

Automated Options Trading Using Machine Learning

Peter Anselmo and Karen Hovsepian* and Carlos Ulibarri and Michael Kozloski

*Department of Management, New Mexico Tech,
Socorro, NM 87801, U.S.A.*

We summarize an experimental study on the viability of several call option trading strategies that rely on our earlier work with machine-learning-based detection and prediction of heightened volatility periods. The proposed trading strategies make use of the connection between call options prices and volatility in the underlying.^{1,2} As part of these strategies, the trader would purchase call options as soon as a period of heightened volatility has been detected, and, alternatively, predicted. After that, the strategies suggest the ideal time to sell the previously purchased options contracts in order to maximize profit. The approach is evaluated with simulated option trading experiments, results of which show that the idea has some merit. Furthermore, the experiments provide an empirical basis for choosing the optimal strategy from among the strategies we considered.

1. Introduction

In this paper, we make use of our previous work applying Machine-Learning-based classification for detection/prediction of relative volatility clusters (RVCs).^{3,4} Our methodology consists of the following steps: first we fit a GARCH(1,1) model⁵ to the historical data and use the model to identify segments within the data with GARCH volatility levels significantly higher than the average GARCH volatility computed over the entire length of the historical data. Once these segments have been identified, the machine learning (ML) algorithm is trained on the basis of these segments.

Instead of the important task of forecasting an exact volatility,⁶⁻⁸ we use machine learning for pattern-recognition to either detect (the more naive framework) or predict (the less-naive framework) periods of heightened GARCH volatility, as data arrive in a simulated actual-time segments, using a sliding-windows method^a. We therefore conduct our experiments

*Corresponding author: karen@nmt.edu

^aIn a sliding-windows method, a window/segment of some size is continuously processed

with two distinct approaches to the problem of using pattern-recognition based on GARCH model's definition of relative volatility. The prediction approach consists of a 3-step process: prediction, identification, completion — when we determine that the volatility cluster has ended. The simpler detection approach consists of only the identification and completion steps. We should note that we do not formally control for jump processes, nor do we assess alternative (to GARCH(1,1)) model formulations.

2. RVC detection/prediction

In this section we describe the methodology for detecting and predicting relative volatility clusters, using machine-learning-based classification. In the case of detection, we partition the the universe of all time-series segments (of any length) into two categories/classes: **relatively volatile cluster** (RVC), and **relatively non-volatile** (RNV). Based on this assumption, when a classifier determines that a given time-series segment is **relatively volatile**, we can state that either the entire segment, or some part of it, is a RVC.

Prediction is slightly more complex. Instead of just two classes, we consider that every time-series segment is in one of three categories: **pre-[relatively] volatile** (PRV), **relatively volatile cluster** (RVC), and **non-prevolatile**(NPV), which is the class of segments that are neither PRV nor RVC. As before, when the classifier identifies a given segment as PRV, we have essentially an early warning that an RVC is coming sometime in the future.

The ML classification algorithm we employ is the current industry-standard *Support Vector Machines* (SVM) algorithm.⁹ SVM can be described as an algorithm that implements the Structural Risk Minimization¹⁰ statistical inductive principle, and is simultaneously connected to Tikhonov Regularization and ridge regression.¹¹ It has been shown to be a very robust algorithm with good generalization characteristics. A number of excellent tutorial/books on SVM exist, among which the tutorial by Burges (1998)¹² is quite helpful. Due to space limitation, we shall only touch upon the aspects of SVM most relevant to the discussion in the paper.

2.1. Setup of the SVM classification module

The task of the SVM algorithm — indeed any classification algorithm — is to compute a function that can estimate the class (output) of any given ob-

as it is shifted forward one frame/tick at time.

servation (input). As an inductive learning algorithm, SVM computes this classification function by ‘learning’ from an example set of input observations and their ‘actual’ or preferred outputs. An important part of using SVM is the process of selecting a suitable example set, also called a training set.

In the context of applying SVM to time-series, it is worth noting the following: that the input observations used by the out-of-box SVM learning algorithm need to be vectors of equal size. Therefore, all time-series segments used as SVM input observations are vectorized by making each tick an element in the vector — a time-series segment 20 ticks long is represented as a vector of 20 numbers.

