MS&E 448 Midterm Presentation
High Frequency Algorithmic Trading

Francis Choi George Preudhomme Nopphon Siranart
Roger Song Daniel Wright

Stanford University

May 9, 2017

Overview

Order Book and Message book

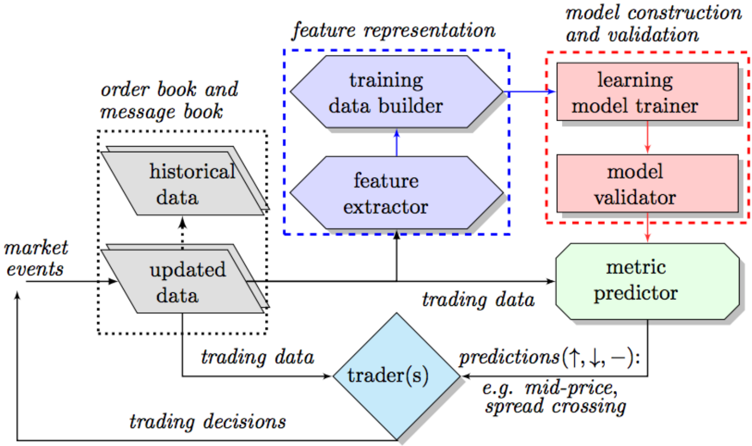
Message book

	Time(sec)	Price($)	Volume	Event Type	Direction
$k-1$	34203.011926972	585.68	18	execution	ask
k	34203.011926973	585.69	16	execution	ask
...	...	...	...	...	...
$k+4$	34203.011988208	585.74	18	cancellation	ask
$k+5$	34203.011990228	585.75	4	cancellation	ask
...	...	...	...	...	...
$k+8$	34203.012050158	585.70	66	execution	bid
$k+9$	34203.012287906	585.45	18	submission	bid
$k+10$	34203.089491920	586.68	18	submission	ask

Order book

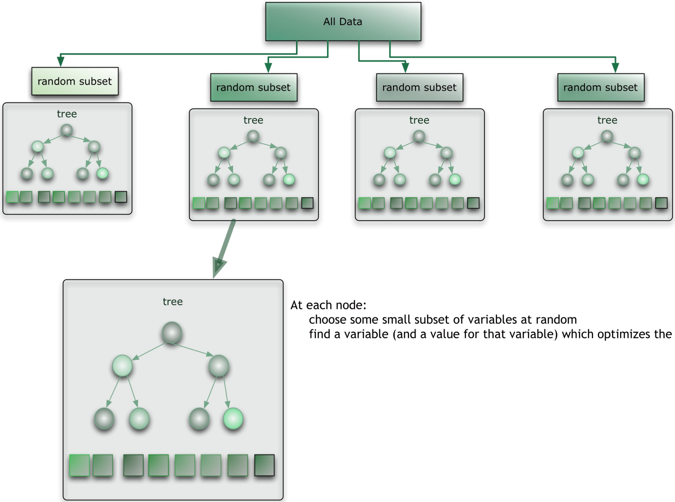
	Ask^1		Bid^1		Ask^2		Bid^2		Ask^3		Bid^3		...
	Price	Vol.	Price	Vol.	Price	Vol.	Price	Vol.	Price	Vol.	Price	Vol.	...
$k-1$	585.69	16	585.44	167	585.71	118	585.40	50	585.72	2	585.38	22	...
k	585.71	118	585.44	167	585.72	2	585.40	50	585.74	18	585.38	22	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...
$k+4$	585.71	118	585.70	66	585.72	2	585.44	167	585.75	4	585.40	50	...
$k+5$	585.71	118	585.70	66	585.72	2	585.44	167	585.80	100	585.40	50	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...
$k+8$	585.71	100	585.44	167	585.80	100	585.40	50	585.81	100	585.38	22	...
$k+9$	585.71	100	585.45	18	585.80	100	585.44	167	585.81	100	585.40	50	...
$k+10$	585.68	18	585.45	18	585.71	100	585.44	167	585.80	100	585.40	50	...

