Ensemble methods for measuring dynamics of limit order books

Jian Wang

wangjian790@gmail.com

Financial math Ph.D Candidate
Florida State University

Fall 2016 advanced financial seminar

Table of contents

Contents

- Our main goal is to use ensemble machine learning methods to predict the limit order book price cross over opportunity.
- Use the high frequency data to predict relatively long time future price changing trend(eg. 5 seconds later).
- Features selection: choose what kind of data as our independent variables(choose x_i s).
- Compare the f1 score and calculation time among different machine learning methods, and show that ensemble methods can improve the predicting performance significantly.

Contents

High frequency trading

High frequency trading is a specialized case of algorithmic trading involving the frequent turnover of many small positions of a security.

Positive impact

- Increased liquidity
- Narrowing spreads
- Improve market efficiency
- Increase fees for Exchanges

Negative impact

- Impact on the institutional investors.
- Increase volatility (2010 flash crash)
- Disadvantages to the small Investors(asymmetric information)

HFT Strategies:

Market Making

Place bets on both sides of the trade by placing a limit order to sell slightly above the current market price, or to buy slightly below the current market price, thereby profiting from the difference between the two.

Statistical Arbitrage

Firms and traders looking to make profits from market arbitrage essentially exploit the momentary inconsistencies in factors such as rates, prices, and other conditions between different exchanges or asset classes

Liquidity Rebate Trading

look for large orders, fill a part of that order, and then offer these shares back to the market by placing a limit order, which makes them eligible to collect the rebate fee for providing liquidity,with or without them making a capital gain.

Momentum Ignition

Ignition strategies involve initiating and canceling a number of trades and orders with a certain security in a particular direction, which may ignite a rapid market price movement.

Contents

Dataset

Limit order book data

The dataset contains limit order book prices of specific stocks from NASDAQ. For each stock, it divided into two major components: the message book and the order book.

- Message book: Contains Time, Prices, Volume, Event Type, Direction
- Order book: Contains price levels, price and volume in each level for every event.
- Sample sizes: AAPL(400391),AMZN(269748),GOOG(147916),INTC(624040), MSFT(668765)
- Date: 2012-06-21

Message Book

AAPL as example:

Time(sec)	Type	Order ID	Volume	Price($)	Direction
34200.004241176	1	16113575	18	5853300	1
34200.00426064	1	16113584	18	5853200	1
34200.004447484	1	16113594	18	5853100	1
34200.025551909	1	16120456	18	5859100	-1
34200.025579546	1	16120480	18	5859200	-1
34200.025613151	1	16120503	18	5859300	-1
34200.050241056	1	16127688	100	5850000	1
34200.201517942	1	16166035	100	5859300	-1
34200.201735987	3	16113594	18	5853100	1
34200.201742395	3	16113584	18	5853200	1
34200.201743336	3	16120456	18	5859100	-1
34200.201768069	3	16120503	18	5859300	-1
34200.201780978	3	16120480	18	5859200	-1
34200.20196619	1	16166175	2	5849900	1

Time is in sec and minimum time change is nanosecond, Price is in dollars and each tick is one cent, 5 Event type, such as execution, cancellation and so on, 2 Direction ask and bid.

Order book types:

Type	Description
1	Submission of a new limit order
2	Cancellation (Partial deletion)
3	Deletion (Total deletion of a limit order)
4	Execution of a visible limit order
5	Execution of a hidden limit order

Order book directions:

Direction	Description
-1	Sell limit order
1	Buy limit order

Order Book:

Ask_level 1		Bid_level 1		Ask_level 2		Bid_level 2		Ask_level 3		Bid_level 3	
Price	Vol	Price	Vol	Price	Vol	Price	Vol	Price	Vol	Price	Vol
5859400	200	5853300	18	5859800	200	5853000	150	5861000	200	5851000	5
5859400	200	5853300	18	5859800	200	5853200	18	5861000	200	5853000	150
5859400	200	5853300	18	5859800	200	5853200	18	5861000	200	5853100	18
5859100	18	5853300	18	5859400	200	5853200	18	5859800	200	5853100	18
5859100	18	5853300	18	5859200	18	5853200	18	5859400	200	5853100	18
5859100	18	5853300	18	5859200	18	5853200	18	5859300	18	5853100	18
5859100	18	5853300	18	5859200	18	5853200	18	5859300	18	5853100	18
5859100	18	5853300	18	5859200	18	5853200	18	5859300	118	5853100	18
5859100	18	5853300	18	5859200	18	5853200	18	5859300	118	5853000	150
5859100	18	5853300	18	5859200	18	5853000	150	5859300	118	5851000	5

From level 1 to level 10, where the first level is the best bid and ask.

Contents

Methodology

Logistic regression

$$ln\frac{F(x)}{1-F(x)} = \beta_0 + \sum_i \beta_i x_i$$

Ridge regression

$$\hat{\beta}^{ridge} = argmin_\beta\{\sum_{i=1}^{p}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\}$$

Lasso regression

$$\hat{\beta}^{lasso} = argmin_\beta\{\sum_{i=1}^{p}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\}$$

Methodology

Comparison of L1 and L2 Penalized Model

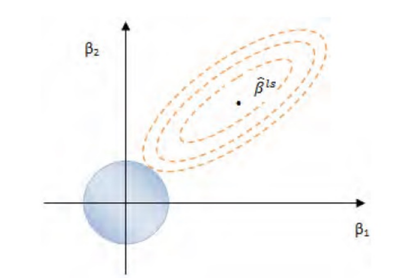
Ridge regression

$$\hat{\beta}^{ridge} = argmin_{\beta}\{\sum_{i=1}^{p}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\}$$

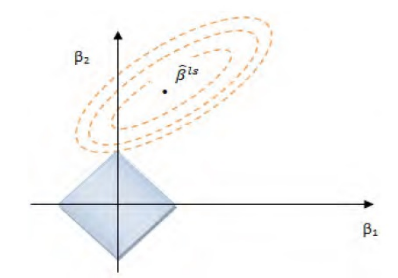
Lasso regression

$$\hat{\beta}^{lasso} = argmin_{\beta}\{\sum_{i=1}^{p}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|\}$$

