Volatility Forecasting with Sparse Bayesian Kernel Models

Peter Tiňo[1] **Nikolay Nikolaev**[2] **Xin Yao**[1]

[1]The University of Birmingham Birmingham B15 2TT, UK
[2]Goldsmiths College, University of London, London SE14 6NW, UK

Abstract

Motivated by previous findings that discretization of financial time series can effectively filter the data and reduce the noise, this experimental study, performed in a realistic setting of trading straddles via predicting volatility, compares trading performances of symbol-based models with those of probabilistic models operating on real-valued sequences. We show that carefully designed probabilistic models trained in a Bayesian framework of automatic relevance determination can achieve superior trading performances.

1 Introduction

It has been frequently argued in the past that quantizing real-valued financial time-series into symbolic streams and subsequent use of predictive models on such sequences can be of great benefit in many financial tasks (e.g. [1, 3, 7]. This is predominantly due to the inherently noisy and non-stationary nature of financial data. Careful quantization can reduce the noise component in the data while preserving the underlying predictable patterns in the stochastic process[1].

Recently, we performed large comparative studies of various model classes used to predict daily volatility differences in order to trade (on a daily basis) straddles on the DAX and FTSE 100 indexes [8, 7]. The straddles were traded based on predictions of daily (implied[2]) volatility differences in the underlying indexes. Continuous models (such feed-forward neural networks, mixture density networks, AR models) operating on the original real-valued sequences of volatility differences, as well as symbolic models (fixed order Markov models, variable memory length Markov models, fractal prediction machines) trained and tested on their quantized counterparts were considered. Two key observations were made: **(1)** quantization technique significantly improves the overall profit, and **(2)** quantization into just two symbols representing the *sign* of daily volatility differences gave the best results.

In this contribution, we would like to add another token to the 'discretize vs. do not discretize' debate by applying continuous models inherently capable of dealing with noise in the data. In particular, we use a probabilistic model, the Noninformative Prior Relevance Vector Machine (NPRVM), based on the Relevance Vector Machine (RVM) [9].

[1]Recently, this issue has been a source of some controversy and confusion. Due to the space limitation, in this contribution we will not follow this issue any further.

[2]calculated using Black-Scholes model

2 RVM with noninformative prior – NPRVM

Modeling of financial time series may be regarded as a multivariate regression problem. Given a series of scalar observables (volatility differences): $..., x_t, x_{t+1}, ..., x_T$ sampled at discrete time intervals, the goal is to find a model that accurately describes how future values depend on the past values. The future values are predicted based on model operating on lagged delay vectors:

$$\hat{x}_{t+1} \equiv y_t = f(\mathbf{x}_t) = f(x_{t-(d-1)\tau}, x_{t-(d-2)\tau}, ..., x_t) \quad (1)$$

where d is the embedding dimension and τ is the delay time. The original series is transformed into a data set $D = \{(\mathbf{x}_t, y_t)\}_{t=1}^{N}$, where the $\mathbf{x}_t \in \mathcal{R}^d$ are independent variable vectors and $y_t \in \mathcal{R}$ is the dependent variable. In a generalized linear kernel regression formulation [6]:

$$f(\mathbf{x}) = \sum_{n=1}^{M} w_n K(\mathbf{x}, \mathbf{x}_n), \qquad (2)$$

where w_n are the weight parameters and $K(\cdot, \cdot)$ is the kernel basis function. We use Gaussian kernels of width s^2: $K(\mathbf{x}, \mathbf{x}_n) = exp[-(\mathbf{x} - \mathbf{x}_n)^T(\mathbf{x} - \mathbf{x}_n)/(2s^2)]$. The future values are modeled as $x_{t+1} = f(\mathbf{x}_t)+\varepsilon_t$, where ε_t is i.i.d. zero-mean Gaussian noise with (unknown) variance σ^2.

