Noname manuscript No.
(will be inserted by the editor)

Multiple Kernel Learning with Fisher kernels for High Frequency Currency Prediction

Tristan Fletcher · John Shawe-Taylor

Received: date / Accepted: date

Abstract Financially motivated kernels based on the EURUSD currency data are constructed from limit order book volumes, commonly used technical analysis methods and canonical market microstructure models - the latter in the form of Fisher kernels. These kernels are used through their incorporation into Support Vector Machines (SVM) to predict the direction of price movement for the currency over multiple time horizons. Multiple Kernel Learning (MKL) is used to replicate the signal combination process that trading rules embody when they aggregate multiple sources of financial information. Significant outperformance relative to both the individual SVM and benchmarks is found, along with an indication of which features are the most informative for financial prediction tasks. An average accuracy of 55% is achieved when classifying the direction of price movement into one of three categories for a 200s predictive time horizon.

Keywords multiple kernel learning · support vector machines · limit order books · kernel methods · market microstructure · Fisher kernels

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 $P_{t+\Delta t}^{Bid} > P_t^{Ask}$, sell it because $P_{t+\Delta t}^{Ask} < P_t^{Bid}$ or do nothing because $P_{t+\Delta t}^{Bid} < P_t^{Ask}$ and $P_{t+\Delta t}^{Ask} > P_t^{Bid}$.

In this paper we examine the problem of currency prediction by briefly reviewing previous methods that have been used over the last few decades, explaining their shortcomings and then proposing a methodology that aims to deal with these weaknesses - namely Multiple Kernel Learning (MKL) based on features extracted from an exchange's limit order book volume data, previous price action and Fisher features based on canonical market microstructural models.

There is no evidence of published research that uses order book volumes away from the front Bid or Offer 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. Lastly, there is no evidence of using Fisher features, in any context, in order to predict future price action.

T. Fletcher
Department of Computer Science University College London Gower Street London WC1E 6BT United Kingdom Tel.: +44-7890101644
E-mail: t.fletcher@cs.ucl.ac.uk

J. Shawe-Taylor
Department of Computer Science University College London Gower Street London WC1E 6BT United Kingdom Tel.: +44-2076797680
E-mail: J.Shawe-Taylor@cs.ucl.ac.uk

2 Price Prediction Approaches

When the forecast horizon Δt of a prediction represents medium to long term timescales, methods focusing on fundamental macro-economic trends are best placed to make predictions on how the price of the currency might evolve. These techniques are often complemented with technical analysis to either make the outright predictions themselves or more commonly to ascertain turning points in these trends and hence the most suitable times to enter into or exit from trades (e.g. [11], [36], [41], [33] and [42]). The majority of technical analysis is concerned with incorporating previous price action of the time series in question, occasionally incorporating simple measures of volume, for example to confirm trends [40]. However, the efficacy of the traditional technical analysis toolset can arguably said to to be in decline (e.g. [43]).

Nowadays, the majority of currency trading takes place on Electronic Communication Networks (ECNs) [4]. Continuous trading takes place on these exchanges via the arrival of market and limit orders. The latter specify 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 order book update (tick) 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. There is a great deal of research on order books and the related field of market microstructure but it is heavily based on stocks and often relates to characterising features such as liquidity, volatility and spreads (the difference between the best bid and ask prices) instead of attempting to predict future price action, see for example [35], [20], [47], [26] and [34].

A popular, more recent method for making predictions in currency exchange markets, sometimes incorporating technical analysis, is that of Artificial Neural Networks (ANN). ANN embody a set of thresholding functions connected to each other with adaptive weights that is trained on historical data in order that it may be used to make predictions in the future, see for example [29], [58] and [53]. These techniques are often criticised for the stochastic nature of their weight initialisation, the fact that they cannot be guaranteed to provide optimal solutions (they fail to converge on global optima) and that they are prone to overfitting.

A more novel method that is not subject to these criticisms but that is nevertheless well placed to deal with the high dimensional data sets that order books reveal are Support Vector Machines (SVM) [9].

3 Support Vector Machines

For the purpose of explaining SVM and MKL, the general prediction/learning problem can be expressed in the form:

$$y = \sum_{i=1}^{N} w_i \phi(\boldsymbol{x}) + b \tag{1}$$

where $\phi(\boldsymbol{x})$ represents a non-linear mapping of the input instance $\boldsymbol{x}$ into a higher dimensional feature space (i.e. a basis function), y is the corresponding prediction and $\boldsymbol{w}$ and b are parameters learned from the N instances of training data.

It is the choice of kernel κ in (1) that determines the non-linear mapping $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$ where $\kappa(\boldsymbol{x_i}, \boldsymbol{x_j}) = \langle \phi(\boldsymbol{x_i}), \phi(\boldsymbol{x_j}) \rangle$. Two commonly used kernel functions are the linear kernel where $\kappa(\boldsymbol{x_i}, \boldsymbol{x_j}) = \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle$ and the radial basis function kernel $\kappa(\boldsymbol{x_i}, \boldsymbol{x_j}) = \exp\left(- \|\boldsymbol{x} - \boldsymbol{x}'\|^2 / \sigma^2\right)$ [10].

In the case of classification (i.e. where $y = \pm 1$) the parameters $\boldsymbol{w}$ and b are found by using Quadratic Programming (QP) optimization to first find the α_i which maximize:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$$

$$\text{where} \quad \alpha_i \geq 0 \ \forall_i, \ \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2}$$

The α_i are then used to find:

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\boldsymbol{x}_i) \tag{3}$$

and the set of support vectors $\mathbb{S}$ is represented by the indices i where $\alpha_i > 0$. The bias b can then be calculated:

$$b = \frac{1}{N_s} \sum_{s \in \mathbb{S}} \left(y_s - \sum_{m \in S} \alpha_m y_m \phi(\boldsymbol{x}_m) \cdot \phi(\boldsymbol{x}_s) \right) \tag{4}$$

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. [54], [55], [7], [28], [48], [24], [19], [21], [46], [22], [23], [17] and most notably [8].

Despite the many advantages of using SVM and their resulting popularity, one of the main problems of the SVM approach for real-world problems is the selection of the feature-space mapping through the choice of kernel, which is often selected empirically with little theoretical justification.

4 Multiple Kernel Learning

A method which deals with the problem of kernel selection is that of Multiple Kernel Learning (MKL) (e.g. [32], [3]). This technique mitigates the risk of erroneous kernel selection to some degree by taking a set of kernels and deriving a weight for each kernel such that predictions are made based on a weighted sum of several kernels. Furthermore, 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 [27]). This model combination process is similar conceptually to Multiple Kernel Learning, where individual kernels based on common trading signals are created to represent the constituent sources of information.

Multiple Kernel Learning considers convex combinations of K kernels:

$$\boldsymbol{\kappa}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{t=1}^{K} d_t \kappa_t(\boldsymbol{x}_i, \boldsymbol{x}_j) \text{ where } \ d_t \geq 0, \ \sum_t d_t = 1. \tag{5}$$

It is often useful when looking at high dimensional datasets (for example in the world of computer vision) to summarise them by extracting features (e.g. [5]). Taking this into account, the principle of MKL can be extended by constructing a set of kernels that not only includes different kernel functions applied to the input data but also different features of the data itself. The weightings that the MKL allocates to each kernel method / feature combination highlights its usefulness in representing the input space for the task at hand.

The majority of the research using SVM in financial prediction tasks deals with the problem of kernel selection in a purely empirical manner with little to no theoretical justification. The exceptions to this being Wang and Zhu (2008) [59], who use a two step kernel-selection/SVM procedure and Luss and D'Aspremont (2008) [37] who use Multiple Kernel Learning (MKL) to classify the impact of news for a financial prediction task. 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 (other than [16]), with all previous research using features based on previous price movements (e.g. [57]).

For the purposes of exploring MKL on order book data the main MKL technique that will be investigated is Rakotomamonjy *et al.'s* (2008) SimpleMKL [51].

