SVM: AN ALTERNATIVE WAY IN VOLATILITY PREDICTION

A PREPRINT

Yule Liu
School of Information Science and Technology
ShanghaiTech University
liuyl4@shanghaitech.edu.cn

Yucheng Jiang
School of Information Science and Technology
ShanghaiTech University
jiangych2@shanghaitech.edu.cn

Zhenjie Xu
School of Information Science and Technology
ShanghaiTech University
xuzhj1@shanghaitech.edu.cn

December 16, 2022

ABSTRACT

Financial time series forecasting is one of the most challenging applications of modern time series analysis. Financial time series are deterministically chaotic, noisy, and non-stationary by nature. Support Vector Machine (SVM), a semiparametric tool to perform regression estimation, is combined with GARCH model in our project to perform forecasting assignments on real-life stock market. To be specific, we choose NASDAQ index, Chinese Ten-year Treasury bonds and Tesla as our main data sources. The Structural Risk Minimization Principle, which tries to reduce the generalization error rather than the training error, is implemented by SVMs, so we expect a scenario where more generality will be achieved as a consequence than using traditional methods.

Keywords ARCH · GARCH · SVM · Volatility Prediction

1 Introduction

Volatility is a statistical measure of the dispersion of returns for a given security or market index. In most cases, the higher the volatility, the riskier the security. Volatility is often measured from either the standard deviation or variance between returns from that same security or market index. In the securities markets, volatility is often associated with big swings in either direction. For example, when the stock market rises and falls more than one percent over a sustained period of time, it is called a "volatile" market. An asset's volatility is a key factor when pricing options contracts.

There already exist several mature methods in terms of predicting volatility and estimating stock performance in global financial markets. Two widely used models are the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model [Bollerslev, 1987] and ARCH model [Bollerslev, 1986] . It has been frequently utilized to capture the time-varying volatility of the data and is a generalization of ARCH. ARCH models were created in the context of econometric and finance problems having to do with the amount that investments or stocks increase (or decrease) per time period, so there's a tendency to describe them as models for that type of variable. An ARCH model could be used for any series that has periods of increased or decreased variance. This might, for example, be a property of residuals after an ARIMA model has been fit to the data. A GARCH (generalized autoregressive conditionally heteroscedastic) model uses values of the past squared observations and past variances to model the variance at time t.

In these cases, maximum likelihood estimation (MLE) is typically used to estimate both symmetric and asymmetric GARCH models [Bollerslev and Wooldridge, 1992] . In order to estimate the parameters of the model, this estimation needs a certain distribution of innovations, including the Normal distribution, the Student's t distribution, and the Generalized Error Distribution (GED). As a result, finding the best fit distribution of innovation is a challenge for researchers, making the estimating procedure time-consuming, expensive, and difficult.

The support vector machine (**SVM**) estimates a function by non-linearly mapping the input space into a high-dimensional hidden space and then running the linear regression in the output space [Vapnik, 2006] . Thus, the linear regression in the output space corresponds to a nonlinear regression in the low-dimensional input space. Since they do not rely on any assumptions about the functional form, several semiparametric techniques have fortunately been developed to improve estimate for GARCH models. Support vector machines (SVM) are among the most popular and effective semi-parametric models. In recent works, there has been evidence that combining GARCH and SVM (i.e. GARCH-SVM) to do forecasting is a preferred choice in real market scenarios [Pérez-Cruz et al., 2003] , though the conventional GARCH does perform better when the data is normal and large.

However, these methods mainly focus on estimating the result of the first-order model like GARCH(1,1). Because of the lack of ability to integrate higher-order time series information, the predicting power is still limited by the first-order model. In this project, we focus on estimating the high-order GARCH model to achieve better predicting power. This is our attempt to determine the most accurate approach for forecasting the stock market, which makes sense given that SVM has been demonstrated to be effective.

