\%$; VoV=γ = 50%; ρ = -50%.

In Section 6 below, I will compare the price of such call spread obtained using the calibrated LSVM to the prices obtained using two benchmark models: (1) Black-Scholes (with constant volatility of 30%), and (2) a pure SVM (i.e., stochastic volatility model with the leverage function set to a constant value of 1.0).

5.2 Testing Model Sensitivity to Key Inputs and Processing Components

I. I first tested the model sensitivity to the following **key inputs (parameters)**:

(1) Sensitivity to option 'moneyness' (option strike values):

I computed estimates of the (vanilla) call prices (maturity T=1) in the calibrated LSVM for strikes equal to 60, 70, 80, 90, 100, 110, 120, 130, 140, and 150, and inverted the Black-Scholes formula to get the corresponding estimation of the implied volatilities σ(T,K), for each of the obtained (from LSVM) call option prices. If the LSVM works well, I expect the resulting implied BS volatilities to be close to the observed market surface of 30%. Appendix 2 shows the results of this sensitivity check. I also summarize the results below:

```
Strike: 60
Call Price (LVM): 40.69
Imp. vol. is 0.332783

Strike: 70
Call Price (LVM): 31.48
Imp. vol. is 0.303302

Strike: 80
Call Price (LVM): 23.46
Imp. vol. is 0.297052

Strike: 90
Call Price (LVM): 16.73
Imp. vol. is 0.291855

Strike: 100
Call Price (LVM): 11.42
Imp. vol. is 0.287158

Strike: 110
Call Price (LVM): 7.57
Imp. vol. is 0.285364
```

```
Strike: 120
Call Price (LVM): 4.60
Imp. vol. is 0.276248

Strike: 130
Call Price (LVM): 2.80
Imp. vol. is 0.273969

Strike: 140
Call Price (LVM): 1.67
Imp. vol. is 0.272196

Strike: 150
Call Price (LVM): 0.98
Imp. vol. is 0.271151
```

The calibrated LSVM matches the implied market volatilities reasonably well - note that all 'recovered' implied volatilities above are indeed roughly around 30% (i.e., matching the observed (given) flat market volatility surface), indicating a reasonably successful leverage function calibration to the market data.

However, as the options get deeper in- or out-of the money (at such strikes as 60 or above 120; with the current spot of 100), there is some deviation from the market volatility level of 30%. Thus, a further fine-tuning of the LSVM is needed (such as better kernel approximation technique; a much higher number of simulation paths used for the local volatility calibration[17] etc).

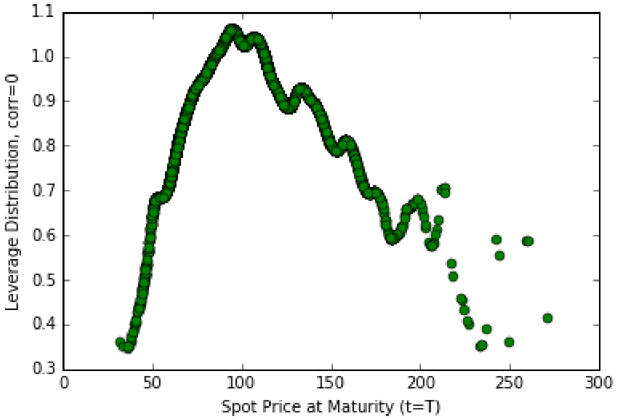
Despite this weakness, the calibrated LSVM performed much better than a ‘pure’ SVM. The latter could not reproduce the observed flat market volatility surface of 30%. As shown in Section 6.1 (1), Figure 4, a ‘pure’ SVM resulted in the pronounced volatility skew for non-ATM (non-at-the-money) options.

(2) Sensitivity to forward values at maturity:

Similarly to the exercise described in (1) above, but in more general (non-product-specific) settings, I tested the sensitivity of the LSVM to the forward prices at maturity (S_T). (See Appendix 2, section (b), “bullet 1” for the code / implementation). Figure 1 below shows the levels of the calibrated leverage (local volatility) functions at different levels of S_T:[18]

[17] This would require a much more powerful machine than my old personal laptop (e.g., an actual industry setting).
[18] At correlation set to zero. (Similar patterns were observed at different correlation levels). As discussed in Section 4.2 above, I used a cubic interpolation method to interpolate leverage function values between the ‘grid’ values. I used increments of forward price at maturity of about 2. See Appendix 2.

