ECONSTOR
Make Your Publications Visible.

A Service of

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Li, Yushu

Working Paper

Estimating and Forecasting APARCH-Skew-t Models by Wavelet Support Vector Machines

Working Paper, No. 2012:13

Provided in Cooperation with:
Department of Economics, School of Economics and Management, Lund University

Suggested Citation: Li, Yushu (2012) : Estimating and Forecasting APARCH-Skew-t Models by Wavelet Support Vector Machines, Working Paper, No. 2012:13, Lund University, School of Economics and Management, Department of Economics, Lund

This Version is available at:
https://hdl.handle.net/10419/260039

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

Working Paper 2012:13

Department of Economics
School of Economics and Management

Estimating and Forecasting APARCH-Skew-*t* Models by Wavelet Support Vector Machines

Yushu Li

May 2012

LUND UNIVERSITY

Estimating and Forecasting APARCH-Skew-*t* model by Wavelet Support Vector Machines

Yushu Li[1][2]

Department of Economics, Lund University

Abstract:

This paper concentrates on comparing estimation and forecasting ability of Quasi-Maximum Likelihood (QML) and Support Vector Machines (SVM) for financial data. The financial series are fitted into a family of Asymmetric Power ARCH (APARCH) models. As the skewness and kurtosis are common characteristics of the financial series, a skew *t* distributed innovation is assumed to model the fat tail and asymmetry. Prior research indicates that the QML estimator for the APARCH model is inefficient when the data distribution shows departure from normality, so the current paper utilizes the nonparametric-based SVM method and shows that it is more efficient than the QML under the skewed Student's *t*-distributed error. As the SVM is a kernel-based technique, we further investigate its performance by applying a Gaussian kernel and a wavelet kernel. The wavelet kernel is chosen due to its ability to capture the localized volatility clustering in the APGARCH model. The results are evaluated by a Monte Carlo experiment, with accuracy measured by Normalized Mean Square Error (*NMSE*). The results suggest that the SVM based method generally performs better than QML, with a consistently lower *NMSE* for both in sample and out of sample data. The outcomes also highlight the fact that the wavelet kernel outperforms the Gaussian kernel with a lower *NMSE* , is more computation efficient and has better generation capability.

JEL classification: C14, C53, C61

Keywords: SVM, APARCH, wavelet kernel, Monte Carlo Experiment.

1. Introduction:

Since the ARCH model was proposed in a seminal paper by Engle (1982), related research has grown rapidly and various forms and specifications of the ARCH model have emerged to represent the three typical "stylized characteristics" in the financial series: volatility clustering, fat tail leptokurtosis and the asymmetric leverage effect. The ARCH and GARCH

[1] The author gratefully acknowledges funding from the Swedish Research Council (421-2009-2663)

[2] The author gratefully acknowledges comments from professor David Edgerton, Fredrik NG Andersson and Abdullah Almasri

(Bollerslev, 1986) models successfully managed these first two factors but failed in handling the leverage effect, which is a common phenomenon in financial markets due to insufficient information. To resolve this problem, Ding *et al.* (1993) proposed the Asymmetric Power ARCH (APARCH) model, which rapidly gained popularity due to its ability to capture the asymmetric impact of volatility corresponding to positive and negative news. Instead of assuming the correlation in the second order term of the innovation in GARCH models, the APARCH model allows the correlation to exist in other power forms and can further capture the leverage effect between asset return and volatility. Moreover, compared with GARCH model, which assumes a linear relationship between the return and volatility, the APARCH model allows more flexible autoregressive structure of the returns. However, the flexibility in the APARCH model also complicates the estimation due to the higher dimension and the identification problem of the parameters. As with the GARCH model, the estimation of the ARARCH model is generally based on the Maximum Likelihood (ML) under normal distribution or Quasi-Maximum Likelihood (QML) for non-normal densities. As the normal distribution lacks the ability to capture skewness (3rd moment) and kurtosis (4th moment) in high frequency financial data, Fernández and Steel (1998) proposed a Skewed Student's *t*-distribution to model the excess of kurtosis and asymmetric effects. The problem arises in cases where, for example, the QML estimator becomes inefficient with the inefficiency increasing as the degree of skewness increases (Engle and González-Rivera, 1991). The current paper will attempt to improve model fitting and forecasting when encountering skewed density by applying a distribution-free approach: Support Vector Machine (SVM)-based regression. The SVM is a pure data driven technique and does not need *a priori* assumptions of the model structure or distribution properties. It is also a kernel-based methodology which can achieve computational sparsity when faced with the high dimensional data, which makes it an attractive approach for estimating the APARCH model when high power terms are introduced. Furthermore, the implementation of the SVM will generally attain high accuracy without requiring large sample sizes, which makes it more efficient than the QMLE, especially when distribution information is not available.

Previous research has used the SVM and the extended methods to estimate and predict the volatility in financial markets: Tay and Cao (2002) have used *C*-ascending SVM in financial time series forecasting; Préz-Cruz *et al.* (2003) estimate the GARCH model by ε insensitive SVM, Chen *et al.* (2010) apply a Recurrent SVM procedure to forecast volatility under a GARCH framework and Ou and Wang (2010) suggest a similar Relevance Vector Machine

(RVM) to deal with GARCH, EGARCH and GJR models. This research shows that the SVM is generally considered to be a better predictor of volatility when assessing the outcomes by various criteria. However, one important issue in applying the SVM technique is that its performance will be influenced by kernel selection. When estimating and predicting the volatility in financial data, Tang *et al.* (2009) suggest that the wavelet kernel can better capture the volatility clustering than the generally applied Gaussian kernel as the wavelet kernel is constructed on an orthonormal wavelet basis on $L^2(R)$ space through horizontal floating and flexing, so that it has a more accurate localized property and can approximate curves in quadratic continuous integral space better than the Gaussian kernel. No application of the SVM and wavelet kernel to the APARCH type model has been performed in previous researches, and the present paper will be the first to apply the SVM to estimate the APARCH model, which contains the GARCH, GJR, TSGARCH and TGARCH models. In addition, we will further investigate whether a wavelet-based kernel will outperform the commonly applied Gaussian kernel in the APARCH framework when using the SVM.