The RVM [9] provides a framework for Bayesian learning of sparse kernel regression models. Starting with the complete model, having basis functions centered on all given data points, it adapts simultaneously the weights and their associated hyperparameters. It is assumed that the prior distribution $p(\mathbf{w}|\alpha)$ over the weights $\mathbf{w}$ is an M-dimensional Gaussian of zero mean and covariance matrix $\mathbf{\Gamma}(\alpha) = diag(\alpha_1^{-1}, \alpha_2^{-1}, ..., \alpha_M^{-1})$. The hyperparameters $\alpha = (\alpha_1, \alpha_2, ..., \alpha_M)$ quantify the prior belief in the possible ranges of weight values, and the hyperparameter β quantifies the (inverse variance of) output noise.

Posterior distribution over the weights reads

$$p(\mathbf{w}|\mathbf{y}, \alpha, \beta^{-1}) = \frac{p(\mathbf{y}|\mathbf{x}, \mathbf{w}(\alpha), \beta^{-1})p(\mathbf{w}|\alpha)}{p(\mathbf{y}|\alpha, \beta^{-1})} \qquad (3)$$

where $p(\mathbf{y}|\mathbf{x}, \mathbf{w}(\alpha), \beta^{-1})$ is the likelihood and $p(\mathbf{w}|\alpha)$ is the prior over weights; $p(\mathbf{y}|\alpha, \beta^{-1})$ is the normalizing factor and $\mathbf{w}(\alpha)$ denotes explicit dependence on the argument. The RVM carries out re-estimation of the prior hyperparameters and weights. During the learning process some hyperparameters grow thus causing their corresponding weights to shrink toward zero. In practice, all training points $\mathbf{x}_n$ with the corresponding hyperparameter α_n bellow a (predefined) threshold α_{MAX} are pruned out from the model.

The RVM performs adaptation by iterative maximization of the marginal likelihood of the hyperparameters $p(\mathbf{y}|\alpha, \beta^{-1})$ by following the evidence procedure of [4]. However, the resulting formulae can be problematic to compute [5] as they are sensitive to the numerical accuracy of the computers[3] Therefore, we use a different noninformative prior over the weight parameters to obtain more stable hyperparameter estimates. Following [2], we use the Jeffreys' prior in order to adjust the hyperparameters, while keeping the noise variance fixed.

[3]Even if the intermediate quantities in these formulae are computable, they are numerically unstable and can incur errors in the learning process.

After adapting the hyperparameters, the MAP estimates for weights can be obtained analytically (posterior over weights, $p(\mathbf{w}|\mathbf{y}, \alpha, \beta^{-1})$, is a Gaussian):

$$\mathbf{w} = \beta(\beta\mathbf{K}^T\mathbf{K} + \mathbf{A})^{-1}\mathbf{K}^T\mathbf{y} \qquad (4)$$

where $\mathbf{A} = \mathbf{\Gamma}^{-1}(\alpha) = diag(\alpha_1, \alpha_2, ..., \alpha_M)$ is the diagonal matrix with the weight prior hyperparameters, and $\mathbf{K}$ is the kernel design matrix with entries $K(\mathbf{x}_i, \mathbf{x}_j)$, $1 <= i, j <= N$.

Having individual regularizers causes different ranges among the weight parameters. The overall effect of this is inducing model sparseness and hence smoothing of the model. The selected basis functions are usually only a fraction of the full superposition of kernels centered on all given data. Given a test input $\mathbf{x}_*$, the mean of the output distribution $p(y_*|\mathbf{x})$ is $y_* = \mathbf{w}^T\mathbf{k}(\mathbf{x}_*)$, where $\mathbf{k}(\mathbf{x}_*) = [K(\mathbf{x}_*, \mathbf{x}_1), K(\mathbf{x}_*, \mathbf{x}_2), ..., K(\mathbf{x}_*, \mathbf{x}_M)]^T$, and $\mathbf{w} = [w_1, w_2, ..., w_M]^T$ is the weight vector.

3 Data, trading strategy and experimental results

First data set is a series of daily closing values of the German stock index DAX together with a series of daily closing prices of call and put options on the DAX with different maturities and exercise prices. In particular, the first in-the-money and the first out-of-the money call and put option maturing next month are available. The at-the-money point is assumed to be the value of the DAX at that time[4]. Straddle prices are obtained by adding call and put prices. The series start on 22 August 1991 and end on 8 June 1998 which corresponds to a period of 1700 trading days.