Figure 1



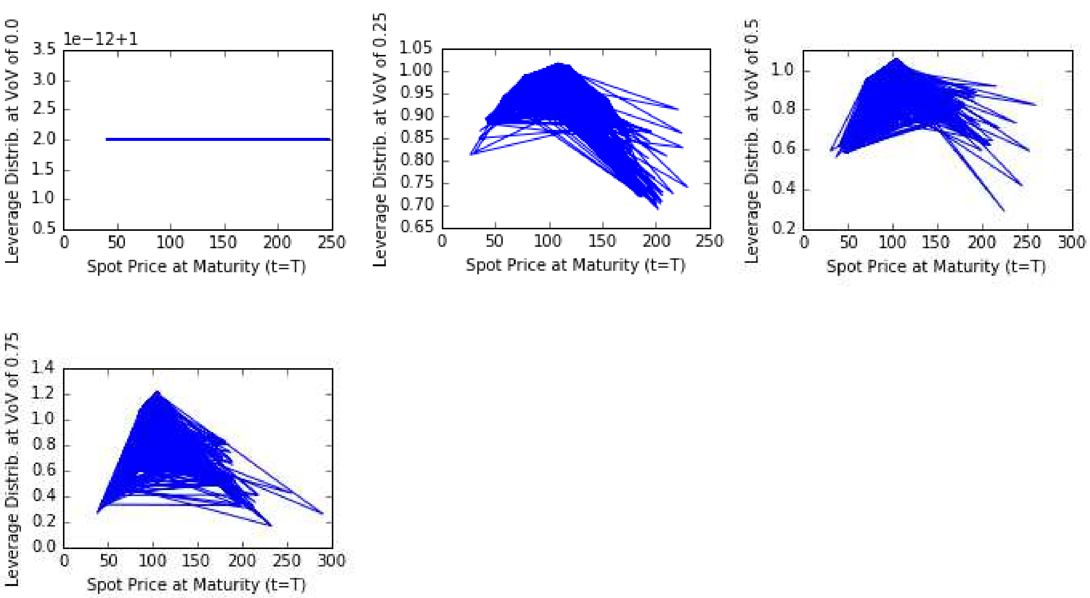
The resulting leverage function has an expected shape. Specifically, as for this chart I set correlation to 0%, one would expect the leverage reaches it maximum level of about 1 at the current spot price of 100 (as at that level, the observed market volatility would match a pure SVM volatility (i.e., LSVM with leverage equal 1 and correlation of zero).

Consistent with the findings in (1) above, the calibration does not perform as well at far forward values (e.g., at S_T above 200). For such high values, we see an inconsistent 'scattered' pattern of the resulting leverage function, as Figure 1 above shows.

(3) Sensitivity to volatility of volatility (VoV):

Next, let us test the sensitivity of the LSVM to the VoV. Let us pick some correlation level (e.g., a 'neutral' level of 0%), and plot the calibrated local volatility (leverage) at different S_T and VoV levels:

Figure 2



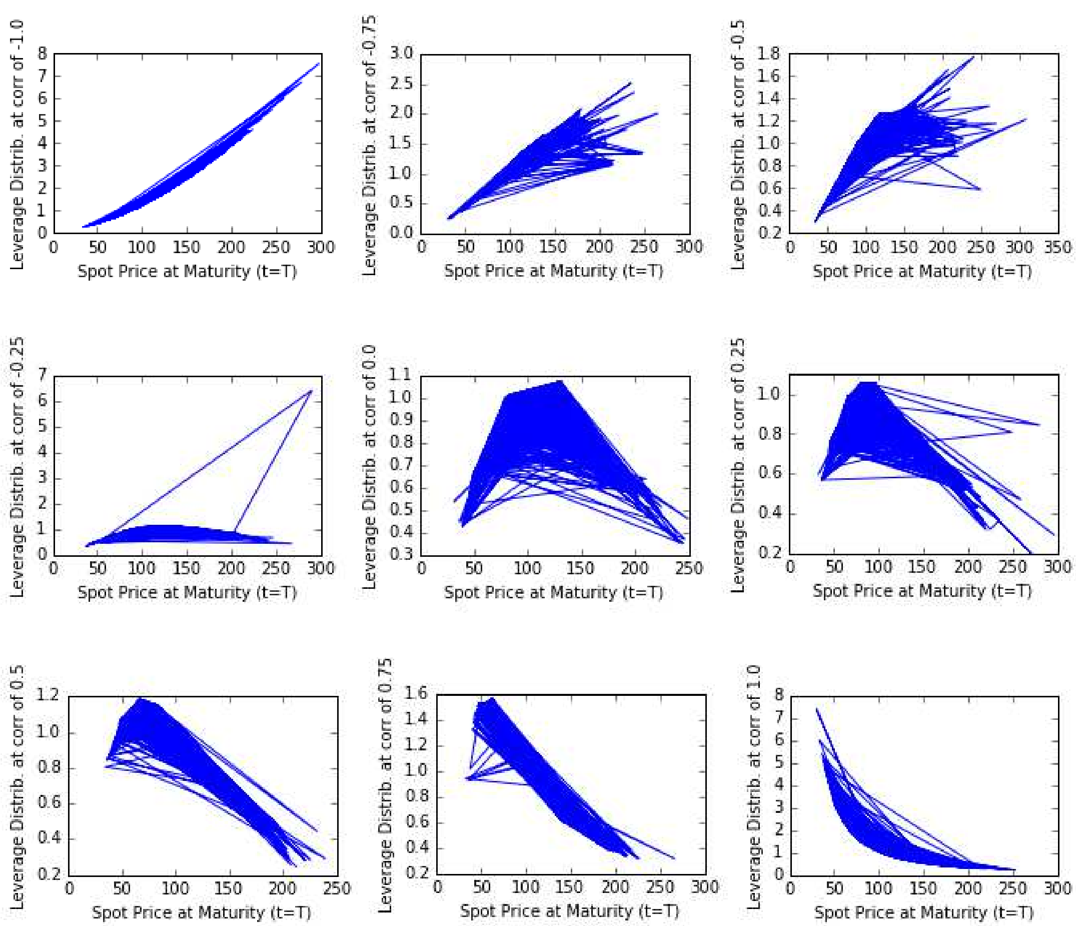
The shapes of the resulting leverage functions shown above are as expected. Specifically, at VoV=0 (i.e., constant volatility), the leverage function is constant at 1.0 (as shown in Figure 2 (first chart)). At relatively low VoV values, leverage function is (as expected) relatively flat (see Figure 2, second chart). As VoV increases, the leverage function becomes sizably steeper, i.e., its slope increases (reflecting the need for the more pronounced calibration at the times when the asset volatility changes drastically (i.e., high VoV values).

This should be contrasted with a much weaker performance (in fact, inability) of the pure SVM to properly capture the changes in VoV. See Section 6.1 (2), Figure 5: a 'pure' SVM resulted in the pronounced volatility skews, which become even more pronounced at higher VoV values.

(4) Sensitivity to correlation:

Let us now test the sensitivity of the LSVM to the correlation between the underlying asset process and its volatility process, ρ. For the given (market) level of VoV=50%, let us plot the calibrated local volatility (leverage) at different S_T and correlation levels:

Figure 3



The shapes of the leverage distributions/functions shown above are as expected. Specifically, with perfect negative correlation, we see that the leverage function is *linearly increasing* in spot (forward) prices: as spot prices/strikes increase, we see rapidly increasing leverage function values. This happens as the leverage function needs to 'compensate' for the fact that spot price and volatility processes are negatively correlated, i.e., the leverage 'calibration' function helps to 'pull' the modeled forward price process back to the observed market data.

Similarly, with perfect positive correlation, we see a *linearly decreasing* leverage function:[19] i.e., as spot prices/strikes increase, we see rapidly decreasing leverage function values - at higher spot prices, volatility will be also higher, thus, the calibrated leverage function is dropping to low values (as there is no need to 'compensate'; there is lower sensitivity of the spots to the market 'disturbances'/ events at high spot values in such a case).

And, as expected, at the correlation levels ‘in-between’, the leverage function shows the corresponding changing (‘rotating’) shape at changing correlation levels.

