Detection and Prediction of Relative Clustered Volatility in Financial Markets

Karen Hovsepian Peter C. Anselmo Subhasish Mazumdar
[1] New Mexico Institute of Mining and Technology Socorro, New Mexico 87801 USA

Abstract

We present a methodology for analysis of differences in variation levels in time series data. Our focus is on automatic detection, as well as prediction, of periods of relatively increased volatility in the time series data. We accomplish this using a synthesis of three methods from the fields of computer science – *support vector classifiers* (SVC), statistics – *generalized autoregressive conditional heteroskedasticity* (GARCH), and signal analysis – *periodogram*. The outcome is a multi-layer tool, which has shown the capability to accomplish detection and prediction tasks with tractable results. An example of identification and prediction of volatility clusters in currency-exchange time returns is summarized.

Keywords: volatility, machine learning, SVC

1. Introduction

Our overall objective was creation of a methodology for swift detection and prediction of relative volatility clusters in time-series data. The desired outcome was a tool that might augment and enhance the analysis of relative clustered volatility in dynamic time-series and facilitate experts' decision making in academic as well as industrial settings.

The basis of our approach is the assumption that relative volatility clusters have fundamentally different patterns from sections with relatively low levels of volatility. The term 'patterns' is used here to describe the elements of the hidden, or latent, structure and composition of objects. This lends to virtual partitioning of the universe of all time-series segments into two classes, relatively volatile (RV) and relatively nonvolatile (NV). This approach sets the stage for deployment of support vector classifiers (SVC), which is a supervised learning technique for binary classification [1]. Using SVC we are able to categorize a given section of financial time-series data into one of the two mentioned classes: RV and NV. More details on the algorithms and the approach in general can be obtained in [7].

2. Algorithms

SVC is a member of the more extensive machine-learning technique known as support vector machines (SVM). Like SVC, SVM has derivations that tackle function estimation/regression and density estimation. The general SVM technique and all its derivations have been applied to numerous problems and have, in many cases, systematically performed equal to or better than other pattern recognition and data analysis techniques [2].

The simplest description of SVC is that it 'learns' to classify observations into one of two classes from prior pre-classified examples. This characteristic puts this method in the category of *supervised* statistical/machine learning methods, which follow a similar learning-by-examples framework. As such, the process consists of two phases: training and testing. In the training phase, the algorithm is presented with a data set of observations (in our case these are time-series segments of currency-exchange return data), which have been pre-classified into one of the two categories of interest [1][3][4], which, in this case, we have denoted as RV and NV.

SVC requires a training set of observations for which the class labels (in this case, RV and NV) have been pre-determined. Unlike many other SVC applications, where real-world labels/classes are obvious and/or easy to attain, it is not always obvious whether a given time-series segment is relatively volatile. This first step of the process – initial identification of volatility clusters – was done using a GARCH [5][6].

GARCH(p,q) defines the conditional variance at time t, $\sigma(t)^2$, as a function of p past conditional variances $\sigma(t-i)^2$ and q past squared shocks/changes $\varepsilon(t-i)^2$, called innovations:

$$\begin{aligned}\sigma(t)^2 &= \beta_0 + \beta_1\sigma(t-1)^2 + \ldots + \beta_p\sigma(t-p)^2 \\ &\quad + \gamma_1\varepsilon(t-1)^2 + \ldots + \gamma_q\varepsilon(t-q)^2\end{aligned} \quad (1)$$

Changes in the conditional variance $\sigma(t)^2$ are compared with an overall segment variance using a χ^2 test. Statistically significant peaks in segments of sufficient length (in this example, the minimum length required for designation as RV was 10 observations) constitute a volatility cluster. We select examples of relative volatility segments and relatively nonvolatile segments from within any time-series, to define the two classes for the SVC training phase.

