Machine Learning and Order Book Dynamics

Dr Tristan Fletcher

Introduction

- Aim is to show how traditional models of market behaviour can be incorporated into Machine Learning framework.

- Run through:
  - Predictive problem
  - Order books / features
  - Fisher Kernels
  - Three models of market behaviour
  - Multiple Kernel Learning
  - Parameter Learning
  - Preliminary experiments

- Assume little knowledge of finance but significant ML knowledge.

Problem

- It would obviously be useful to be able to predict which way an asset might move over the short term:
  - Buy low, sell high.
- It is common to assume that previous price action of asset in question has prognostic ability – e.g. that there is momentum / mean reversion.
- Set of tools commonly termed as Technical Analysis which vary in their complexity and plausibility (e.g. moon cycles!) deal with this area.
- There are also usually strong economic reasons why asset prices might relate to each other that can also be informative:
  - E.g. equity prices of companies in similar industry, different currencies against same base etc.
- However, many of these techniques are useful over longer time horizons (days to months) and less useful over the very short term where the effects they pick up on haven't had a chance to be borne out.
- Here we look at shorter term predictions (seconds to minutes).

Limit Order books

- What is useful over the second to minute range is the aggregate behaviour of buyers and sellers who are trading the asset in question.
- In many markets, the participants interact on an exchange where the trading takes place.
- Limit orders for an asset, e.g. a currency such as EURUSD (€/$), specify whether a party wishes to buy or sell the currency, the amount (volume) desired, and the price the transaction will occur at.
- These orders are accumulated in a limit order book, with buy orders sitting on the *Bid* side and sell orders sitting on the *Ask*:

	Price	Volume ($M)
	1.4752	1
Ask	1.4751	8
	1.4750	3
Spread ⇕		
	1.4749	15
Bid	1.4748	5
	1.4747	9

Simple Volume Features

- Starting point is to describe the volumes resting on the order book at time t as a vector $\boldsymbol{V}_t$.

- Construct very simple set of features that seem plausible for predicting direction currency might move.

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

- From practical experience (≈ 5 years ago), individually these features typically explain 5-10% of the variance when regressed (linearly) on a predictive horizon Δt of 10s in a liquid currency pair.

Support Vector Machines

- One problem of using standard techniques, e.g. linear regression, is that due to their simplicity they are commonly employed
  - This erodes their efficacy.

- Furthermore, their use does not always encourage a statistically rigorous approach to the problem.
  - e.g. turn models off for circumstantial reasons and miss pocket of good performance.

- One obvious way of capitalising on the dimensionality of the data (and the skillset of my then PhD supervisor) is to use Support Vector Machines (SVM) (Cortes and Vapnik 1995).

- This is an optimisation which finds the α_i that maximises:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j) \text{ where } \alpha_i \geq 0 \; \forall_i, \; \sum_{i=1}^{N} \alpha_i y_i = 0$$

- SVM are quick to run and find global optima.

- Kernel mapping often empirically selected.

Fisher Kernels

- The Fisher Kernel (Jaakkola and Haussler 1998) represents a method for incorporating generative probability models into discriminative framework.

- More importantly, it allows us to take some models of market microstructure (i.e. order book dynamics) and integrate them into SVM.

- 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 $\partial\mathcal{L} / \partial\theta_i$.

- The Fisher kernel incorporates this principle by creating a kernel composed of values of $\partial\mathcal{L} / \partial\theta_i$ 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 to be $\log \mathcal{L}(x)$, the Fisher score of x is the vector gradient of $\log \mathcal{L}(x)$:

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

- The practical Fisher kernel is then defined as:

$$\kappa(x, z) = \mathbf{g}(\boldsymbol{\theta}, x)' \mathbf{g}(\boldsymbol{\theta}, z)$$

Informed Vs Noise Traders

- Literature (and experience) suggest that one should expect stochastic clustering in the rate of price changes for a financial asset.
- Ill-informed traders trade randomly according to a stochastic process (e.g. Poisson), 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.
- One should therefore see clustering of trading following an information event because of the increased numbers of informed traders.

