Volatility Forecasting with Sparse Bayesian Kernel Models

Peter Tiño
University of Birmingham, UK

Nikolay Nikolaev
Goldsmiths College, University of London, UK

Xin Yao
University of Birmingham, UK
Some motivations

- Quantizing real-valued financial time-series into symbolic streams and subsequent use of predictive models on such sequences - Good/Bad?

- Careful quantization can reduce the noise component in the data while preserving the underlying predictable patterns in the stochastic process (Buhlmann 1998, Giles 1997, Schittenkopf 2002).

- Still a controversial topic (Lin 2004).

- 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 (Tino 2001, Schittenkopf 2002).
Previous work (Tino 2001, Schittenkopf 2002)

- (At-the-money) straddles traded based on predictions of daily (implied/historical) volatility differences in the underlying indexes. Backtesting, but a realistic trading setting.

- **Continuous models** - operating on the original real-valued sequences of volatility differences (feed-forward neural networks, mixture density networks, AR models).

- **Symbolic models** operating on the quantized sequences (fixed order Markov models, variable memory length Markov models, fractal prediction machines).
Previous work cont’d

Two key observations:

- quantization technique significantly improves the overall profit
- quantization into just two symbols representing the sign of daily volatility moves gave the best results.

Model formulation

Time series of scalar observables (volatility differences): \( x_1, \ldots, x_t, \ldots, x_T \)

Predictive model operating on lagged delay vectors:

\[
\hat{x}_{t+1} \equiv y_t = f(x_t) = f(x_{t-(d-1)\tau}, x_{t-(d-2)\tau}, \ldots, x_t),
\]

where \( d \) is the embedding dimension and \( \tau \) is the delay time.

Generalized linear kernel regression formulation

\[
f(x) = \sum_{n=1}^{M} w_n K(x, x_n),
\]

\( K(\cdot, \cdot) \) is the kernel basis function, e.g.

\[
K(x, x_n) = \exp[-\|x - x_n\|^2/(2s^2)].
\]
**RVM**

The future values of $x$ are modeled as

$$x_{t+1} = f(x_t) + \varepsilon_t,$$

$\varepsilon_t$ is i.i.d. zero-mean Gaussian noise with (unknown) variance $\sigma^2$.

**RVM:**

- Start with basis functions centered on all given data points.
- ARD framework for weights $w_n$: prior $p(w|\alpha)$ over the $M$ weights is an $M$-dim Gaussian of zero mean and covariance matrix $\Gamma(\alpha) = \text{diag}(\alpha_1^{-1}, \alpha_2^{-1}, \ldots, \alpha_M^{-1})$. The Hyperparameters $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_M)$ quantify the prior belief in the possible ranges of weight values. Hyperparameter $\beta$ quantifies the (inverse variance of) output noise.
RVM cont’d

Posterior distribution over the weights:

\[ p(w|y, \alpha, \beta^{-1}) = \frac{p(y|x, w(\alpha), \beta^{-1})p(w|\alpha)}{p(y|\alpha, \beta^{-1})} \] (3)

RVM carries out re-estimation of the weights and hyperparameters (by maximizing marginal likelihood of the hyperparameters \( p(y|x, \alpha, \beta^{-1}) \)).

Some hyperparameters \( \alpha_n \) grow, causing their corresponding weights \( w_n \) to shrink toward zero. In practice, all training points \( x_n \) with the corresponding hyperparameter \( \alpha_n \) above a (predefined) threshold \( \alpha_{MAX} \) are pruned out from the model.
A stronger bias for sparseness ...

Figueiredo (2003): increase the pressure for model sparseness by considering a Laplacian prior (instead of Gaussian)

\[ p(w|\kappa) = \left( \frac{\kappa}{2} \right)^M \exp(-\kappa \|w\|_1). \]

Can be motivated by assuming that each weight \( w_n \) has a zero-mean Gaussian prior with variance \( \alpha_n^{-1} \) (RVM) and that each variance \( \alpha_n^{-1} \) has an exponential hyper-prior (hierarchical Bayes)

\[ p(\alpha_n|\gamma) = \frac{\gamma}{2} \exp \left( -\frac{\gamma}{2\alpha_n} \right). \]

Hyperparameter \( \gamma \) controls the degree of model sparseness.
Get rid of $\gamma$

Figueiredo (2003): Replace the exponential hyper-prior on variances $\alpha_n^{-1}$ by a non-informative Jeffreys hyper-prior

$$p(\alpha_n^{-1}) \propto \frac{1}{\alpha_n^{-1}}, \quad \text{hence} \quad p(\alpha_n) \propto \alpha_n.$$ 

Treat variances $\alpha_n^{-1}$ as hidden data. EM-style weight update:

$$\hat{w}(t) = \beta(\beta K^T K + A(t))^{-1} K^T y,$$

(4)

where $K$ is the kernel design matrix $K(x_i, x_j), 1 \leq i, j \leq N$, and

$$A(t) = \text{diag}(|w_1(t)|^{-2}, |w_2(t)|^{-2}, \ldots, |w_M(t)|^{-2}).$$

$A$ plays the role of $\Gamma(\alpha)^{-1} = \text{diag}\{\alpha_n^{-1}\}$ in RVM weight estimation. Effectively, variances $\alpha_n$ at time $t$ are estimated as $|w_n(t)|^2$. 