4.1 SimpleMKL

SimpleMKL learns the kernel weightings (i.e. d_t of (5)) along with the α's by using semi-infinite linear programming to solve the following constrained optimisation:

$$\min_d J(d) = \left\{ \begin{array}{l} \min\limits_{\{f\},b,\xi} \ \frac{1}{2} \sum_{t=1}^{K} \frac{1}{d_t} \left\| f_t \right\|^2 + C \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad y_i \sum_{t=1}^{K} f_t(\boldsymbol{x_i}) + y_i b \geq 1 - \xi_i \\ \qquad\quad \xi_i \geq 0, \ i = 1, \ldots, m \end{array} \right\} \tag{6}$$

where:

$$f_t(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i \kappa_t(\boldsymbol{x_i}, \boldsymbol{x}) + b\right) \tag{7}$$

The associated dual problem is very similar to (5):

$$\begin{aligned} &\min_{d_t} \left\{ \max_{\alpha} -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_t d_t \kappa_t(x_i, x_j) + \sum_i \alpha_i \right\} \\ &\text{s.t.} \ \sum_t d_t = 1, \ \ d_t \geq 0, \ \ \sum_i \alpha_i y_i = 0, \ \ C \geq \alpha_i \geq 0 \end{aligned} \tag{8}$$

This problem can be solved using a standard SVM optimisation solver for a fixed weighting of kernels, to generate an initial solution for $\boldsymbol{\alpha}$. After this step one can fix $\boldsymbol{\alpha}$ and use a linear programming technique to solve for the d's and hence find the weights of the kernels. This 2-step process is repeated until a convergence criterion is met.

5 Fisher kernels

The Fisher kernel represents a method for incorporating generative probability models into discriminative classifiers such as SVM. It also facilitates the inclusion of some canonical market micro-structural models, all of which are generative by nature, into the discriminative machine learning domain. It is this incorporation of traditional, often empirically based, market microstructural (MM) models into the machine learning framework that represents the main contribution of this work.

When one adapts the parameters of a model to incorporate a new data point so that the model's likelihood $\mathcal{L}$ will increase, a common approach is to adjust each parameter θ_i by some function of $d\mathcal{L}/d\theta_i$. The Fisher kernel [25] incorporates this principle by creating a kernel composed of values of $d\mathcal{L}/d\theta_i$ for each of the model's parameters and therefore comparing data instances by the way they stretch the model's parameters.

Defining the log likelihood of a data item x with respect to a model for a given setting of the parameters $\boldsymbol{\theta}$ to be $\log \mathcal{L}_{\boldsymbol{\theta}}(x)$, the Fisher score of x is the vector gradient of $\log \mathcal{L}_{\boldsymbol{\theta}}(x)$:

$$\mathbf{g}(\boldsymbol{\theta}, x) = \left(\frac{\partial \log \mathcal{L}_{\theta_i}(x)}{\partial \theta_i}\right)_{i=1}^{N} \tag{9}$$

Commonly adopted Market Microstructure models are based around three main families: Autoregressive Conditional Duration models, Poisson processes and Wiener processes.

5.1 Autoregressive Conditional Duration model

The literature on time deformation in financial markets suggests that sometimes time, as observed through the rate of a particular financial transaction occurring, flows very rapidly while in other periods it moves slowly. The market microstructure literature, for example [13], shows that one should expect stochastic clustering in the rate of price changes for a financial asset. An explanation for this is that ill-informed traders trade randomly according to a stochastic process such as a Poisson process, while informed traders enter the market only after observing a private, potentially noisy signal. The agents providing the prices (market makers) will slowly learn of the private information by watching order flow and adjust their prices accordingly. Informed traders will seek to trade as long as their information has value. Hence one should see clustering of trading following an information event because of the increased numbers of informed traders.

Engle and Russel's (1998) Autoregressive Conditional Duration (ACD) model [14] captures this stochasticly clustering arrival rate by expressing the duration of a price (how long a financial asset's price remains constant) as a function of previous durations. The model is used commonly throughout the market microstructure literature, e.g. to measure the duration of financial asset prices [12]. It is defined as follows:

$$h_t = w + \sum_{i=1}^{L} q_i x_{t-i} + \sum_{i=1}^{L} p_i h_{t-i} \tag{10}$$

$$x_t = h_t \epsilon_t, \quad \epsilon_t \sim Exp(\lambda) \tag{11}$$

where x_t is the duration of the price at time t, h_t is its expected duration, L is the lag of the autoregressions and w, $\mathbf{p}$, $\mathbf{q}$ and λ are constants.

Simplifying (10) to be an order 1 autoregressive process, the likelihood of the model can be expressed:

$$\mathcal{L} = \lambda \exp\left[-\lambda x_t (w + q x_{t-1} + p h_{t-1})^{-1}\right] \tag{12}$$

Differentiating $\mathcal{L}$ wrt each of the parameters:

$$\frac{\partial \mathcal{L}}{\partial w} = \lambda^2 x_t (w + q x_{t-1} + p h_{t-1})^{-2} \exp\left[-\lambda x_t (w + q x_{t-1} + p h_{t-1})^{-1}\right] \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial q} = \lambda^2 x_t x_{t-1} (w + q x_{t-1} + p h_{t-1})^{-2} \exp\left[-\lambda x_t (w + q x_{t-1} + p h_{t-1})^{-1}\right] \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial p} = \lambda^2 x_t h_{t-1} (w + q x_{t-1} + p h_{t-1})^{-2} \exp\left[-\lambda x_t (w + q x_{t-1} + p h_{t-1})^{-1}\right] \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = (1 - \lambda x_t (w + q x_{t-1} + p h_{t-1})^{-1}) \exp\left[-\lambda x_t (w + q x_{t-1} + p h_{t-1})^{-1}\right] \tag{16}$$

The chain rule can then be used to express these differentials as functions of the log likelihood ($\mathcal{LL}$):

$$\frac{d\left(\ln\left[\mathrm{f}(x)\right]\right)}{dx} = \frac{\mathrm{f}'(x)}{\mathrm{f}(x)} \Rightarrow$$

$$\frac{\partial \mathcal{LL}}{\partial w} = \lambda x_t (w + q x_{t-1} + p h_{t-1})^{-2} \tag{17}$$

$$\frac{\partial \mathcal{LL}}{\partial q} = \lambda x_t x_{t-1} (w + q x_{t-1} + p h_{t-1})^{-2} \tag{18}$$

$$\frac{\partial \mathcal{LL}}{\partial p} = \lambda x_t h_{t-1} (w + q x_{t-1} + p h_{t-1})^{-2} \tag{19}$$

$$\frac{\partial \mathcal{LL}}{\partial \lambda} = \lambda^{-1} - x_t (w + q x_{t-1} + p h_{t-1})^{-1} \tag{20}$$

A search algorithm, such as the simplex method [30], can be used to find estimates for p, q and w based on the observed durations $\mathbf{x}_{1:T}$ and then these estimates can be used in (17) - (20) in order to derive the Fisher score of a new observation for each of the four parameters. This will be done for price durations on the front bid x_{Bid} and x_{Ask} sides so that an 8 dimensional Fisher score vector can be calculated for each data instance:

$$\mathbf{g}_t^{ACD} = \left\{ \frac{\partial \mathcal{LL}_t}{\partial w_{Bid}}, \frac{\partial \mathcal{LL}_t}{\partial q_{Bid}}, \frac{\partial \mathcal{LL}_t}{\partial p_{Bid}}, \frac{\partial \mathcal{LL}_t}{\partial \lambda_{Bid}}, \frac{\partial \mathcal{LL}_t}{\partial w_{Ask}}, \frac{\partial \mathcal{LL}_t}{\partial q_{Ask}}, \frac{\partial \mathcal{LL}_t}{\partial p_{Ask}}, \frac{\partial \mathcal{LL}_t}{\partial \lambda_{Ask}} \right\} \tag{21}$$

5.2 Poisson processes

Poisson processes permeate the market microstructure literature in their descriptions of limit order arrival rates, volume changes and order cancellations - see for example [52]. However, their modelling of trade arrival rates are subject to criticism and it is suggested that renewal processes might be better placed to do this - see for example [39], [38], [49] and [50].

