Multiple Kernel Learning on the Limit Order Book

AMS 520

Jason Bohne, Jarryd Scully, and Paul Vespe

Applied Mathematics and Statistics Department
Stony Brook University

December 12, 2022

Overview

1. Introduction
2. Classification and SVM
3. Statistical Inference
4. High-Frequency Quote Analysis
5. Numerical Results

Group Composition

- Jason Bohne; Quantitative Finance Ph.D. Student
- Jarryd Scully; Quantitative Finance M.S. Student
- Paul Vespe; Statistics M.S. Student

Reference Paper

- *Multiple Kernel Learning on the Limit Order Book, 2010*
- Authored by Fletcher, Hussain,and Shawe-Taylor
- Affiliated with the University College of London
- Published in the Journal of Machine Learning Research Proceedings

Linear Classification

- Decides the class y_i using a linear decision function on the observation vector x_i
- Binary linear classification specifies two classes; *wlog* $y_i = 1, -1$
- Linear classifier has the following form, where w and b are unknown a priori

$$x \rightarrow sgn(w^T x - b)$$

- Solving the inference problem allows for estimates of w and b for model calibration
- Natural to extend this to multi-class formulations via one vs. one, one vs. all

Support Vector Machines

- Specific formulation of a linear classifier; common in machine learning literature
- Identifies optimal hyperplane $w^T x - b$ to split the feature space into classes
- In multi-class formulations for p features the hyperplane will have dimension $p - 1$
- Hard margin constrains the model to have exact accuracy; often overfitting
- Regularization via soft margin allows for misclassifications with hinge loss function

$$\max(0, 1 - (w^T x - b)y_i)$$

Objective Functions for Single Kernel SVM

- Primal formulation solves for the optimal decision function $f(x) + b$

$$\min_{f,b,\varepsilon} \frac{1}{2}||f||^2_{\mathcal{H}} + C\sum_i \varepsilon_i$$

$$y_i f(x_i) + y_i b \geq 1 - \varepsilon_i,\ \ \varepsilon_i \geq 0$$

- Solving for Lagrangian dual; objective is reformulated to a maximization problem

$$\max_{\alpha} -\frac{1}{2}\sum_{(i,j)} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_i \alpha_i$$

Multiple Kernel Support Vector Machines

- In the primal objective the decision function is decomposed into distinct $f_m(x)$
- Each $f_m(x)$ corresponding to a reproducing kernel Hilbert space $\mathcal{H}_m$ with kernel K_m
- Common kernel basis have Gaussian, polynomial, or sigmoid functional forms
- As the dual is convex and differentiable; can solve optimization with gradient descent
- For a fixed coefficient vector d_i determine the α^* that maximizes the dual objective
- Compute gradient of the dual $\frac{\partial J}{\partial d_m}$ and descent direction D in closed-form
- Update current estimate with a step size γ determined by line search

Primal Objective for Multiple Kernel SVM

- Primal problem assumes classification function $f(x)$ is of form

$$f(x) + b = \sum_m f_m(x) + b$$

- Where each function $f_m(x)$ corresponds to a kernel K_m

Primal MKL Problem:

$$\min_{f_m} \frac{1}{2}\left(\sum_m \frac{1}{d_m}||f_m||^2_{\mathcal{H}}\right)^2 + C\sum_i \varepsilon_i$$

$$y_i \sum_m f_m(x_i) + y_i b \geq 1 - \varepsilon_i,\ \ \varepsilon_i \geq 0,\ \sum_m d_m = 1,\ \ d_m \geq 0$$

Dual Objective for Multiple Kernel SVM

- Solving for the Lagrangian dual objective of the primal problem

$$J(d) = \max_{\alpha} -\frac{1}{2} \sum_{(i,j)} \alpha_i^* \alpha_j^* y_i y_j \sum_m d_m K_m(x_i, x_j) + \sum_i \alpha_i^*$$

- Gradient of the dual $\frac{\partial J}{\partial d_m}$ computed for evaluation in descent direction

$$\frac{\partial J}{\partial d_m} = -\frac{1}{2} \sum_{(i,j)} \alpha_i^* \alpha_j^* y_i y_j K_m(x_i x_j)$$