P. Tiño, N. Nikolaev, X. Yao
Data

• **DAX**: Daily closing values of DAX (August 1991 – June 1998); daily closing prices of call and put options on DAX with different maturities and exercise prices. 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. The prices of call and put options are added to obtain straddle prices.

• **FTSE 100**: Intraday bid-ask prices of American options on FTSE 100 at LIFFE (May 1991 – December 1995). Trading at 3 pm on normal trading days and 12 pm otherwise. 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 current FTSE 100 are extracted. For these options (roughly at-the-money), the average of bid-ask quotes is an approximation of the option price.
**Trading strategy**

Only straddles maturing the following month are traded (avoids the influence of strong price movements towards the end of contracts).

Every trading day, predict the change in volatility for the next trading day. If volatility is predicted to increase, buy near-the-money straddles (strike price closest to the at-the-money point) worth a fixed amount of money, otherwise sell them.

On the next trading day, close the position and restart by predicting the next volatility change.

Fixed but otherwise arbitrary investment – facilitate the interpretation of results with respect to transactions costs.
Dealing with non-stationarity

‘Sliding window technique’ (shift by 5 days). Within each window:

- **Training set**: 500 trading days. 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 $d = 10$) are estimated.

- **Validation set**: 125 trading days. Accumulated profit of the estimated models is checked on the validation set – criterion for selecting a model class representative.

- **Test set**: 5 trading days. Out-of-sample profit of the model class representative is determined on the test set.
Volatility Forecasting with Sparse Bayesian Kernel Models

Experimental setup

series of daily volatility differences

Train Valid Test

series of daily test set profits

block 1 block 2 block 3 block n

series of average block-profits

P. Tiño, N. Nikolaev, X. Yao
Estimating significance of the results

The test sets are non-overlapping, test set profits are concatenated to form a large series of out-of-sample profits.

Divide the daily profits into disjoint blocks of length 60 and 40 for DAX and FTSE series, respectively.

The average block profit can be assumed to be normally distributed (central limit theorem). Jarque-Bera test did not reject the null hypothesis of a normal distribution at any reasonable significance level.

Subject the series of average block profits to $t$-tests.
‘Simple’ and Compound models

**Simple**: pick one of the four trivial strategies – ‘Always Sell’, ‘Always Buy’, 'Copy the last trading decision' and 'Reverse the last trading decision' – based on the validation set profit. 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.
Alternative model classes

- **NN(10)**: 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).
We also report ...

- ... profits that would be made by the hypothetical always-correct-predictor, ACP, perfectly predicting all the volatility changes. ... See if the trading setup makes sense...
- ... the maximal transaction costs $T_C$ 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.

First DAX, then FTSE 100 results...
<table>
<thead>
<tr>
<th>Model class</th>
<th>% profit per-day</th>
<th>Highest TC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>ACP Simple</td>
<td>1.310</td>
<td>0.652</td>
</tr>
<tr>
<td>NPRVM</td>
<td>0.514</td>
<td>0.434</td>
</tr>
<tr>
<td>NPRVM+Simple</td>
<td>0.526</td>
<td>0.572</td>
</tr>
<tr>
<td>NN(10)</td>
<td>0.033</td>
<td>0.576</td>
</tr>
<tr>
<td>NN(10)+Simple</td>
<td>0.405</td>
<td>0.554</td>
</tr>
<tr>
<td>MM(5)</td>
<td>0.262</td>
<td>0.603</td>
</tr>
<tr>
<td>MM(5)+Simple</td>
<td>0.430</td>
<td>0.458</td>
</tr>
<tr>
<td>MM(10)</td>
<td>0.208</td>
<td>0.668</td>
</tr>
<tr>
<td>MM(10)+Simple</td>
<td>0.425</td>
<td>0.578</td>
</tr>
<tr>
<td>Model class</td>
<td>% profit per-day</td>
<td>Highest TC</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>ACP</td>
<td>2.706</td>
<td>1.109</td>
</tr>
<tr>
<td>Simple</td>
<td>1.562</td>
<td>1.135</td>
</tr>
<tr>
<td>NPRVM</td>
<td>3.234</td>
<td>2.804</td>
</tr>
<tr>
<td>NPRVM+Simple</td>
<td>2.018</td>
<td>1.615</td>
</tr>
<tr>
<td>NN(10)</td>
<td>1.331</td>
<td>1.095</td>
</tr>
<tr>
<td>NN(10)+Simple</td>
<td>1.432</td>
<td>1.131</td>
</tr>
<tr>
<td>MM(5)</td>
<td>1.551</td>
<td>0.833</td>
</tr>
<tr>
<td>MM(5)+Simple</td>
<td>1.551</td>
<td>0.833</td>
</tr>
<tr>
<td>MM(10)</td>
<td>1.490</td>
<td>0.894</td>
</tr>
<tr>
<td>MM(10)+Simple</td>
<td>1.489</td>
<td>0.894</td>
</tr>
</tbody>
</table>
Conclusions

Contrary to our previous conclusions (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 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.