The structure of the paper can be divided into the following parts: section 2 is an introduction of SVM-based regression and wavelet kernels; section 3 is a description of the model and the experimental design; section 4 applies the Monte Carlo experiment to assess the results and the final section contains conclusions and discussion.

2. Brief description of SVM regression and wavelet kernels

2.1. Theory of SVM based regression

The SVM algorithm is a nonlinear extension of the generalized portrait algorithm developed by Vladimir Vapnik (Vapnik and Lerner, 1963) and based on the ground theory of statistical learning theory introduced by Vapnik and Chervonenkis (1974). It aims to minimize the structure risk in model fitting and prediction, and the solution can be uniquely and globally achieved by solving a linearly constrained quadratic optimizing problem. The SVM was originally used in classification and pattern recognition problems, while its utility for nonlinear regression becomes apparent after the introduction of the ε-insensitive loss function （Vapnik, 1995）due to the high accuracy and computation sparseness of the SVM. The framework of the ε-insensitive Support Vector Regression (SVR) begins with a training data set $\{(\mathrm{x}_1, y_1),...,(\mathrm{x}_l, y_l)\} \subset \Re^d \times \Re$ with $\mathrm{x}_i \in \Re^d$ denoting the input vector and $y_i \in \Re$ being the output scalar; the goal of this regression is to find a function $f(x)$ that has at least

ε deviation from the output scalar y_i while at the same time showing optimal smoothness. To achieve this goal, SVM nonlinearly maps the input space into a higher dimension feature space $\Re^{df}$, where $df > d$. The linear regression can then be employed in this feature space and the nonlinear relations in the input space can be approximated by the linear regression in the higher dimension feature space, with the accuracy of the approximation increasing with feature space dimension. Generally, given training data $\{(\mathrm{x}_1, y_1),...,(\mathrm{x}_l, y_l)\}$, the regression function in the feature space can be expressed as:

$$f(\mathrm{x}) = w^T \varphi(\mathrm{x}) + b, \tag{1}$$

where $\varphi(\mathrm{x})$ is the nonlinear mapping function, which maps the input vector x into the future space at which the linear function $f(\mathrm{x})$ is defined. The smoothness of $f(\mathrm{x})$ corresponds to the norm of the regression coefficients $w = [w_1,..., w_{df}]^T$. Here, we will refer to the Euclidean norm $\|w\|^2$ with a smaller $\|w\|^2$ indicating a flatter $f(\mathrm{x})$ as a minimum $\|w\|^2$ is equal to the maximum of the separation margin $1/\|w\|^2$, which corresponds to the generalization ability (Smola and Scholkopf, 1998). The minimization should be performed while controlling the structure risk function under the ε-insensitive band constrain condition as follows:

$$\text{Minimize } \frac{1}{2}\|w\|^2 + \frac{C}{l}\sum_{i=1}^{l} L(f(\mathrm{x}_i), y_i);\ L(f(\mathrm{x}_i), y_i) = \begin{cases} |y_i - f(\mathrm{x}_i)| - \varepsilon \text{ for } |y_i - f(\mathrm{x}_i)| > \varepsilon \\ 0 \quad \text{otherwise} \end{cases}. \tag{2}$$

Function $L(f(\mathrm{x}_i), y_i)$ is the ε-insensitive loss function defined by Vapnik (1995). The ε-insensitive band constraint sets a penalty to the empirical risk: $e = y - w^T \varphi(\mathrm{x}) - b$: training data with an empirical error lower than ε will not be penalized, and training data with error larger than ε will be linear penalized. Thus, the training points within the ε-tube will not provide information for decisions. Only the data outside of the ε-tube are applied as support vectors to construct $f(\mathrm{x})$, resulting in prediction generalization and computational sparsity. Furthermore, slack variables ξ_i, ξ_i^* are introduced to denote the errors outside ε-tube, and equation (2) becomes the following:

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*), \text{ subject to } \begin{cases} y - w^T \varphi(\mathrm{x}) - b \le \varepsilon + \xi_i \\ w^T \varphi(\mathrm{x}) + b - y \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}, \tag{3}$$

where penalty parameter C in the second term determines to which extent the empirical error can be tolerated. The first term (the regularization term) denotes the smoothness of the

regression function. By choosing an appropriate C and setting a trade-off of the empirical error and generalization error, the regression can both fit the historical data well and make reliable predictions about future values. Both C and ε are free parameters and should be predetermined empirically according to the given data. In general, the value of C and ε are determined by cross-validation, which can guarantee sufficient generalization on the data set used for prediction. Equation (3) is called the primal objective function; solving the primal objective function is difficult due to the large variable set. Thus, a set of dual variables is introduced and Lagrange multipliers are applied to transfer the primal problem to dual problems of optimization. By constructing a Lagrange function from the primal objective function and the corresponding constraints (see Mangasarian, 1969; McCormick, 1983), the resulting formulation is:

$$\begin{aligned} L = & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ & - \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i - y_i + \langle w, \varphi(\mathrm{x}_i)\rangle + b) \qquad \text{subjects to } \alpha_i, \alpha_i^*, \eta_i, \eta_i^* > 0, \\ & - \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle w, \varphi(\mathrm{x}_i)\rangle - b) \end{aligned} \tag{4}$$

where L is the Lagrange function and $\alpha_i, \alpha_i^*, \eta_i, \eta_i^*$ are Lagrange multipliers. The partial derivatives of L with respect to the primal variables w, b, ξ_i, ξ_i^* must be removed for optimality as follows:

$$\partial_b L = \sum_{i=1}^{L}(\alpha_i^* - \alpha_i) = 0, \tag{5}$$

$$\partial_w L = w - \sum_{i=1}^{L}(\alpha_i^* - \alpha_i)\varphi(\mathrm{x}_i) = 0, \tag{6}$$

$$\partial_{\xi_i^{(*)}} L = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0. \tag{7}$$