This should be contrasted with a weaker performance of the ‘pure’ SVM to properly capture the changes in correlation. See Section 6.1 (3), Figure 6: a ‘pure’ SVM resulted in the pronounced volatility skews, which become even more pronounced at higher correlation values.

II. Next, I tested the model sensitivity to the following **key processing components:**

(5) Sensitivity to the choice of kernel and kernel-related parameters:

I tested the model for the use of both Gaussian kernel[20] and quartic kernel.[21] I have not noticed a sizable difference in the performance. I settled on the quartic kernel, as also recommended by Guyon and Henry-Labordere (2014).

One has to calibrate the kernel to the market data. The kernel parameter that is used for that is κ. As discussed in Appendix 2, I settled on κ of 1.5 as it gave the best fit to the observed market data[22] (i.e., market implied volatility surface flat at 30%). This choice of κ is consistent with the recommendation in Guyon and Henry-Labordere (2014).

(6) Sensitivity to the choice of a forward values grid in estimating the expected stochastic variance:

As discussed in Section 4.2, kernel is used to estimate the expected stochastic variance given a forward value, i.e., $E^Q[a_t^2 \mid f_t = f]$ (see equation (11)) via a non-parametric kernel (Nadaraya – Watson) regression. I have tried two ways to estimate such expected stochastic variance. The first way, or *precise* estimation, is an estimation of the expected variance for every simulated forward value via the non-parametric

[19] It may not be entirely ‘linear-looking’ just due to the estimation error.
[20] See Appendix2, “gauss_kern” function in Section (a).
[21] See Appendix2, “quar_kern” function in Section (a).
[22] Effectively, ***assumed*** market data for the purposes of this model validation exercise.

kernel regression. The second way, or an approximation based on creating a grid of forward values of a certain size,[23] is an estimation of the expected variance for the chosen grid of the forward values. This approximation saves some computational time but resulted in the poorer calibration, even with a substantially higher number of simulation paths (I.e., 7,000 paths for the precise calibration vs. 30,000 paths for the grid-based approximation). The calibrated implied volatilities were about 10-20% off the market data of 30% when I used a grid-based approximation. Hence, I chose to implement via the LSVM using the precise non-parametric regression estimation, as shown in Appendix 2.

(7) Sensitivity to the Monte Carlo simulation implementation (discretization method, discretization interval, and number of simulation paths):

For simulation of the underlying asset and volatility processes, I relied on a Monte Carlo simulation. There are a number of ways to implement such simulation, and I tested the model's sensitivity to the key components of a Monte Carlo simulation.

As discussed in Section 4.2 above, I tested the model 's sensitivity to a discretization scheme using both a regular Euler discretization scheme and a log-Euler discretization scheme. The latter resulted in a higher precision estimates.[24] This is also consistent with the academic and industry literature; see, for example, Andersen & Piterbarg (2010) and Glasserman (2004).

For simulation of stochastic processes (such as underlying asset (forward) and volatility processes in this case), in addition to the discretization scheme, one has to choose the discretization step. I tried a range of steps and settled on the value of 0.1 years (less than 3 calendar days). The smaller discretization steps (such as 0.05 and 0.01 years) resulted in sizably higher computational times and did not produce a particularly noticeable gain in the model precision.

As the number of simulation paths, I used 100,000 paths for the pricing of derivatives. A higher number of paths led to substantial increase in computation time without a corresponding gain in precision. For calibration of the leverage function, I used 7,000 paths. A higher number of paths (such as 10,000 - 30,000) did result in even better (more accurate) calibration to the market data (i.e., for the deep out-of-the-money and deep in-the-money options) but required prohibitively long computational time on my computer (at least a few hours, and even longer with a higher number of simulations). For an actual industry implementation, I recommend to increase a number of simulation paths ("particles") needed for the leverage function calibration to over 10,000 (and preferably as high as 30,000 or even higher, depending on the complexity (range) of the actual market data).

5.3 Stress Testing

As part of the sensitivity checks described above, I tried a few stress testing scenarios. The most important ones are described below:

---

[23] Such as dividing the range of all possible simulated forward values in 2000 equal size 'bins', of approx. size 2 each. See Appendix 2.

