Currency Forecasting using Multiple Kernel Learning with Financially Motivated Features

Tristan Fletcher, Zakria Hussain and John Shawe-Taylor
Centre for Computational Statistics and Machine Learning
Department of Computer Science
University College London, UK
{t.fletcher,z.hussain,jst}@cs.ucl.ac.uk

Abstract

Multiple Kernel Learning (MKL) is used to replicate the signal combination process that trading rules embody when they aggregate multiple sources of financial information when predicting an asset's price movements. A set of financially motivated kernels is constructed for the EURUSD currency pair and is used to predict the direction of price movement for the currency over multiple time horizons. MKL is shown to outperform each of the kernels individually in terms of predictive accuracy. Furthermore, the kernel weightings selected by MKL highlights which of the financial features represented by the kernels are the most informative for predictive tasks. [1]

1 Introduction

A trader wishing to speculate on a currency's movement is most interested in what direction he believes the price of that currency P_t will move over a time horizon Δt so that he can take a position based on this prediction. Any move that is predicted has to be significant enough to cross the difference between the buying price (bid) and selling price (ask) 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 buy the currency pair because he believes $P^{Bid}_{t+\Delta t} > P^{Ask}_t$, sell it because $P^{Ask}_{t+\Delta t} < P^{Bid}_t$ or do nothing because $P^{Bid}_{t+\Delta t} < P^{Ask}_t$ and $P^{Ask}_{t+\Delta t} > P^{Bid}_t$.

When making trading decisions such as whether to buy or sell a currency, traders typically combine the information from many models to create an overall trading rule (see for example [1]). The aim of this work is to represent this model combination process through Multiple Kernel Learning, where individual kernels based on common trading signals are created to represent the constituent sources of information.

There has been much work in using kernel based methods such as the SVM to predict the movement of financial time series, e.g. [2–15]. 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, and makes no attempts to either use financially plausible kernels or indeed to combine kernels in any manner.

[1] This work is closely related to a presentation titled *Multiple Kernel Learning on the Limit Order Book* given at the Workshop on Applications of Pattern Analysis 2010.

2 Financially Motivated Features

2.1 Price-based Features

The following four features are based on common price-based trading rules (which are described briefly in the Appendix):

$$\mathcal{F}_1 = \left\{ EMA_t^{L_1}, \ldots, EMA_t^{L_N} \right\}$$

$$\mathcal{F}_2 = \left\{ MA_t^{L_1}, \ldots, MA_t^{L_N}, \sigma_t^{L_1}, \ldots, \sigma_t^{L_N} \right\}$$

$$\mathcal{F}_3 = \left\{ P_t, \max_t^{L_1}, \ldots, \max_t^{L_N}, \min_t^{L_1}, \ldots, \min_t^{L_N} \right\}$$

$$\mathcal{F}_4 = \left\{ \Uparrow_t^{L_1}, \ldots, \Uparrow_t^{L_N}, \Downarrow_t^{L_1}, \ldots, \Downarrow_t^{L_N} \right\}$$

where $EMA_t^{L_i}$ denotes an exponential moving average of the price P at time t with a half life L_i, $\sigma_t^{L_i}$ denotes the standard deviation of P over a period L_i, $MA_t^{L_i}$ its simple moving average over the period L_i, $\max_t^{L_i}$ and $\min_t^{L_i}$ the maximum and minimum prices over the period and $\Uparrow_t^{L_i}$ and $\Downarrow_t^{L_i}$ the number of price increases and decreases over it.

2.2 Volume-based Features

The majority of currency trading takes place on Electronic Communication Networks (ECNs). Continuous trading takes place on these exchanges via the arrival of 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 when making trading decisions and suggests the use of kernel methods on this higher dimensional data.