If we denote the volume at each depth i of the order book ($i \in [1 \dots 2D]$ assuming D levels on the bid side and D on the ask) at time t as V_t^i, we can use $\mathit{\Delta} V_i$ to represent the rate of change of volume at a depth over a given time interval τ:

$$\mathit{\Delta} V_i = \frac{\left|V_{t+\tau}^i - V_t^i\right|}{\tau} \tag{22}$$

We can model $\mathit{\Delta} V_i$ using a Poisson process, i.e. $\mathit{\Delta} V_i \sim Poi(\lambda_i)$:

$$P(\mathit{\Delta} V_i = x) = \frac{e^{-\lambda_i \tau}(\lambda_i \tau)^x}{x!} \tag{23}$$

Setting the time interval τ to 1, the log likelihood of a rate x_i observed at depth i for a model parameterised by λ_i can be expressed:

$$\begin{aligned}\mathcal{LL} &= \log\left(\frac{e^{-\lambda_i}(\lambda_i)^{x_i}}{x_i!}\right)\\ &= -\lambda_i + x_i\log(\lambda_i) - \log(x_i!) \end{aligned} \tag{24}$$

Differentiating (24) with respect to λ_i we can derive the Fisher score:

$$\frac{\partial \mathcal{LL}}{\partial \lambda_i} = -1 + \frac{x_i}{\lambda_i} \tag{25}$$

The parameter λ_i needs to be estimated at each depth i. The Maximum Likelihood (ML) estimate of λ_i can be calculated by adjusting (24) to take into account a set of N observations:

$$\begin{aligned}\mathcal{LL}_N &= \sum_{j=1}^{N}\log\left(\frac{e^{-\lambda_i}(\lambda_i)^{x_i^j}}{x_i^j!}\right)\\ &= -N\lambda_i + \left(\sum_{j=1}^{N}x_i^j\right)\log(\lambda_i) - \sum_{j=1}^{N}\log(x_i^j!) \end{aligned} \tag{26}$$

Differentiating (26) wrt λ_i and setting this to zero yields our ML estimate of λ_i:

$$\begin{aligned}\frac{\partial \mathcal{LL}_N}{\partial \lambda_i} &= -N + \left(\sum_{j=1}^{N}x_i^j\right)\frac{1}{\lambda_i} = 0\\ \Rightarrow \hat{\lambda}_i &= \frac{1}{N}\sum_{j=1}^{N}x_i^j \end{aligned} \tag{27}$$

$\hat{\lambda}_i$ is substituted into (25) and a $2D$ dimensional Fisher score vector created for the D levels on each side:

$$\mathbf{g}_t^{Poiss} = \left\{\frac{\partial \mathcal{LL}_t}{\partial \lambda_1}, \ldots, \frac{\partial \mathcal{LL}_t}{\partial \lambda_{2D}}\right\} \tag{28}$$

5.3 Wiener process barrier model

Lancaster's (1992) Wiener process barrier model [31] assumes that the price evolution of an asset follows a Wiener process:

$$dp_t = \mu dt + \sigma dz \tag{29}$$

where the price of the asset p_t follows a random walk with drift μ and variance σ^2.

This means that the price movement of an asset over a time period t will be distributed :

$$p_t - p_{t-1} = \Delta p_t \sim N(\mu t, \sigma^2 t) \tag{30}$$

and that the ML estimates of μ and σ can be ascertained from a sequence of N such price movements:

$$\hat{\mu} = \frac{1}{N}\sum_{t=1}^{N}\Delta p_t \tag{31}$$

$$\hat{\sigma^2} = \frac{1}{N}\sum_{t=1}^{N}(\Delta p_t - \hat{\mu})^2 \tag{32}$$

Modelling the price with such a process means that the likelihood of a limit order priced at a distance α from the current mid price surviving at least t time units without being hit is:

$$\mathcal{L} = \Phi\left(\frac{\alpha - \mu t}{\sigma\sqrt{t}}\right) - \exp\left[\frac{2\mu\alpha}{\sigma^2}\right]\Phi\left(\frac{-\alpha - \mu t}{\sigma\sqrt{t}}\right) \tag{33}$$

where Φ represents the standardised cumulative Gaussian distribution function.

Using the substitutions $x = \dfrac{\alpha - \mu t}{\sigma\sqrt{t}}$, $y = \dfrac{-\alpha - \mu t}{\sigma\sqrt{t}}$ and $z = \exp\left[\frac{2\mu\alpha}{\sigma^2}\right]$ so that:

$$\mathcal{L} = \Phi(x) - z\Phi(y) \tag{34}$$

the derivatives of $\mathcal{L}$ wrt μ and σ can be calculated:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mu} &= \frac{\partial x}{\partial \mu}\frac{\partial(\Phi(x))}{\partial x} - \frac{\partial z}{\partial \mu}\Phi(y) - z\frac{\partial y}{\partial \mu}\frac{\partial(\Phi(y))}{\partial y} \\
&= -\frac{\sqrt{t}}{\sigma}\phi(x) - \frac{2\alpha}{\sigma^2}z\Phi(y) + z\frac{\sqrt{t}}{\sigma}\phi(y) \qquad (35)\\
\frac{\partial \mathcal{L}}{\partial \sigma} &= \frac{\partial x}{\partial \sigma}\frac{\partial(\Phi(x))}{\partial x} - \frac{\partial z}{\partial \sigma}\Phi(y) - z\frac{\partial y}{\partial \sigma}\frac{\partial(\Phi(y))}{\partial y} \\
&= -\left(\frac{\alpha - \mu t}{\sigma^2\sqrt{t}}\right)\phi(x) + \frac{4\mu\alpha}{\sigma^3}z\Phi(y) + z\left(\frac{-\alpha - \mu t}{\sigma^2\sqrt{t}}\right)\phi(y) \\
&= -\frac{x}{\sigma}\phi(x) + \frac{4\mu\alpha}{\sigma^3}z\Phi(y) + z\frac{y}{\sigma}\phi(y) \qquad (36)
\end{aligned}$$

where ϕ represents the standardised Gaussian distribution function.
The chain rule can then be used to express these differentials as functions of the log likelihood:

$$\frac{\partial \mathcal{LL}}{\partial \mu} = \frac{-\frac{\sqrt{t}}{\sigma}\phi(x) - \frac{2\alpha}{\sigma^2}z\Phi(y) + z\frac{\sqrt{t}}{\sigma}\phi(y)}{\Phi(x) - z\Phi(y)} \tag{37}$$

$$\frac{\partial \mathcal{LL}}{\partial \sigma} = -\frac{-\frac{x}{\sigma}\phi(x) + \frac{4\mu\alpha}{\sigma^3}z\Phi(y) + z\frac{y}{\sigma}\phi(y)}{\Phi(x) - z\Phi(y)} \tag{38}$$

$\hat{\mu}$ from (31) and $\hat{\sigma^2}$ from (32) can be substituted into (37) and (38) in order to derive the 4 dimensional Fisher score vector for a new set of observations:

$$\mathbf{g}_t^{Wiener} = \left\{\frac{\partial \mathcal{LL}_t}{\partial \mu_{Bid}}, \frac{\partial \mathcal{LL}_t}{\partial \sigma_{Bid}}, \frac{\partial \mathcal{LL}_t}{\partial \mu_{Ask}}, \frac{\partial \mathcal{LL}_t}{\partial \sigma_{Ask}}\right\} \tag{39}$$

6 Experiments

6.1 Feature sets investigated

In the experiments that are described below, the following four price-based features based on common price-based trading rules (which are described briefly in Section 10.2 of the appendix) were constructed:

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

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

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

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

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.