[24] By about 10%-15% in terms of the implied volatility (recovered from the LSVM-priced vanilla options) matching the market volatility (set at 30% for the model validation purposes).

- VoV set to 75%. This corresponds to a period of *high market volatility*. As discussed in Section 5.2, the LSVM performed fairly well under such stressed conditions. The model was able to calibrate to the stressed market conditions, and priced the derivatives reasonably well.
- *High correlation* between the forward and volatility process. Under normal market conditions, such correlation is generally not expected to exceed certain values (such as 25%). In times of the market crises, the correlation between different types of assets (and also volatility clustering and correlation between underlying assets and their volatilities) tends to increase.[25] I tested the model's performance at high correlation levels, including correlation of 100%. As discussed above, the LSVM performed reasonably well: i.e., it was able to calibrate to such stressed market data with a reasonable degree of precision.

[25] See, for example, discussion in Sandoval & Franca (2010).

6 Comparison with Benchmark Models

6.1 Comparison to a 'Pure' Stochastic Volatility Model (SVM)

Below, I describe the comparison of the key LSVM outputs with a closely competing, a predecessor model of "pure" stochastic volatility ('pure' SVM). Unlike the LSVM which allows one to calibrate to the market conditions and data directly via a leverage function, a 'pure' SVM would need to be calibrated in a more 'traditional' way via fitting a number of parameters to the observed derivatives' prices.

I implemented a 'pure' SVM using all the same assumptions and inputs as the LSVM with one difference: instead of calibrating the leverage function (as it is done in the LSVM), I effectively removed the local volatility (leverage) calibration mechanism from the LSVM by setting the leverage function to a constant of 1 in the 'pure' SVM model.

Below, I describe the performance of the 'pure' SVM and compare it to the findings for the LSVM (as described in Section 5.2 above).

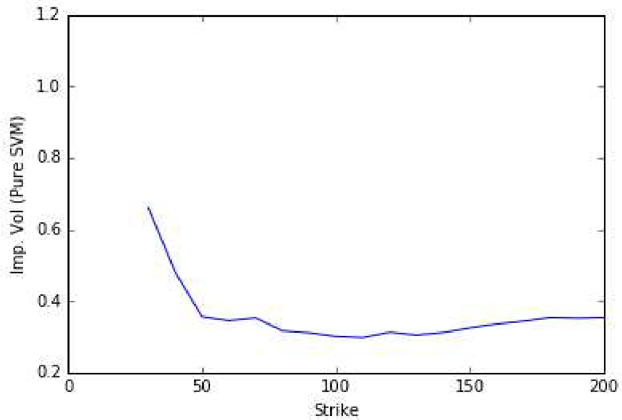
(1) Performance of the LSVM vs 'pure' SVM in terms of calibrating to (and recovering) the market data for different levels of option 'moneyness' (strikes):

As described in Section 5.2 (1), the LSVM performed reasonably well in recovering market data for a wide range of vanilla option strike levels, with a possible exception of very deep in-the-money and very deep out-of-the-money options. Nonetheless, even for such options, the LSVM recovered the market (assumed to be flat) volatility surface within the 10% precision level.[26] I performed the same type of the test for the 'pure' SVM model, and the results are shown below in Figure 4. Instead of the expected flat (or nearly flat, within some reasonable approximation bound) volatility surface assumed to be inherent in the market (at 30% level), the 'pure' SVM resulted in a pronounced volatility smile. For example, at strikes of about 40-50, the recovered (from the 'pure' SVM) volatility is 40% and higher as compared to the actual market volatility of 30%.

Thus, the LSVM shows a clearly superior performance as compared to the 'pure' SVM model.

[26] See Section 5.2 (1) and Appendix 2. For example, at strike 150, the LVSM recovered the market implied volatility to be at about 27% vs the actual value of 30%.