Autoregressive Conditional Duration Model

- Engle and Russel's (1998) Autoregressive Conditional Duration model 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:

$$h_t = w + \sum_{i=1}^{L} q_i x_{t-i} + \sum_{i=1}^{L} p_i h_{t-i}$$
$$x_t = h_t \epsilon_t, \quad \epsilon_t \sim Exp(\lambda)$$

- Where x_t is the duration of the price at time t, h_t is its expected duration, L is the lag of the auto-regressions and w, p, q and λ are constants.

- Simplifying this to be an order 1 auto-regressive process, the likelihood of the model can be expressed:

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

Autoregressive Conditional Duration Model

- The derivative of the log likelihood w.r.t .each parameter can then be derived:

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

- Estimates for p, q and w based on the observed durations $x_{1:T}$ can be found and then these estimates can be used in order to derive the Fisher score of a new observation for each of the four parameters.

- This is done for price durations on the front bid x_{Bid} and ask x_{Ask} sides so that an 8 dimensional Fisher score vector can be calculated for each data instance:

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

Poisson Processes

- Poisson processes permeate the market microstructure literature in their descriptions of limit order arrival rates, volume changes and order cancellations.

- Denote the volume at each depth i of the order book ($i \in [1 \ldots 2D]$ assuming D levels on the bid side and D on the ask) at time t as $V^{\mathrm{i}}{}_t$.

- Use ΔV_i to represent the rate of change of volume at a depth over a given time interval τ:

$$\Delta V_i = \frac{\left| V^i_{t+\tau} - V^i_t \right|}{\tau}$$

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

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

Poisson Processes

- 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:

$$\mathcal{LL} = -\lambda_i + x_i \log(\lambda_i) - \log(x_i!)$$

- This gives rise to a Fisher score for each depth i:

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

- So that the $2D$ Fisher score vector created for the D levels on each side is:

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

Wiener Process Barrier Model

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

$$dp_t = \mu dt + \sigma dz$$

- 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:

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

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

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

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

Wiener Process Barrier Model

- 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:

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

- Where Φ represents the standardised cumulative Gaussian distribution function.

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

- Gives Fisher scores of :

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

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

- So that the 4-dimensional Fisher score vector for a new set of observations would be:

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

Financial Features

- Motivated by common trading rules, other features can be created that look at previous price action.
- These can be investigated along with the simple volume based features and the Fisher scores:
  - Exponential Moving Average Crossover $\mathcal{F}_1 = \left\{ EMA_t^{L_1}, \ldots, EMA_t^{L_N} \right\}$
  - Bollinger Bands $\mathcal{F}_2 = \left\{ MA_t^{L_1}, \ldots, MA_t^{L_N}, \sigma_t^{L_1}, \ldots, \sigma_t^{L_N} \right\}$
  - Donchian Trend System $\mathcal{F}_3 = \left\{ P_t, \max_t^{L_1}, \ldots, \max_t^{L_N}, \min_t^{L_1}, \ldots, \min_t^{L_N} \right\}$
  - Relative Strength Index $\mathcal{F}_4 = \left\{ \Uparrow_t^{L_1}, \ldots, \Uparrow_t^{L_N}, \Downarrow_t^{L_1}, \ldots, \Downarrow_t^{L_N} \right\}$
  - Simple Volume Features $\mathcal{F}_{5\ldots8} = \left\{ \boldsymbol{V}_t, \frac{\boldsymbol{V}_t}{\|\boldsymbol{V}_t\|_1}, \boldsymbol{V}_t - \boldsymbol{V}_{t-1}, \frac{\boldsymbol{V}_t - \boldsymbol{V}_{t-1}}{\|\boldsymbol{V}_t - \boldsymbol{V}_{t-1}\|_1} \right\}$
  - Fisher Score Features $\mathcal{F}_{9\ldots11} = \left\{ \mathbf{g}_{\mathbf{ACD}}, \mathbf{g}_{\mathbf{Poiss}}, \mathbf{g}_{\mathbf{Wiener}} \right\}$

Multiple Kernel Learning

- In an analogous manner to the combination of trading rules that a systematic trader might employ, one can attempt to extract the information from these features simultaneously.

- One way to do this is to turn the features into kernels and combine them using Multiple Kernel Learning (MKL) (Lanckriet *et al*. 2004).