Substituting (5), (6), and (7) into (4) yields the dual optimization as follows:

$$\begin{aligned} & \text{minimize } \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle\varphi(\mathrm{x}_i), \varphi(\mathrm{x}_j)\rangle - \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) + \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \\ & \text{subjects to } \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned}. \tag{8}$$

The nonlinear minimization in equation (4) is under the inequality constraint. Thus, the Karush-Kuhn-Tucker (KKT) conditions (Karush, 1939; Kuhn and Tucker, 1951) must be satisfied. The KKT conditions require that at the solution points, the product between dual variables and constraints must be removed as follows:

$$\begin{aligned}
&\alpha_i(\varepsilon+\xi_i-y_i+<w,\mathrm{x}_i>+b)=0\\
&\alpha_i^*(\varepsilon+\xi_i+y_i-<w,\mathrm{x}_i>-b)=0\,.\\
&(C-\alpha_i)\xi_i=0,\ (C-\alpha_i^*)\xi_i^*=0
\end{aligned} \tag{9}$$

Equation (9) indicates that for $\left|f(\mathrm{x}_i)-y_i\right|<\varepsilon$, α_i and α_i^* should be 0, which indicates that only the sample points associated with nonzero coefficients are referred to as support vectors and are used in deriving the function. Furthermore, equation (6) leads to $w=\sum_{i=1}^{L}(\alpha_i^*-\alpha_i)\varphi(\mathrm{x}_i)$ so that the regression function is rewritten as follows:

$$f(\mathrm{x})=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\left\langle\varphi(\mathrm{x}_i),\varphi(\mathrm{x})\right\rangle+b\,, \tag{10}$$

where $\left\langle\varphi(\mathrm{x}_i),\varphi(\mathrm{x})\right\rangle$ is the inner product of vectors in the feature space. To avoid the complexity of computing the nonlinear mapping φ, we can replace the dot product using kernel functions in the feature space. Equation (10) is as follows:

$$f(\mathrm{x})=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(\mathrm{x}_i,\mathrm{x})+b^*\,, \tag{11}$$

where the kernel function $K(\mathrm{x},\mathrm{y})=\left\langle\varphi(\mathrm{x}),\varphi(\mathrm{y})\right\rangle$ satisfies Mercer's theorem (Mercer, 1909). The qualified kernels will correspond to the inner product in the feature space. By applying the kernel function to replace the inner product, the issues relating to dimension are alleviated and only the kernel function requires specification, which can be performed without knowledge of the form of the nonlinear mapping. The following kernels that can be selected as admissive kernels in SVM include:

Linear kernel: $K(\mathrm{x}_i,\mathrm{x})=\mathrm{x}_i^T\,\mathrm{x}$,

Polynomial kernel: $K(\mathrm{x}_i,\mathrm{x})=(\kappa\,\mathrm{x}_i^T\,\mathrm{x}+1)^d$,

Gaussian kernel: $K(\mathrm{x}_i,\mathrm{x})=\exp\left(\dfrac{-\left\|\mathrm{x}-\mathrm{x}_i\right\|^2}{2\sigma^2}\right)$,

Sigmoid kernel: $K(\mathrm{x}_i,\mathrm{x})=\tanh(\kappa\,\mathrm{x}_i^T\,\mathrm{x}+r)$.

In addition to the free parameters C and ε, hyper parameters d, σ^2, κ and r in the above kernels must be determined in advance. There is no analytical method to determine the most suitable kernel for a particular data set other than certain general rules: the linear kernel is suitable for large sparse data vectors, the polynomial kernel is used in image processing, and the sigmoid kernel is preferred as a proxy for neural networks. When applying a kernel to data without knowledge of its form, the Gaussian kernel is considered a reasonable first choice.

The Gaussian kernel is also a general kernel that contains the linear and sigmoid kernel by setting restrictions on the penal parameter (Keerthi and Lin (2003)). As both Polynomial and Sigmoid kernels have more hyper parameters that need to be specified, compared to only one hyper parameter in the Gaussian kernel, the current paper will apply the Gaussian kernel and later compare it to the wavelet kernel proposed by Zhang *et al.* (2004). Zhang *et al.* combine the wavelet theory and support vector machines to show that the wavelet kernel achieves more accurate approximation for nonlinear functions. The current paper aims to adopt their proposed Morlet wavelet kernel when applying SVM to manage the APARCH model and compare the outcome with that of the Gaussian model. The following section will provide a brief introduction of the wavelet theory and wavelet kernel.