Denoting the volume at time t at each of the price levels of an 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), these price-based features were augmented by four more sets of features constructed as follows :

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

Finally Fisher kernels as described in (21), (28) and (39) are constructed :

$$\mathcal{F}_{9\ldots11} = \{\mathbf{g_{ACD}}, \mathbf{g_{Poiss}}, \mathbf{g_{Wiener}}\}$$

6.2 Kernel Mappings

A mapping set consisting of the following kernel types is used :

$$\mathcal{K}_{1:5}(RBF_{1:5}) = \left\{\exp\left(-\left\|\mathbf{x}-\mathbf{x}'\right\|^2/\sigma_1^2\right), \ldots, \exp\left(-\left\|\mathbf{x}-\mathbf{x}'\right\|^2/\sigma_5^2\right)\right\}$$
$$\mathcal{K}_{6:10}(Poly_{1:5}) = \left\{\left(\langle\mathbf{x},\mathbf{x}'\rangle+1\right)^{d_1}, \ldots, \left(\langle\mathbf{x},\mathbf{x}'\rangle+1\right)^{d_5}\right\}$$
$$\mathcal{K}_{11:15}(INN_{1:5}) = \left\{\frac{2}{\pi}\sin^{-1}\left(\frac{2\mathbf{x}^T\Sigma_1\mathbf{x}'}{\sqrt{(1+2\mathbf{x}^T\Sigma_1\mathbf{x})(1+2\mathbf{x}'^T\Sigma_1\mathbf{x}')}}\right), \ldots, \frac{2}{\pi}\sin^{-1}\left(\frac{2\mathbf{x}^T\Sigma_5\mathbf{x}'}{\sqrt{(1+2\mathbf{x}^T\Sigma_5\mathbf{x})(1+2\mathbf{x}'^T\Sigma_5\mathbf{x}')}}\right)\right\}$$

This means that altogether there are $|\mathcal{F}| \times |\mathcal{K}| = 11 \times 15 = 165$ individual kernels along with an average kernel constructed by taking the average of the 165 kernel matrices, giving 166 individual kernels altogether[1].

6.3 Experimental design

In the problem of currency prediction, any move that is predicted has to be significant enough to cross the spread in the appropriate direction if the trader is to profit from it and here we are concerned with a multiclass classification of that movement. We attempt to predict whether the trader should go long by buying the currency pair because $P^{Bid}_{t+\Delta t} > P^{Ask}_t$, go short by selling it because $P^{Ask}_{t+\Delta t} < P^{Bid}_t$ or do nothing because $P^{Bid}_{t+\Delta t} < P^{Ask}_t$ and $P^{Ask}_{t+\Delta t} > P^{Bid}_t$.

With this in mind, three SVM are trained on the data with the following labelling criteria for each SVM:

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

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 $\boldsymbol{y}_t = [\pm1, \pm1, \pm1]$. 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.

Predictions for time horizons (Δt) of 5, 10, 20, 50, 100 and 200 seconds into the future are created. Training and prediction are 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 became the current window's in-sample set. The data consists of 6×10^4 instances of order book updates sampled at a frequency of 1 second for the EURUSD currency pair from the EBS exchange on 2/11/2009.

6.3.1 Reduced kernel set

With the aim of understanding what happens when the MKL set is reduced, two kernel sub-sets will also be investigated :

- *Selection based on Stability / Highest Weightings (Reduced Set A)* A reduced set will be formed by investigating the weightings from a standard MKL run and taking the N_A most commonly selected individual kernels along with the N_A kernels with the most significant weightings for each of the six time horizons.

- *Selection based on Efficacy (Reduced Set B)* A reduced set will be formed by taking the N_B most effective individual SVM for each of the six time horizons.

N_A and N_B will be selected to give roughly the same size subset in both cases (which will not simply be $N \times 6$ because of overlap, for example because some kernels may be allocated high weightings for multiple time horizons etc).

[1] Note that the average kernel is only used as a benchmark and is not included in the set of kernels used for combination by MKL.

6.3.2 Benchmark

All the results will be compared with a very simple benchmark, namely a classifier which predicts for the entire out-of-sample region, whatever the most common class was in the in-sample period. For example, if the price was mostly moving up in the in-sample period and had hence been assigned '+1' in the majority of cases, then this benchmark classifier would assign '+1' to the entire out-of-sample period. This benchmark embodies both the trend-following technical indicator and the principle of a random classifier (which is significantly more effective than simply predicting each class with a probability of a third).

7 Results

7.1 MKL weightings

Figure 7 in the appendix shows the average weighting that MKL allocated each of the individual SVM over the different time horizons. In order to understand whether there is a particular kernel mapping, feature or class of feature which is consistently awarded the highest weightings, Figures 1, 2 and 3 show these weightings summarised in these three different manners.

Fig. 1 Average weighting of individual SVM by feature type

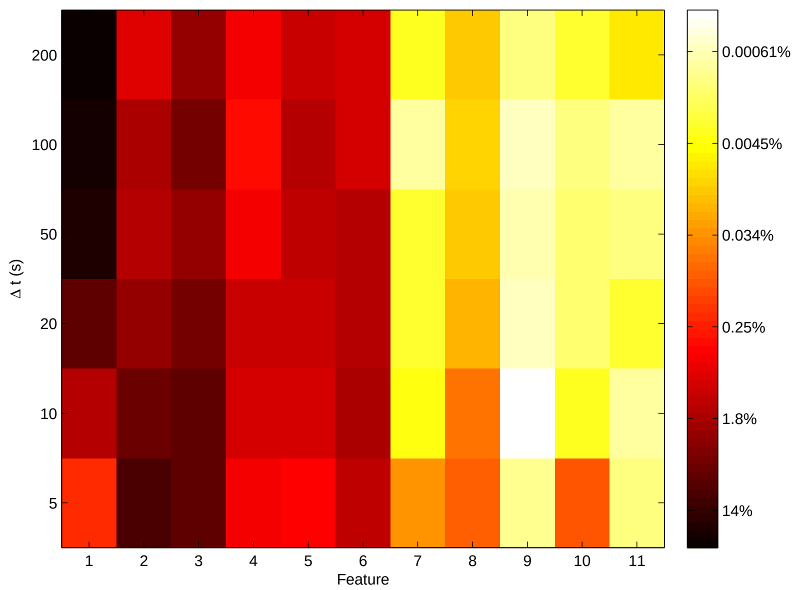


Fig. 2 Average weighting of individual SVM by kernel mapping

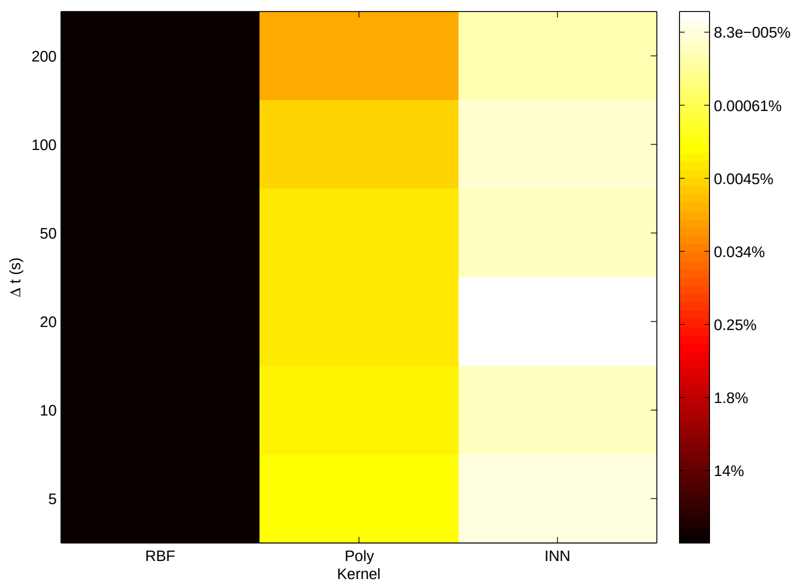


Fig. 3 Average weighting of individual SVM by feature class

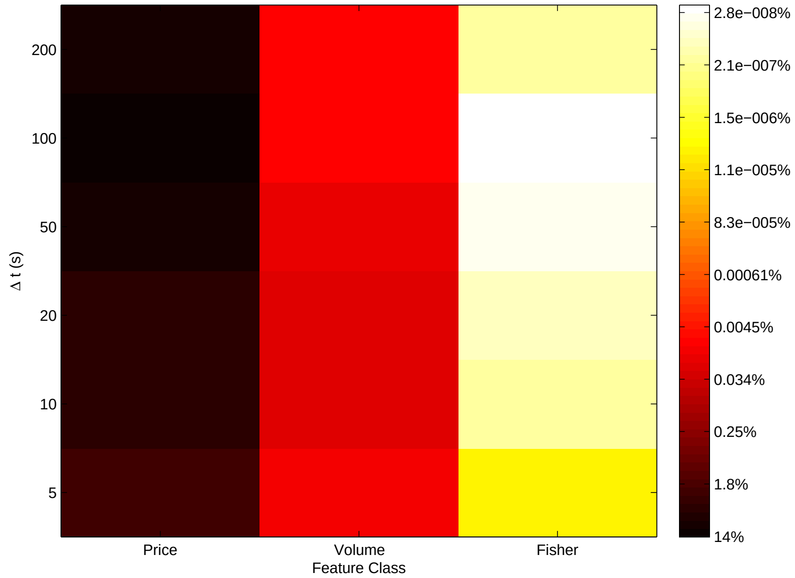


Figure 1 indicates that the first 6 features (F_1 to F_6) are awarded significantly higher weightings on average than the others. One can also see that the RBF kernel mapping is allocated much higher weightings than the Polynomial and INN mappings from Figure 2. Furthermore, Figure 3 shows that price based features are favoured by MKL significantly more than volume or Fisher based ones, the latter being awarded very low weightings. In all cases, there is a great deal of consistency in these weightings across the different time horizons.