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 further set of four features can be constructed:

$$\mathcal{F}_{5\ldots8} = \left\{ \boldsymbol{V}_t, \frac{\boldsymbol{V}_t}{\|\boldsymbol{V}_t\|_1}, \boldsymbol{V}_t - \boldsymbol{V}_{t-1}, \frac{\boldsymbol{V}_t - \boldsymbol{V}_{t-1}}{\|\boldsymbol{V}_t - \boldsymbol{V}_{t-1}\|_1} \right\}$$

3 Experimental Design

Radial Basis Function (RBF) and polynomial kernels have often been used in financial market prediction problems, e.g. [7] and [15]. Furthermore, Artificial Neural Networks (ANN) are often used in financial forecasting tasks (e.g. [16–18]) and for this reason a kernel based on Williams (1998) [19] infinite neural network with a sigmoidal transfer function is also employed (see $\mathcal{K}_{11:15}$ below). A feature mapping set consisting of 5 of each of these kernel types with different values of the relevant hyperparameter (σ, d or Σ) along with the linear kernel is used:

$$\mathcal{K}_{1:5} = \left\{ \exp\left(-\left\| \boldsymbol{x} - \boldsymbol{x}' \right\|^2 / \sigma_1^2 \right), \ldots, \exp\left(-\left\| \boldsymbol{x} - \boldsymbol{x}' \right\|^2 / \sigma_5^2 \right) \right\}$$

$$\mathcal{K}_{6:10} = \left\{ \left(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + 1 \right)^{d_1}, \ldots, \left(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + 1 \right)^{d_5} \right\}$$

$$\mathcal{K}_{11:15} = \left\{ \frac{2}{\pi} \sin^{-1}\left(\frac{2\boldsymbol{x}^T \Sigma_1 \boldsymbol{x}'}{\sqrt{(1 + 2\boldsymbol{x}^T \Sigma_1 \boldsymbol{x})(1 + 2\boldsymbol{x}'^T \Sigma_1 \boldsymbol{x}')}} \right), \ldots, \frac{2}{\pi} \sin^{-1}\left(\frac{2\boldsymbol{x}^T \Sigma_5 \boldsymbol{x}'}{\sqrt{(1 + 2\boldsymbol{x}^T \Sigma_5 \boldsymbol{x})(1 + 2\boldsymbol{x}'^T \Sigma_5 \boldsymbol{x}')}} \right) \right\}$$

$$\mathcal{K}_{16} = \{ \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \}$$

Table 1: Percentage of time predictions possible

Δt	**SimpleMKL**	$\mathcal{F}_8\mathcal{K}_{16}$	$\mathcal{F}_1\mathcal{K}_1$	$\mathcal{F}_1\mathcal{K}_3$
5	26.1	24.7	26.1	24.7
10	41.1	40.4	39.8	37.7
20	50.2	49.1	48.1	45.0
50	46.3	44.1	44.8	45.5
100	32.8	33.5	34.6	35.3
200	27.0	24.9	26.6	27.4

This means that altogether there are $|\mathcal{F}| \times |\mathcal{K}| = 8 \times 16 = 128$ feature / kernel combinations. We will adopt notation so that for example the combination $\mathcal{F}_1\mathcal{K}_1$ is the moving average crossover feature with a RBF using the scale parameter σ_1^2.

Three SVM are trained on the data with the following labeling criteria for each SVM:

$$\begin{aligned}
&\text{SVM 1:} P^{Bid}_{t+\Delta t} > P^{Ask}_t && \Rightarrow y^1_t = +1, \text{otherwise } y^1_t = -1 \\
&\text{SVM 2:} P^{Ask}_{t+\Delta t} < P^{Bid}_t && \Rightarrow y^2_t = +1, \text{otherwise } y^2_t = -1 \\
&\text{SVM 3:} P^{Bid}_{t+\Delta t} < P^{Ask}_t, P^{Ask}_{t+\Delta t} > P^{Bid}_t && \Rightarrow y^3_t = +1, \text{otherwise } y^3_t = -1
\end{aligned}$$

In this manner, a three dimensional output vector $\boldsymbol{y}_t$ is constructed from y^1_t, y^2_t and y^3_t for each instance such that $\boldsymbol{y}_t = [\pm 1, \pm 1, \pm 1]$. Predictions are only kept for instances where exactly one of the signs in $\boldsymbol{y}_t$ is positive, i.e. when all three of the classifiers are agreeing on a direction of movement. For this subset of the predictions, a prediction is deemed correct if it correctly predicts the direction of spread-crossing movement (i.e. upwards, downwards or no movement) and incorrect if not.