2.2. Introduction to the wavelet and the wavelet kernel.

Wavelet methods have been widely applied in the field of signal and image processing after their theoretical development in the 1980s (Grossmann and Morlet, 1984; Mallat, 1989). Wavelet methods adopt a basis of spatially localized functions as their transform filters, based on wavelet filtering of the original signal through shifting and dilations. The wavelet transformation can capture the characteristics of data series both in the frequency domain and the temporal domain using a two dimensional resolution. Corresponding to sinusoidal waves in the Fourier transform, the orthonormal wavelet bases $\{\psi_{k,a} : k, a \in R\}$ used in the wavelet transform are generated by translations and dilations of a basic mother wavelet $\psi \in L^2(R)$ and can be expressed as $\psi_{k,a}(x) = \frac{1}{\sqrt{a}}\psi(\frac{x-k}{a})$. For the signal $f(x)$, the wavelet transform is $\gamma(k,a) = \langle f, \psi_{k,a} \rangle = \int f(x)\psi^*_{k,a}(x)dx$. When the mother wavelet that satisfy the condition $\int_0^\infty \frac{|H(\omega)|^2}{|\omega|} d\omega < \infty$, with $|H(\omega)|$ as the Fourier transform of the $\psi(x)$, we can reconstruct $f(x)$ using the inverse wavelet transform, $f(x) = \iint \gamma(k,a)\psi_{k,j}(x)dkda$, or using finite terms to approximate the function, $\hat{f}(x) = \sum_{i=1}^{l} W_i \psi_{k,a}(x)$. For multi-dimensional data, which will be encountered in SVM, applying the tensor theory from Zhang and Benveniste (1992) results in a multi-dimensional wavelet function defined as $\psi_d(\mathrm{x}) = \prod_{j=1}^{d} \psi(x_j)$ where $\{\mathrm{x}=(x_1,...,x_d) \in \Re^d\}$.

The fundamental motivation to combine the wavelet and the SVM is that by constructing a wavelet kernel that satisfies the Mercer theorem, any arbitrary function can be optimally approximated in the space spanned by the multi-dimensional wavelet basis. Zhang *et al.* (2004) proposed two types of wavelet kernel, the dot productive kernel and the translation invariant kernel, which are calculated as follows:

Dot-product wavelet kernel: $K(\mathrm{x},\mathrm{x}') = K(<\mathrm{x},\mathrm{x}'>) = \prod_{j=1}^{d} \psi(\frac{x_j - k_j}{a}) \psi(\frac{x_j' - k_j'}{a})$.

Translation invariant kernel: $K(\mathrm{x},\mathrm{x}') = K(\mathrm{x}\text{-}\mathrm{x}') = \prod_{j=1}^{d} \psi(\frac{x_j - x_j'}{a})$

Zhang *et al.* (2004) also set the necessary and sufficient conditions for the kernels so that they satisfy Mercer's theorem and can be applied as admissible SV kernels in Hilbert space. Based on those conditions, Zhang *et al.* (2004) construct a translation invariant kernel using the Morlet wavelet function and show that it is superior to the Gaussian function based kernel in both unitary and binary examples. Moreover, compared with the Gaussian kernel, which is correlative and redundant, the wavelet kernel is orthonormal or approximately orthonormal. This property can lead to increased training speed and will be superior when managing high dimensional data. The current paper utilizes the Morlet wavelet kernel with the kernel function $\psi(x) = \cos(1.75x)\exp(\frac{x^2}{2})$ and assesses its performance when combined with SVM in estimating APARCH model.

3. Model Specification and experiment design

A short description of the standard GARCH (1,1) model is presented for further generalization in the APARCH model. The form of the standard GARCH (1,1) model is as follows:

$$\begin{aligned} u_t &= \eta_t \sqrt{h_t};\ \eta_t \sim i.i.d.(0,1) \\ h_t &= w + \alpha u_{t-1}^2 + \beta h_{t-1} \end{aligned}, \qquad (12)$$

where $w > 0,\ \alpha \geq 0, \beta \geq 0$ to ensure a positive conditional variance and condition $\alpha + \beta < 1$ should be satisfied such that the GARCH series is weakly stationary. he stochastic process h_t is the conditional variance of u_t with $u_t \left| \mathrm{I}_{t-1} \right. \sim D(0, h_t)$, where D is the distribution and I_{t-1} denotes the available information at time $t-1$. Volatility h_t can be predicted by a weighted average of the constant long run unconditional variance, the first lag of the squared residual, and the lag one conditional variance, with the weights w, α and β. The restriction

$w+\alpha+\beta=1$ is imposed to ensure that the long run unconditional variance $\frac{w}{1-\alpha-\beta}$ is equal to 1. The standard GARCH model has been widely applied due to its ability to capture volatility persistence and clustering. However, as its linear structure only allows correlations to exist in squared residuals, negative shocks and positive shocks to the series will result in the same impact in predicting the volatility. As the volatility in financial return series tends to be more affected by negative events, relative to positive events of similar magnitude, the linear GARCH model is not able to manage this "leverage effect". To resolve this problem, Ding *et al*. (1993) introduced the APARCH model, which can capture the asymmetric effect of "negative news" and "positive news" in the stock market. The model structure is then the following:

$$\begin{aligned} &u_t=\eta_t\sqrt{h_t};\quad \eta_t\sim i.i.d.(0,1)\\ &h_t^{\delta/2}=w+\alpha(\left|u_{t-1}\right|-\gamma u_{t-1})^{\delta}+\beta h_{t-1}^{\delta/2}, \end{aligned} \tag{13}$$

where $\delta>0,-1<\gamma<1,\ w>0,\alpha>0,\ \beta>0$, and the conventional stationary condition is $\alpha(1+\gamma^2)+\ \beta<1$. This model introduces the power coefficient δ and the leverage coefficient γ. The power term δ allows other power digits in the data transform instead of only the second order in the GARCH model, and parameter γ controls the asymmetric volatility response to positive and negative returns. The APARCH model is a general class of model, which consists of a family of models such as the GARCH model with δ=2 and γ=0, the GJR-GARCH model by Glosten *et al.* (1993) with δ=2, the TS-GARCH model of Taylor (1986) and Schwert (1989) with δ=1 and γ=0 and the T-ARCH model of Zakoian (1993) with δ=1. Although the various models have special applications in particular circumstances, the estimations of the APARCH model are generally measured by Maximum Likelihood when D is a normal distribution or Quasi Maximum Likelihood when D is a non-normal distribution. Bollerslev and Wooldridge (1992) show that QML provides consistent estimators; however, QML is inefficient and cannot provide the best estimate for finite sample sizes. More flexible tools are required when skewness and kurtosis are detected in the series, and the pure data based SVM may be an elegant choice. The current paper investigates the estimation performance and forecasting ability of the QML and SVM when the distribution of the data is set as a skewed Student-t distribution, to capture both fat tails and asymmetry in financial series.