- This uses the same optimisation as SVM but with an additional stage that selects a convex combination of kernels.

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

- Literature shows that most implementations of SVM use a kernel mapping (often Radial Basis Function) that is selected circumstantially / with little theoretical basis.
  - MKL mitigates this to some extent.

Parameter Learning

- It is possible to extend MKL to learning the hyperparameter values of the kernel mappings and the parameters used in the Fisher kernels themselves.

- Referring back to the MKL optimisation we use to find the set or kernel weightings d_t which minimise the following quantity:

$$\beta = \max_{\alpha} \left\{ -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_t d_t \kappa_t(x_i, x_j) + \sum_i \alpha_i \right\}$$

- Once d_t and α have been selected, this is equivalent to maximising:

$$\sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_t d_t \kappa_t(x_i, x_j) \equiv \boldsymbol{\alpha}^T \boldsymbol{Y} \mathcal{K}(\boldsymbol{\theta}) \boldsymbol{Y} \boldsymbol{\alpha}$$

- This means that for a given kernel $\mathcal{K}$ parameterised by a hyperparameter / Fisher parameter set $\boldsymbol{\theta}$, the optimum value of $\boldsymbol{\theta}$ is:

$$\max_{\boldsymbol{\theta}} \boldsymbol{\alpha}^T \boldsymbol{Y} \mathcal{K}(\boldsymbol{\theta}) \boldsymbol{Y} \boldsymbol{\alpha}$$

$$= \max_{\boldsymbol{\theta}} \mathcal{C}(\boldsymbol{\theta})$$

- Where $\boldsymbol{\alpha}$ is ascertained from the support vectors of an SVM trained with the target values in $\boldsymbol{Y}$.

Parameter Learning

- Calculating derivatives of $\mathcal{C}(\boldsymbol{\theta})$ with respect to each of the components in $\boldsymbol{\theta}$, namely $\nabla\mathcal{C}(\boldsymbol{\theta})$:

$$\begin{aligned}\nabla C(\boldsymbol{\theta}) &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(\boldsymbol{\alpha}^T \boldsymbol{Y} K(\boldsymbol{\theta}) \boldsymbol{Y} \boldsymbol{\alpha}\right) \\ &= \boldsymbol{\alpha}^T \boldsymbol{Y} \frac{\partial K(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \boldsymbol{Y} \boldsymbol{\alpha}\end{aligned}$$

- Allows one to use a gradient search method in as part an iterative process to solve for the parameter set as well as the Kernel weightings.

- However, we found that the increased computational complexity of the parameter learning on the parameter values was not justified by the very limited improvement in performance.
  - This was due to over-fitting – but results are not discussed here.

- This was relative to using a range of hyperparameters spaced evenly in log space and maximum likelihood estimates for the Fisher parameters.

Experiments

- Investigated performance of using 11 features described before passed through 15 different kernel mappings (RBF, linear and Infinite Neural Network (Williams 1998) – each with five different sets of hyperparameters):

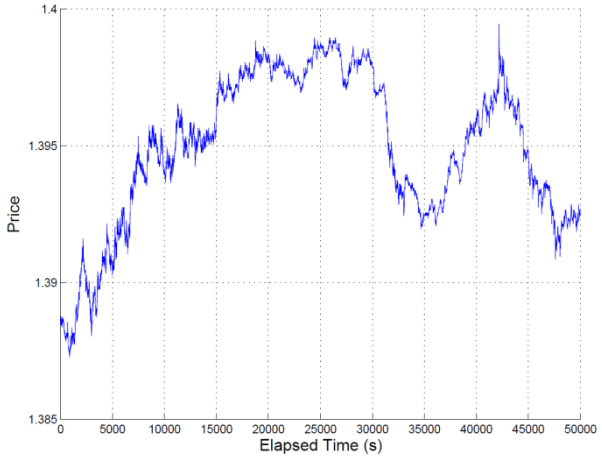
$$\mathcal{K}_{1:5} = \left\{ \exp\left(-\left\| x - x' \right\|^2 / \sigma_1^2 \right), \ldots, \exp\left(-\left\| x - x' \right\|^2 / \sigma_5^2 \right) \right\}$$

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

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

- $|\boldsymbol{F}| \times |\boldsymbol{K}|$ = 11 x 15 = 165 individual mapping / feature combinations.