A standard SVC implementation requires the input observations to be vectors of one common size, with each component in the vectors measuring the value of the corresponding feature. Thus, time-series segments are input into SVC's training algorithms as vectors, where each tick in the series is a component in the vectors. This, however, poses a problem in the analysis of financial (and, in fact, most) time series because there is no reason to expect relatively volatile and nonvolatile segments to be of the same length.

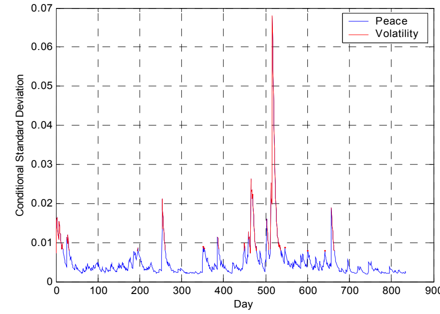
To overcome this issue, the second stop of our process is utilization of the periodogram signal processing tool. Once GARCH has provided relatively volatile and nonvolatile segments of arbitrary length, they are input to a periodogram, which returns the power spectrum density estimate (PSDE) for each segment. We then substitute the original segments with their corresponding PSDEs, which solves the vector dimensionality issue. In addition to solving the problem of unequal segments, computing the PSDEs transforms the data from the time domain into the domain of global structural features [7].

3. Implementation

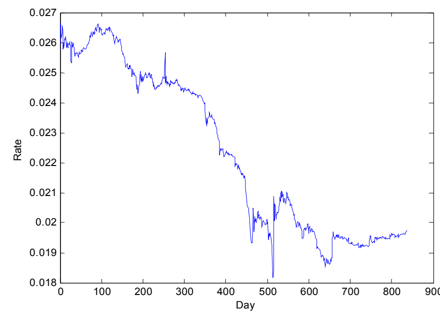
In our implementation, we started with raw daily currency-exchange time series data, obtained from the *Pacific Exchange Rate Service* [8]. The first phase of our approach is the use of GARCH to select the RV and NV examples. This is the most tedious part of the method due to the trial-and-error nature of fitting GARCH models. Human expertise in this phase is necessary in most cases.

The data were foreign exchange daily closing rates for around 70 currencies [8]. We deleted series that did not contain RV segments. In the end, we had 45 time-series, each spanning the available time-line of the particular currency.

After fitting the GARCH model and obtaining the conditional variances, we selected the sections that correspond to RV and NV segments. Fig. 1 illustrates this phase on the Philippines' peso to US Dollar raw time-series data (Fig. 1.a) and identified volatility segments – in red – over time (Fig. 1.b).



1.a



1.b

Fig. 1: Plots of the price data and conditional standard deviations with highlighted RV clusters for PHP/USD over 860 daily observations.

To standardize the lengths of these segments we compute their PSDEs. Fig. 2 plots the PSDEs for the segments shown in Fig. 1.b.

The third and final phase is the training of the SVC decision function using the training data obtained in the previous two steps (GARCH + PSDEs). It is worthwhile to note that in many cases a NV segment from one currency was in fact more volatile than a RV segment of another. This fact illustrates the innate complexity of defining a general approach for identification and prediction of volatility clusters. To counter the possible bias in the training/testing phase of SVC due to differences in base volatility levels of the segments, we added a new feature to each vector, measuring the unconditional variance of the entire time-series that each individual segment is from. Once training was completed, we tested the accuracy of the trained decision function on a dataset that also contains observations with pre-determined classes. We "tuned" the SVC parameters[1] and continued re-training and re-testing until the best fit between predicted and actual RV and NV segments was attained. This iterative process provides a decision function that is set to be used for identification of past RV segments and for detection of present and about-to-occur clusters in a real-time context.

[1] Kernel function and overfitting control parameter C.

It is paramount to note that in the application phase our methodology is completely automated. A computer simply needs to take a time-series segment under question and classify it using the optimal decision function. For the task of identification of past RV segments one can input past segments to find out whether they are RV.