To estimate the parameters in APARCH with QML, the log-likelihood function provides a maximized conditional on a set of samples when the distributions for the innovations are specified. For the nonparametric SVM estimation, there is no specified parameter that must be estimated, and the most important issue is identifying the output and input variables for function $f(\mathrm{x}_t)$. Applying SVM to estimate the APARCH model is not purely nonparametric, as the model framework must specify the output scalar and the input vector. As the primary goal in the present research is to forecast volatility, the output variable is naturally chosen to be $h_t^{\delta/2}$, and the variable δ is already known based on the model types. The input vectors will vary based on whether γ are available or not. If γ is given, the input x_t is $[(|u_{t-1}|-\gamma u_{t-1})^{\delta},\ h_{t-1}^{\delta/2}]$; if γ is not known, the power term is expanded to a linear form and the input will differ according to the model types as follows: $\mathrm{x}_t=[u_{t-1}{}^2,\ h_{t-1}]$ for GARCH model, $\mathrm{x}_t=[|u_{t-1}|,\ h_{t-1}^{1/2}]$ for TS-GARCH, $\mathrm{x}_t=[|u_{t-1}|,\ u_{t-1},\ h_{t-1}^{1/2}]$ for TARCH and $\mathrm{x}_t=[u_{t-1}{}^2,\ |u_{t-1}|u_{t-1},\ h_{t-1}]$ for GJR-GARCH model. Another important issue is that although the volatility h_t is available and can be used directly in simulated data; for empirical series obtained from financial market, the volatility h_t is unobservable. A feasible resolution is suggested by Perez-Gruz *et al.* (2003), where they set $h_t^{'}=\frac{1}{5}\sum_{k=0}^{4}u_{t-k}^2$ as the measurement for h_t. Because our simulation results show that $h_t^{'}=\frac{1}{5}\sum_{k=0}^{4}u_{t-k}^2$ will result in an over-smoothing of the volatility and reduce the asymmetric style of the series, we choose the formula $h_t^{'}=\frac{1}{3}\sum_{k=0}^{4}u_{t-k}^2$. However, the actual volatility h_t can be utilized later when we evaluatethe result by the normalized mean square error: $NMSE_h=\dfrac{\frac{1}{n}\sum_{t=1}^{n}(\hat{h}_t^{'}-h_t)^2}{\frac{1}{n-1}\sum_{t=1}^{n}(h_t-\bar{h})^2}$ where $\bar{h}=\sum_{t=1}^{n}h_t$ and $\hat{h}_t^{'}$ are estimated by SVM. For real data where only u_t is available because $u_t=\eta_t\sqrt{h_t}$ and $\eta_t\sim i.i.d.(0,1)$ is independent with h_t, we obtain $Eu_t^2=E\eta_t^2h_t=Eh_t$, and thus, the criteria can

be set as $NMSE_u = \dfrac{\frac{1}{n}\sum_{t=1}^{n}(\hat{h}_t^{'} - u_t^2)^2}{\frac{1}{n-1}\sum_{t=1}^{n}(u_t^2 - \bar{u}^2)^2}$ where $\bar{u}^2 = \sum_{t=1}^{n} u_t^2$. The evaluation for the performance of the estimation will be performed considering two aspects: the in-sample training data are used to evaluate model fitting, while the out-of-sample test data are applied to evaluate the predictive ability.

4. Monte Carlo Experiment and result comparison.

We first need to parameterize the four models before generating data, and the parameters are set to be weakly stationary:

GARCH(1,1) model with δ =2 and γ =0: $w = 0.2, \alpha = 0.5, \beta = 0.3$;

TS-GARCH model with δ =1 and γ =0: $w = 0.2, \alpha = 0.5, \beta = 0.3$;

GJR-GARCH model with δ =2: $w = 0.2, \alpha = 0.5, \beta = 0.3, \gamma = 0.3$;

T-ARCH model with δ =1: $w = 0.2, \alpha = 0.5, \beta = 0.3, \gamma = 0.5$.

The distributions of the innovations are Student's-t distributions with six degrees of freedom with the non-central parameter μ set to (-0.5, 0.5). Parameter μ controls the asymmetry of the distribution with μ >0 denoting a heavier right tail. The sample size for the series is 1000, with the first half as training data and last half as testing data. The free parameters *C* and ε are tuned by 10-fold cross-validation error. The combinations that minimize the validation error are chosen to adjust the weights α_i based on the training data. The same *C* and ε are applied for both the Gaussian kernel-based SVM and the wavelet kernel-based SVM for further comparison. The hyper parameter σ in the Gaussian kernel is determined based on suggestion from Caputo *et al.* (2002), where the optimal values are any values between the 0.1 and 0.9 quantiles of $\left\| x - x^{'} \right\|^2$. We first simulate one data set from the GJR model and graph the estimation and prediction performance of three approaches.