Statistical Inference

Algorithm Gradient Descent Algorithm (SimpleMKL)

set $d_m = \frac{1}{M}$ for $m = 1 \ldots M$
while $\mathrm{P}_{obj} - D_{obj} > \epsilon$ **do**
  Evaluate $J(d)$ using SVM with $K = \sum_m d_m K_m$
  Compute $\frac{\partial J}{\partial d_m}$ and descent direction D
  set $\mu = \arg\max_m d_m, J^* = 0, d^* = 0, D^* = 0$
  while $J^* < J(d)$ **do**
    $d = d^*,\ D = D^*$
    $\nu = \arg\min \frac{-d_m}{D_m}\ \gamma_{\max} = \frac{-d_\nu}{D_\nu}$
    $d^* = d + \gamma_{\max} D,\ \ D^*_\mu = D_\mu - D_\nu\ D^*_\nu = 0$
    Compute J^* using SVM with $K = \sum_m d^*_m K_m$
  end while
  Line search candidate γ_i for $d \leftarrow d + \gamma_{opt} D$
end while

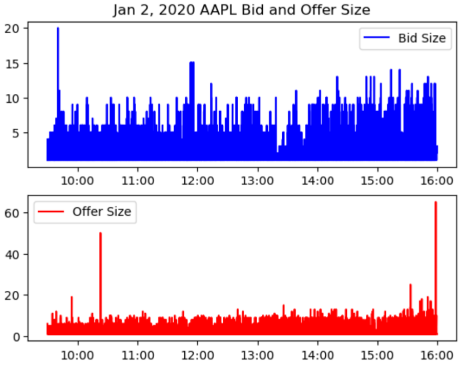
Exploratory Quote Analysis

- Quotes are contracts to buy or sell an asset at a specific price and size
- AAPL top of the book quote data between 01/01/2020 and 01/30/2020.
- Each observation is composed of the best bid and best offer price and size.
- Observations reindexed to participant timestamp; time for which arrives to SIP.
- Removed quotes outside market hours or with a zero bid/offer price/inverted spread.
- Cleaned dataset consists of 11 million observations.

	Exchange	Symbol	Best_Bid_Price	Best_Bid_Size	Best_Offer_Price	Best_Offer_Size
2020-01-02 09:30:00.134062	P	AAPL	296.24	2.0	296.29	1.0
2020-01-02 09:30:00.134336	K	AAPL	296.21	1.0	296.29	1.0
2020-01-02 09:30:00.134532	K	AAPL	296.10	1.0	296.29	1.0
2020-01-02 09:30:00.136081	K	AAPL	296.10	1.0	296.29	1.0
2020-01-02 09:30:00.234474	K	AAPL	296.11	1.0	296.29	1.0

Figure: Sample Quotes

Statistical Quote Analysis



(a) AAPL Bid and Offer Size Sample Date

Table: Summary Statistics

Condition	Mean	Standard Deviation
Best Bid Size	2.43	1.68
Best Offer Size	2.66	1.45
Change in Bid Size	-2e-5	1.42
Change in Offer Size	-5e-5	1.62
Spread	0.05	0.10

(b) Summary Statistics for AAPL Quotes

Paper Motivation

- Modern day trading takes place electronically in a high-frequency environment.
- Available liquidity on limit order book can be indicative of future market dynamics.
- Size, imbalance, and change of liquidity are interpretable features.
- Paper attempts to forecast short term trend direction off limit order book data.
- Profitable trading strategy arises from consistent short term trend prediction.

Defining the Problem

- Directionality of short term trend can be formulated as a classification problem
- Classes can be up trend, down trend, or no trend; specified below
- Predictive signals over time would allow for profitable trading strategy
- Requires the identification of features relevant to future price movement

Classification outcomes:

$$P^{Bid}_{t+\delta t} > P^{Ask}_{t} \rightarrow \text{Up trend}$$
$$P^{Bid}_{t+\delta t} < P^{Ask}_{t} \rightarrow \text{Down Trend}$$
$$P^{Bid}_{t+\delta t} < P^{Ask}_{t}, P^{Ask}_{t+\delta t} > P^{Bid}_{t} \rightarrow \text{No Trend}$$

Our Features

- Feature set composed of 1-second aggregated price and size features
- *Spread* (S): Difference between the current ask price and bid price.
- *Spread-Change* (SC): Difference in current spread and previous spread.
- *Weighted Spread* (WS): Difference in average volume-weighted ask/bid price.
- *Weighted Spread Anomaly* (WSA): If weighted spread is above 95% quantile.

$$F = \left(V_t, V_t - V_{t-1}, S_t, SC_t, WS_t, WSA_t \right)$$

Methods

- For computational feasibility partition data into batches of 300 observations
- From 1-second feature aggregation this corresponds to 5 minute intervals
- Train SVMs across batches using subsequent batch for evaluation
- SVMs utilize single kernel and multiple kerne; functions
- Allows for short term memory along with efficient computation

Single Kernel Training

- Train single kernel SVMs using linear and Gaussian kernels
- Employ a grid search to fit optimal bandwidth parameter in Gaussian Kernel.
- Train SVMs across multivariate feature vector to predict price direction
- Compare accuracy and weighted precision across subsequent evaluation batch