Architecture of Our Model Framework



Random Forest

-



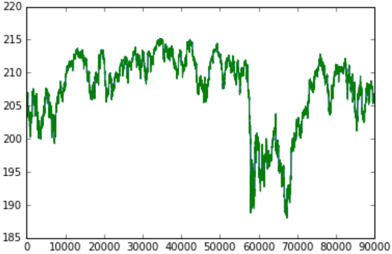
At each node:

choose some small subset of variables at random

find a variable (and a value for that variable) which optimizes the split

Data Summary

- IVV Stock Prices in Year 2015 containing 89,869 datapoints.
- Minute-by-minute (Except beginning/End of trading days)
- We used the first 80% for Training Set and the last 20% for Test Set



Features

Basic Set	Description(i = *level index*)
$v_1 = \{P_i^{ask},\ V_i^{ask},\ P_i^{bid},\ V_i^{bid}\}_{i=1}^n,$	*price and volume(n levels)*

Time-insensitive Set	Description(i = *level index*)
$v_2 = \{(P_i^{ask} - P_i^{bid}), (P_i^{ask} + P_i^{bid})/2\}_{i=1}^n,$	*bid-ask spreads and mid-prices*
$v_3 = \{\max P_i^{ask} - \min P_i^{ask},\ \max P_i^{bid} - \min P_i^{bid}\}_{i=1}^n,$	*max-min price differences*
$v_4 = \{\frac{1}{n}\sum_{i=1}^n P_i^{ask},\ \frac{1}{n}\sum_{i=1}^n P_i^{bid},\ \frac{1}{n}\sum_{i=1}^n V_i^{ask},\ \frac{1}{n}\sum_{i=1}^n V_i^{bid}\},$	*mean prices and volumes*
$v_5 = \{\sum_{i=1}^n (P_i^{ask} - P_i^{bid}),\ \sum_{i=1}^n (V_i^{ask} - V_i^{bid})\},$	*accumulated differences*

Time-sensitive Set	Description(i = *level index*)
$v_6 = \{dP_i^{ask}/dt,\ dP_i^{bid}/dt,\ dV_i^{ask}/dt,\ dV_i^{bid}/dt\}_{i=1}^n,$	*price and volume derivatives*
$v_7 = \{\lambda_{\Delta t}^{la},\ \lambda_{\Delta t}^{lb},\ \lambda_{\Delta t}^{ma},\ \lambda_{\Delta t}^{mb},\ \lambda_{\Delta t}^{ca},\ \lambda_{\Delta t}^{cb}\ \}$	*average intensity of each type*
$v_8 = \{\mathbf{1}_{\{\lambda_{\Delta t}^{la} > \lambda_{\Delta T}^{la}\}},\ \mathbf{1}_{\{\lambda_{\Delta t}^{lb} > \lambda_{\Delta T}^{lb}\}},\ \mathbf{1}_{\{\lambda_{\Delta t}^{ma} > \lambda_{\Delta T}^{ma}\}},\ \mathbf{1}_{\{\lambda_{\Delta t}^{mb} > \lambda_{\Delta T}^{mb}\}}\},$	*relative intensity indicators*
$v_9 = \{d\lambda^{ma}/dt,\ d\lambda^{lb}/dt,\ d\lambda^{mb}/dt,\ d\lambda^{la}/dt\},$	*accelerations(market/limit)*

Label

- Originally, binary classification (mid-price change)
- Changed to three-way classification (spread crossing)

Time	Bid	Ask	Upward	Downward	Label
1	207.29	207.32	-0.07	-0.02	-1
2	207.25	207.27	-0.01	0.02	0
3	207.26	207.27	0.04	0.08	1

Table: Labels using Upward cross and Downward Cross

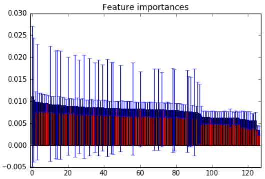
- Our data: 26,105 ups (+1), 37,870 zeros (0), 25,894 downs (-1) - Evenly distributed

Results

Key Observations

- We notice that those features in the lower levels are more important than those higher ones
- Volume and number of orders are the most significant features

Ranking	Features	Score
1	bids size 1	0.010127
2	asks size 2	0.010027
3	asks nord 0	0.010024
4	asks size 1	0.009959
5	bids nord 3	0.009956
6	asks nord 3	0.009895



Results

- Out of 17,973 entries, there are 1,947 ups (+1), 14,715 zeros (0), 1,311 downs (-1) in our predicted labels - biased towards zero
- Below shows the confusion matrix

		Prediction		
		1	0	-1
Truth	1	420	4063	543
	0	503	6579	829
	-1	388	4073	575

Table: Confusion Matrix of the Predictions

- Not good enough !! $\Rightarrow$ Set the threshold

Trading Strategy

- Consider likelihoods that model predicts for each new data point
- If the highest likelihood is -1 or 1 and that likelihood is sufficiently large enough (above our threshold), then trade in that direction
- For example, if the threshold = 0.40, only time 1 and time 2 have the likelihood above the threshold. However, we only open position at time 2 as our predicted label at time 1 is 0