7.1.1 Kernel subset

Selection based on Stability / Highest Weightings (Reduced Set A)
Using $N_A = 3$, the overall set contained the following 7 members:

- F_1RBF_1 EMA Price based feature
- F_2RBF_1 [SMA SD] Price based feature
- F_3RBF_1 [Mins Maxs] Price based feature
- F_3RBF_2 [Mins Maxs] Price based feature
- F_3RBF_5 [Mins Maxs] Price based feature
- F_5RBF_5 $\mathbf{V}_t$ Volume based feature
- F_6RBF_5 $\frac{\mathbf{V}_t}{\|\mathbf{V}_t\|_1}$ Volume based feature

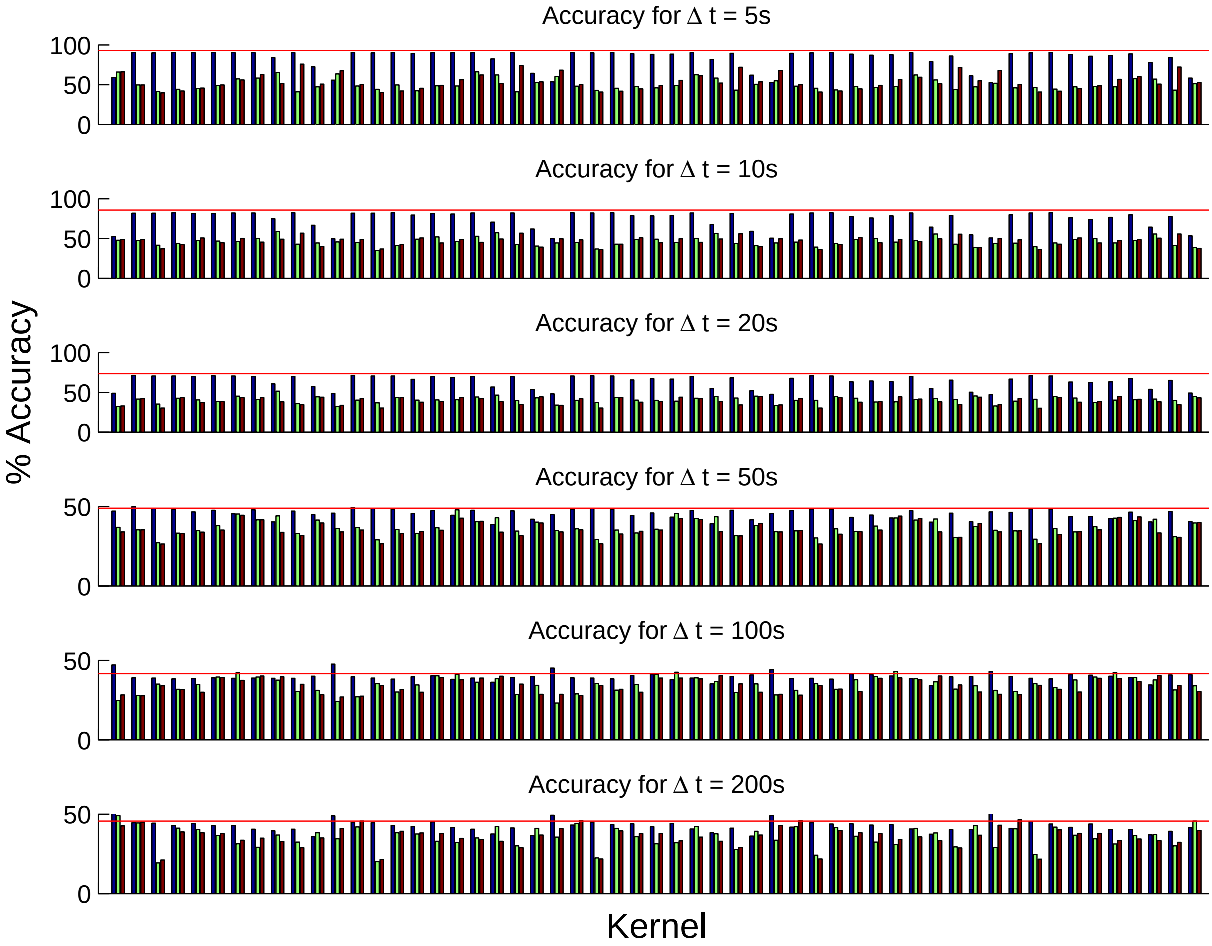
Selection based on Efficacy (Reduced Set B)
Using $N_B = 2$, the overall set contained the following 8 members:

- F_1RBF_1 EMA Price based feature
- F_2RBF_1 [SMA SD] Price based feature
- F_4RBF_1 [Ups Downs] Price based feature
- F_6RBF_1 $\frac{\mathbf{V}_t}{\|\mathbf{V}_t\|_1}$ Volume based feature
- $F_{10}RBF_1$ Poisson Fisher feature
- F_1RBF_2 EMA Price based feature
- F_2RBF_2 [SMA SD] Price based feature
- F_1RBF_5 EMA Price based feature

7.2 Performance

In order to investigate which kernel mapping / feature combinations were the best performers when used individually, the percentage accuracy of each out-of-sample period was calculated over the different predictive time horizons. Figure 4 shows the average of this accuracy for all of the 165 kernels for each of the horizons along with the performance of the benchmark (shown as the horizontal red line). One can see that the benchmark outperforms all the individual SVM for shorter time horizons and the majority of them for the longer horizons.

Fig. 4 % accuracy of individual SVM against benchmark



In order to be able to compare the MKL methods with the set of individual SVM in terms of percentage accuracy and frequency of predictions, the individual SVM with the highest percentage accuracy for each time horizon were selected and referred to as if they were one individual SVM. This aggregate kernel *Individual Best* was composed as follows:

Table 1 SVM that *Individual Best* is composed from

Δt	**Kernel**
5	F_6RBF_1
10	$F_{10}RBF_1$
20	F_2RBF_1
50	F_2RBF_1
100	F_1RBF_2
200	F_1RBF_5

Figures 5 and 6 compare the percentage accuracy and proportion of instances predictions were possible for the benchmark, average kernel, the full MKL set and subsets A and B.