Coefficients:



Coefficients:



Methodology

Comparison of L1 and L2 Penalized Model

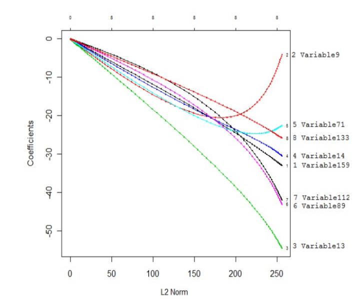
Ridge regression

$$\hat{\beta}^{ridge} = argmin_{\beta}\{\sum_{i=1}^{p}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\}$$

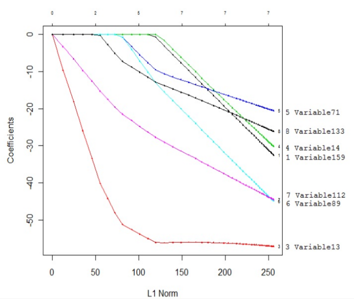
Lasso regression

$$\hat{\beta}^{lasso} = argmin_{\beta}\{\sum_{i=1}^{p}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|\}$$

Path::



Path::



Methodology

Support vector machine

- Introduced in COLT-92 by Boser, Guyon & Vapnik. Became rather popular since.
- Theoretically well motivated algorithm: developed from Statistical Learning Theory (Vapnik & Chervonenkis) since the 60s.
- Empirically good performance: successful applications in many fields (bioinformatics, text, image recognition, . . .)

Try to maximize the margin:

$r = 1/||w||, y_j = 1, -1$

primal form:

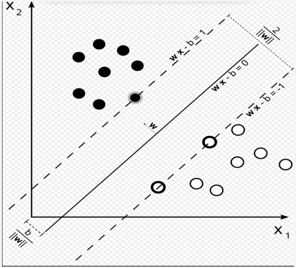
$$\max_{W,b} \ r = 1/||W||$$

$$s.t.(W^T x_j + b)y_j >= 1$$

Dual form:

$$\max_{\alpha_1,\ldots,\alpha_M} \sum \alpha_l - \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{M} \alpha_j \alpha_k y_j y_k < X_j, X_k >$$

$$\text{s.t.} \alpha_l \geq 0, \sum_{l=1}^{M} \alpha_l y_l = 0$$



Methodology

Kernel functions

We can use the kernel function to calculate the inner product in high dimensional cases in its original feature spaces.

Example:two dimension polynomial

$$k(x, z) = (x^T z)^2$$
$$= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T (z_1^2, \sqrt{2}z_1 z_2, z_2^2)$$
$$= \Phi(x)^T \Phi(z)$$

Kernel functions that we used

- Linear kernel: $k(x, y) = x^T y + c$
- Polynomial Kernel: $k(x, y) = (\alpha x^T y + c)^d$
- Radial basis function kernel(RBF): $k(x, y) = exp(-\gamma||x - y||^2)$

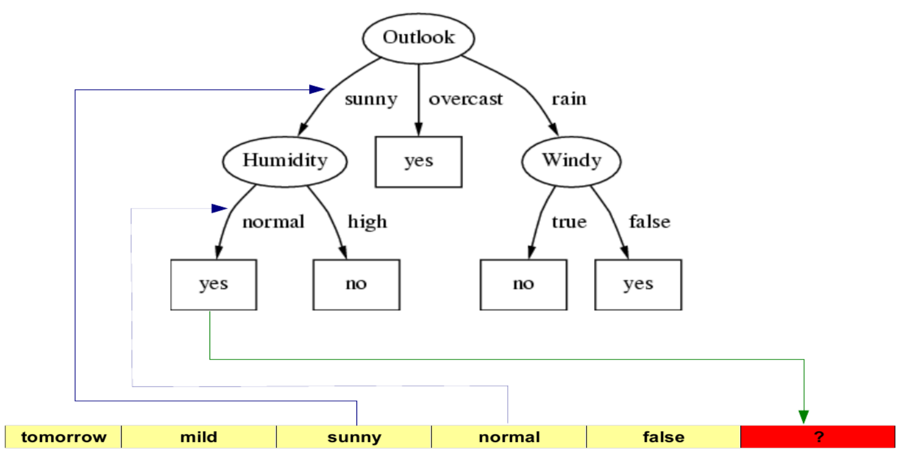
Methodology

Decision tree Example: Know the history of playing golf or not, given new data, make prediction

Day	Temperature	Outlook	Humidity	Windy	Play Golf?
07-05	hot	sunny	high	false	no
07-06	hot	sunny	high	true	no
07-07	hot	overcast	high	false	yes
07-09	cool	rain	normal	false	yes
07-10	cool	overcast	normal	true	yes
07-12	mild	sunny	high	false	no
07-14	cool	sunny	normal	false	yes
07-15	mild	rain	normal	false	yes
07-20	mild	sunny	normal	true	yes
07-21	mild	overcast	high	true	yes
07-22	hot	overcast	normal	false	yes
07-23	mild	rain	high	true	no
07-26	cool	rain	normal	true	no
07-30	mild	rain	high	false	yes

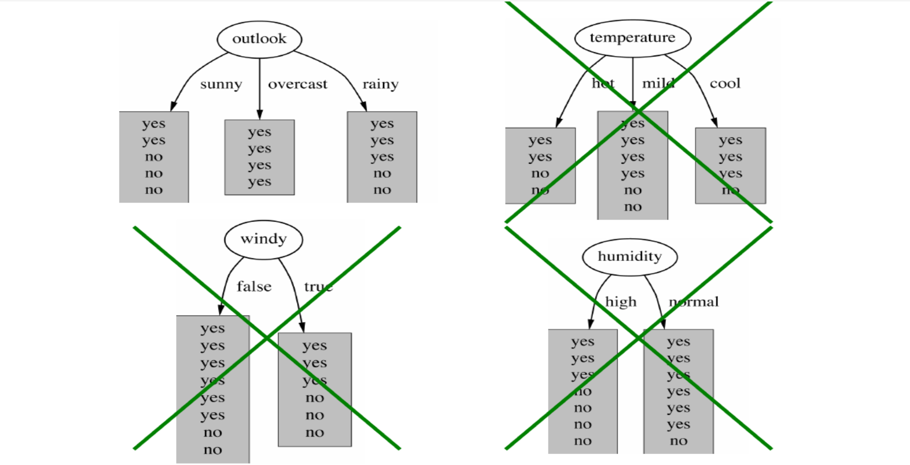
today	cool	sunny	normal	false	?
tomorrow	mild	sunny	normal	false	?