Time	**-1**	**0**	**-1**	**Predicted**	**Threshold**	**Position**
1	0.253	0.479	0.266	0	Yes	No
2	0.301	0.269	0.428	-1	Yes	Sell
3	0.358	0.303	0.337	1	No	No

Table: Likelihood of our random forest model

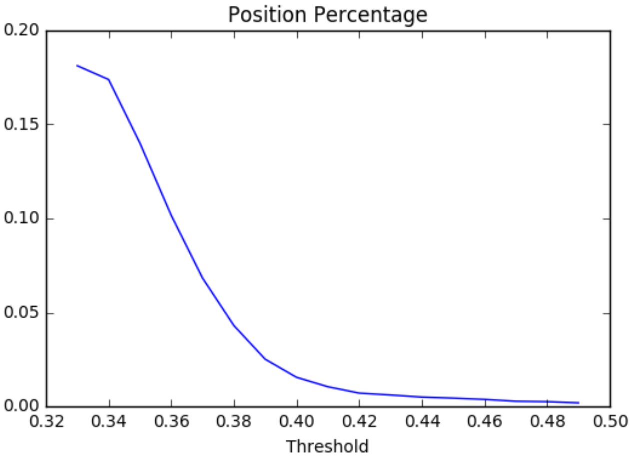
Trading Strategy - Calculating Profit

- Once we open a position, we will close it in the next time step
- For example, total profit of the positions in the following table is $0 + (-0.02) + 0.04 = 0.02$

Time	**Bid**	**Ask**	**Upward**	**Downward**	**Position**	**Profit**
1	207.29	207.32	-0.07	-0.02	0	0
2	207.25	207.27	-0.01	0.02	-1	-0.02
3	207.26	207.27	0.04	0.08	1	0.04

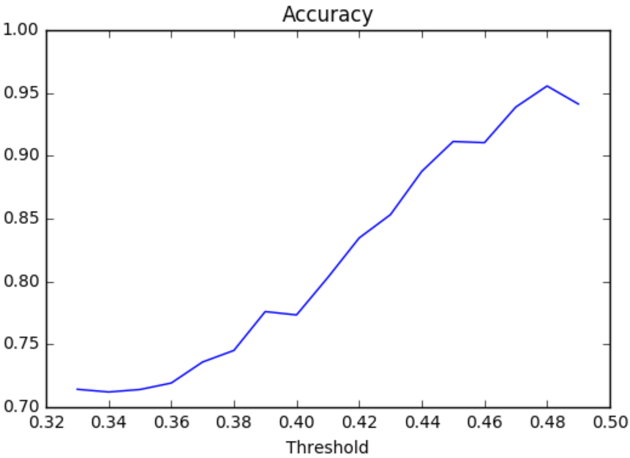
Results

- As the threshold increases, the number of total positions decreases



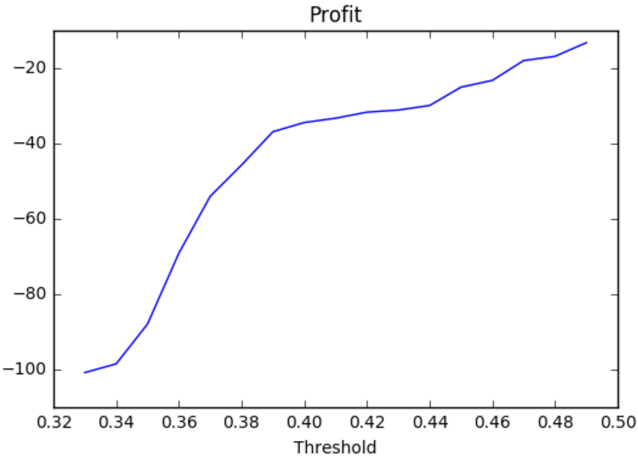
Results

- Accuracy is measured as follows. If we our position is 1, and the true label is either 1 or 0, then we say it is accurate and vice versa for -1.
- As the threshold increases, the accuracy also increases as well



Results

- As the threshold increases, the profit also increases as well



Next Steps

- Tuning Hyperparameters e.g. Max Depth, Number of Trees in Random Forest, Threshold etc.
- Try different prediction models such as SVM, Regression, Time Series
- Running the strategy using the simulator