Fig. 5 Accuracy comparison of different dethods

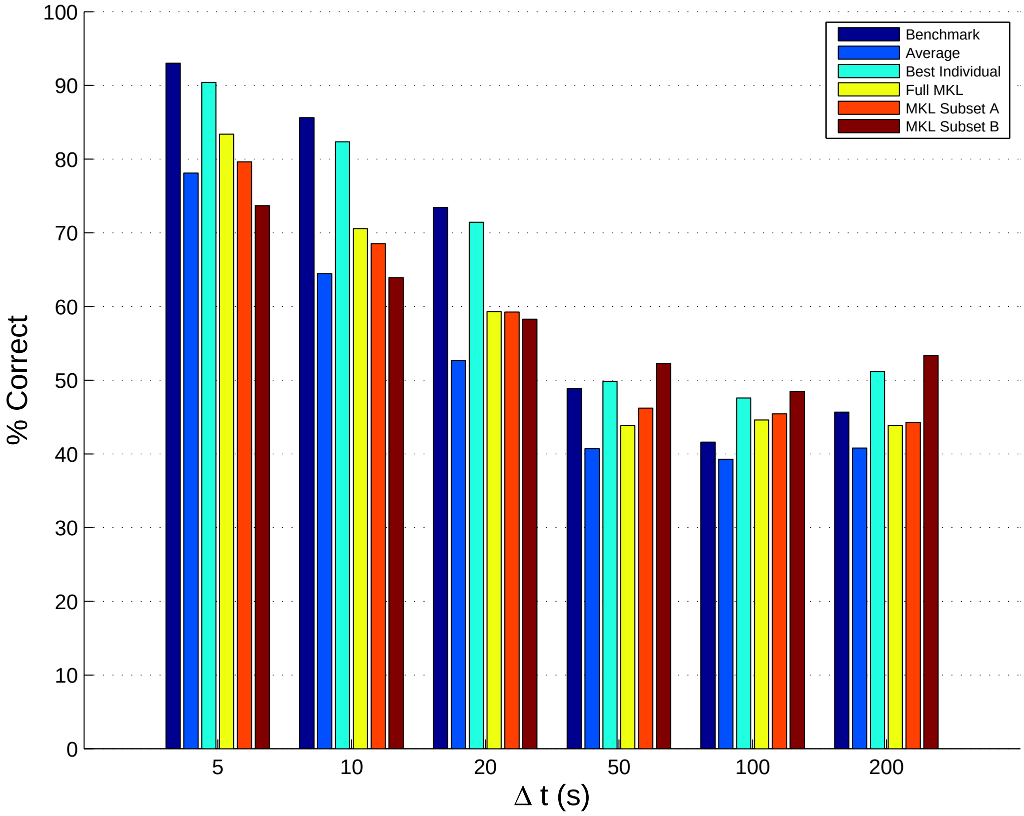


Fig. 6 Proportion of times predictions possible for each method

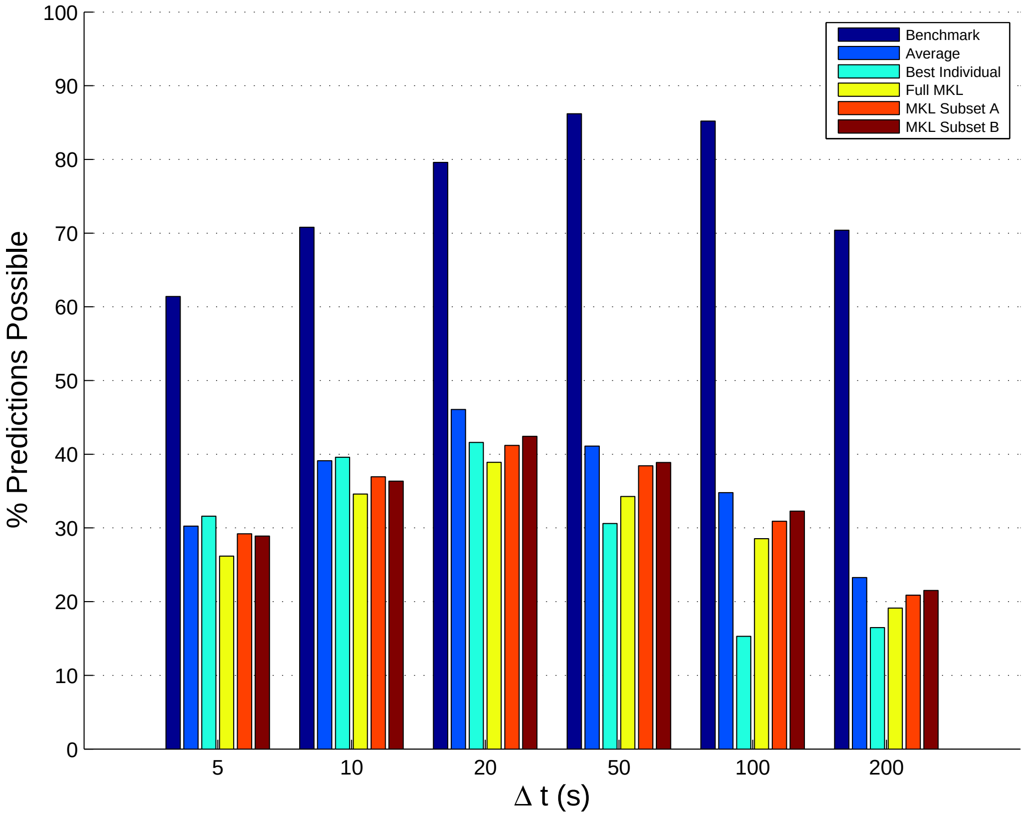


Figure 5 shows that the percentage accuracy of both subsets is better than the full MKL set for the longer time horizons ($\Delta t \geq 50$), but worse for the shorter ones. Subset A outperforms Subset B for the shorter time horizons, but this is reversed for longer ones, where Subset B is the most effective predictor. The benchmark outperforms all other methods for shorter time horizons, but is generally worse for longer ones. The performance of the average kernel is worse than all the other methods over all time horizons. The most effective individual SVM is only marginally less effective than the benchmark for the shorter time horizons and then of Subset B for the longer ones.

Figure 6 shows that the benchmark is able to make predictions significantly more frequently than any of the other methods. Following this, the average kernel is able to make predictions more frequently than any of the other techniques. Less noticeably, methods based on Subsets A and B are able to make predictions slightly more frequently than those based on the full MKL set. All methods are able to make predictions most frequently over medium time horizons (i.e. $20 \leq \Delta t \leq 50$), with the number of predictions possible falling as the time horizon is increased or decreased.

8 Conclusion

Referring to Figure 4 one can clearly see that the benchmark has a very high percentage accuracy for short time horizons, exceeding that of all the individual SVM for $\Delta t \leq 20$s. If one considers the benchmark's construction, this significant accuracy over shorter time periods is not surprising; one would expect trends that are evident in the 100s of in-sample training data to persist for a reasonable time with the effect tailing off with the longer forecast horizons over which the trend is less likely to have persisted to. The fact that the accuracies of the individual SVM are so different to the benchmark for the shorter time horizons highlights the fact that they are doing something different to simply trend following, and for longer time horizons, actually something more effective.

SimpleMKL clearly favours features F_1 to F_6 (all the price based features and half of the volume-based features) and the RBF mapping with much higher weightings across all time horizons - as can be seen in Figures 1, 2 and 3. The choice of the RBF kernel mapping can be explained through its significant outperformance in predictive accuracy, but the bias towards features F_1 to F_6 can not be explained through their predictive accuracy.

The kernel subsets described in Section 7.1.1 consist mostly of price-based features in both cases, with the addition of a couple of volume-based features in subset A (where the kernels were selected based on their weighting in the full-set SimpleMKL) and one volume based feature and one Fisher feature in subset B (where the kernels were selected based on the performance as individual SVM). Subset A's composition reflects Simple MKL's proclivity for price based features, whilst subset B's reflects the fact that a price-based feature often had the highest predictive accuracy for any given time horizon. The kernel mapping is always RBF in both subsets, reflecting Simple MKL's strong bias to selecting it as a mapping and its significant predictive outperformance relative to the other kernel mappings.

Looking at Figure 5, one can see that reducing the size of the set from which SimpleMKL chooses its kernels to focus on ones that one would suspect would improve performance, either because full-set SimpleMKL allocated them higher weightings or because they had higher predictive accuracies when used individually, does increase performance for the longer time horizons. The fact that it does not do so for the shorter ones suggests again that a large proportion of the predictive ability of the SVM based approach, be it individual SVM or in combination through SimpleMKL, is based on short term trend-following and that the larger set used in full-set MKL is better able to capitalise on this than smaller sets based around more potentially effective predictors. The fact that the benchmark outperforms the other methods for shorter time horizons, but that this is reversed for the longer ones, lends greater credence to this idea. The same is also true for Subset A's shorter time horizon outperformance / longer time under performance relative to Subset B; it is possible that the kernels that Subset B is composed of, being selected because of their outperformance as individual SVM, are doing more genuine prediction over and above the bias component of trend-following than their counterparts in Subset A.

The outperformance of all three MKL methods relative to the average kernel at all time horizons indicates that the weighting procedure that SimpleMKL employs is much more effective when it comes to the resulting predictor than simply allocating all the individual kernels in the full set an equal weighting. There are clearly some individual kernels in this full set that significantly reduce the performance of any predictor that includes them and SimpleMKL is able to weed these out.