The MKL method of SimpleMKL [20] is investigated along with standard SVM based on each of the 128 kernels / feature combinations individually. Predictions for time horizons (Δt) of 5, 10, 20, 50, 100 and 200 seconds into the future are created. Training and prediction is carried out by training the three SVM on 100 instances of in sample data, making predictions regarding the following 100 instances and then rolling forward 100 instances so that the out of sample data points in the previous window become the current window's in sample set. The data consists of 6×10^4 instances of order book updates for the EURUSD currency pair from the EBS exchange starting on 2/11/2009. [2]

4 Results and Conclusions

When comparing the predictive accuracy of the kernel methods when used individually to their combination in MKL one needs to consider both how often each method was able to make a prediction as described above and how correct the predictions were overall for the whole dataset. In the tables and figures that follow, for the sake of clarity only three of the 128 individual kernels are used when comparing SimpleMKL to the individual kernels. 10-fold cross-validation was used to select the three kernels with the highest predictive accuracy for the dataset, namely $\mathcal{F}_8\mathcal{K}_{16}$, $\mathcal{F}_1\mathcal{K}_1$ and $\mathcal{F}_1\mathcal{K}_3$.

Table 1, which shows how often each of the methods were able to make a prediction for each of the time horizons, indicates that SimpleMKL was very similar in the frequency with which it was able to make predictions as the three individual kernel / feature combinations highlighted. Table 2 shows each of the method's predictive accuracy over the entire dataset when a prediction was actually possible. The results indicate that SimpleMKL has higher predictive accuracy than the most effective individual kernels for all time horizons under 200 seconds and is only marginally less effective than $\mathcal{F}_1\mathcal{K}_3$ for the 200 second forecast horizon.

P-values for the null hypothesis that the results reported could have occurred by chance were calculated (the methodology for doing this is explained in the Appendix). It was found that for both

[2] EURUSD was selected as the currency pair to investigate because it is the world's most actively traded currency pair, comprising 27% of global turnover [21]. Consequently, the EBS exchange was selected for this analysis because it is the primary ECN for EURUSD.

