JMLR: Workshop and Conference Proceedings 11 (2010) 167–174 Workshop on Applications of Pattern Analysis

Multiple Kernel Learning on the Limit Order Book

Tristan Fletcher T.FLETCHER@CS.UCL.AC.UK
Zakria Hussain Z.HUSSAIN@CS.UCL.AC.UK
John Shawe-Taylor JST@CS.UCL.AC.UK
University College London, UK

Editors: Tom Diethe, Nello Cristianini, John Shawe-Taylor

Abstract

Simple features constructed from order book data for the EURUSD currency pair were used to construct a set of kernels. These kernels were used both individually and simultaneously through the Multiple Kernel Learning (MKL) methods of SimpleMKL and the more novel LPBoostMKL to train multiclass Support Vector Machines to predict the direction of future price movements. The kernel methods outperformed a trend following benchmark both in their predictive ability and when used in a simple trading rule. Furthermore, the kernel weightings selected by the MKL techniques highlight which features of the EURUSD order book are the most informative for predictive tasks.

1. Introduction

The majority of currency trading takes place on Electronic Communication Networks (ECNs). Continuous trading takes place on these exchanges via the arrival of orders (known as limit orders) specifying whether the party wishes to buy or sell, the amount (volume) desired, and the price the transaction will occur at. While traders had previously been able to view the prices of the highest buy (best bid) and lowest sell orders (best ask), a relatively recent development in certain exchanges is the real-time revelation of the total volume of trades sitting on the ECN's order book at both these price levels and also at price levels above the best ask and below the best bid. This exposure of order books' previously hidden depths allows traders to capitalize on the greater dimensionality of data available to them at every time step (order book update) when making trading decisions and allows techniques that are more sophisticated than the standard time series analysis toolset to be used when forecasting a currency's value.

In this paper we investigate the usage of kernel methods on this higher dimensional data in order to find patterns which can be exploited with the aim of forecasting the currency's movement. Standard SVM classification techniques are investigated with different kernels along with two Multiple Kernel Learning (MKL) techniques: SimpleMKL (Rakotomamonjy et al., 2008) and LPBoostMKL (Hussain and Shawe-Taylor, 2009).

2. Related Work

A trader wishing to speculate on a currency's movement is most interested in what direction he believes that currency will move over a time horizon Δt so that he can take a position based on this prediction. Note that any move that is predicted has to be significant enough to cross the spread (difference between the best bid and ask prices) in the appropriate direction if the trader is to profit from it. If we view this as a three class classification task, then we can simplify this aim into an attempt to predict whether the trader should take one of the following actions:

$$\begin{aligned} P^{Bid}_{t+\Delta t} > P^{Ask}_{t} \Rightarrow &\quad \text{Buy the currency (\textit{long position})} \\ P^{Ask}_{t+\Delta t} < P^{Bid}_{t} \Rightarrow &\quad \text{Sell the currency (\textit{short position})} \\ P^{Bid}_{t+\Delta t} < P^{Ask}_{t}, P^{Ask}_{t+\Delta t} > P^{Bid}_{t} \Rightarrow &\quad \text{Take no position} \end{aligned}$$

There has been much work in using SVM and other similar single-kernel based methods to predict the movement of financial time series, e.g. Tay and Cao (2001). However the majority of the previous work in this area deals with the problem of kernel selection in a purely empirical manner with little to no theoretical justification. An exception to this being Wang and Zhu (2010) who use a two step kernel-selection/SVM procedure.

There is no evidence of published research that uses order book volumes when making financial market predictions. All previous research uses features based on previous price movements and in this sense this research is completely novel. Furthermore, there is very scant evidence of research using MKL in financial market prediction and no evidence of work based on using MKL on order book volume data.

3. Experimental Design

Representing the volume at time t at each of the price levels of the order book on both sides as a vector $\boldsymbol{V}_t$, where $\boldsymbol{V}_t \in \mathbb{R}^6$ for the case of three price levels on each side, a set of features was constructed:

$$\mathcal{F} = \left\{ \boldsymbol{V}_t, \frac{\boldsymbol{V}_t}{\|\boldsymbol{V}_t\|_1}, \boldsymbol{V}_t - \boldsymbol{V}_{t-1}, \frac{\boldsymbol{V}_t - \boldsymbol{V}_{t-1}}{\|\boldsymbol{V}_t - \boldsymbol{V}_{t-1}\|_1} \right\}$$

Radial Basis Function kernels have often proved useful in financial market prediction problems, e.g. Huang et al. (2005). For this reason, a set consisting of five radial kernels with different values of σ^2 along with a linear kernel was used:

$$\mathcal{K} = \left\{ \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\sigma_1^2} \right), \ldots, \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\sigma_5^2} \right), \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \right\}$$