Figure 4

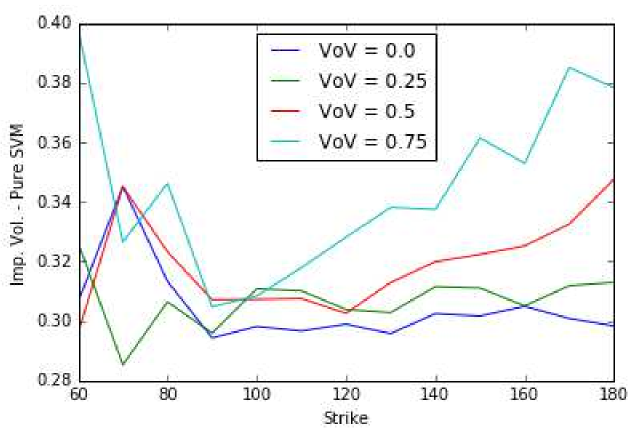


(2) Performance of the LSVM vs 'pure' SVM at different levels of market uncertainty (as proxied by different levels of volatility-of-volatility):

As described in Section 5.2 (3), the LSVM performed well under different levels of market uncertainty (i.e., as proxied by different levels of VoV). In contrast to that, and as shown in Figure 5 below, as the uncertainty (stress market conditions) became more acute, the 'pure' SVM performed worse and worse in its ability calibrate to the actual market data. Specifically, as the VoV levels increased, the more pronounced the volatility skew generated by the 'pure' SVM became; see Figure 5 below.[27]

Thus, again, the LSVM shows a clearly superior performance as compared to the 'pure' SVM model.

Figure 5



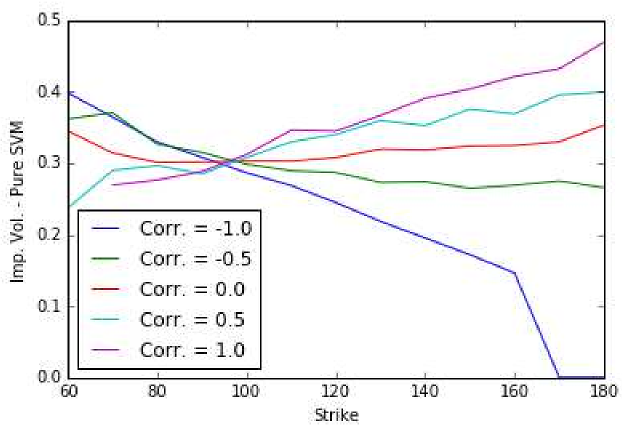
(3) Performance of the LSVM vs 'pure' SVM at different levels of correlation between the underlying asset and its volatility processes

[27] As a reminder, the actual (assumed) implied market surface is flat, so the more pronounced the volatility skew resulting from the model, the less accurate and less applicable the model is.

Finally, let us compare the LSVM performance at different correlation levels (ρ) (see Section 5.2 (4)) to the corresponding performance of the 'pure' SVM. Unlike the LSVM, which performed well in adjusting to different correlation levels, the 'pure' SVM failed to recover the observed flat volatility surface, and this failure became even more pronounced at higher correlation levels. See Figure 6 below.

Thus, the LSVM performed better than the 'pure' SVM model in this key model aspect as well.

Figure 6



(4) Performance of the LSVM vs 'pure' SVM in pricing a forward-starting call spread:

I used both LSVM and 'pure' SVM to price the forward-starting swap described in Section 5.1 above. I tried 10 different simulations (batch samples) and estimated 'batched' mean and variance of such estimation, as shown below:

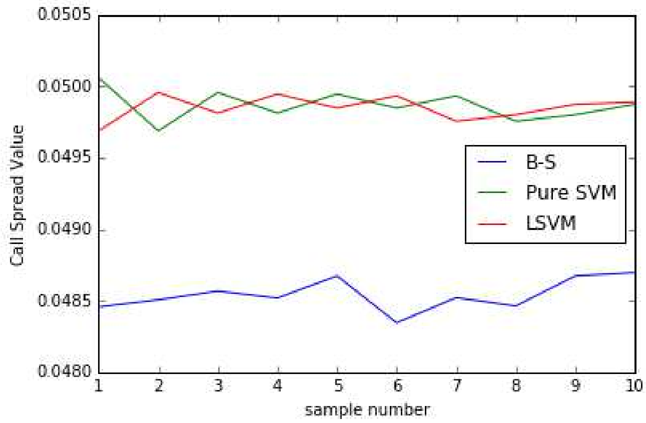
Pure SVM estimation variance:

1.07751885818e-09

LSVM estimation variance:

6.84560071368e-10

Figure 7



As Figure 7 above shows, call spread prices are fairly similar [in the same order of magnitude] between the 'pure' SVM model and the LSVM model. However, LSVM model results in a more stable estimate (note that the LSVM estimation variance is sizably lower as compared to the pure SVM estimation variance), and a more realistic pattern of the forward volatility skew (as LSVM is better suited to capture the actual forward volatility than a simpler (and non-calibrated) pure SVM).