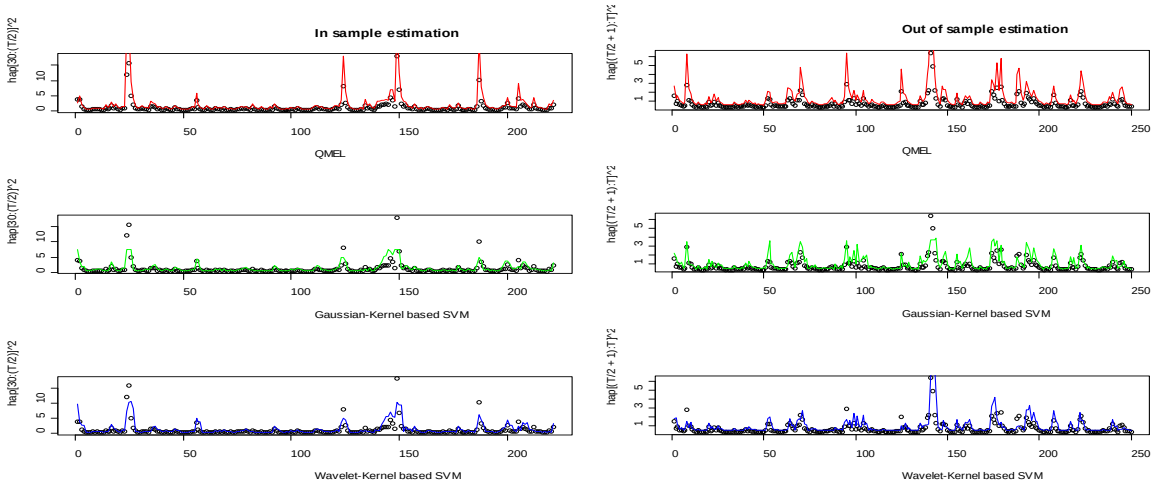

Figure 1: Estimating and forecasting results for one data trial

Figure 1 depicts the actual and estimated or predicted conditional variance h_t for both training and testing data. We see that the QML-based method can capture all of the volatility but tends to exaggerate the volatility to a greater extent than the estimated volatility from SVM. This performance partially confirms research by Acosta *et al.* (2002), where they mention that the ML estimation of the GARCH type of model tended to overestimate the volatility magnitude. For the Gaussian kernel-based SVM, although it failed to capture the large volatility in the training data, it provided better predictions in the testing dataset. However, even for the in-sample data, the overall performance of the Gaussian kernel-based SVM is better than the QML estimation. The wavelet-based SVM, although it slightly underestimates the volatility for the training data, provides the best prediction in the out-of-sample data in the three cases. Next, we run 100 independent trials with the above-mentioned parameter settings and choose the median and mean values and the smallest value of the *NMSE* for comparison. The results are reported in the following three tables.

Table 1: Estimated result based on the QML method

μ	*In sample*						*Out of sample*					
	$NMSE_h$			$NMSE_u$			$NMSE_h$			$NMSE_u$		
	Avg	Med	Low	Avg	Med	Low	Avg	Med	Low	Avg	Med	Low
GARCH												
-0.5	0.842	0.683	0.123	1.131	1.161	0.587	0.801	0.737	0.138	56.45	2.460	0.701
0.5	0.821	0.630	0.069	1.135	1.158	0.392	0.719	0.749	0.025	13.50	3.370	0.949
TSGARCH												
-0.5	0.657	0.633	0.371	0.934	0.942	0.862	0.734	0.864	0.317	2.578	1.761	0.896
0.5	0.622	0.589	0.407	0.929	0.931	0.875	0.716	0.698	0.014	3.202	1.877	1.016
GJR												
-0.5	0.864	0.762	0.120	1.056	0.888	0.221	0.840	0.801	0.101	257.5	4.345	0.831
0.5	0.882	0.687	0.021	0.827	0.881	0.230	0.888	0.856	0.035	35.90	1.340	0.940
TGARCH												
-0.5	0.959	0.968	0.790	0.990	0.997	0.909	0.987	0.829	0.041	61.97	2.333	0.633
0.5	0.589	0.525	0.330	0.877	0.887	0.760	1.117	1.081	0.043	1.423	1.383	1.014

To obtain the *NMSE* for in-sample and out-of-sample data, we need the fitted and forecasted volatility $\hat{h}_t'$. The fitted $\hat{h}_t'$ values are derived from the QML, while the forecasted $\hat{h}_t'$ is calculated from the APARCH model with estimated parameters based on the testing data. Table 1 indicates that for the $NMSE_h$, the out-of-sample and the in-sample values are similar to each other. However, for the $NMSE_u$, the out-of-sample values are much larger and less stable. When we verify the average values of the $NMSE_u$, we observe that many values are quite large and the range of the value varies significantly, supporting the notion that the QML-based method is not efficient in finite samples. This inefficiency is in part due to the normal departure distribution of the data; for the skewed Student-*t* distribution, it is common to see outliers, even for small samples of data. As the extreme value could not be captured by the fixed model structure, the prediction is likely to be unstable.

We next use the SVM approach to train the data, and the fitted $\hat{h}_t'$ and forecasted $\hat{h}_t'$ are determined by specifying the input vectors according to different types of APARCH models. The results are shown in Table 2:

Table 2: Estimated result based on the Gaussian-Kernel based SVM method

μ	*In sample*						*Out of sample*					
	$NMSE_h$			$NMSE_u$			$NMSE_h$			$NMSE_u$		
	Avg	Med	Low	Avg	Med	Low	Avg	Med	Low	Avg	Med	Low
GARCH												
-0.5	0.559	0.591	0.289	0.753	0.771	0.571	1.243	0.982	0.622	1.045	1.042	0.973
0.5	0.343	0.345	0.182	0.400	0.418	0.221	2.836	0.976	0.522	1.075	1.065	0.927
TSGARCH												
-0.5	0.415	0.437	0.234	0.704	0.735	0.440	0.723	0.787	0.194	1.130	1.115	0.978
0.5	0.386	0.380	0.166	0.770	0.781	0.604	0.679	0.687	0.333	1.127	1.131	0.996
GJR												
-0.5	0.263	0.255	0.071	0.508	0.536	0.252	1.090	0.921	0.569	1.041	1.034	0.949
0.5	0.589	0.612	0.387	0.848	0.866	0.751	0.905	0.905	0.470	35.69	1.040	0.978
TGARCH												
-0.5	0.758	0.814	0.425	0.856	0.881	0.618	0.909	0.980	0.432	1.030	1.024	0.975
0.5	0.803	0.803	0.368	0.847	0.863	0.598	0.878	0.869	0.615	1.154	1.139	0.971