- EURUSD for an entire day predicting over time horizons Δt of 5, 10, 20, 50, 100 and 200s.



Experiments

- Method used 100 in sample data points to predict movement over time horizon:
  - *(t-100):t* $\longrightarrow$ *t+*Δt

- Investigated prognostic ability of 3 class prediction by training three predictors in parallel and ignoring predictions when they disagree (i.e. when more than one of the output signs is +ve).

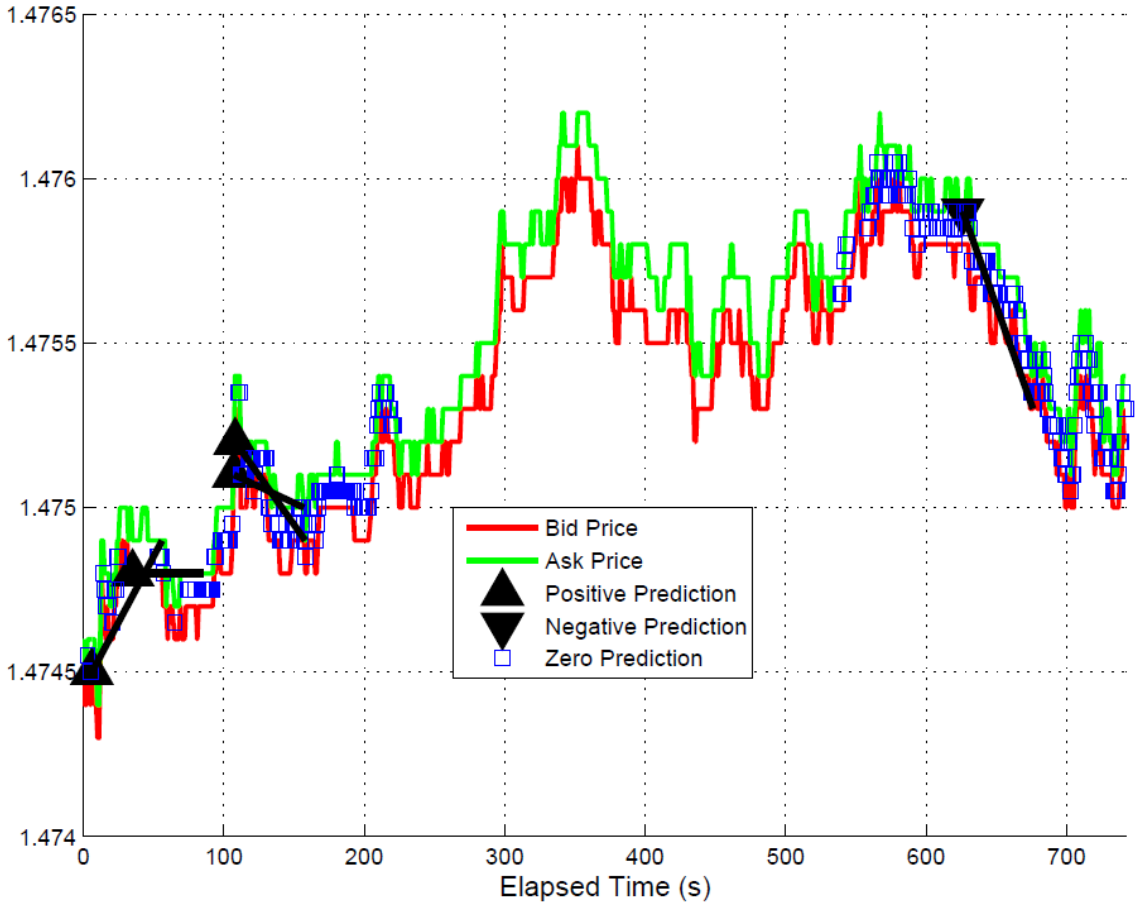
$$\begin{aligned}
&\text{SVM 1:} P^{Bid}_{t+\Delta t} > P^{Ask}_{t} && \Rightarrow y^1_t = +1, \text{otherwise } y^1_t = -1 \\
&\text{SVM 2:} P^{Ask}_{t+\Delta t} < P^{Bid}_{t} && \Rightarrow y^2_t = +1, \text{otherwise } y^2_t = -1 \\
&\text{SVM 3:} P^{Bid}_{t+\Delta t} < P^{Ask}_{t}, P^{Ask}_{t+\Delta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

- Mappings investigated individually as well as in combination through MKL.