Our main contributions of our project are listed below:

- We use SVM to estimate the high-order GARCH model and achieve a better performance compared to all the previous baselines.
- We conduct extensive experiments to find out the best setting of SVM to obtain the best penalty parameter and validate the effectiveness of SVM in the area of generalization power.
- We obtain the regression form of SVM, which is previously used for classification in most cases.

2 Preliminary

2.1 ARCH model

Suppose that we are modeling the variance of a series y_t . The ARCH(1) model for the variance of model y_t is that conditional on y_{t-1}, the variance at time t is

$$\mathbf{Var}(y_t \mid y_{t-1}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 \tag{1}$$

where constraints such as $\alpha_0 \geq 0$ and $\alpha_1 \geq 0$ are imposed to avoid negative variance.

If we assume that the series has mean = 0 (which can be done by demeaning), the ARCH model could be written as

$$y_t = \sigma_t \epsilon_t \text{ with } \sigma_t = \sqrt{\alpha_0 + \alpha_1 y_{t-1}^2} \text{ and } \epsilon_t \sim^{i.i.d.} (\mu = 0, \sigma^2 = 1) \tag{2}$$

For inference (and maximum likelihood estimation) we would also assume that the are normally distributed. An ARCH(m) process is one for which the variance at time t is conditional on observations at the previous m times, and the relationship is

$$\mathbf{Var}(y_t \mid y_{t-1}, \ldots, y_{t-m}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \ldots + \alpha_m y_{t-m}^2 \tag{3}$$

With certain constraints imposed on the coefficients, the y_t series squared will theoretically be AR(m).

2.2 GARCH

A GARCH (generalized autoregressive conditionally heteroscedastic) model uses values of the past squared observations and past variances to model the variance at time t. As an example, a GARCH(1,1) is

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{4}$$

In the GARCH notation, the first subscript refers to the order of the y^2 terms on the right side, and the second subscript refers to the order of the σ^2 terms.

2.3 SVM

Given a training dataset(x_t, y_t), where input vector $x_t \in R^p$ and output scalar $y_t \in R^1$. Indeed, the desired response y, known as a 'teacher', represents the optimum action to be performed by the **SVM**. We aim at finding a sample regression function f(x), or denoted by y^, as below to approximate the latent, unknown decision function $g(x)$:

$$f(x) = w^T \phi(x) + b \tag{5}$$

where the superscript T is a transposing operator that should be differentiated from the sample size T of the time series used later in this paper.

$$\phi(x) = [\phi_1(x), ..., \phi_l(x)]^T, w = [w_1, ..., w_l]^T \tag{6}$$

where denotes the nonlinear transfer function, l is the dimension of the feature space, W denotes a set of linear weights connecting the feature space to the output space, and b is the threshold. To get the function f(x), the optimal w* and b* have to be estimated from the data. First, we define a linear ϵ-insensitive loss function, Lϵ:

$$L_\epsilon(x, y, f(x)) = \begin{cases} |y - f(x)| - \epsilon & for |y - f(x)| \geq \epsilon \\ 0 & otherwise \end{cases} \tag{7}$$

This function indicates the fact that it does not penalize errors below ϵ.

3 Methodology

In this section, we will first introduce the problem statement. Then, we will present our SVM regression to forecast volatility. There are two main components in our framework: (1) The methodology on how SVM works (2) The empirical module that is used for volatility forecasting with the help of SVM regression.

3.1 Problem Statement

- The data is denoted by a timing sequence $\{\sigma_1, \sigma_2, ..., \sigma_t\}$, where σ_i is the volatility of time i. In particular, σ_i is computed as the average of the square of the past five days.
- The volatility forecasting problem is to find a regressor $\mathcal{F} : \mathrm{R}^m \rightarrow \mathrm{R}$, such that the output of the model is closer to σ_{t+1} as much as possible.