For detailed description of the steps involved in producing the training dataset and training the SVM is given in our earlier papers.^{3,4} First step of the setup is fitting the GARCH(1,1) model to returns^b of the historical underlying price data and computing the conditional variances, which can be used to explore the cases when shocks have influenced other shocks by way of propagating volatility. Clusters of volatility are groups of highly interrelated shocks and they will show up as peaks in the conditional variances. It is then up to us to make a decision, using statistical inference tools and input domain knowledge, whether or not these peaks are significantly^c above the average conditional variance for the entire series.

Another user chosen parameter is the **minimum required length** of every segment that is to be added to the training dataset. A conditional variance peak that is too short in length can be deemed to be a jump, rather than a volatility cluster, which are assumed by most experts to differ from short jumps. Thus, the return segments that correspond to conditional variance peaks of sufficient height and sufficient length are labeled as RVC.

For detection, all sufficiently long returns sections with conditional variances within the norm are labeled as RNV. For prediction, however, we have the special PRV class. For this class we choose returns sections that start some time (**prediction horizon**) before every RVC segment and end right before the start of that RVC segment. The prediction horizon is another parameter that the user can modify to vary the nature of prediction. There is, therefore, a 1-1 correspondence between the number of RVC and PRV examples in the prediction dataset. The last ‘neither’ class examples are chosen to be sufficiently long returns sections that have no overlap with

^bThe returns are computed as log of the ratio of the current price to previous price.

^cDuring the significance tests on the conditional variance peaks, we set the significance threshold $\alpha = 0.05$

any of the RVC or PRV segments.

The example time-series segments, belonging to any of the classes, that we selected during the previous steps could well have varying lengths, and, hence, the vectors representing these segments could also be of varying lengths. Due to the requirement, mentioned earlier, that all SVM training observations be vectors of equal length, we find that the example time-series segments cannot be used as is, and either the data or the algorithm must be modified to make it work.

We find that a simple and effective solution is to modify the data by transforming all example segments (returns) into the frequency domain, via the *periodogram* function. The result is that in the frequency domain all the segments are of equal length, equal to the number of frequency components, which means that all vectors representing these segments are also equally sized. After this transformation, the training dataset is used to train the SVM to classify any given periodogram-transformed returns segment. In addition to giving the predicted class of a new segment, the SVM model can estimate the posterior probability of every class in the problem given the new observation, $p(\hat{y} = c|x^*)$, where x^* is the new observation, \hat{y} is its predicted class, and c is the index scalar for one of the classes in the problem. The predicted class of the new segment is the one with the highest posterior probability, $\hat{y} = \arg \max_c (p(\hat{y} = c|x^*))$.

2.2. Setup for applying the SVM model

The setup for detection or prediction of RVCs is illustrated in Fig. 1. The first step is combining the newly available price tick with past price ticks to construct a window of some size. In the next two steps we compute the returns of this price window, using the same formula as the one used in the classification module setup, and compute the periodogram of the returns. The periodogram is then vectorized and fed to the trained SVM model to find its class. In the case of simple detection, the model outputs

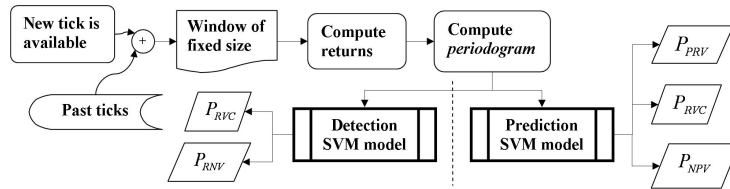


Fig. 1. A simple RVC detection system setup. The optimally trained SVM model is already defined.

two posterior probabilities: probability that the class of the periodogram-transformed segment is RVC, represented in the condensed form as P_{RVC} , and a probability that the class is RNV, P_{RNV} . In the case of prediction, we have three posterior probabilities: that the window is pre-[relatively] volatile, P_{PRV} , relatively volatile, P_{RVC} , and neither, P_{NPV} . In the next subsection we present the strategy for trading options that makes direct use of these posterior probabilities.

