The Application of SVM to Algorithmic Trading

Johan Blokker, CS229 Term Project, Fall 2008
Stanford University

Abstract

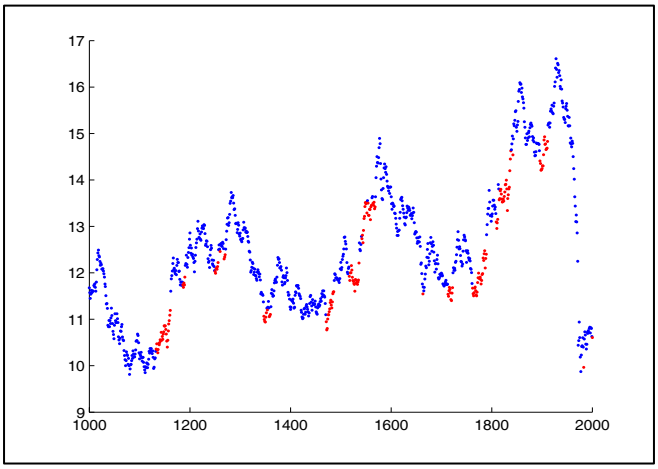
A Support Vector Machine (SVM) was used to attempt to distinguish favorable buy conditions on daily historical equity prices. The SVM used a Gaussian kernel and was optimized over sigma and the margin classifier using cross validation. Although encouraging results were initially obtained through optimization, applying the results to obtain a single trading classifier was unsuccessful. Variation in the data made it impossible to find a useful setting for sigma and C that would produce consistent results. The results are a confirmation of the Weak Efficient Market Theory that predicts there is no information within the market that can be used to predict future prices.

Introduction

Support Vector Machines (SVM) are supervised learning algorithms for classification of data. They have been used successfully in a broad range of purposes from detecting email spam to hand writing recognition. SVM's have become popular because of their ease of implementation in problems requiring classification of data. SVM's separate highly dimensional training data with hyper-planes that maximize the margin between the classification of the data set. Then, from the determination of the best hyper-planes, new data can be classified. In this way, SVM's are a maximum likelihood estimation of the classification.

Method

In this study a SVM was used to attempt to classify equity prices to form a successful algorithmic trading system. The problem consists of creating a training data set $\{X : X \in \mathbb{R}^n\}$ along with a corresponding set of data classification $\{Y : Y \in \{-1,1\}\}$ used to train the SVM algorithm and then test its recognition error rate. The training data set was composed of historical price data from 1980 to 2008, and included 20-day and 200-day moving averages along with their standard deviations. The data was classified as being in a favorable buy condition if the price increased by 10% within the next 30 days. This classification works well when the underlying equity has at least an average volatility. The graph below illustrates in red the favorable buy points of a sample data set created from this classification.

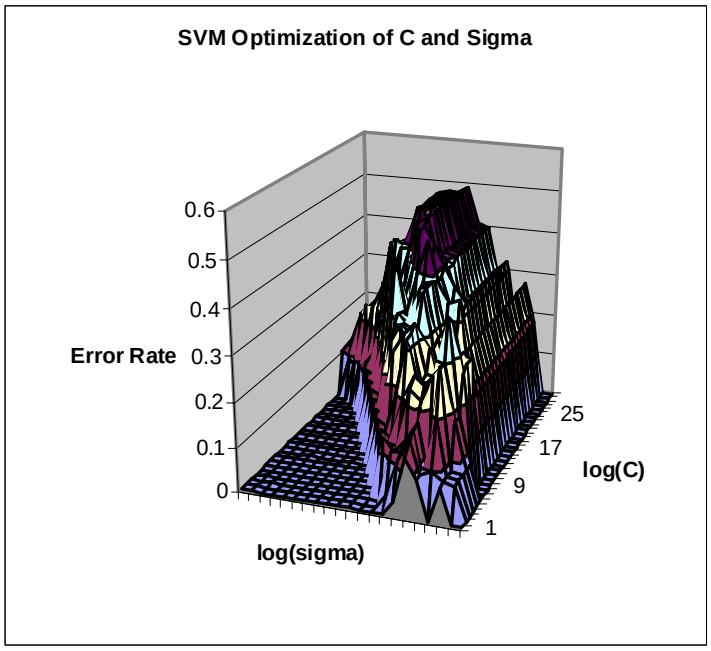


An error rate function was chosen to count the percent of the buy recommendations that were incorrect in all the buy recommendations. In other words, an error was counted when the SVM predicted a *buy* but the data set was really classified as *don't buy*. The opposite condition is not as important because it only represents a missed buy opportunity, not a potential loss of value.