3.2 Support Vector Machine regression

Suppose we are given a training set $(x_1, y_1), ..., (x_m, y_m) \subset R^d \times R$, where x_i's are the regressors and y_i's are the observations. In "ϵ - insensitive" measure, our goal is find a function f(x) that has at most ϵ deviation from the actually obtained target y_i's for all the training data, and at the same time, is as flat as possible. In other words, we do not care about errors as long as they are less than ϵ, but will penalize it otherwise. Consider the linear function

$$f(x) =< w, x > +b \tag{8}$$

Structural risk is the upper boundary of empirical loss, denoted by ϵ-insensitive loss function, plus the confidence interval (or called margin), which is constructed in equation (8). The primal constrained optimization problem of SVM is obtained below:

$$\min_{w \in R^t, \xi(') \in R^{2T}, b \in R} C(w, b, \xi_t, \xi_t') = \frac{1}{2}||w||^2 + C \sum_{t=1}^{T} (\xi_t + \xi_t') \tag{9}$$

with $w \in R^d, b \in R$. Flatness means that one seeks small $||w||_2 2$. Formally we can write this problem as a convex optimization problem by requiring:

$$min \frac{1}{2}||w||^2$$
$$s.t. \begin{cases} y_t - < w, x_t > -b \leq \epsilon \\ < w, x_t > +b - y_t \leq \epsilon \end{cases} \tag{10}$$

The tacit assumption was that such a function f actually exists that approximates all pairs (x_t, y_t) with ϵ precision, or in other words, that the convex optimization problem is feasible. Analogously to the "soft margin" loss function, one can introduce slack variables ξ_t, ξ_t^* to cope with otherwise infeasible constraints of the optimization problem. Hence we arrive at the formulation

$$min \frac{1}{2}||w||^2 + C \sum_{t=1}^{T} (\xi_t + \xi_t^*)$$
$$s.t. \begin{cases} y_t - < w, x_t > -b \leq \epsilon + \xi_i \\ < w, x_t > +b - y_t \leq \epsilon + \xi_t^* \\ \xi_t, \xi_t^* \geq 0 \end{cases} \tag{11}$$

The formulation of the cost function C(·) is in perfect accord with the SRM principle. The penalty parameter C>0 controls the penalizing extent on the sample points which lie outside ϵ-tube. Both ϵ and C, the free parameter of SVM,

must be selected by the user.The corresponding dual problem of the SVM can be derived from the primal problem by using the Karush–Kuhn–Tucker conditions [Fletcher, 2013] as follows:

$$\min_{\alpha^{(\prime)}{}_t \in R^{2T}} \frac{1}{2}\sum_{s=1}^{T}\sum_{t=1}^{T}(\alpha'_s - \alpha_s)(\alpha'_t - \alpha_t)K(x_s \cdot x_t) + \epsilon\sum_{t=1}^{T}(\alpha'_t + \alpha_t) - \sum_{t=1}^{T} y_i(\alpha'_t - \alpha_t) \tag{12}$$

where α_t and α'_t(or $\alpha_t^{(\prime)}$) are the Lagrange multipliers. The dual problem can be solved more easily than the primal problem [Schölkopf et al., 2002] . Making use of any solution of α_t and α'_t, the optimal solutions of the primal problem can be calculated in which w* is unique and expressed as follows:

$$w* = \sum_{t=1}^{T}(\alpha'_t - \alpha_t)\phi(x_t) \tag{13}$$

However, b* is not unique and formulated in terms of different cases. If $i \in t|\alpha_t \in (0, C)$, then

$$b* = y_t - \sum_{t=1}^{T}(\alpha'_t - \alpha_t)K(x_t \cdot x_i) + \epsilon \tag{14}$$

If $j \in \{t|\alpha'_t \in (0, C)\}$, then

$$b* = y_j - \sum_{t=1}^{T}(\alpha'_t - \alpha_t)K(x_t \cdot x_i) - \epsilon \tag{15}$$