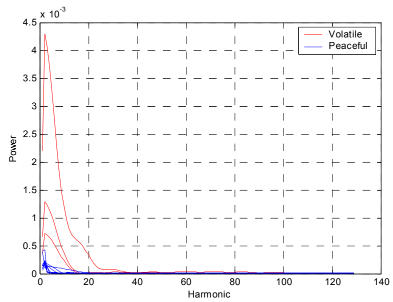


Fig. 2: PSDEs of the conditional standard deviations.

For the task of detection of RV in a simulated real-time context, we propose the following scenario. The computer adds each new tick – in this case, a daily return observation – to the end of a window of some (variable) size. This window is then passed on to our method and a decision on its class is obtained. If this window is classified as RV, then we may have effectively identified the beginning of a RV segment as it occurs. As we describe below, the majority of RV segments we tested with such rolling window experiments were caught on the first observation of the RV segments.

4. Experimental Results

After GARCH and PSDEs steps, we had obtained 1483 observations of segments of common length, around 824 of which were relative volatility segments and 659 were relatively nonvolatile. In order to assess the accuracy of our three-step methodology, four fifths of the entire dataset was randomly allocated for training and the remaining one fifth was allocated for testing. The highest accuracy of classifications on the testing set was 99.5%. The optimal parameters for the SVC were C: 4096 and Kernel: RBF|s = 8192.

False positives (missed relatively nonvolatile segments) and false negatives (missed reltively volatile segments) occurred in 0.07% and 0.04% of the segments, respectively. This means only 0.04% of the relatively volatile segments in the test examples were not classified correctly.

Another set of experiments was performed in which, rather than objectively testing the performance of our approach on a testing set produced the same way as the training set, we generated random segments, varying in lengths, from randomly chosen time series. The classes of these segments were then decided by the optimal SVC decision function. The purpose of these experiments was for visual/subjective validation of the method. Fig. 3 lists the results of experiments performed on 2 of the time-series. The plots display the original time series, with the randomly chosen segments color-coded according to the class, as decided by the SVC.

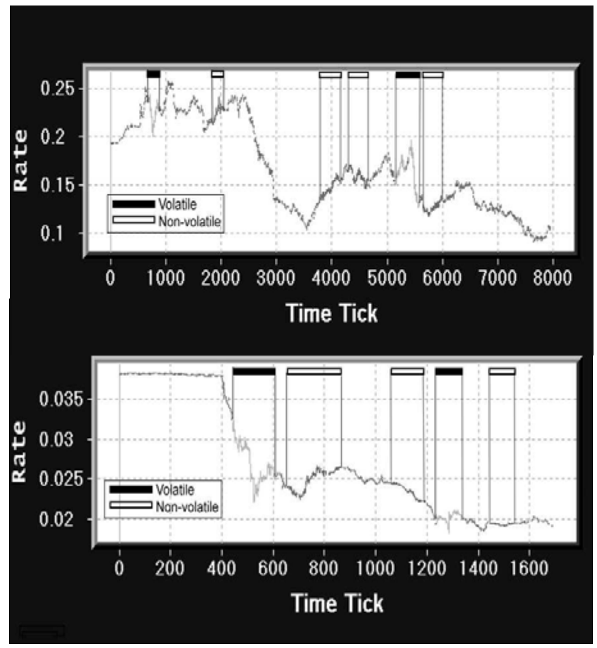


Fig. 3: Random segments classified as RV or NV.

While the above results indicate the ability of SVC to accurately classify relatively volatile segments, we were more curious with how the SVC would fare when only a part of a volatility segment was available as in the case of detecting presently occurring clusters in a real-time environment.

To test this we performed a series of rolling window experiments on a set of known RV segments. We let a window of fixed size begin a few ticks before each RV segment and classified it as it rolled over the boundary and into the RV segment. This experiment was repeated on various window sizes in order to empirically determine the optimal size for this overall dataset. Fig. 4 contains the histograms of the detection delays for several sizes. As the figure demonstrates, in many cases our implementation detected the cluster the moment the window passed the boundary (delay of 1 observation or tick). The experiments showed that the optimal window size for this application was *10*. In addition they validated SVC as truly capable of recovering complex patterns, even when statistical information content is relatively scarce.