3. Trading Strategies

Our first trading strategy makes use of simpler detection framework, while the second one uses the three-class prediction framework briefly described above. The strategy that uses detection is summarized in the diagram on Fig. 2. As soon as we have detected an RVC event with the probability that it is indeed an RVC event greater than some cutoff p' , we buy a single call option contract at the top-of-book ask price. We hold the contracts until the moment when we detect a RNV period, again with the probability above some cutoff p'' . At that point we sell all options at the top-of-book bid price. The cutoff constants p' and p'' allow us to control the responsiveness of the strategy to the classification outputs.

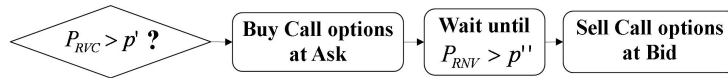


Fig. 2. Diagram outlining the options trading strategy based on detection.

The strategy based on prediction is outlined in the diagram on Fig. 3. Here we buy options when we have identified a PRV period, as before with some level of certainty $P_{PRV} > p'''$. From this point on, there are two choices for the point of sale. Under option A, we sell when we detect an RVC period with probability above cutoff p'''' , while under option B we hold on to the options until we detect the ‘other’ class NPV, with probability above cutoff p'''' .

4. Data

To test the effectiveness of our strategies we performed a paper trading simulation using USD/EUR price tick data, spanning the month of March,

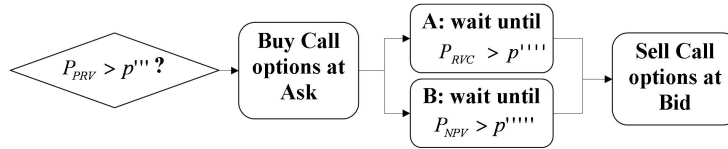


Fig. 3. Diagram outlining the options trading strategy based on prediction.

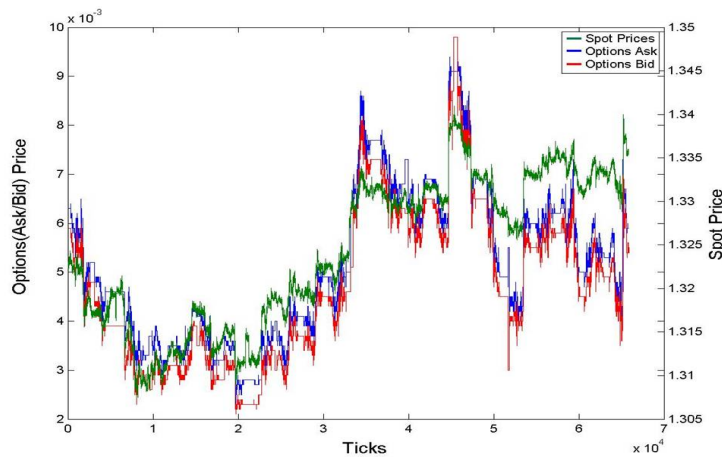


Fig. 4. Plot of the 30 second interval spot prices and ask/bid options prices.

2007^d, and corresponding call option quote data, with strike price of \$1.35 and expiration of May, 2007, The simulated call option trades were made during the period of March 11th to March 30th, 2007 (weeks 3,4, and 5). Volatility clusters were detected/predicted within the USD/EUR underlying price data for the same period. The SVM models, for prediction as well as for detection, were trained using the underlying prices during the period of March 1st to March 9th (end of week 1 and week 2) — prior to start of trading. Prior to their use, both options and spot price data were pre-processed to create fixed interval data at intervals of 15, 30, and 60 (1 min) seconds. Figure 4 contains the plots of the 30 second interval option prices (bid and ask) and spot prices for entire month of March, 2007.

^dThe data were purchased from Olsen Financial Technologies (<http://www.olsendata.com>).