6.2 Comparison to Black-Scholes (BS) Model

Finally, in addition to pricing the forward-starting call spread with the 'pure' SVM and LV SM as discussed above, I priced it using a much simpler Black-Scholes (BS) model. I also priced it using the BS in 10 different simulations (batch samples) and estimated 'batched' mean and variance of such estimation, as shown below and in Figure 7 above:

BS estimation variance:

1.13515827006e-09

LSVM estimation variance:

6.84560071368e-10

Just as in the case of the LSVM vs 'pure' SVM, the BS model resulted in a higher estimation variance as compared to the LSVM. Also, the BS performance is substantially worse than either 'pure' SVM or LSVM. Specifically, as Figure 7 above shows, the BS suffers from a downward bias: it consistently underestimated the forward-starting call spread.[28]

[28] As discussed with Prof. Guyon and the TA in the "Computational Methods for Finance", the PDE-based estimation would result in the forward-starting call spread price of about 0.05. As Appendix 2 shows, the LVSM resulted in a precise estimate of 0.050003; the BS model consistently underestimated the price to be around 0.048696. (The 'pure' SVM resulted in the price 0.049731 in one of the simulations I tried, see Appendix 2).

6.3 Conclusion Regarding the LSVM Performance as Compared to the Benchmark Models

Thus, the calibrated LSVM is much better suited to capture an actual market dynamics as compared to simpler models such as BS model and SVM. Also, unlike simpler, "static" models such as BS model and even a 'pure' SVM, the LSVM is much more flexible: it can be easily modified to price more complex (path-dependent) derivatives i.e., *at varying (and even stressed) market conditions*.

7 Model Strengths and Weaknesses

This section summarizes the strengths and weaknesses of the LSVM (particularly, in comparison to simpler benchmark models such as BS model and 'pure' SVM).

7.1 Strengths

- As discussed in Sections 5 and 6, the LSVM was able to calibrate to the various market data quite well, even under (simulated) stressed market conditions.
- The LSVM was able to price such relatively complex instruments as forward-starting call spreads well, and with a clearly higher precision and/or efficiency than simpler models such as 'pure' SVM and especially BS model.
- The LSVM offers ***a unified volatility modeling mechanism***, a clear improvement to both of its predecessors, the 'pure' SVM and 'pure' LVM. As discussed in Section 3 above, a standard, 'pure' SVM can handle only a *finite* number of parameters (e.g., volatility-of-volatility (VoV), volatility mean reversion etc), and, thus, by definition, cannot be calibrated to the whole implied volatility surface. On the other hand, a standard, 'pure' LVM can calibrate to the observed volatility surface but cannot consistently model the observed properties of volatility such as mean reversion and VoV. ***The LSVM overcomes both the weaknesses of SVMs and LVMs, and, thus, at least theoretically, offers a superior option pricing and risk management tool***.
- Although seemingly complex, due to the developments in the statistical physics (the particle method), the LSVM can now be *relatively* easily (and efficiently) implemented,[29] and offers a stable, *precise* calibration to the market data.

7.2 Weaknesses

- The LSVM is a very new model. My understanding is that it has been first comprehensibly introduced by Guyon & Henry-Labordere in 2014.[30] As Guyon & Henry-Labordere (2014) note, the uniqueness and existence of the LSVM solution is not guaranteed (i.e., it has not been fully proven; only some "partial results" exist).[31] Thus, a more extensive theoretical and empirical research is needed for this model. Similarly, a much more extensive model validation effort is required (i.e., in the actual industry settings) before any firm conclusions are drawn regarding the validity of this model.
- The LSVM is more complex and likely harder to implement than all of its predecessors, such as stochastic and local volatility models. Although all models suffer from some subjectivity, there are a number of parameters and techniques in the LSVM (e.g., choice of kernel) where a model developer has to make a certain modeling choice and then also apply substantial fine-tuning/

---

[29] At least, in industry settings.