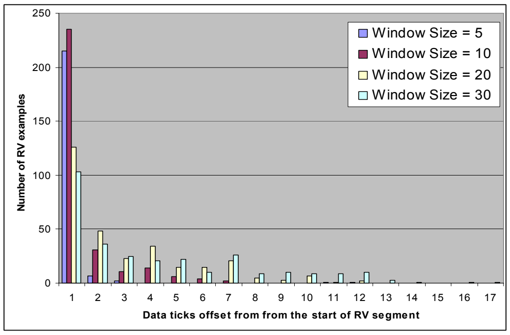


Fig. 4: Detection delays for rolling window experiments with several window sizes.

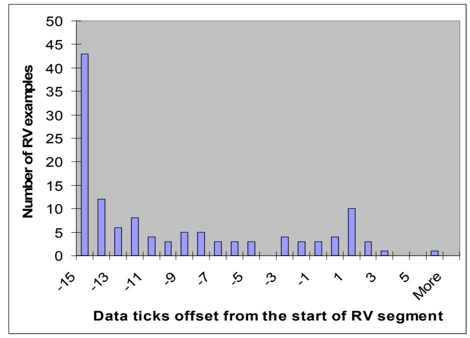


Fig. 5: Prediction delays for rolling window experiments for window sizes equal to 10.

So far we have shown that our approach can be applied to accurately identify volatile clusters that have occurred in the past and to swiftly detect volatility periods occurring at the present moment, which is highly relevant in financial practice. The next progression is modifying this approach to the task of predicting volatility clusters. The advantage of this methodology is that extension to prediction can be accomplished with relative ease. The only difference between detection and prediction – as described in the previous section – is the choice of the training/testing data sets. Rather than choosing examples of relatively volatile segments – for the first class – we choose suitable and representative pre-volatility segments, which we hypothesized to contain patterns differentiating segments occurring right before relative volatility bursts from other segments. For the second class, we found that best choice was a dataset of time-series segments from currencies that do not contain relative volatility clusters at all. Using this datasets, the accuracy of the selected SVC decision function for prediction of volatility bursts in a simulated real-time basis reached 98% (the optimal parameters for the SVC were C: 1024; Kernel: RBF|s = 542488).

Finally, we replicate the sliding windows experiments for the prediction task. Fig. 5 is the histogram of prediction delays. Negative delays signal the successful prediction of the cluster that is about to occur.

5. Conclusions

In this report we have described how we use SVC, together with other powerful analytical tools, to provide an efficient and highly accurate solution to a dynamic, complex computational problem.

Several extensions are possible and indeed necessary to make this approach more useful in the real-time context.

The first is the ability to detect more than just one level of volatility. The use of multiple SVCs appears as the simplest way to accomplish this. Several specifications for multiple SVCs have been proposed [9]: the most notable among them are the 1-vs-1, 1-vs-all and DAGSVM methods. We believe that our approach can be easily extended to handle multiple classes: by defining training sets with segments from the categories or classes of interest.

References

[1] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[2] http://www.clopinet.com/isabelle/Projects/SVM.

[3] http://www.kernel-machines.org/

[4] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

[5] R. F. Engle, "Autoregressive conditional heteroskedasticity with estimates of the variance of united kingdom inflation," *Econometrica*, vol. 50, pp. 987-1007, 1982.

[6] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, pp. 307-327, 1986.

[7] K. Hovsepian, P. Anselmo, and S. Mazumdar, "Support Vector Classifier Approach for Detection of Clustered Volatility in Dynamic Time-Series," New Mexico Tech, Tech. Rep., 2005. [Online]. Available: http://www.nmt.edu/~karoaper/AI_RVC.pdf

[8] http://pacific.commerce.ubc.ca/xr/

[9] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multi-class support vector machines," Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, Tech. Rep., 2001.