5. Simulation Results

The design of the simulation experiments was straightforward. Upon training the detection and prediction SVM models on the first two weeks of underlying price data, we followed the setup outlined in section 2.2 using sliding-windows to simulate the process of adding a new price to past prices to construct a window (cf. Fig. 1). To explore the effect of different window sizes on the outcome of the trading, we let the window sizes vary from 5 to 30 ticks, in increments of 5 ticks.

The simulated trades followed the strategies outlined in section 3. The sliding-windows simulation and the simulated trades spanned the period of March 11th to March 30th (weeks 3,4, and 5).

As with the window sizes, we wanted to vary the interval frequency of the data, to see if it has any impact on the efficacy of our approach. We had 3 interval frequencies to experiment with: 15 seconds, 30 seconds, and 60 seconds. Together with the 6 windows size choices, we had a 3×6 experiment grid. Furthermore, for each of the experiments in this grid, we tried several cutoff probabilities p' , p'' , p''' , p'''' , and p''''' . All six probability cutoffs were allowed to vary from 0.5, smallest probability that is still higher than the probabilities of the other classes, to 0.99, almost certain.

The profit at each selling point was computed simply as $Ns - \sum_i^N b_i$, where N is the number of previously purchased option contracts at the selling point, s is the bid price of the option at the selling point, and b_i is the ask price at the point of the i^{th} purchase. The total profit over the entire three week period is that the sum of all profits from sales. We assume that the market is sufficiently liquid to enable the sell transactions at the bid price whenever we want to sell.

In Table 1, we list the results of our simulation experiments, comparing the results across interval scales, window sizes, and probability cutoffs. The table clearly shows that most profit is achieved when we trade based on prediction of relative volatility clusters by the SVM model. The highest profit of \$11.61 was registered in the 15 second interval data, with windows 20 ticks long (5 minutes), using the prediction strategy under option A. It is important to note that the prediction strategy under option B does not result in equally high profits. This is in line with trading logic. At the time when a segment of class PNV has been identified, the market has had ample time to react to the heightened volatility and correct itself. In contrast, at the moment when we detect a volatility cluster the market has just been hit by the effect of heightened volatility resulting in higher options prices, but it has not had enough time to correct itself. The table also shows

Table 1. Highest total profits, in dollars, (negative means losses) for options trades over the entire space of probability cutoffs, compared across all intervals and window sizes.

Intervals		Window Sizes					
		5	10	15	20	25	30
Detect	15	-3.3123	-2.7376	-2.234	-1.8367	-1.5575	-1.18
	30	-1.9353	-1.6313	-1.3645	-1.2338	-1.1648	-0.9222
	60	-0.584	-0.1584	-0.1864	<u>2.9315</u>	<u>2.0008</u>	<u>2.172</u>
Predict A	15	<u>0.0003</u>	<u>0.7376</u>	<u>9.1464</u>	<u>11.611</u>	<u>1.0271</u>	<u>1.1925</u>
	30	-0.7076	<u>1.7449</u>	<u>1.4761</u>	<u>3.8512</u>	<u>0.4974</u>	<u>0.6378</u>
	60	-0.3234	-0.2123	<u>2.2343</u>	<u>2.9593</u>	<u>0.8253</u>	<u>0.4321</u>
Predict B	15	<u>0.0002</u>	-0.0017	-0.0038	-0.0008	-0.0023	-0.0068
	30	-0.1831	-0.0051	-0.0207	-0.0088	<u>2.7417</u>	<u>0.1304</u>
	60	-0.3212	-0.0431	-0.09321	-0.0004	-0.0074	-0.003

Note: Positive profits are underlined.

that using detection alone achieves the lowest comparable profit. In Fig. 5 we provide a matrix plot of total profits using prediction with option A, graphed against the probability cutoffs, spanning select interval frequencies and window sizes. The graph for prediction shows that the best trading strategy is to buy as soon as we are marginally sure that a pre-volatile segment has been detected ($P_{PRV} > 0.5$), and sell when we are almost certain that a volatile tick has been detected ($P_{RVC} > 0.99$). For plots of the total profits using the other strategies and spanning more window sizes and more interval frequencies, we refer the reader to the following web address: <http://www.nmt.edu/~karen/volatilityanalyst/>.