We have shown that for time horizons $\geq$ 50s, significant outperformance in terms of predictive accuracy relative to a benchmark can be found using MKL on a set of features (subset B) based on previous price action, order book volume data and a simple market microstructural model. An average accuracy of 55% is achieved when classifying the direction of price movement into one of three categories for a 200s predictive time horizon.

Acknowledgements This work would not have been possible without the data that EBS (ICAP) and the Chicago Mercantile Exchange (CME) kindly agreed to supply me with for research purposes.

9 Appendix

9.1 Further Figures

Fig. 7 Average weighting of individual SVM

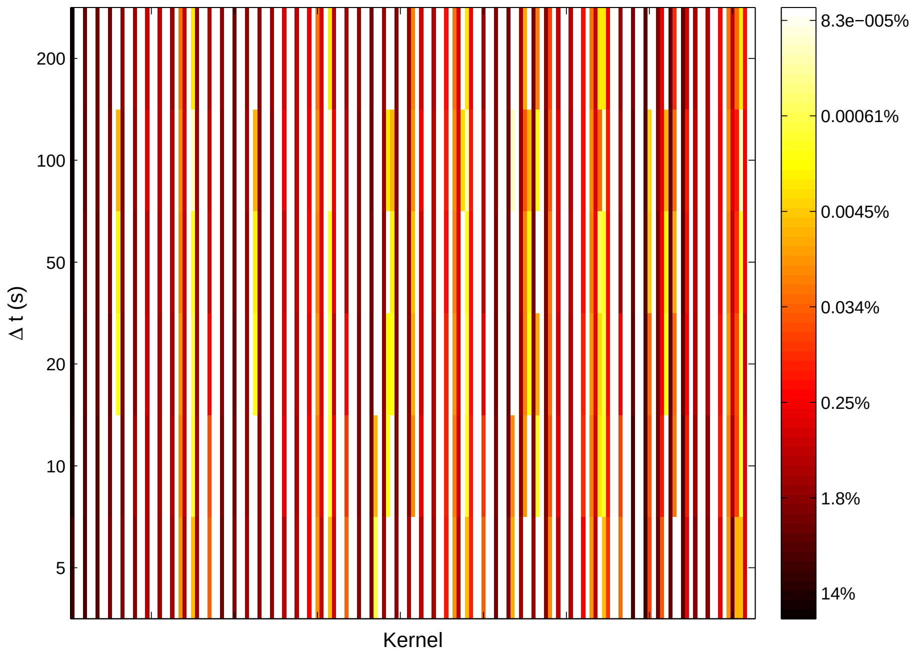


Fig. 8 Average weighting for Subset A

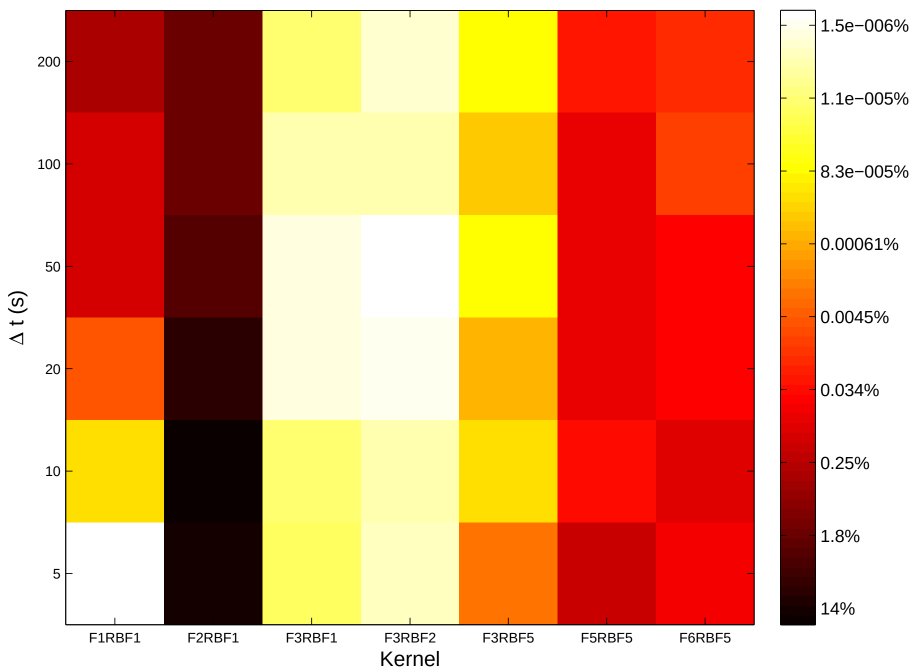
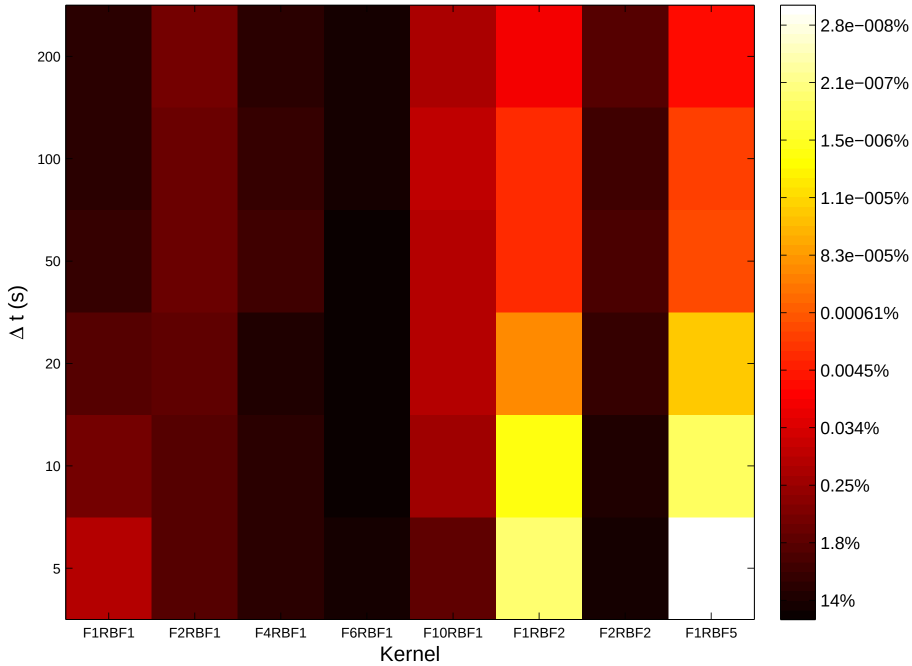


Fig. 9 Average weighting for Subset B



9.2 Trading Rules

- $\mathcal{F}_1$: A common trading rule is the moving average crossover technique (see for example [2]) 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 vice versa.
- $\mathcal{F}_2$: Breakout trading rules (see for example [15]) 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 [6], 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 [15] 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 [60] 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 vice versa.

References

1. Alp, O.S., Buyukbebeci, E., Cekic, A.I., Ozkurt, F.Y., Taylan, P., Weber, G.W.: Cmars and gam & cqp - modern optimization methods applied to international credit default prediction. Journal of Computational and Applied Mathematics **235**(16), 4639 – 4651 (2011)
2. Appel, G.: Technical Analysis: Power Tools for Active Investors. Financial Times (2005)
3. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning, p. 6. ACM, New York, NY, USA (2004). DOI http://doi.acm.org/10.1145/1015330.1015424
4. BIS: Triennial central bank survey of foreign exchange and derivatives market activity in 2007. http://www.bis.org/publ/rpfxf07t.htm (2007)
5. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence **97**(1-2), 245 – 271 (1997)
6. Bollinger, J.A.: Bollinger on Bollinger Bands. McGraw-Hill (2001)
7. Cao, L.: Support vector machines experts for time series forecasting. Neurocomputing **51**, 321–339 (2003)
8. Chalup, S.K., Mitschele, A.: Kernel methods in finance. In: Handbook on Information Technology in Finance, pp. 655–687 (2008)
9. Cortes, C., Vapnik, V.: Support vector networks. In: Machine Learning, pp. 273–297 (1995)