The cases of both $i, j \in t|\alpha_t^{(\prime)} = 0$ and $i, j \in t|\alpha_t^{(\prime)} = C$ rarely occur in reality. Thus the regression decision function f(x) will be computed by using w* and b* in the following forms:

$$\begin{aligned} f(x) &= w*^{T}\,\phi(x) + b* \\ &= \sum_{t=1}^{T}(\alpha'_t - \alpha_t)\phi^T(x_t)\phi(x) + b* \\ &= \sum_{t=1}^{T}(\alpha'_t - \alpha_t)K(x_t, x) + b* \end{aligned} \tag{16}$$

where $K(x_t, x) = \phi^T(x_t)\phi(x)$ is the inner-product kernel function. In fact, the SVM theory considers only the form of $K(x_t, x)$ in the feature space without specifying explicitly $\phi(x)$ and without computing all corresponding inner products. We experiment with two different kernels to investigate the effect of a kernel type [Schölkopf et al., 2002] in Monte Carlo simulation:

$$Linear : K(x_t, x) = x_t^T x \tag{17}$$

$$Gaussian : K(x_t, x) = exp(\frac{-||x - x^t||^2}{2\sigma^2}) \tag{18}$$

where d and σ^2 are the parameters for the polynomial and Gaussian kernel. The appropriate values of the coefficients ϵ, C, d and σ^2 will be discussed in the following part.

3.3 Higer order GARCH estimation

In the low-order GARCH model, the linear function is :

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{19}$$

where only the first-order time series information is used, to estimate a more powerful high-order model, we model the relation as:

$$\sigma_i = \sum_{j=1}^{l}\beta_j \sigma_{i-j} + \epsilon_i \tag{20}$$

where the σ'_is are the implied volatility data and ϵ'_is are the noises. In the simulation, we add in the time to maturity parameter as an additional feature. Therefore the formula changes slightly.

$$\sigma_i = \sum_{j=1}^{l}\beta_j \sigma_{i-j} + \beta_0(T - t_i) + \epsilon_i \tag{21}$$

4 Experiment

In this chapter, we first show our experimental setup, including datasets and the evaluation metrics, different experimental scenarios, the primary baselines. We then designed a bunch of experiments to answer the following questions

Q1: How powerful is the prediction ability of SVM compared to baselines?

Q2: How does the size of training set influence the model result?

Q3: How does penalization parameter C influence the result?

4.1 Experiment Preparation

4.1.1 Datasets

In this project, we use financial data from different markets to test the prediction power.

NDQ Index We download the data of NDQ index in the past 5 years to conduct our experiment. NationalAssociation of Securities Dealers Automated Quotations (NASDAQ) is the second largest stock exchange in the world. And it is one of the most efficient markets in the world.

10CNY.B We download the data of Chinese Ten-year Treasury bonds in the past 5 years, which is a core indicator of Chinese interest rate. It is a benchmark in the bond market.

TSLA We download the data of Tesla in the past 5 years. Tesla is one of the most valuable companies in the world and it is also a representative stock in the market, which is suitable to estimate the power of prediction in stock market.

4.1.2 Metrics and Experiment setting

To exactly evaluate the model prediction power, we use the R-square value as the metrics for the model. This is a classical metric in previous volatility prediction tasks.(cite).

$$R^2 = 1 - \frac{\sum_{t=1}^{N} \left(y_t^2 - \hat{y}_t^2\right)^2}{\sum_{t=1}^{N} \left(y_t^2 - \left(\frac{1}{N}\sum_{s=1}^{N} y_t^2\right)\right)^2} = 1 - \frac{\sum_{t=1}^{N} \left(y_t^2 - \hat{y}_t^2\right)^2}{\sum_{t=1}^{N} \left(y_t^2 - \bar{y}^2\right)^2}. \tag{22}$$

This relative accuracy statistic indicates that the model accounts for over $100 \times R^2$ per cent of the variability in the observations. For example, $R^2 = 0.11$ means that the model accounts for 11% of the variability in the observations. If $R^2 = 0$, then the model is incapable of extracting the deterministic part of the time series, if there is any. If $R^2$is negative this means that the model introduce more variability than the sample mean of the original time series.

To examine the generalization power of the model, we use 50%/50%, 60%/40%, 70%/30% and 90%/10% dataset split respectively.