Methodology



Methodology

Which attribute to select as the root?



Methodology

Entropy:

Entropy is a measure for un-orderedness

$$E(s) = -\sum_{i=1}^{n} p_i log p_i$$

Outlook = sunny: 3 examples yes, 2 examples no

$$E(outlook = sunny) = -\frac{2}{5} log\frac{2}{5} - \frac{3}{5} log\frac{3}{5} = 0.971$$

Outlook = sunny: 4 examples yes, 0 examples no

$$E(outlook = overcast) = -1 log 1 - 0 log 0 = 0$$

Outlook = rainy: 2 examples yes, 3 examples no:

$$E(outlook = sunny) = -\frac{3}{5} log\frac{3}{5} - \frac{2}{5} log\frac{2}{5} = 0.971$$

Methodology

Information Gain for attribute A:

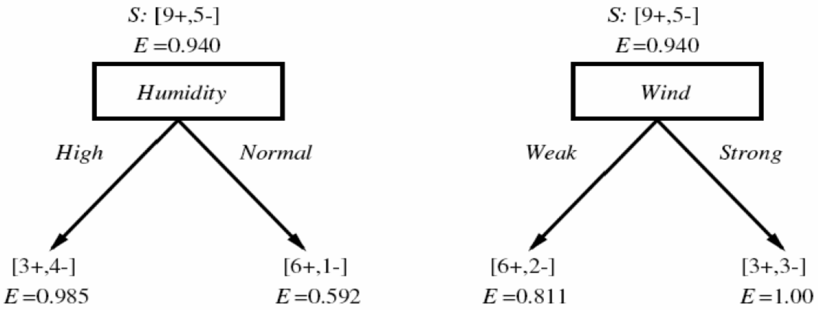
When an attribute A splits the set S into subsets S_i

- we compute the average entropy
- and compare the sum to the entropy of the original set S

$$Gain(S, A) = E(S) - I(S, A) = E(S) - \sum_i \frac{|S_i|}{|S|} E(S_i)$$

The attribute that maximizes the difference is selected

Methodology



Gain (S, Humidity)

$$= .940 - (7/14).985 - (7/14).592$$
$$= .151$$

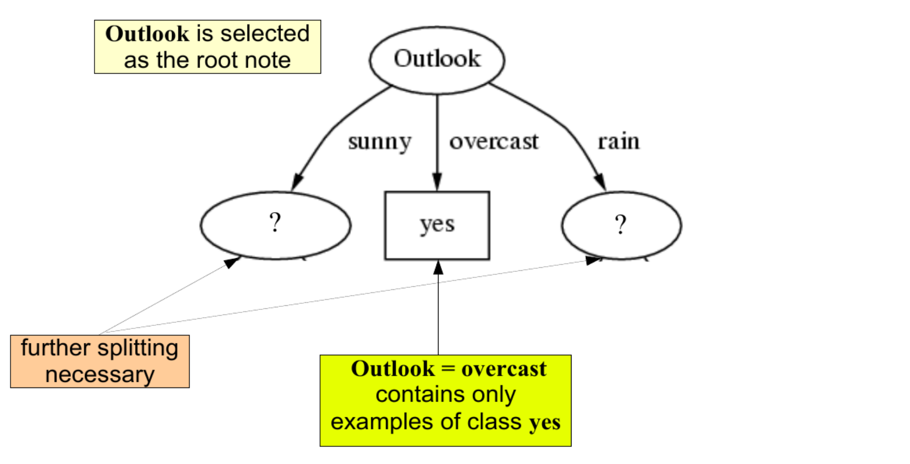
Gain (S, Wind)