Second data set comprises transactions data of FTSE 100 option contracts traded at the London International Financial Futures Exchange (LIFFE). Intraday bid-ask prices of American options on the FTSE 100 between 29 May 1991 and 29 December 1995 are available. This time period corresponds to 1161 trading days. The option prices are recorded synchronously with the FTSE 100 and time-stamped to the nearest second. Since our trading strategy is set up on a daily basis, we must fix a reference point in time on each trading day. This reference point is 3 pm on normal trading days and 12 pm on days

[4]It would be probably more appropriate to take the value of the futures contract with the same maturity as the at-the-money point.

where the stock exchange closes earlier. The first quotes of call and put options maturing the next month with the same strike price as close as possible to the value of the FTSE 100 at that time are extracted for the actual trading day (and also for the next trading day). For these options, which are roughly at-the-money, the average of bid-ask quotes is calculated as an approximation of a reasonable option price. Then the prices of call and put options are added to obtain straddle prices.

The basic trading strategy is simple [8, 7]: Every trading day, predict the change in volatility for the next trading day. If volatility is predicted to increase, buy near-the-money straddles, otherwise sell them. On the next trading day, close the position and restart by predicting the next volatility change. The straddles bought or sold every day are near-the-money, i.e., the strike price closest to the at-the-money point is selected. Therefore, the straddle portfolio is approximately delta-neutral which means that there is no need to delta-hedge the position[5]. Every day, after predicting the direction of change in volatility, straddles worth a fixed amount of money are bought or sold. The choice of a fixed but otherwise arbitrary investment is intended to facilitate the interpretation of results with respect to transactions costs. Finally, only straddles maturing the following month are bought or sold, which avoids the influence of strong price movements towards the end of the contracts.

An important issue when dealing with financial time series analysis is stationarity. A useful method to deal with non-stationarity of data is the 'sliding window technique': The models are estimated and evaluated on the first time window. Then the sliding window is shifted by one or several steps. The models are estimated and evaluated on the second time window and so on.

Each time window consists of three parts: a training set, a validation set, and a test set. The training set is used to estimate the model parameters. The size of the training set is 500 which means that the volatility measure is collected over a period of roughly two years. Several representatives from the class of NPRVM models with different pruning cut values ($\alpha_{MAX} = 0.25$ and $\alpha_{MAX} = 0.5$) and kernel width s^2 (input lag is fixed to $d = 10$) are estimated. Performance of the estimated models (accumulated profit) on the validation set is the criterion for selecting the best model within the model class. Size of the validation set is 125 trading days (roughly half a year). Finally, the out-of-sample profit of the best model is determined on the test set which covers 5 trading days (one week). Then the sliding window with a total size of 630 days is shifted by 5 days, the models are re-estimated etc. In particular, the test sets are non-overlapping and the obtained profits can be concatenated to form a large series of out-of-sample profits.

Following [8, 7]. we tested for significance of the profits by dividing the series into distinct blocks of length 60 and 40 for DAX and FTSE series, respectively. Due to the central limit theorem, the average block profit can be assumed to be normally distributed. Hence we can subject the series of average block profits to t-tests[6]

As in [8, 7], we also consider an additional model class '*Simple*' that picks one of the four trivial strategies – 'Always Sell', 'Always Buy', 'Copy the last trading decision' and 'Reverse the last trading decision' – based on their validation set profit. This model class can gain surprisingly large profits, because it totally eliminates the need for a training set potentially containing old (no longer relevant) data. Compound models '*NPRVM+Simple*' make predictions in the test week using either the more sophisticated model (NPRVM), or 'Simple', depending on which model gained more profit on the validation set.