10. Cristianini, N., Shawe-Taylor, J.: An introduction to support Vector Machines: and other kernel-based learning methods. Cambridge University Press, New York, NY, USA (2000)
11. Dittmar, R., Neely, C.J., Weller, P.: Is technical analysis in the foreign exchange market profitable? a genetic programming approach. CEPR Discussion Papers 1480, C.E.P.R. Discussion Papers (1996)
12. Easley, D., Engle, R.F., O'Hara, M., Wu, L.: Time-varying arrival rates of informed and uninformed trades. Tech. rep., EconWPA (2002)
13. Easley, D., O'Hara, M.: Time and the process of security price adjustment. Journal of Finance **47**(2), 576–605 (1992)
14. Engle, R., Russell, J.: Autoregressive conditional duration: A new model for irregularly spaced transaction data. Econometrica **66**, 1127–1162 (1998)
15. Faith, C.: Way of the Turtle. McGraw-Hill Professional (2007)
16. Fletcher, T., Hussain, Z., Shawe-Taylor, J.: Multiple kernel learning on the limit order book. In: JMLR Proceedings, vol. 11, pp. 167–174 (2010)
17. Fletcher, T., Redpath, F., D'Alessandro, J.: Machine learning in fx carry basket prediction. In: Proceedings of the International Conference of Financial Engineering, vol. 2, pp. 1371–1375 (2009)
18. Gehler, P., Nowozin, S.: Infinite kernel learning (2008)
19. Gestel, T.V., Suykens, J.A.K., Baestaens, D.E., Lambrechts, A., Lanckriet, G., Vandaele, B., Moor, B.D., Vandewalle, J.: Financial time series prediction using least squares support vector machines within the evidence framework. In: IEEE Transactions on Neural Networks, pp. 809–821 (2001)
20. Hasbrouk, J.: Empirical Market Microstructure: The Institutions, Economics, and Econometrics of Securities Trading. Oxford University Press, USA (2006)
21. Hazarika, N., Taylor, J.G.: Predicting bonds using the linear relevance vector machine, chap. 17, pp. 145–155. Springer-Verlag (2002)
22. Huang, S.C., Wu, T.K.: Wavelet-based relevance vector machines for stock index forecasting. In: International Joint Conference on Neural Networks (IJCNN), pp. 603–609 (2006)
23. Huang, S.C., Wu, T.K.: Combining wavelet-based feature extractions with relevance vector machines for stock index forecasting. Expert Systems **25**, 133–149 (2008)
24. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Comput. Oper. Res. **32**(10), 2513–2522 (2005)
25. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: In Advances in Neural Information Processing Systems 11, pp. 487–493. MIT Press (1998)
26. Jondeau, E., Perilla, A., Rockinger, G.M.: Optimal Liquidation Strategies in Illiquid Markets. SSRN eLibrary (2008)
27. Kaufman, P.: The New Trading Systems and Methods. John Wiley & Sons (2005)
28. Kim, K.: Financial time series forecasting using support vector machines. Neurocomputing **55**, 307–319 (2003)
29. Kuan, C.M., Liu, T.: Forecasting exchange rates using feedforward and recurrent neural networks. Journal of Applied Econometrics **10**(4), 347–64 (1995)
30. Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E.: Convergence properties of the nelder–mead simplex method in low dimensions. SIAM J. on Optimization **9**, 112–147 (1998). DOI http://dx.doi.org/10.1137/S1052623496303470. URL http://dx.doi.org/10.1137/S1052623496303470
31. Lancaster, T.: The Econometric Analysis of Transition Data. Cambridge University Press (1992)
32. Lanckriet, G.R.G., De Bie, T., Cristianini, N., Jordan, M.I., Noble, W.S.: A statistical framework for genomic data fusion. Bioinformatics **20**(16), 2626–2635 (2004). DOI http://dx.doi.org/10.1093/bioinformatics/bth294
33. LeBaron, B.: Technical trading rule profitability and foreign exchange intervention. NBER Working Papers 5505, National Bureau of Economic Research, Inc (1996)
34. Linnainmaa, J., Rosu, I.: Time Series Determinants of Liquidity in a Limit Order Market. SSRN eLibrary (2008)
35. Lo, A., MacKinlay, A., Zhang, J.: Econometric Models of Limit-Order Executions. SSRN eLibrary (1997)
36. Lui, Y.H., Mole, D.: The use of fundamental and technical analyses by foreign exchange dealers: Hong kong evidence. Journal of International Money and Finance **17**(3), 535–545 (1998)
37. Luss, R., d'Aspremont, A.: Predicting abnormal returns from news using text classification (2009)
38. Mainardi, F., Gorenflo, R., Scalas, E.: A fractional generalization of the poisson processes. Vietnam Journal of Mathematics **32**, 53–64 (2004)
39. Mainardi, F., Raberto, M., Gorenflo, R., Scalas, E.: Fractional calculus and continuous-time finance ii: the waiting-time distribution. Physica A: Statistical Mechanics and its Applications **287**, 468 (2000)
40. Marney, C.: Building robust fx trading systems. FX Trader Magazine (2010)
41. Neely, C.J.: Technical analysis and the profitability of u.s. foreign exchange intervention. Review (Jul), 3–17 (1998)
42. Neely, C.J., Weller, P.A.: Intraday technical trading in the foreign exchange market. Working Papers 1999-016, Federal Reserve Bank of St. Louis (2001)
43. Neely, C.J., Weller, P.A., Ulrich, J.M.: The adaptive markets hypothesis: Evidence from the foreign exchange market. Journal of Financial and Quantitative Analysis **44**(02), 467–488 (2009)
44. Özöğür-Akyüz, S., Weber, G.W.: Modelling of kernel machines by infinite and semi-infinite programming. In: A. Halim Hakim, P. Vasant, & N. Barsoum (ed.) American Institute of Physics Conference Series, *American Institute of Physics Conference Series*, vol. 1159, pp. 306–313 (2009)
45. Özöğür-Akyüz, S., Weber, G.W.: On numerical optimization theory of infinite kernel learning. J. of Global Optimization **48**, 215–239 (2010)
46. P. Tino, N.N., Yao, X.: Volatility forecasting with sparse bayesian kernel models. In: Proc. 4th International Conference on Computational Intelligence in Economics and Finance, Salt Lake City, UT, pp. 1150–1153 (2005)
47. Parlour, C., Seppi, D.: Limit order markets: A survey. In: A. Boot, A. Thakor (eds.) Handbook of Financial Intermediation and Banking, chap. 3, pp. 61–96. Elsevier Science, Amsterdam, The Netherlands (2008)
48. Perez-cruz, F., Afonso-rodriguez, J., Giner, J.: Estimating garch models using support vector machines. Quantitative Finance **3**(3), 163–172 (2003)
49. Politi, M., Scalas, E.: Activity spectrum from waiting-time distribution. Physica A: Statistical Mechanics and its Applications **383**(1), 43 (2007)
50. Politi, M., Scalas, E.: Fitting the empirical distribution of intertrade durations. Physica A: Statistical Mechanics and its Applications **387**(8-9), 2025 (2008)

51. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: Simplemkl. Journal of Machine Learning Research **9**, 2491–2521 (2008)
52. Rydberg, T., Shephard, N.: A modelling framework for the prices and times of trades made on the New York stock exchange. Cambridge University Press (2000)
53. Shadbolt, J., Taylor, J.G. (eds.): Neural networks and the financial markets: predicting, combining and portfolio optimisation. Springer-Verlag, London, UK (2002)
54. Tay, F., Cao, L.: Application of support vector machines in financial time series forecasting. Omega **29**, 309–317 (2001)
55. Tay, F., Cao, L.: Modified support vector machines in financial time series forecasting. Neurocomputing **48**, 847–861 (2002)
56. Taylan, P., Weber, G.W., Beck, A.: New approaches to regression by generalized additive models and continuous optimization for modern applications in finance, science and technology. Optimization **56**(5-6), 675–698 (2007)
57. Ullrich, C., Seese, D., Chalup, S.: Foreign exchange trading with support vector machines. In: Advances in Data Analysis, pp. 539–546. Springer Berlin Heidelberg (2007)
58. Walczak, S.: An empirical analysis of data requirements for financial forecasting with neural networks. J. Manage. Inf. Syst. **17**(4), 203–222 (2001)
59. Wang, L., Zhu, J.: Financial market forecasting using a two-step kernel learning method for the support vector regression. Annals of Operation Research **174**, 103–120 (2010)
60. Wilder, J.W.: New Concepts in Technical Trading Systems. Trend Research (1978)