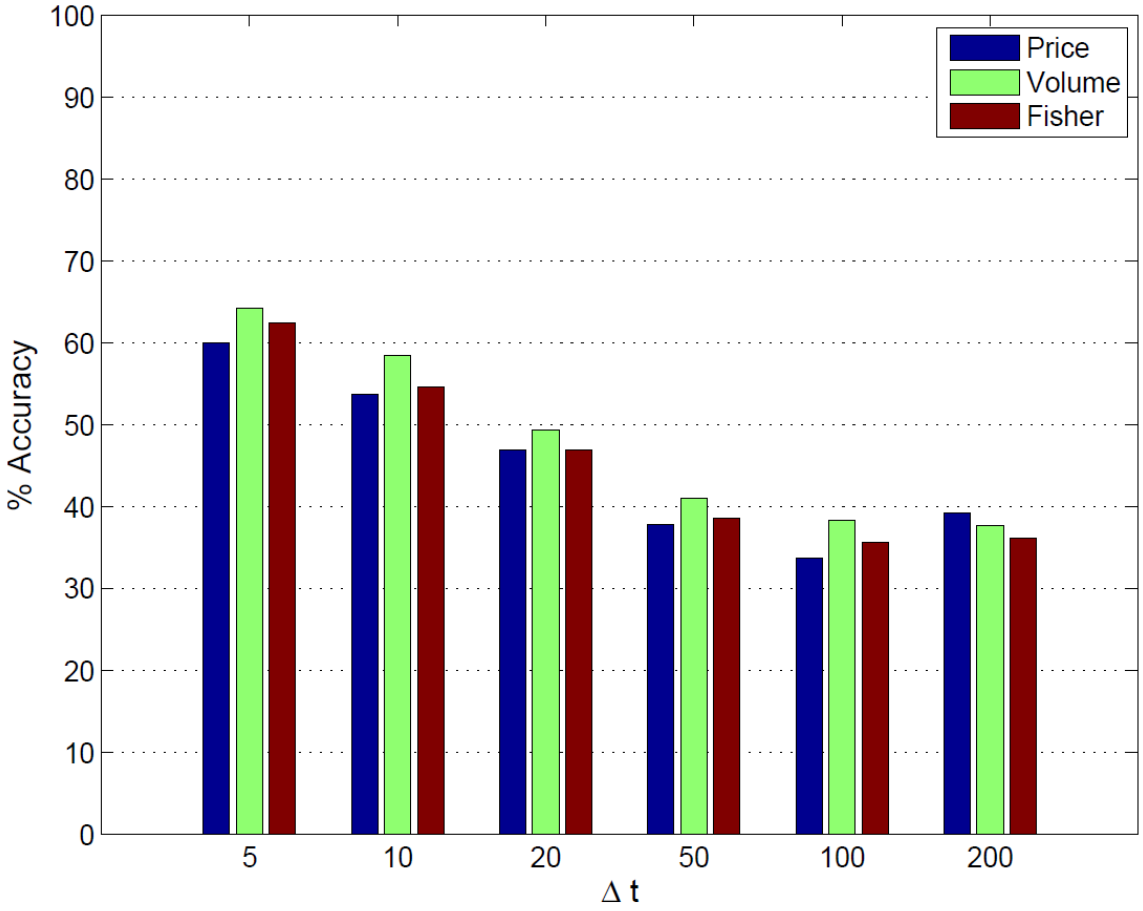
- Two subsets formed based on best performing individual kernel / feature mappings (*A*) and those given highest weighting in MKL (*B*).

- Benchmarked against a classifier which makes prediction based on most common move in the in sample period.