For accurate test results, it is essential that no future data points creep into the

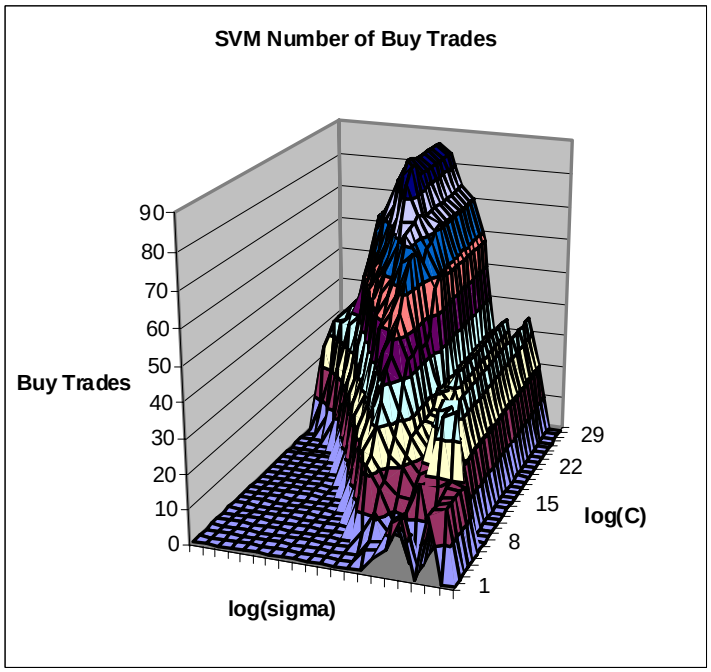
calculation of the SVM. To prevent this, a test was developed with a dummy set of data that had a step function at one point in the data. Looking at the resulting output, it was confirmed that the data formatting algorithm was correctly allied without fence-post errors.

This study used the SVM-KMToolbox[1] written for MATLAB to perform the SVM calculations using a Gaussian kernel. The data was segmented into eleven sections of 200 data points for Cross Validation training. Training was performed on section n, then test performed on section n+1 for ten steps. A grid search was performed across the margin classifier (C) and sigma to see if an optimum setting of the SVM parameters could be found. Promising results were obtained with many points having an error rate of less that 50%.



Problems

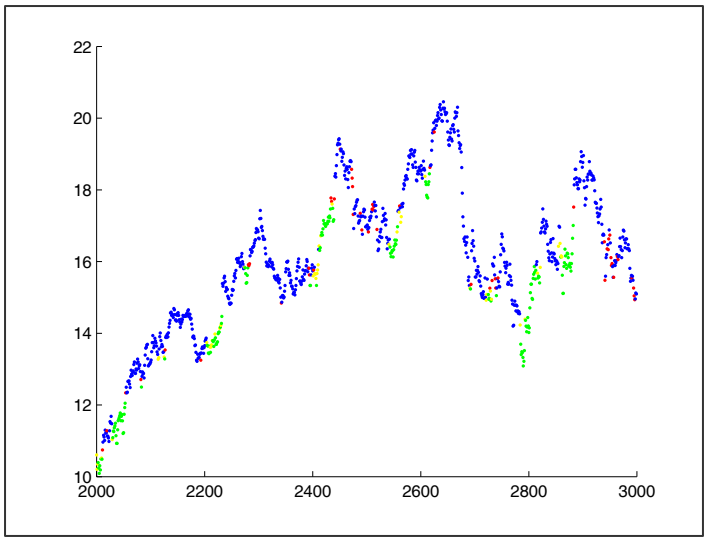
One initial problem was that at points with low error rate, there were not many trades attempted. As the number of trades increased, the error rate rose steeply. Tests were made on different settings for sigma and C, with training sets of 500, 1000, 1500, and 2000 data points, but the results varied erratically, from having zero trades predicted to having greater than 50% error rate. In effect, the location of the steep edge of the error rate function varied erratically in time. Yet there were two distinct regions of interest. Sigma = 1000 corresponded to the peak error rate. This also corresponded to



$\sqrt{\frac{1}{m}\sum x^T x}$ for this data set, which is where the Gaussian kernel would have its greatest discrimination. The other region of interest was when sigma=50 where a flat point in the error rate occurred below the 50% level.