This meant that altogether there were 24 feature / kernel combinations constructed from the order book data. Three sets of labels for the in-sample training data were constructed as follows:

$$\begin{aligned} &\text{Label A:} \quad P^{Bid}_{t+\Delta t} > P^{Ask}_{t} && \Rightarrow +1, \text{ otherwise} && \Rightarrow -1 \\ &\text{Label B:} \quad P^{Ask}_{t+\Delta t} < P^{Bid}_{t} && \Rightarrow +1, \text{ otherwise} && \Rightarrow -1 \\ &\text{Label C:} \quad P^{Bid}_{t+\Delta t} < P^{Ask}_{t}, P^{Ask}_{t+\Delta t} > P^{Bid}_{t} && \Rightarrow +1, \text{ otherwise} && \Rightarrow -1 \end{aligned}$$

In this manner, a three dimensional output vector was produced for each instance $\boldsymbol{y}_t = [\pm1, \pm1, \pm1]$.

Both MKL methods described above were investigated along with standard SVM based on the 24 kernels individually. Predictions for time horizons (Δt) of 5, 10, 20, 50, 100, and 200 seconds into the future were created. The data consisted of 6×10^4 order book updates (ticks) for the EURUSD currency pair from the EBS exchange.

4. Results

Predictions were only kept for instances where exactly one of the signs in $\boldsymbol{y}_t$ was positive, i.e. when all three of the classifiers were agreeing on a direction of movement. For this subset of the predictions, a prediction was deemed correct if it correctly predicted the direction of spread-crossing movement (i.e. upwards, downwards or no movement) and incorrect if not.

When describing the predictive accuracy of the three different kernel methods (SimpleMKL, LPBoostMKL and the individual kernels) several factors need to be considered: how often each method was able to make a prediction as described above, how correct the predictions were overall for the whole dataset and how the predictive accuracy varied depending on the direction of movement predicted, e.g. how many predicted upward movements actually transpired, etc. In the tables and figures that follow, for the sake of clarity only one of the 24 individual kernels is used when comparing the two MKL techniques to the individual kernels. The kernel chosen is the one with the most significant weightings from the SimpleMKL and LPBoostMKL methods, namely the radial basis function mapping with the smallest scale parameter (σ^2) on the simple volumes feature (Kernel 1).

A simple trend following benchmark was employed for the sake of comparison. A moving average crossover technique consisting of a signal constructed from moving averages of two rolling windows, MA_{long} and MA_{short}, was used. When $MA_{long} < MA_{short}$ the signal was to go long (+1) and when $MA_{long} > MA_{short}$ the signal was to go short (-1). In contrast to the kernel based methods, this rule produced a continuous non-zero signal. The window lengths chosen for the two periods were those that gave the highest predictive accuracy for the data set. Table 1 shows how often each of the methods were able to make a prediction for each of the time horizons and Table 2 shows each of the method's predictive accuracy over the entire dataset when a prediction was actually possible.

Table 1: % Number of Instances Predictions Possible

Δt	**SimpleMKL**	**LPBoostMKL**	**Kernel 1**	**Moving Average**
5	27	24	26	100
10	43	38	42	100
20	51	50	50	100
50	44	43	43	100
100	35	34	35	100
200	26	25	21	100

Tables 3, 4 and 5 break down the predictive accuracy of each method conditioned on whether a positive, negative or no movement (i.e. not spread-crossing) were predicted. Note that *NA* indicates that no predictions for that particular value of Δt were possible.

Figures 1 and 2 show the average weighting that SimpleMKL and LPBoostMKL assigned to each of the 24 individual kernels.

Table 2: % Accuracy of Entire Dataset

Δt	SimpleMKL	LPBoostMKL	Kernel 1	Moving Average
5	95	95	95	4
10	90	90	89	6
20	81	81	81	10
50	66	66	66	17
100	51	51	51	24
200	45	46	46	30

Table 3: % Accuracy of Positive Movement Predictions

Δt	SimpleMKL	LPBoostMKL	Kernel 1	Moving Average
5	NA	NA	NA	2
10	NA	NA	NA	5
20	12	33	6	9
50	30	32	30	17
100	27	27	27	24
200	35	35	38	30

Table 4: % Accuracy of Negative Movement Predictions

Δt	SimpleMKL	LPBoostMKL	Kernel 1	Moving Average
5	NA	NA	NA	2
10	NA	NA	NA	4
20	16	17	16	8
50	16	17	16	16
100	19	21	21	23
200	34	37	39	31

Table 5: % Accuracy of Zero Movement Predictions

Δt	SimpleMKL	LPBoostMKL	Kernel 1	Moving Average
5	95	95	95	NA
10	90	90	89	NA
20	82	81	82	NA
50	70	70	70	NA
100	59	60	60	NA
200	50	50	50	NA