Table 2 shows better outcomes, in general, relative to the QML-based method for both in-sample and out-of-sample values. A more stable average value, especially for the $NMSE_u$, was observed in the Gaussian Kernel-based SVM method than in the QML analysis, indicating more efficient prediction when using the Gaussian Kernel-based SVM method. The results are unsurprising because, when training by SVM, the models are adjusted to an optimal density that fits the data best and simultaneously minimizes the prediction error. Moreover, the entire procedure is purely data driven and can be flexible even when including the extreme values and outliers. One could argue that current software packages commonly include the QMLE under the non-normal distribution, and the model can still be estimated

based on the parametric structure. However, in such situations, more parameters (e.g., skew parameters and degrees of freedom) must be specified and misspecification will undermine both estimation and forecasting. The SVM based methods generate these parameters, and only the structures of the input data are required. A more general case is illustrated in Tay and Cao (2002), where the SVM is applied as a pure nonparametric method where no information about the model's structure is needed. Such a design would be especially suitable for data that cannot be described by any single specific model.

The following will show the application of the wavelet kernel in the SVM approach; the results are shown in Table 3.

Table 3: Estimated result based on Wavelet-Kernel based SVM method

μ	*In sample*						*Out of sample*					
	$NMSE_h$			$NMSE_u$			$NMSE_h$			$NMSE_u$		
	Avg	Med	Low	Avg	Med	Low	Avg	Med	Low	Avg	Med	Low
GARCH												
-0.5	0.286	0.294	0.134	0.673	0.694	0.417	0.615	0.715	0.085	1.092	1.000	0.873
0.5	0.218	0.220	0.043	0.551	0.549	0.225	0.585	0.629	0.072	1.322	1.339	0.804
TSGARCH												
-0.5	0.312	0.324	0.096	0.735	0.774	0.407	0.638	0.685	0.188	1.195	1.101	0.778
0.5	0.317	0.324	0.117	0.763	0.785	0.497	0.573	0.569	0.118	1.191	1.115	0.999
GJR												
-0.5	0.217	0.225	0.063	0.603	0.614	0.202	0.864	0.757	0.257	1.283	1.340	0.859
0.5	0.446	0.461	0.263	0.777	0.791	0.534	0.761	0.757	0.340	1.010	1.126	0.900
TGARCH												
-0.5	0.522	0.567	0.110	0.742	0.775	0.345	0.666	0.849	0.014	1.360	1.063	0.968
0.5	0.911	0.889	0.359	0.860	0.867	0.650	0.879	0.932	0.536	1.186	1.172	0.996

The wavelet kernel-based SVM outperforms the Gaussian kernel-based SVM with a larger number of smaller values of *NMSE* and no extreme average values. Table 2 contains one extreme value (35.69) for the $NMSE_h$ in GJR model with $\gamma = 0.5$, while Table 3 shows that the average values are all less than 1.5. One explanation for this result is that the wavelet kernels are constructed to capture the local characteristics in the series and that the wavelet kernel can capture the non-stationary dynamics of the data, such as the structure break and abrupt values. This ability is driven by the volatility clustering model under the fat tail distribution, as the wavelet can handle both local volatility and outliers quite well. It is also interesting to compare these results to those of Chen *et al.* (2010), where they compared the linear kernel, the polynomial kernel and the Gaussian kernel and concluded that no single kernel dominated the volatility predictions. The present paper shows that the wavelet kernel provides consistently better results than the Gaussian kernel in the APARCH model.

In general, for both in-sample and out-of-sample data, most $NMSE_h$ and $NMSE_h$ values present a decreasing trend from Table 1 to Table 3, indicating that the SVM based methods

outperform the QMLE. Moreover, the wavelet kernel-based SVM is more adept at volatility estimation and forecasting than the Gaussian kernel-based SVM. Another appealing property of the wavelet kernel is that the number of support vectors is generally lower than those used in the Gaussian kernel:

Table 4: Number of support vectors in Gaussian and Wavelet kernels

	Gaussian kernel		Wavelet kernel	
	$\mu = -0.5$	$\mu = 0.5$	$\mu = -0.5$	$\mu = 0.5$
GARCH	271	267	231	201
GJR	198	181	141	139
TSGARCH	468	300	468	205
TGARCH	174	346	146	305

Table 4 shows that under the same free parameter setting, the number of support vectors in the wavelet kernel-based SVM is lower than in the Gaussian kernel-based SVM. The number of support vectors is important in SVM application, as a well-performed SVM is expected to approximately outline an entire dataset from a small fraction of input data (see Xiao *et al*., (2005)). For the training data, fewer support vectors will lead to sparse data sets when solving the quadratic programming optimization problem. For testing data, fewer support vectors can provide smaller test error, as the expectation value of the prediction error will be no larger than the ratio between the expectation value of the number of support vectors and the number of training vectors: $E[\Pr(error)] \leq \frac{E[Nr.(Support\ Vectors)]}{Nr.(Training\ Vectors)}$. Compared with the Gaussian kernel, the wavelet kernel provides fewer support vectors in all the cases and indicates more computational efficiency and better generation capability.

5. Conclusion.

The present paper primarily uses the SVM based technique to estimate and predict volatility in the APARCH type of model when the data are skewed Student-*t* distributed. We compare the outcomes with results from the QML estimation, and Monte Carlo simulations show that the SVM based methods outperform the QMLE in both estimation and prediction. As the performance of the SVM depends on the kernel choice in a given circumstance, we further evaluate the SVM with Gaussian and Wavelet kernels. Based on the results, we observe that the wavelet kernel is consistently more accurate than the Gaussian kernel in the APARCH model framework, as the local identification character in the wavelet kernel is well equipped to capture the volatility clustering style for the conditional volatility. Moreover, by applying

the wavelet kernel, fewer support vectors are needed, which simplifies the computation and improves the prediction ability.