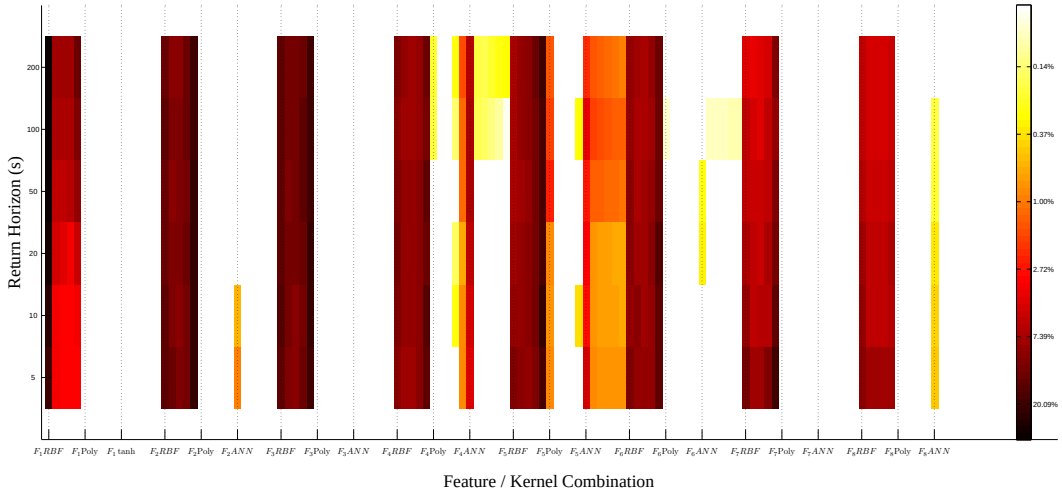

Figure 1: MKL Kernel weightings

Table 2: Percentage accuracy of predictions

Δt	**SimpleMKL**	$\mathcal{F}_8\mathcal{K}_{16}$	$\mathcal{F}_1\mathcal{K}_1$	$\mathcal{F}_1\mathcal{K}_3$
5	94.7	94.7	93.0	92.8
10	89.9	89.6	88.4	84.6
20	81.7	81.3	79.5	72.3
50	67.1	65.4	65.5	61.1
100	61.1	51.1	60.7	59.9
200	58.9	45.0	58.8	61.3

SimpleMKL and the individual kernels highlighted for all forecast horizons, the null hypothesis could be rejected for a significance level of $< 10^{-5}$.

As reflected in Figure 1, the kernel / feature combinations $\mathcal{F}_1\mathcal{K}_1$, $\mathcal{F}_2\mathcal{K}_5$ and $\mathcal{F}_3\mathcal{K}_5$ are consistently awarded the highest weightings by SimpleMKL and hence are the most relevant for making predictions over the data set. These kernels are the RBF mapping with the smallest scale parameter on the exponential moving average crossover feature, the RBF mapping with the largest scale parameter on the price standard deviation / moving average feature and the RBF mapping with the largest scale parameter again on the minimums / maximums feature.

The vertical banding of colour (or intensity) highlights the consistency of each of the kernel / feature combination's weightings across the different time horizons: in almost all cases the weighting for a particular combination is not significantly different between when being used to make a prediction for a short time horizon and a longer term one. One can also see from Figure 1 that although all 8 of the features have weightings assigned to them, in most cases this is only in conjunction with the RBF kernels - the polynomial (*Poly*) and infinite neural network (*ANN*) based mappings being assigned weightings by MKL for only the fourth and fifth features.

The most successful individual kernels as selected by cross-validation are awarded very low weights by SimpleMKL. This reflects a common feature of trading rules where individual signals can drastically change their significance in terms of performance when used in combination. Furthermore, the outperformance of SimpleMKL to the individual kernels highlighted indicates that MKL is an effective method for combining a set of price and volume based features in order to correctly forecast the direction of price movements in a manner similar to a trading rule.

Acknowledgments

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

References

[1] Perry Kaufman. *The New Trading Systems and Methods*. John Wiley & Sons, 2005.

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

[3] F.E.H. Tay and L.J. Cao. Modified support vector machines in financial time series forecasting. *Neurocomputing*, 48:847–861, 2002.

[4] L.J. Cao. Support vector machines experts for time series forecasting. *Neurocomputing*, 51:321–339, 2003.

[5] Kyoungjae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55:307–319, 2003.

[6] Fernando Perez-cruz, Julio Afonso-rodriguez, and Javier Giner. Estimating garch models using support vector machines. *Quantitative Finance*, 3(3):163–172, 2003.

[7] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.*, 32(10):2513–2522, 2005.

[8] Tony Van Gestel, Johan A. K. Suykens, Dirk-Emma Baestaens, Annemie Lambrechts, Gert Lanckriet, Bruno Vandaele, Bart De Moor, and Joos Vandewalle. Financial time series prediction using least squares support vector machines within the evidence framework. In *IEEE Transactions on Neural Networks*, pages 809–821, 2001.

[9] Neep Hazarika and John G. Taylor. *Predicting bonds using the linear relevance vector machine*, chapter 17, pages 145–155. Springer-Verlag, 2002.

[10] N. Nikolaev P. Tino and X. Yao. Volatility forecasting with sparse bayesian kernel models. In *Proc. 4th International Conference on Computational Intelligence in Economics and Finance, Salt Lake City, UT*, pages 1150–1153, 2005.

[11] Shian-Chang Huang and Tung-Kuang Wu. Wavelet-based relevance vector machines for stock index forecasting. In *International Joint Conference on Neural Networks (IJCNN)*, pages 603–609, 2006.