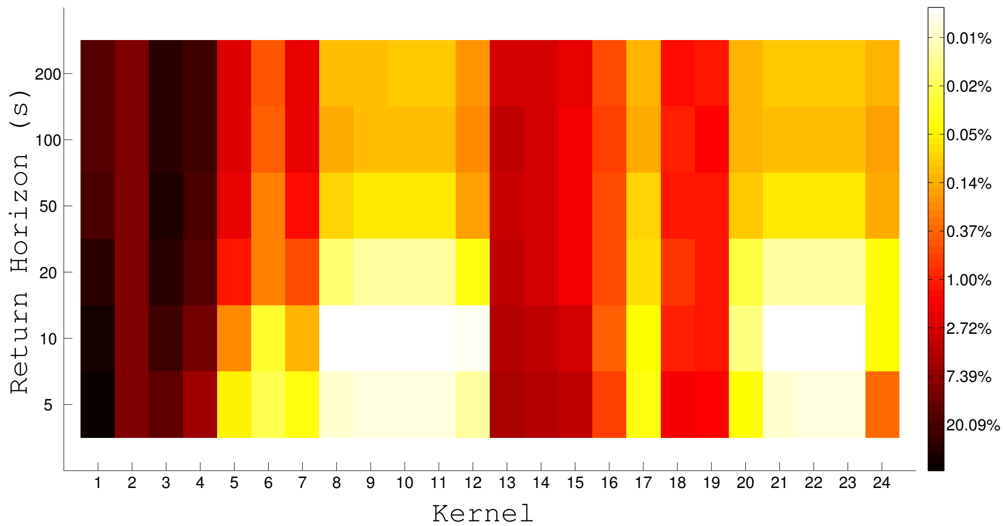

Figure 1: SimpleMKL Kernel Weightings

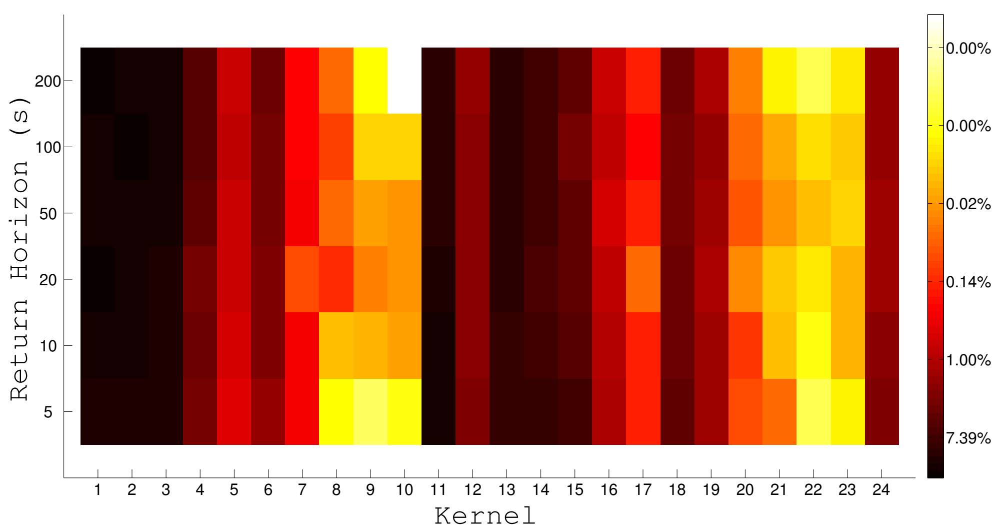

Figure 2: LPBoostMKL Kernel Weightings

5. Conclusions

Table 1 indicates that both MKL methods and the individual kernel were able to make predictions between a third and half the time with the MKL methods slightly outperforming the individual kernels in the majority of cases. Table 2 appears to show that the the individual kernels and MKL methods are significantly better than the MA benchmark, despite it having had the unfair advantage of having had its two parameters optimised on this metric. However, the main reason for this is because the MA technique is not able to predict zero movements. When the predictive accuracy is conditioned on the direction of the movement predicted, as shown in Tables 3, 4 and 5, one can see that although the kernel methods still outperform the trend following technique for positive and negative movement predictions in the majority of cases, the outperformance is less significant.

Figure 3 shows a 700 second snap shot of the positive and negative predictions generated by the SimpleMKL method for $\Delta t = 50$. The triangles denoting predictions of positive or negative movement (at the best bid or ask price respectively) have black lines linking them to the opposing (spread-crossing) price at the end of each prediction's 50 second forecast horizon. In this particular example, one can see that the SimpleMKL method commences by making four predictions for positive movement only the first of which is correct, followed by a successful downward movement prediction. The many zero movement predictions shown do not have their resulting predictions plotted for the sake of clarity. If nothing else, this graphically depicts both the low success rate of these techniques' predictions (which is still nevertheless greater than the benchmark) and the relative frequency of zero movement predictions.