$$= .940 - (8/14).811 - (6/14)1.0$$
$$= .048$$

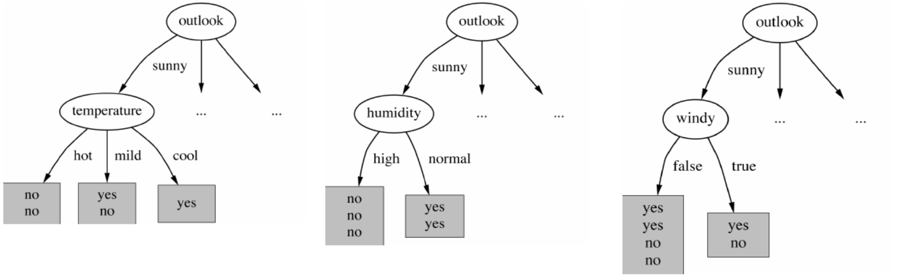
$Gain(S, Outlook) = 0.246$

$Gain(S, Temperature) = 0.029$

Methodology



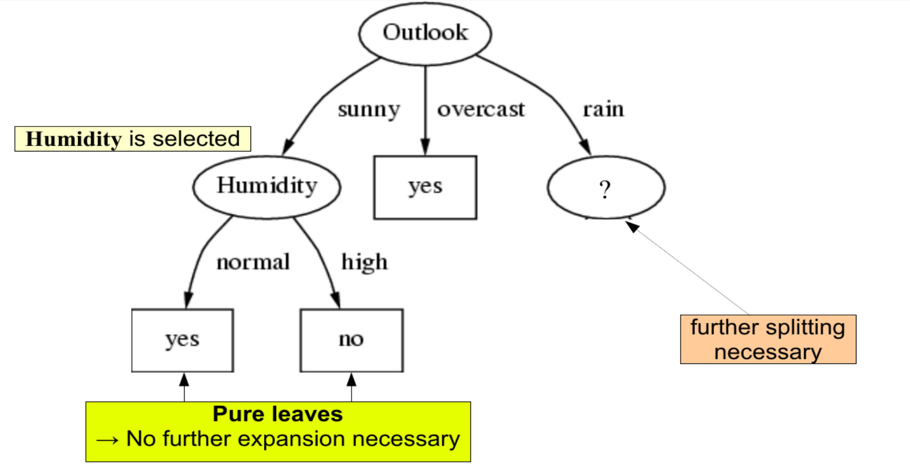
Methodology



$$\left.\begin{aligned}\text{Gain}(\textit{Temperature}) &= 0.571 \text{ bits}\\ \text{Gain}(\textit{Humidity}) &= 0.971 \text{ bits}\\ \text{Gain}(\textit{Windy}) &= 0.020 \text{ bits}\end{aligned}\right\}$$

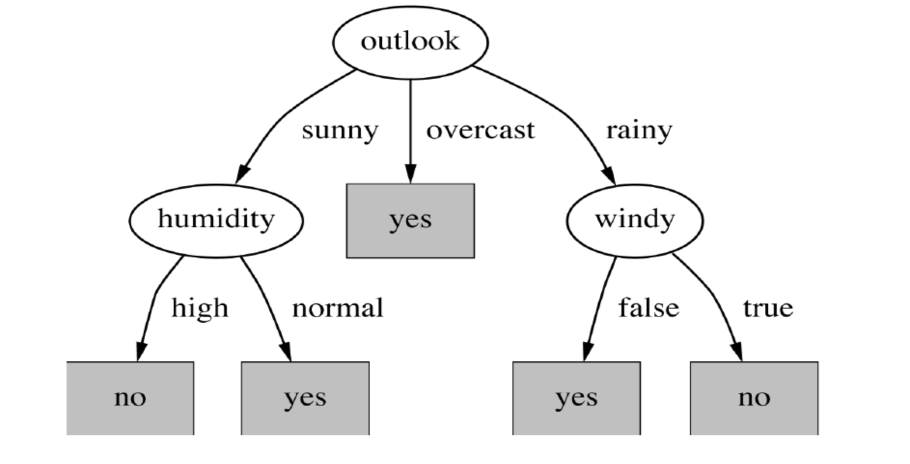
Humidity is selected

Methodology



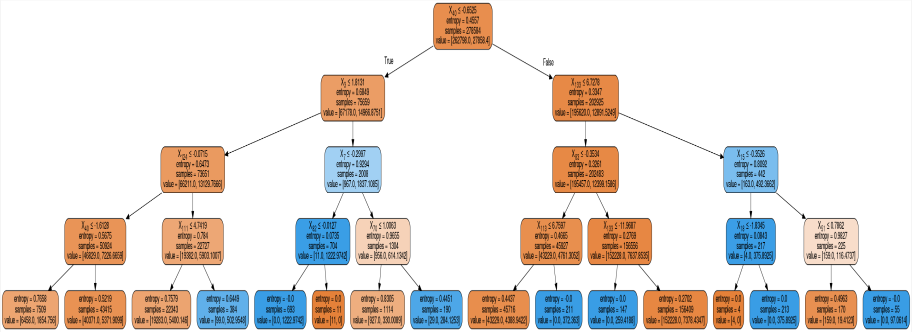
Methodology

Final structure:



Methodology

In our case: limit order book prediction model, choose depth as 6 and criterion as entropy, decision tree example as follows:



We will see the accuracy rate of decision tree in model fit part.

Methodology

Ensembling methods: the most important part

IDEA:

- Do not learn a single class but learn a set of classifiers
- Combine the predictions of multiple classifiers

Motivation:

- Reduce variance: results are less dependent on peculiarities of a single training set
- Reduce bias: a combination of multiple classifiers may learn a more expressive concept class than a single classifier

KEY STEP:

- Formation of an ensemble of diverse classifiers from a single training set

Methodology

Why do ensembles work?

Suppose there are 25 base classifiers:

- Each classifier has error rate, $\epsilon = 0.35$
- Assume classifiers are independent

Probability that the ensemble classifier makes a wrong prediction:

- The ensemble makes a wrong prediction if the majority of the classifiers makes a wrong prediction
- The probability that 13 or more classifiers err is:

$$\sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1-\epsilon)^{25-i} \approx 0.06 \ll \epsilon$$

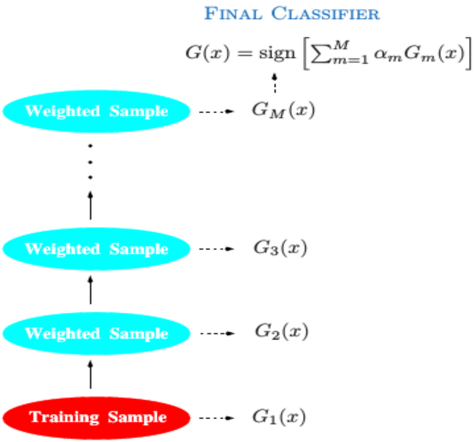
Methodology

First ensemble method: AdaBoost method

- Introduced in 1990s
- Originally designed for classification problems
- Later extended to regression
- Motivation - a procedure that combines the outputs of many "weak" classifiers to produce a powerful "committee"
- Put more weight on mis-classification data each time

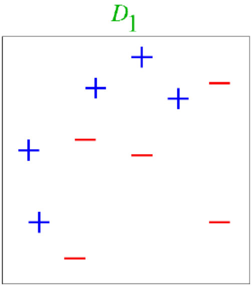
Methodology

Boosting methods



Methodology

AdaBoost example: TOY example:

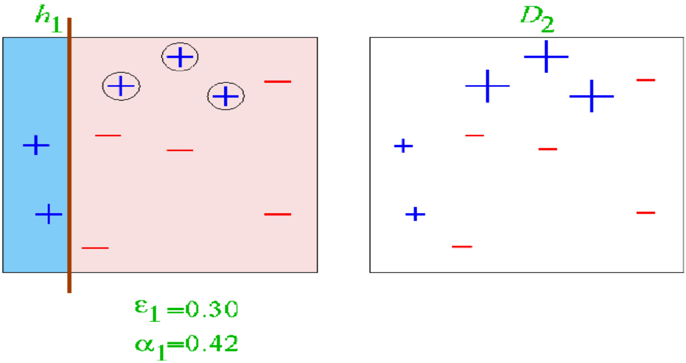


(taken from Verma & Thrun, Slides to CALD Course CMU 15-781, Machine Learning, Fall 2000)

Methodology

Round 1:

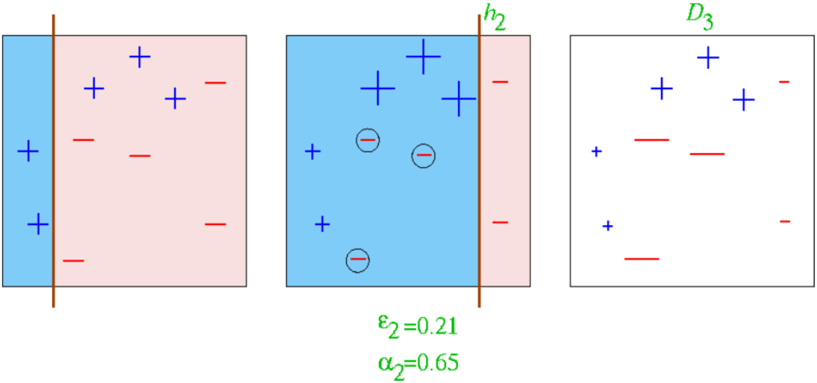
AdaBoost example: TOY example:



Methodology

Round 2:

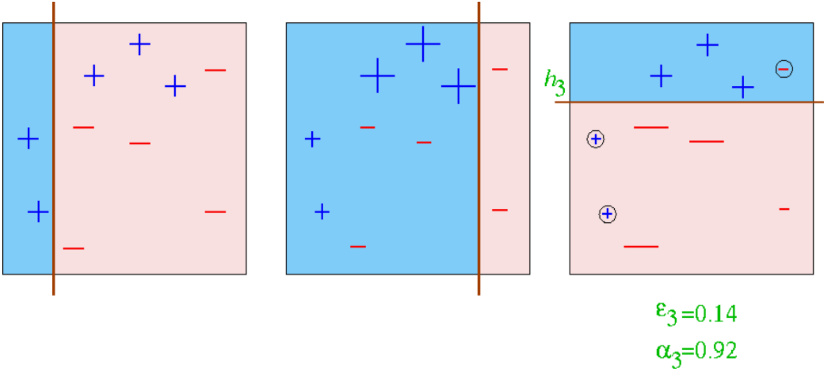
AdaBoost example: TOY example:



Methodology

Round 3:

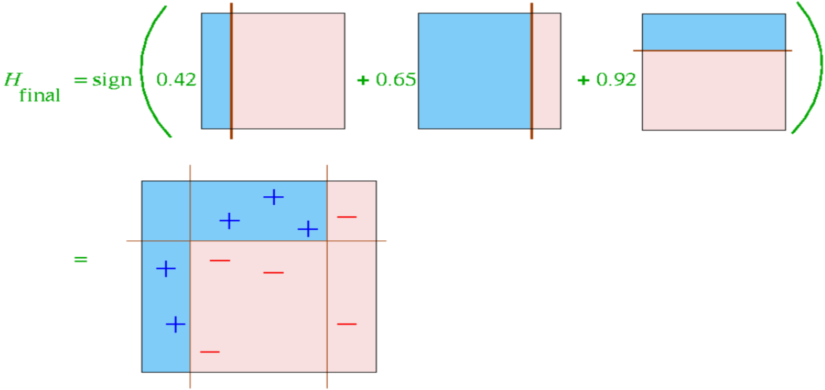
AdaBoost example: TOY example:



Methodology

Final round:

AdaBoost example: TOY example:



Methodology

Second ensemble method: Random forest

Dataset: N samples, each having M attributes (features)
A value m<M is chosen, $m \approx \sqrt{M}$ or $m \approx logM$
Growing one tree:

- Select N samples randomly with replacement (bootstrap)
- At each node, m attributes are selected randomly from the M
- The best binary split from the m attributes (based on information gain) is chosen
- The tree is fully grown, no pruning

Loop the above process several times. Given an observation:

- Each decision tree votes for a class
- The class with most votes is the final result

Adaboosting algorithm:

1 Initialize the observation weights ω_i=$1/N$,i=1,2,...,N;
2 **for** *m=1 to M* **do**

Fit a classifer $G_m(x)$ to the training data using weights ω_i;
Compute

$$err_m = \frac{\sum_{i=1}^{N} \omega_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} \omega_i}$$

Compute $\alpha_m = log((1 - err_m)/err_m)$;
Set $\omega_i \leftarrow \omega_i \cdot exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, ..., N$;

3 Output $G(x) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$

source:ESL

- Put more weights on the false classification data
- Average each classifer based on error to get the strong classifer
- Maybe the strongest classifer among the out of box classifers

Random forest algorithm:

1 **for** *b=1 to B* **do**

(a) Draw a bootstrap sample Z^* of size N from the training data.
(b) Grow a random-forest tree T_b to the bootstrapped data, by re- cursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
i. Select m variables at random from the p variables.
ii. Pick the best variable/split-point among the m.
iii. Split the node into two daughter nodes.