Further Testing

Next, the SVM algorithm was reorganized to model real world conditions. A set of data was used to train the SVM and a prediction was made on the next data point. This was repeated, shifting down the data to produce a set of trade predictions. The graph on the right shows the correct predictions in green, incorrect predictions in red, and missed predictions in yellow. This test was performed with 200 and 500 training samples with sigma = 50 and 1000. The best results were obtained with sigma = 50. The number of training samples did not have a significant effect on the results.

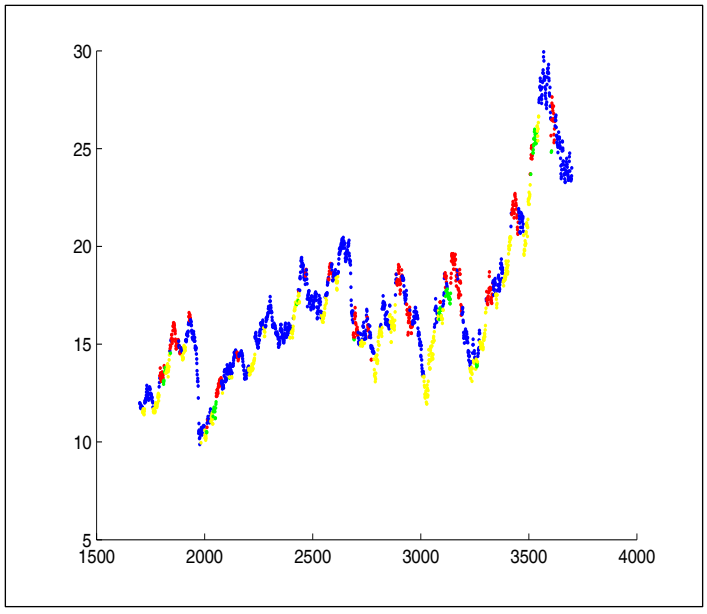


The predicted data was incorporated into a trading model that bought on the buy recommendations and sold on a fixed set of trading rules. The annual rate of return obtained was 55% which was too good to be true. Further analysis explained the hidden source of error.

Hidden Source of Error

The hidden source of error in the model above came from the fact that the classification data of the training data set relied on future information. The spectacular results were a result of SVM having indirect knowledge about the future. Although this simulation was faulty, it demonstrates the SVM algorithm was implemented correctly and could classify data that contains useful information about the future.

To remove the source of error, a 30 day gap was added between the training data set and the predicted data. The second graph shows the results when a 30-day data gap is



introduced. There is a significant increase in errors and missed buy points. When the new trading predictions were incorporated into a trading model, the resulting rate of return was within one percent of the return on a portfolio of random trades. There was no significant benefit from the SVM decisions.

Conclusion

It was my hypothesis that statistical fluctuations in prices could be taken advantage of by using a computerized trading algorithm. The use of an SVM algorithm, in an effort to find information in market data that could be useful for predicting profitable buy conditions, failed. According to the Efficient Market Theory, building a computerized trading system should not be possible. This is because all information that is publicly available that could affect the market has already been taken into account. This study is a confirmation of this theory that the market is a Martingle.

References

1. S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, "SVM and Kernel Methods Matlab Toolbox ", Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005.

2. B. Scholkopf and A. J. Smola, Learning with Kernals, The MIT Press, Cambridge, MA, 2002.

3. C. H. Hsu and C. J. Lin, "A Simple Decomposition Method for Support Vector Machines", Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2008.

4. C. H. Hsu and C. J. Lin, "A Practical Guide to Support Vector Classification", Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2008.

5. J. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization" Microsoft Research, Redmond, WA, 1998.