References:

Acosta, E., Fernández, F. and Pérez, J. (2002): "Volatility bias in the Garch model: a simulation study", *Working paper 20026-02*, University of Las Palmas de Gran Canaria.

Bollerslev T., (1986): "Generalised Autoregressive Conditional Heteroskedasticity", *Journal of Econometrics*, Vol. 31, pp. 307-327.

Bollerslev, T. and Wooldridge, J.M. (1992): "Quasi-Maximum Likelihood Estimation and Inference in Dynamic Models with Time-Varying Covariances", *Econometric Reviews*, Vol. 11, pp. 143-172.

Caputo, B., Sim, K., Furesjo, F., Smola, A. (2002): "Appearance-based Object Recognition using SVMs: Which Kernel Should I Use?", Proceedings of NIPS Workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision, Whistler, 2002.

Chen, S.Y., Karl, W. (2010): "Forecasting volatility with support vector machine-based GARCH model", *Journal of Forecasting*, Vol. 29, Issue 4, pp. 406-433.

Ding, Z., Granger, C.W.J., Engle, R.F. (1993): "A Long Memory Property of Stock Market Returns and a New Model", *Journal of Empirical Finance*, Vol. 1, pp. 83-106.

Engle, R.F. (1982): "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation", *Econometrica*, Vol. 50, pp. 987-1007.

Engle, R., González-Rivera, G., (1991): "Semi parametric ARCH models", *Journal of Business and Economic Statistics*, Vol. 9, pp. 345-359.

Fernández, C. and Steel, M.F.J. (1998): "On Bayesian modelling of fat tails and skewness", *Journal of the American Statistical Association*, Vol. 93, pp. 359-371.

Glosten, L., Jagannathan, R., Runkle, D. (1993): "On the Relation Between Expected Value and the Volatility of the Nominal Excess Return on Stocks", *Journal of Finance*, Vol. 48, pp. 1779-1801.

Grossman, A. and Morlet, J. (1984): "Decomposition of Hardy functions into square integrable wavelets of constant shape", *Society for Industrial and Applied Mathematics Journal on Mathematical Analysis*, Vol. 15, pp. 732-736.

Karush, W. "Minima of functions of several variables with inequalities as side constraints", Master's thesis, Dept. of Mathematics, Univ. of Chicago, 1939.

Keerthi, S.S. and Lin, C.J., (2003): "Asymptotic behaviors of support vector machines with Gaussian kernel", *Neural Comput*, Vol. 15, pp. 1667-1689.

Kuhn, H.W. and Tucker, A.W. (1951): "Nonlinear programming". In Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics, pp. 481-492, Berkeley, University of California Press.

Mallat, S.G. (1989): "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", Pattern Analysis and Machine Intelligence, IEEE Transactions. Vol. 11, No. 7, pp. 674-693.

Mangasarian, O.L. (1969): *Nonlinear Programming*, McGraw-Hill, New York.

McCormick, G.P. (1983): *Nonlinear Programming: Theory, Algorithms and Applications*, John Wiley and Sons, New York.

Mercer, J. (1909): Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society, London, A 209:415-446.

Schwert, G.W. (1989): "Why does stock market volatility change over time?", *Journal of Finance*, Vol. 44, pp.1115-1153.

Smola, A.J. and Schölkopf, B. (1998a): "On a kernel-based method for pattern recognition, regression, approximation and operator inversion" *Algorithmica*, Vol. 22, pp. 211-231.

Taylor, S. (1986): *Modelling Financial Time Series*, Wiley, New York.

Tang, L, Sheng, H.Y.and Tang, L.X. (2009): "Forecasting Volatility based on wavelet support vector machine", *Expert Systems with Applications*, Vol.36, Issue 2, pp.2901-2909.

Tay, F.E.H. and Cao, L. (2002): "Modified support vector machines in financial time series forecasting", *Neurocomputing*, Vol. 48, pp. 847-861.

Ou, P. and Wang, H.S. (2010): "Financial Volatility Forecasting by Least Square Support Vector Machine Based on GARCH, EGARCH and GJR Models: Evidence from ASEAN Stock Markets", *International Journal of Economics and Finance*, Vol. 2, No. 1, pp. 51-64.

Préz-Cruz, F., Afonso-Rodriguez, J.A. and Giner, J. (2003): "Estimating GARCH models using support vector machines", *Journal of Quantitative Finance*, Vol. 3, pp. 1-10.

Vapnik, V. (1995): *The Nature of Statistical Learning Theory*. Springer, New York.

Vapnik, V. and Chervonenkis, A. (1974): *Theory of Pattern Recognition*, (in Russian). Nauka, Moscow; German translation: *Theorie der Zeichenerkennung*, Akademie Verlag, Berlin, 1979.

Vapnik, V. and Lerner, A. (1963): "Pattern recognition using generalized portrait method", *Automation and Remote Control*, Vol. 24, pp.774-780.

Xia, X.L., Michael, R.L., Lok, T.M. and Huang, G.B. (2005): "Methods of Decreasing the Number of Support Vectors via k-Mean Clustering", *Lecture Notes in Computer Science*, Vol. 3644, pp.717-726.

Zakoian, J.M. (1994): "Threshold Heteroskedasticity Models", *Journal of Economic Dynamics and Control*, Vol. 15, pp. 931-955.

Zhang L., Zhou, W.D. and Jiao, L.C. (2004): "Wavelet Support Vector Machine", Systems, Man, and Cybernetics—Part B: Cybernetics, IEEE Transactions, Vol. 34, No.1, pp. 34-39.

Zhang, Q. and Benveniste, A. (1992): "Wavelet networks", Neural Networks, IEEE Transactions, Vol. 3, Issue 6, pp. 889-898