[30] I was first introduced to this model thanks to Prof. Guyon's class at Courant last semester, "Computational Methods for Finance". (Note that portions of the code shown in Appendix 2 were developed by me during the last semester in that class.)

[31] See p. 274 in Guyon & Henry-Labordere (2014): "it is not clear at all whether an LSVM exists for a given arbitrary arbitrage-free implied volatility surface: some smiles may not be attainable by the model. […] The problem of deriving the set of stochastic volatility parameters for which the LSVM does exist for a given market smile, is very challenging and open." The authors do note that "a partial result exists", particularly "in the case of suitably regularized initial conditions…" and "if the volatility-of-volatility is small enough".

calibration effort. (However, as noted in the "Strengths" sub-section above, such effort may be well worth it; the LSVM – unlike SVMs and LVMs – offer much more superior theoretical and calibration (empirical) possibilities.)

- The LSVM is computationally involved. I believe it is best to be fully implemented, tested / validated in the industry settings, using fairly computationally powerful resources. For example, I was able to successfully implement the model only with a fairly low (7,000) number of simulation paths ("particles") used for the local volatility calibration. Otherwise, the implementation proved to be too computationally burdensome. Some preliminary testing I performed indicated that a substantially higher number of simulation paths ("particles") used for the local volatility calibration would lead to even more superior, more precise and accurate calibration results.[32]

8 Conclusions and Recommendations

In this section, I offer some conclusions and recommendations in the area of further LSVM validation.

- Although I was able to validate the LSVM in a number of key areas, further on-going work, model monitoring and validation are recommended. Moreover, as I was the one who effectively / developed the model, *an independent third-party should have validated the model* (not just myself). (Such independence of the model validation process from the model development process is prescribed by the SR Letter 11-7 guidelines).
- As discussed above, the LSVM is a fairly new and complex model, and its implementation would be best done *in the actual industry settings* (I believe this could allow the LSVM to demonstrate more accurate/precise results).
- In this model validation exercise, I primarily just assumed certain 'market' data (such as flat market implied volatility surface etc). *This model will need to be tested / validated further with a wider range (i.e., actually observed) market data*.

In summary, further on-going model development, testing, validation and monitoring is recommended, i.e., in the actual industry settings and with a wider range of the market data.

[32] As discussed in Section 5, for example, the fact that the LSVM priced deep in- and out-of-the-money vanilla options only with about 10% precision is due to the low number of 'particles' I was able to efficiently implement on my personal laptop. Thus, the model is recommended to be implemented and fully validated in the actual industry settings.

Appendices

A1. Bibliography

Andersen, Leif B.G, & Piterbarg, Vladimir V. (2010). *Interest Rate Modeling. Volume I: Foundations and Vanilla Models*. Atlantic Financial Press.

Board of Governors of the Federal Reserve System, Office of the Comptroller of the Currency (OCC): ***SR Letter 11-7*** *(including its Attachment), Supervisory Guidance on Model Risk Management*. April 4, 2011. Online publication as distributed in class.

Brigo, Damiano, & Mercurio, Fabio (2006). *Interest Rate Models – Theory and Practice*. Springer Finance.

Glasserman, Paul (2004). *Monte Carlo Methods in Financial Engineering*. Springer (Applications of Mathematics / Stochastic Modelling and Applied Probability Series).

Guyon, Julien, & Henry-Labordere, Pierre (2014). *Nonlinear Option Pricing*. CRC Press (Chapman & Hall / CRC Financial Mathematics Series).

Hull, John C. (2008). *Options, Futures, and Other Derivatives* (7th Ed.). Pearson Prentice Hall.

Joshi, Mark S. (2008). *The Concepts and Practice of Mathematical Finance*. Cambridge University Press.

Sandoval, Leonidas Jr., & Franca, Italo De Paula (2010). *Correlation of Financial Markets in Times of Crisis*. Working Paper.

Wilmott, Paul (2006). *Paul Wilmott on Quantitative Finance*. John Wiley & Sons, Ltd.

A2. Source Code

See attached iPython Notebook (printed in PDF format).