4.1.3 Baseline

ARCH The parameter selection of ARCH model is usually determined by the partial auto-correlation of the data itself. The partial auto correlation of each dataset is shown in the following figure.1 - figure.3. We find that 8 is the most important border for the coefficients. Therefore, we use the ARCH of 8-lags to estimate the volatility.

GARCH The GARCH (1, 1) model provides a simple representation of the main statistical characteristics of a return series for a wide range of assets and, consequently, it is extensively used to model real financial time series. The SVM is also used to estimate a GARCH, therefore, we use the default setting of GARCH(1,1) model.

4.2 Q1: The prediction power

To demonstrate the superiority of the proposed model, we perform several experiments to compare SVM based model to the state-of-the-art methods mentioned above. We trained the model with the same parameters in 60%/40% split on all three datasets, and calculate the R-square value for each train, the final result in Table 1 is the mean of five runs.

We have achieved particularly significant results on all three datasets, where our R-square has improved respectively by 0.19495, 0.04969, and 0.53017 compared to the best baseline.

Among the two baselines, we find that the GARCH model performs relativly well. That is because GARCH is an extension of the ARCH model that incorporates a moving average component together with the autoregressive

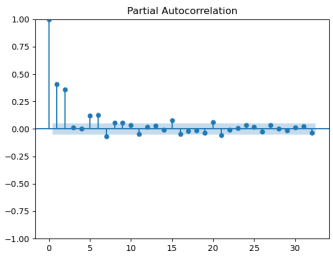

Figure 1: NDQ Index

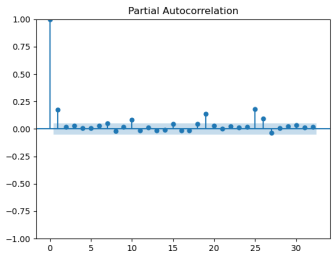

Figure 2: 10CNY.B

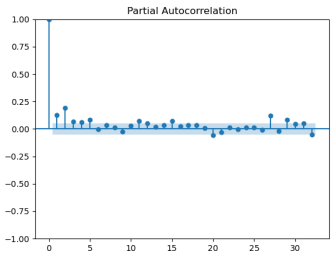

Figure 3: TSLA

Table 1: R-square values of different models

	NDQ	CNY10	TSLA
ARCH	0.05531	0.59933	0.1032
GARCH	0.2692	0.66505	0.12071
SVM-LINEAR	0.66723	0.61744	0.68091
SVM-RBF	0.79341	0.66161	0.72506
SVM-LINEAR	**0.81629**	**0.67278**	**0.73885**

component. GARCH is the "ARMA equivalent" of ARCH, which only has an autoregressive component. GARCH models permit a wider range of behavior more persistent volatility.

We find that the RBF kernel performs well in the index-asset like NDQ and CNY10 Bond, however, it performs really poor in a stock. And the linear kernel performs well in all the data we tests, while it usually generates many outliers which is hard to deal with. Generally, it is more reliable to use SVM-RBF, because it is a relative stable model. The prediction is shown as follows:

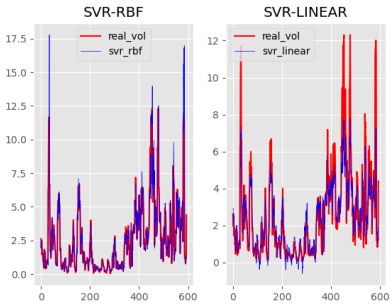

Figure 4: NDQ Index prediction

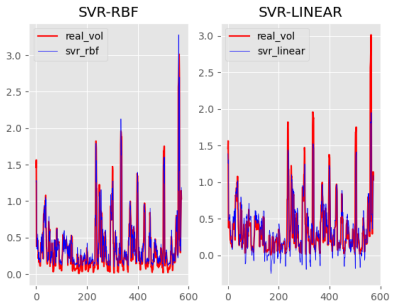

Figure 5: 10CNY.B prediction

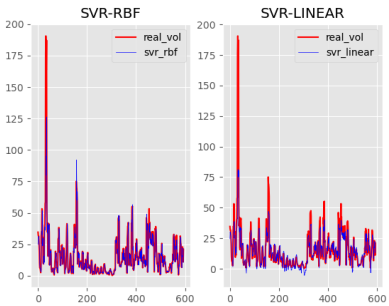

Figure 6: TSLA prediction

4.3 Q2: The generalization power

We test the NDQ dataset of different split type to explore if the model is capable of predicting with different size of training set. We find an interesting result in Figure 7 that even though the R square of SVM-LINEAR roughly increases when training set is larger, other models perform better in a smaller training set. It might be a reason that the first three models are simple to train and larger training set leads to overfitting, while SVM with linear kernel needs more data to fit the kernel to give a better result. One evidence is that linear kernel needs much more time than RBF kernel when training.

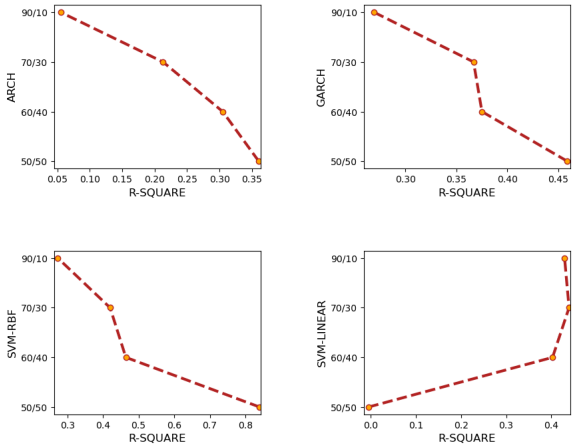

Figure 7: Different split on dataset.

To make the split selection more reasonable, we choose the 60/40 split in our final result, which is adequate in proving the generalization power of the model.

4.4 Q3: The effect of parameter C

We test SVM-Linear and SVM-RBF with different penalty parameters range from 5 to 50. We find an another interesting result in Figure 8 that the linear kernel prefer a smaller C while the RBF kernel prefer a larger C in training. As a result, we choose C=5 for linear kernel and C=40 for RBF kernel to exploit the most power of the SVM itself.

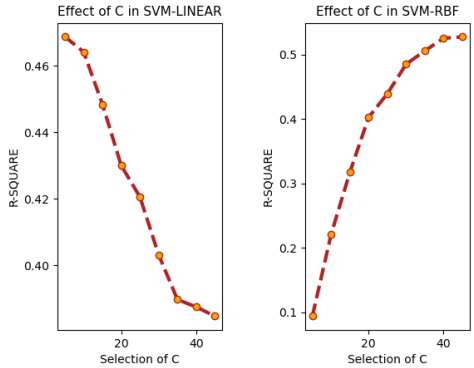

Figure 8: Different split on dataset.

5 Conclusion

We have used the SVM as a linear machine to replace the ML estimation process and tried to obtain a nonlinear estimation using kernels. The SVM has several properties that make it suitable for solving problem in which a linear

and nonlinear dependency has to be estimated from the data. The most relevant property is that the SVM is based on a well established learning theory that is able to give bounds on the expected errors and convergence rate given the number of samples and the machine complexity. In addition, the SVM has two extra desirable properties: the functional to be minimized is quadratic and linearly restricted and the machine architecture is given by the learning procedure. The former property ensures that the solution cannot get trapped in local minima and the latter property precludes the need to look for the best connections and hidden layers because the SVM solution provides it.

References

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

Tim Bollerslev. A conditionally heteroskedastic time series model for speculative prices and rates of return. *The review of economics and statistics*, pages 542–547, 1987.

Tim Bollerslev and Jeffrey M Wooldridge. Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. *Econometric reviews*, 11(2):143–172, 1992.

Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

Fernando Pérez-Cruz, Julio A Afonso-Rodriguez, and Javier Giner. Estimating garch models using support vector machines. *Quantitative Finance*, 3(3):163, 2003.

Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.