Example of Predictions

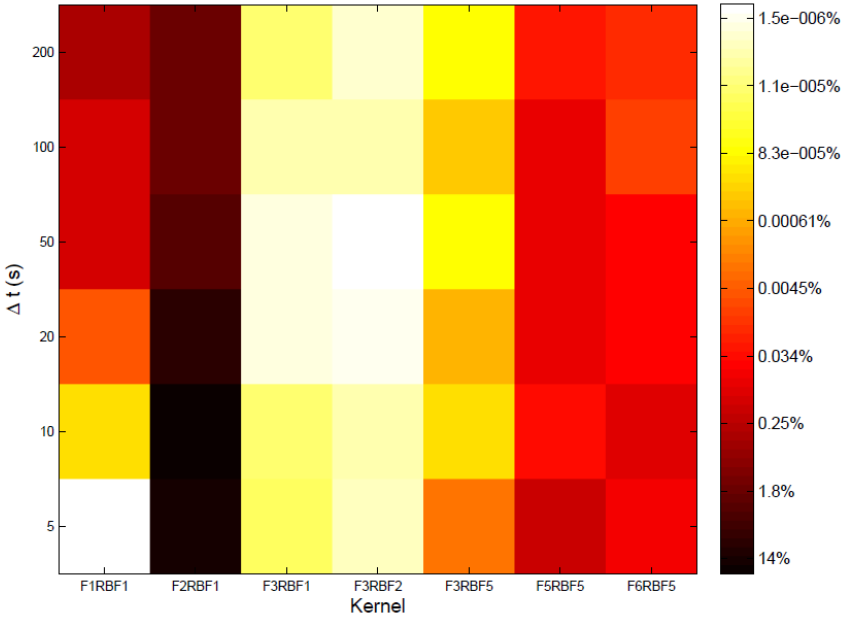


% Accuracy of Individual SVM by Feature Class



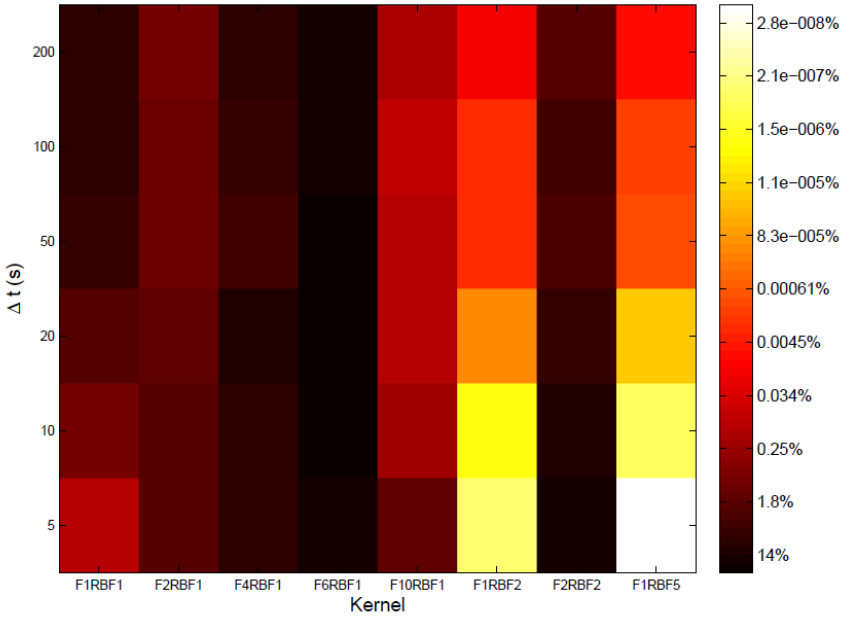
Kernel / Feature Weightings A

- 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 Volume based feature
- F_6RBF_5 WeightedVolumes Volume based feature