2 Output the ensemble of trees $\{T_b\}_1^B$
To make a prediction at a new point x:
3 Let $\hat{C}_b(x)$ be the class prediction of the bth random forest tree. Then $\hat{C}_{rf}^B(x)$= majority vote $\{\hat{C}_b(x)\}_1^B$

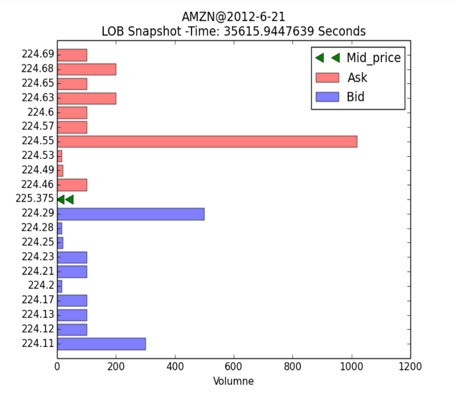
source:ESL

- Combine feature selection and bootstrap methods
- Correct for decision trees' habit of overfitting to their training set

Contents

Model fit

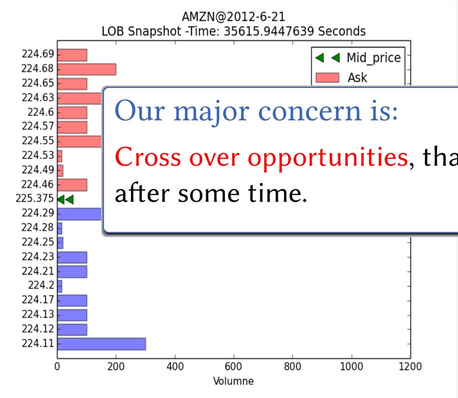
order book snapshot:



- At Time t: $P_t^A > P_t^B$, no arbitrage
- At Time t+ Δt, there are three situations:
  •$P_{t+\Delta t}^A < P_t^B$: ask lower,denote as 1 in our model
  •$P_{t+\Delta t}^B > P_t^A$: bid higher, denote as -1 in our model
  •otherwise(implies thatno direction change)

Model fit

order book snapshot:



- At Time t: $P_t^A > P_t^B$, no arbitrage
- At Time t+ Δt, there are three

enote as

denote

as -1 in our model

•otherwise(implies thatno direction change)

Our major concern is:

Cross over opportunities, that is bid higher or ask lower after some time.

Model fit

Ask low example(5 seconds future):

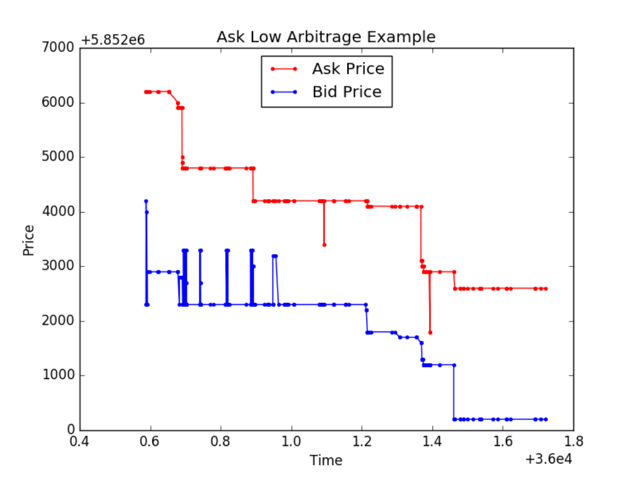


Figure : Ask low arbitrage example

Bid high example(5 seconds future):

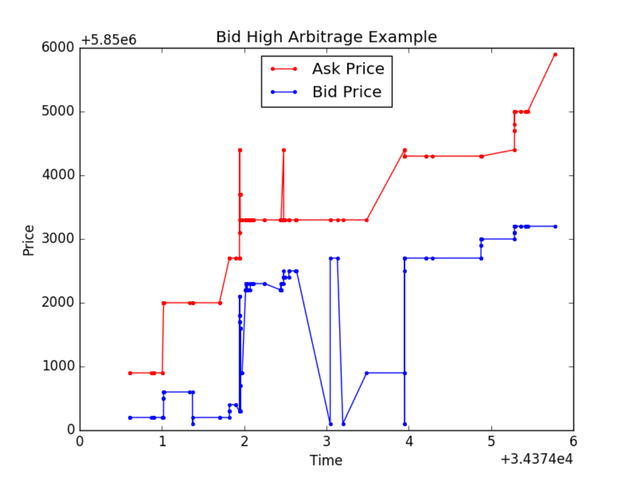


Figure : Bid high arbitrage example

Model fit

No arbitrage example(5 seconds future):

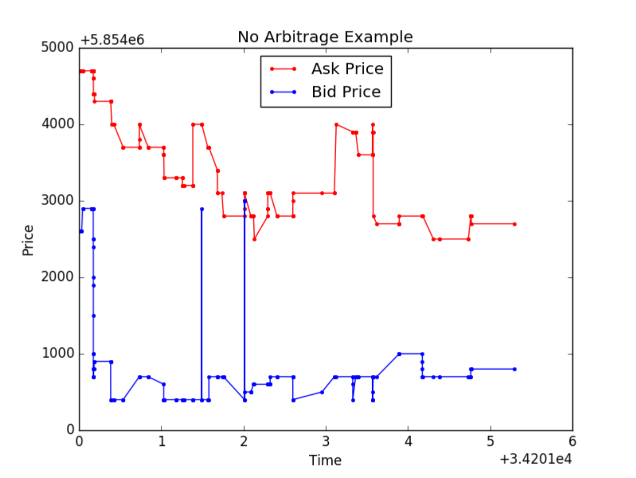
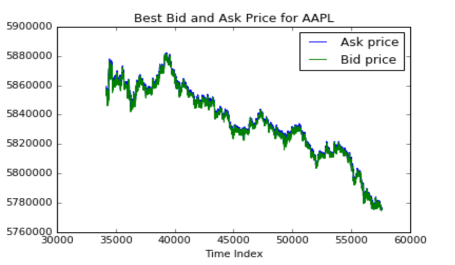