6. Conclusions

This initial experiment shows promise for generation of what are effectively arbitrage profits in a sufficiently-liquid call option market. We will continue to explore more complex trading scenarios and strategies.

References

1. F. Black and M. S. Scholes, *Journal of Political Economy* **81**, 637 (1973).
2. J. Hull and A. White, *The Journal of Finance* **42**, 281 (1987).
3. K. Hovsepian, P. Anselmo and S. Mazumdar, *Support Vector Classifier Approach for Detection of Clustered Volatility in Dynamic Time-Series*, tech.

- rep., New Mexico Tech (September 2007), http://www.nmt.edu/~karen/volatilityanalyst/HovAnsMaz_VolDetection.pdf.
4. K. Hovsepian, P. Anselmo and S. Mazumdar, Detection and prediction of relative clustered volatility in financial markets, in *CIEF 2005*, (Salt Lake City, 2005).
 5. T. Bollerslev, *Journal of Econometrics* **31**, 307 (1986).
 6. T. G. Andersen, T. Bollerslev, F. X. Diebold and P. Labys, *Econometrica* **71**, 579 (March, 2003).
 7. T. G. Andersen, T. Bollerslev, P. F. Christoffersen and F. X. Diebold, *Volatility and Correlation Forecasting*, Handbook of Economic Forecasting Vol. 1 (Elsevier, 2006), ch. 15, pp. 777–878.
 8. S.-H. Poon and C. W. J. Granger, *Journal of Economic Literature* **41**, 478(June 2003).
 9. C. Cortes and V. N. Vapnik, *Machine Learning* **20**, 273 (1995).
 10. V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer-Verlag, 1995).
 11. A. N. Tikhonov, *Dokl. Akad. Nauk.* **39**, 195 (1943).
 12. C. J. Burges, *Data Mining and Knowledge Discovery* **2**, 121 (1998).

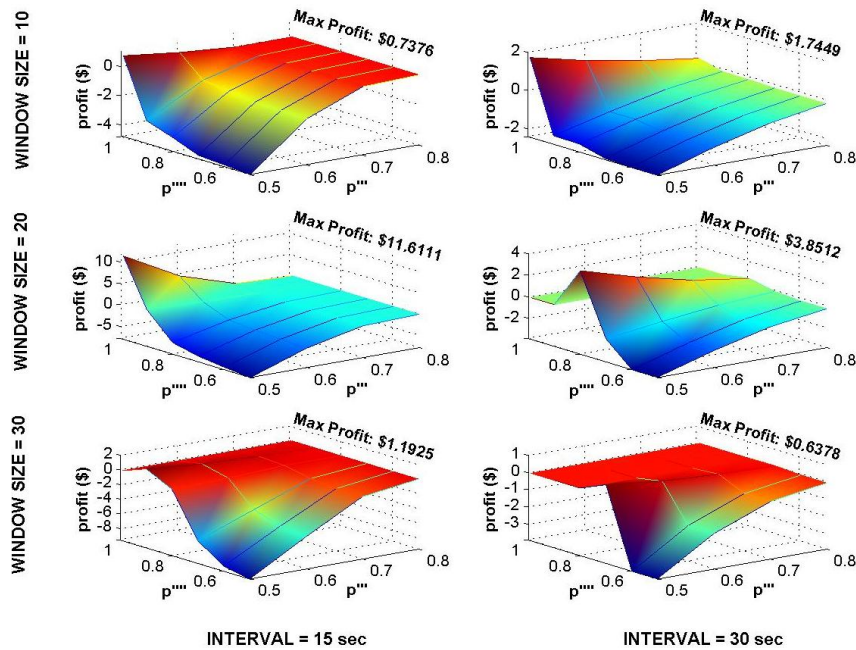


Fig. 5. Matrix plot of total profit using prediction with selling option A. Plots span window sizes of 10, 20 and 30 ticks and intervals of 15 and 30 seconds. Each individual plot graphs the total profit as the Z coordinate and the probability cutoffs p''' and p'''' as X and Y coordinates, respectively