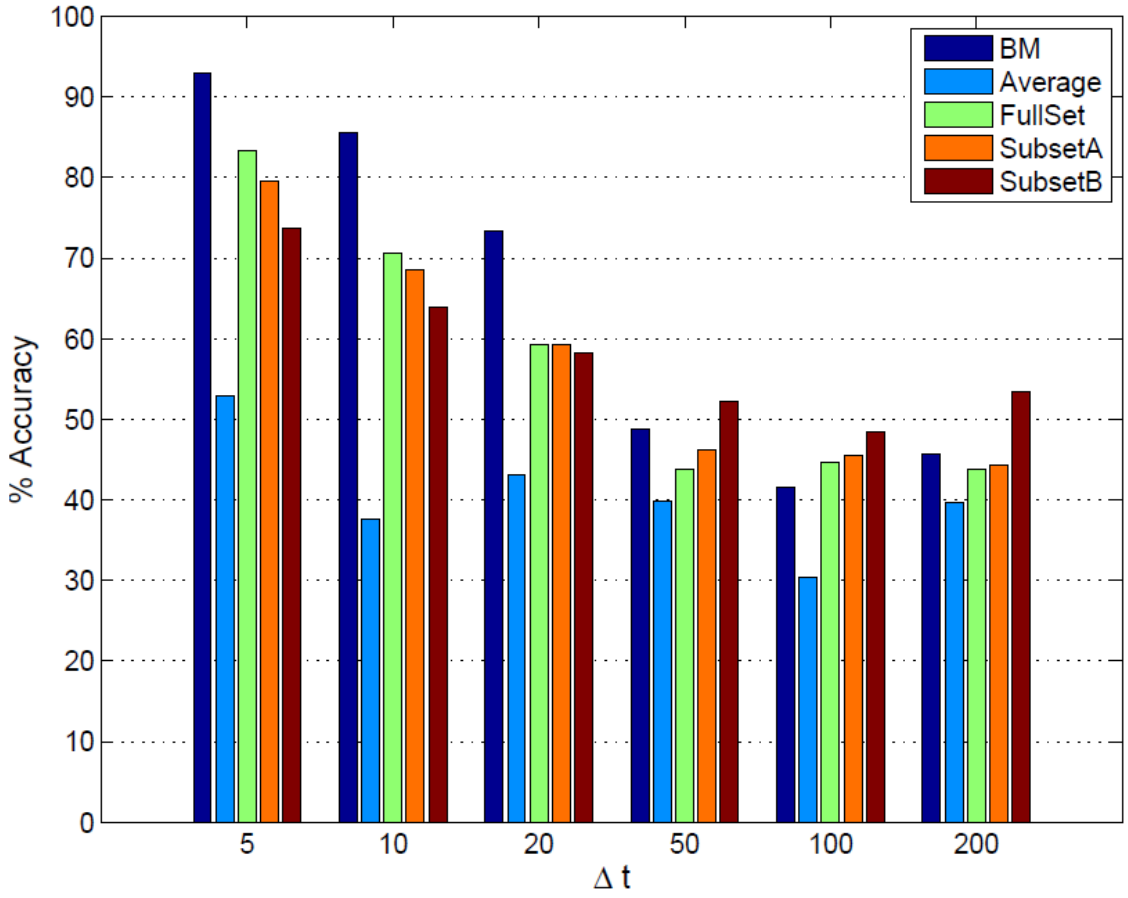


Kernel / Feature Weightings B

- 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 WeightedVolumes 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



Overall Performance



Summary

- Order book volumes are informative for market movements in the short term.
- One can construct market models based on the information they contain and incorporate them into the machine learning framework through Fisher Kernels.
- The performance of individual SVM shows the expected performance profile with increasing return horizon:
  - volume based features optimal for very short time horizons, price based features for longer horizons.
- When kernels are combined using MKL outperformance of a simple bench mark is found.
- Parameter learning was not worth the effort.

Appendix: References

- C. Cortes and V. Vapnik, “Support vector networks,” in *Machine Learning*,1995, pp. 273–297
- T. Jaakkola and D. Haussler, “Exploiting generative models in discriminative classifiers,” in *In Advances in Neural Information Processing Systems 11*. MIT Press, 1998, pp. 487–493.
- R. Engle and J. Russell, “Autoregressive conditional duration: A new model for irregularly spaced transaction data,” *Econometrica*, vol. 66, pp. 1127–1162, 1998.
- T. Lancaster, *The Econometric Analysis of Transition Data*. Cambridge University Press, 1992.
- G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble, “A statistical framework for genomic data fusion,” *Bioinformatics*, vol. 20, no. 16, pp. 2626–2635, 2004.
- C. Williams, “Computation with infinite neural networks,” *Neural Computation*, vol. 10, no. 5, pp. 1203–1216, 1998.