The results are presented in table 1. For comparison, we also include performances of the following models classes [8]: **NN(1)** - feedforward neural networks with 3 hidden nodes (tanh transfer function), and a linear output, the input lag (up to 10) is determined on the validation set); **MM(5)** - Markov models of order up to 5 (the order is determined on the validation set) operating on quantized sequences of volatility differences (binary alphabet); **MM(10)** - the same as MM(5), but the order can now be set to 1,2,...,10 (determined on the validation set).

4 Conclusion

Contrary to our previous conclusions that models built on quantized sequences give superior results, when compared to those constructed on the original real-valued time series, we show that carefully designed probabilistic models trained and regularized

[5]The sensitivities with respect to the interest rate and the time to maturity are negligible and inevitable, respectively.

[6]After calculating the average block profits, a Jarque-Bera test does not reject the null hypothesis of a normal distribution at any reasonable significance level.

Table 1: Profits of the models in the DAX experiment. We also report profits that would be made by the hypothetical always-correct-predictor, `ACP`, perfectly predicting all the volatility changes. To make a more realistic assessment of the achieved profits, we also report the maximal transaction costs (TC) that one can subtract from each daily profit, so that when subjected to the t-test (p=0.05), the average block-profits are still significantly positive.

Model class	% profit per-day Mean	Std.	Highest TC
ACP	1.310	0.652	1.03
Simple	0.477	0.638	0.21
NPRVM	0.514	0.434	0.34
NPRVM+Simple	0.526	0.572	0.29
NN(10)	0.033	0.576	–
NN(10)+Simple	0.405	0.554	0.18
MM(5)	0.262	0.603	0.01
MM(5)+Simple	0.430	0.458	0.24
MM(10)	0.208	0.668	–
MM(10)+Simple	0.425	0.578	0.19

Table 2: Profits of the models in the FTSE 100 experiment.

Model class	% profit per-day Mean	Std.	Highest TC
ACP	2.706	1.109	2.13
Simple	1.562	1.135	1.01
NPRVM	3.234	2.804	1.85
NPRVM+Simple	2.018	1.615	1.22
NN(10)	1.331	1.095	0.77
NN(10)+Simple	1.432	1.131	0.85
MM(5)	1.551	0.833	1.12
MM(5)+Simple	1.551	0.833	1.12
MM(10)	1.490	0.894	1.03
MM(10)+Simple	1.489	0.894	1.03

in a Bayesian framework of automatic relevance determination lead to superior trading performances. Whereas in our previous experiments, the strongest models were the compound models combining flexibility of more sophisticated models with stability of the 'Simple' model class, the NPRVM achieve significantly higher profits ($p < 5\%$) and combination with 'Simple' actually makes their trading performance worse.

References

[1] Buhlmann P. Extreme events from return-volume process: a discretization approach for complexity reduction. Applied Financial Economics 1998; (8): 267-278.

[2] M. Figueiredo, Adaptive sparseness for supervised learning, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.25, N:9, pp.1150-1159, 2003.

[3] Giles CL, Lawrence S, and Tsoi AC. Rule inference for financial prediction using recurrent neural networks. In Proceedings of the conference on Computational Intelligence for Financial Engineering, New York City, NY. 1997, pp. 253-259.

[4] D.J.C. MacKay, Bayesian interpolation, *Neural Computation*, vol.4, N:3, pp.415-447, 1992.

[5] D.J.C. MacKay, Comparison of approximate methods for handling hyperparameters, *Neural Computation*, vol.11, N:5, pp.1035-1068, 1999.

[6] B. Schölkopf and A.J. Smola, *Learning with Kernels*, Cambridge, MA: The MIT Press, 2002.

[7] Ch. Schittenkopf, P. Tino and G. Dorffner, The Benefit of Information Reduction for Trading Strategies, *Applied Financial Economics*, vol.34, N:7, pp.917-930, 2002.

[8] P. Tino, Ch. Schittenkopf, G. Dorffner, Volatility trading via temporal pattern recognition in quantized financial time series, *Pattern Analysis and Applications*, vol.4, N:4, pp.283-299, 2001.

[9] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, *Journal of Machine Learning Research*, vol.1, pp.211-244, 2001.