[12] Shian-Chang Huang and Tung-Kuang Wu. Combining wavelet-based feature extractions with relevance vector machines for stock index forecasting. *Expert Systems*, 25:133–149, 2008.

[13] Stephan K. Chalup and Andreas Mitschele. Kernel methods in finance. In *Handbook on Information Technology in Finance*, pages 655–687. 2008.

[14] Tristan Fletcher, Fabian Redpath, and Joe D'Alessandro. Machine learning in fx carry basket prediction. In *Proceedings of the International Conference of Financial Engineering*, volume 2, pages 1371–1375, 2009.

[15] Christian Ullrich. *Forecasting and Hedging in the Foreign Exchange Markets*. Springer, 2009.

[16] Chung-Ming Kuan and Tung Liu. Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics*, 10(4):347–64, Oct.-Dec. 1995.

[17] Steven Walczak. An empirical analysis of data requirements for financial forecasting with neural networks. *J. Manage. Inf. Syst.*, 17(4):203–222, 2001.

[18] Jimmy Shadbolt and John G. Taylor, editors. *Neural networks and the financial markets: predicting, combining and portfolio optimisation*. Springer-Verlag, London, UK, 2002.

[19] Christopher Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

[20] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, November 2008.

[21] BIS. Triennial central bank survey of foreign exchange and derivatives market activity in 2007. http://www.bis.org/publ/rpfxf07t.htm, 2007.

[22] Gerald Appel. *Technical Analysis: Power Tools for Active Investors*. Financial Times, 2005.

[23] Curtis Faith. *Way of the Turtle*. McGraw-Hill Professional, 2007.

[24] John A. Bollinger. *Bollinger on Bollinger Bands*. McGraw-Hill, 2001.

[25] J. Welles Wilder. *New Concepts in Technical Trading Systems*. Trend Research, 1978.

Appendix

Price-based Features

- $\mathcal{F}_1$: A common trading rule is the moving average crossover technique (see for example [22]) which suggests that the price P_t will move up when its short term moving average EMA_t^{short} crosses above a longer term one EMA_t^{long} and visa versa.
- $\mathcal{F}_2$: Breakout trading rules (see for example [23]) look to see if the price has broken above or below a certain threshold and assume that once the price has broken through this threshold the direction of the price movement will persist. One way of defining this threshold is through the use of Bollinger Bands [24] where the upper/lower thresholds are set by adding/subtracting a certain number of standard deviations of the price movement σ_t^L to the average price MA_L^t for a period L.
- $\mathcal{F}_3$: Another breakout trading rule called the Donchian Trend system [23] determines whether the price has risen above its maximum $\max_t^L$ or below its minimum $\min_t^L$ over a period L and once again assumes that once the price has broken through this threshold the direction of the price movement will persist.
- $\mathcal{F}_4$: The Relative Strength Index trading rule [25] is based on the premise that there is a relationship between the number of times the price has gone up over a period $\Uparrow_t^L$ vs the number of times it has fallen $\Downarrow_t^L$ and assumes that the price is more likely to move upwards if $\Uparrow_t^L > \Downarrow_t^L$ and visa versa.

Calculation of p-values

- For each in sample period, the proportion of occurrences of each of the three classes of movement (up, down or none) over the 100 instances of in sample data was determined.
- Predictions of movement were then generated randomly for each of the instances of the out of sample period where a prediction was deemed possible by SimpleMKL / individual kernel (as explained in section 3), each class having a probability of being assigned based on the in sample proportions.
- This was repeated 10^5 times for each out of sample section with the number of times the randomly generated predictions were correct along with the number of times SimpleMKL / individual kernel was correct for that period recorded each time.
- The proportion of the 10^5 iterations that the number of correct predictions recorded for all the out of sample periods was greater than that reported by SimpleMKL / individual kernel was used to calculate the P-value.
- In the work reported here, not one of the 10^5 iterations of randomly generated predictions outperformed the SimpleMKL / individual kernel methods.