Figure 3: Example of SimpleMKL Predicted Turning Points

The upper half of Figure 4 shows a theoretical profit and loss (P&L) curve for the period constructed from a simple trading rule based on each of the techniques' signals. Amongst other things, it was assumed that one had to pay the spread each time a trade was made, that the trade size each time was \$1M and that each trade was executed immediately at the price submitted (there was no slippage). It highlights the fact that similar predictive accuracies do not necessarily translate into similar P&L curves, with the two MKL techniques clearly outperforming the individual kernel and all three of the kernel methods outperforming the trend following one. The lower half of the figure shows the price of EURUSD over the time period that the dataset encompasses.

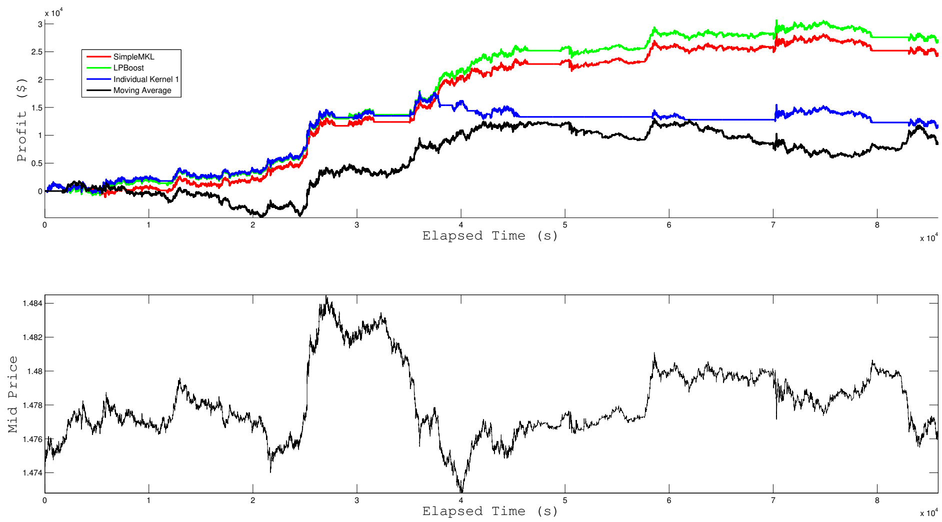

Figure 4: P&L Curve (Above) and EURUSD Price (Below)

Figures 1 and 2 show that for both MKL techniques, kernels 1, 2, 3, 13, 14 and 15 have consistently high weightings and hence were the most relevant for making predictions over the data set. These kernels are the radial basis function mapping with the three smallest scale parameters (σ^2) on the simple volumes feature and the change in volumes feature ($\mathcal{K}_1\mathcal{F}_1$, $\mathcal{K}_1\mathcal{F}_2$, $\mathcal{K}_1\mathcal{F}_3$, $\mathcal{K}_3\mathcal{F}_1$, $\mathcal{K}_3\mathcal{F}_2$ and $\mathcal{K}_3\mathcal{F}_3$ in the notation of Section 3). The fact that both MKL methods selected very similar weightings for the 24 different mapping/feature combinations highlights the consistency of the techniques.

The results of the MKL experiments described here are significant in that no price action or features based on prices is taken into account when predicting future prices - in stark contrast to other research. This means that any trading rules based on this technique are likely to be uncorrelated with existing rules, the majority of which look at previous price action in some manner or other. Furthermore, the out-performance of the kernel-based techniques for long time horizons over the trend following benchmark make them a useful method for locating turning points in time series of EURUSD prices.

In terms of comparing the two MKL methods, it is worth noting that they are both solving the same optimisation problem so one would expect them to give very similar results, as they do in the majority of cases. However, one of the main differences between them is that the continuously improving LPBoost algorithm can be stopped at any point prior to convergence to produce a suboptimal classifier and in this sense one can control the accuracy vs training time trade-off for the method. This aspect of the method is of practical benefit in real-time applications where training time is an important constraint.

Acknowledgments

The authors would like to thank ICAP for making its EBS foreign exchange data available for this research.

References

W. Huang, Y. Nakamori, and S. Wang. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.*, 32(10):2513–2522, 2005.

Z. Hussain and J. Shawe-Taylor. Metric learning analysis. PinView FP7-216529 Project Deliverable Report D3.2, 2009. URL `http://www.pinview.eu/deliverables.php`.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, November 2008.

F. Tay and L. Cao. Application of support vector machines in financial time series forecasting. *Omega*, 29:309–317, 2001.

L. Wang and J. Zhu. Financial market forecasting using a two-step kernel learning method for the support vector regression. *Annals of Operation Research*, 174:103–120, 2010.