Linear vs. Gaussian SVM Performance

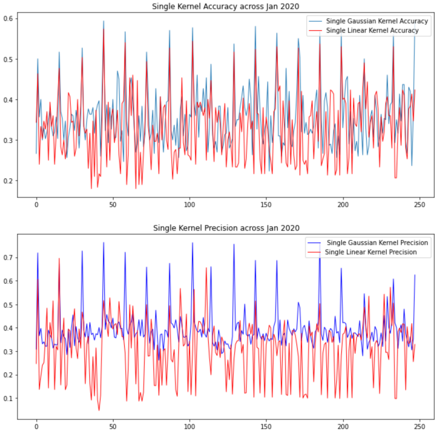


Figure: Accuracy and Precision of Single Gaussian vs. Linear Kernel SVM

Multiple Kernel Training

- Original python implementation applies the gradient descent algorithm
- Quality test is to pass in identical kernels with hopes of resulting in equivalent weights
- Above test is passed for both gaussian and polynomial kernel functions
- Basis set of kernel functions will consist of 3 and 5 Gaussian kernels
- Determine the optimal weight vector for linear combination of candidate kernels

Multiple Kernel SVM Performance

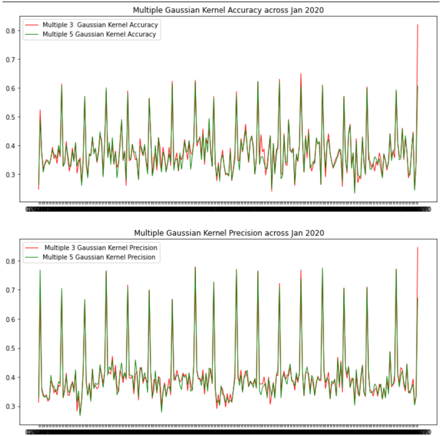


Figure: Accuracy and Precision of Multiple Gaussian Kernel SVM

Single vs. Multiple Kernel Performance

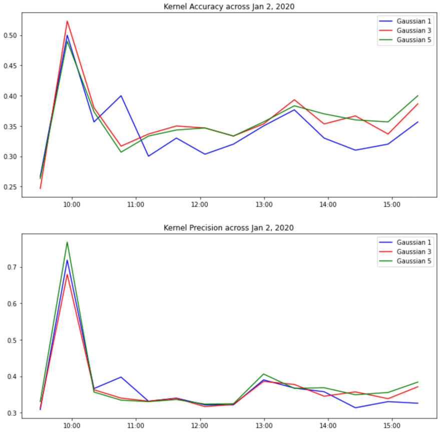


Figure: Accuracy and Precision of Single and Multiple Gaussian Kernel SVM

SVM Accuracy and Precision

- Single and Multiple Gaussian Kernel SVM perform similarly in accuracy and precision
- Significantly outperform naive linear kernel highlighting presence of nonlinearities
- Metrics tend to peak after market open with decay in performance afterwards
- Interpretation for this is stability in trend shortly after the market settles post-open

Kernel	Accuracy
Linear	0.33
Single Gaussian	0.36
Multiple Gaussian (3)	0.38
Multiple Gaussian (5)	0.38

Table: Sample Mean of SVM Accuracy

Kernel	Precision
Linear	0.31
Single Gaussian	0.40
Multiple Gaussian (3)	0.40
Multiple Gaussian (5)	0.40

Table: Sample Mean of SVM Precision

Key Takeaways

- Feature engineering is critical to determine significant predictors in price direction.
- Gaussian kernels outperform linear kernels imply nonlinearities in trend detection.
- Multiple Kernel and Single Kernel SVMs perform similarly for chosen metrics.
- Reoccurring peaks in accuracy and precision observed post market-open.
- Interpretation is the existence of short-tem trend after initial market volatility settles.

Code Repository

To access the data and scripts to reproduce examples, refer to the,
https://github.com/jasonbohne123/Kernel_Learning

References

Tristan Fletcher, Zakria Hussain, Josh Shawe-Taylor
Multiple Kernel Learning on the Limit Order Book
Journal of Machine Learning Research - Proceedings Track 2010

Rakotomamonjy, Alain and Bach, Francis and Canu, Stephane and Grandvalet, Yves
SimpleMKL
Journal of Machine Learning Research 2008

Mehmet gonen and Ethem Alpaydin
Multiple Kernel Learning Algorithms
https://jmlr.csail.mit.edu/papers/volume12/gonen11a/gonen11a.pdf

Wikipedia
Support Vector Machine
https://en.wikipedia.org/wiki/Support_vector_machine

David Cournapeau
scikit-learn
https://scikit-learn.org/stable/