Figure : No arbitrage example

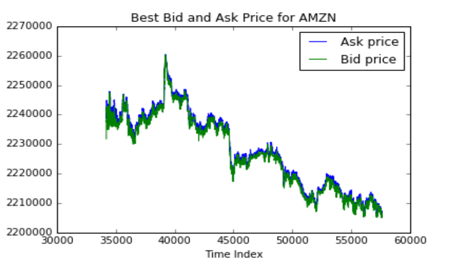
Stock Price

AAPL:

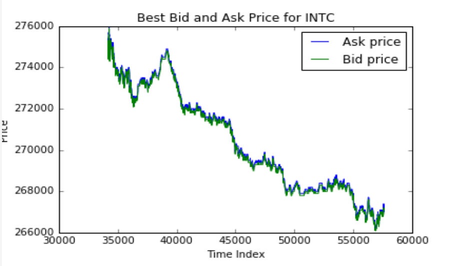


GOOG:

AMZN:



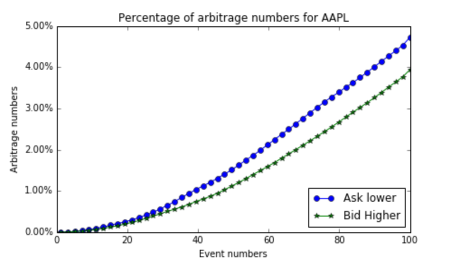
INTC:



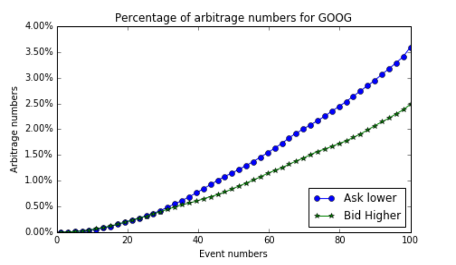
Model fit

Arbitrage opportunities based on future event

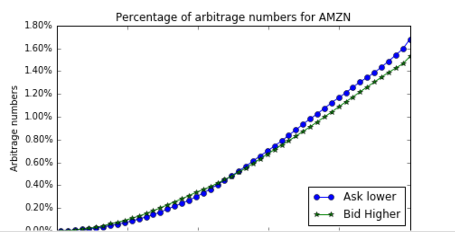
AAPL:



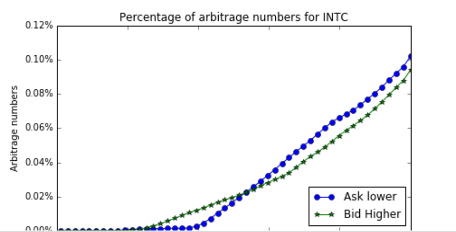
GOOG:



AMZN:



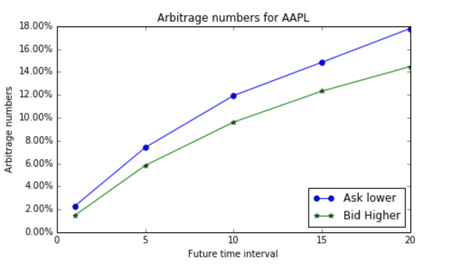
INTC:



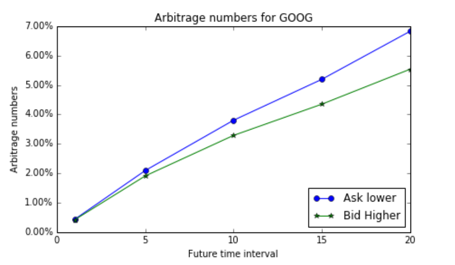
Model fit

Arbitrage opportunities based on future time

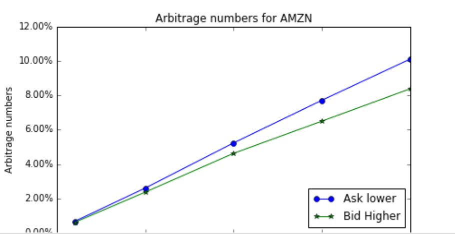
AAPL:



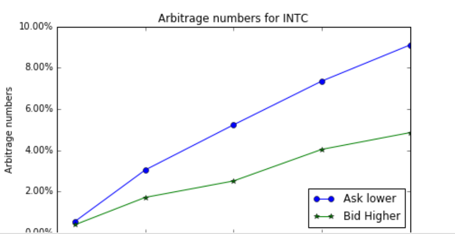
GOOG:



AMZN:

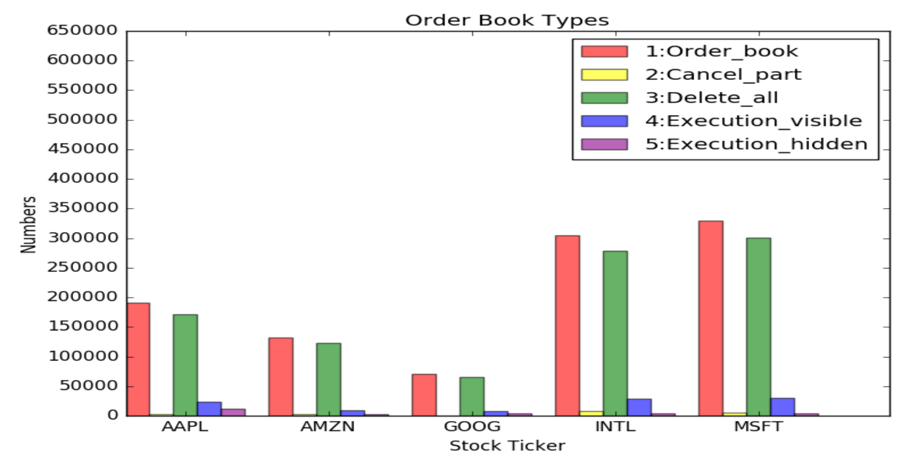


INTC:



Model fit

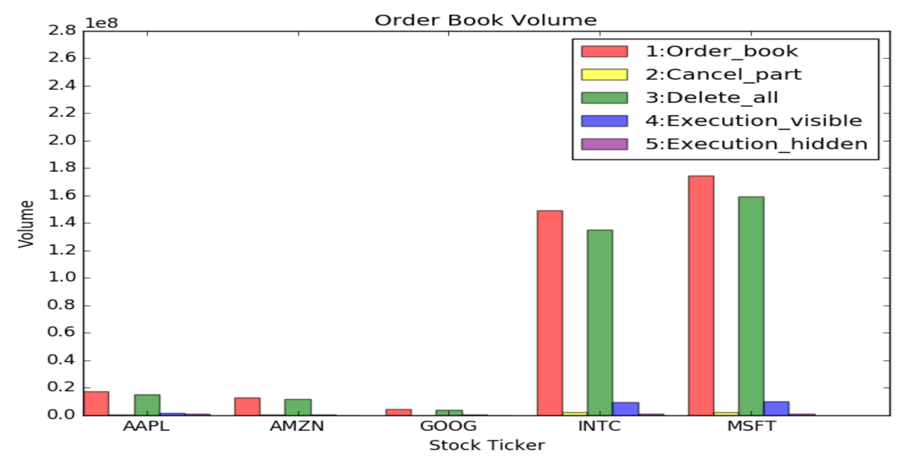
Order book type



Red and green occupy the most, strategy?

Model fit

Order book volume



Still red and green occupy the most

Model fit

Build features:

We use same features that presented by Dr.Kercheval and Yuan Zhang(2015)

Basic Set	Description(i = *level index*, $n = 10$)
$v_1 = \{P_i^{ask}, V_i^{ask}, P_i^{bid}, V_i^{bid}\}_{i=1}^n$,	*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*
$v_3 = \{P_n^{ask} - P_1^{ask}, P_1^{bid} - P_n^{bid}, \lvert P_{i+1}^{ask} - P_i^{ask}\rvert, \lvert P_{i+1}^{bid} - P_i^{bid}\rvert\}_{i=1}^n$,	*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*
$v_8 = \{1_{\{\lambda_{\Delta t}^{la} > \lambda_{\Delta T}^{la}\}}, 1_{\{\lambda_{\Delta t}^{lb} > \lambda_{\Delta T}^{lb}\}}, 1_{\{\lambda_{\Delta t}^{ma} > \lambda_{\Delta T}^{ma}\}}, 1_{\{\lambda_{\Delta t}^{mb} > \lambda_{\Delta T}^{mb}\}}\}$,	*relative intensity indicators*
$v_9 = \{d\lambda^{ma}/dt, d\lambda^{lb}/dt, d\lambda^{mb}/dt, d\lambda^{la}/dt\}$,	*accelerations(market/limit)*

- contain price,volume, bid ask spread, price difference and volume difference for each level, mean of price and volume.
- total 138 variables, can be treated as high dimensional problems.

Model fit

Criteria: Only consider accuaracy? Imbalanced data?

Precision

Precision is the probability that a (randomly selected) retrieved document is relevant.

$$Precision = \frac{True\_positive}{True\_positive + False\_positive}$$

Recall

Recall is the probability that a (randomly selected) relevant document is retrieved in a search.

$$Recall = \frac{True\_positive}{True\_positive + False\_negative}$$

F1 score

A measure that combines precision and recall is the harmonic mean of precision and recall.

$$F_\beta = (1 + \beta^2)\frac{precision \cdot recall}{\beta^2 precision + recall}$$

Numerical results:

AAPL ask low predict(5 seconds):

Table : AAPL Accuracy rate and CPU time

Model	Training time(s)	Training accuracy	Test accuracy	Test f1 score	Test time(s)
Logistic(Lasso penalty)	73	94.50%	94.62%	11.29%	
Logistic(Ridge penalty)	67	94.55%	94.61%	9.65%	
SVM(Poly 2 kernal,10000 estimator)	566	54.86%	54.82%	9.05%	
Decision Tree(10 depth)	21	95.53%	95.54%	35.6%	0.43
Ada boosting(100 estimator)	3841	99.99%	99.76%	97.91%	
Random forest(100 estimator)	214	97.23%	97.15%	66.67%	0.43

remark: training samples 278584 and test samples 30954. The estimation number for AdaBoost and random forest is 100.Computer is 8G memory and Intel Xeon E3 processor(4 cores)

Numerical results:

AAPL Multi-class predict(5 seconds):

Models	One against one		One against rest	
	Trian accuracy	Test accuracy	Train accuracy	Test accuracy
Random Forest	95.86%	95.55%	95.89%	95.68%

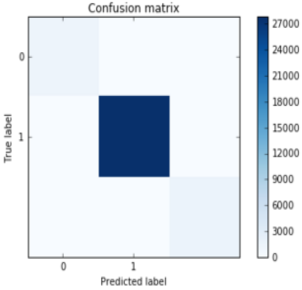
note:one against rest method need C models and one against one method need $\frac{C(C-1)}{2}$ models, here C is the number of classes.

Methodology

Classification matrix for multi-class classification results:

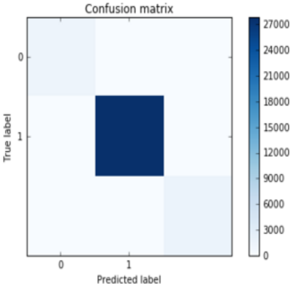
One against One

$$\begin{bmatrix} 851 & 575 & 0 \\ 2 & 27802 & 3 \\ 0 & 798 & 923 \end{bmatrix}$$



One against Rest

$$\begin{bmatrix} 833 & 593 & 0 \\ 3 & 27802 & 2 \\ 0 & 738 & 983 \end{bmatrix}$$



Contents

PnL

PnL is the profit gain and loss through the transaction, formula of PnL can be written as follows:

$$PnL = \begin{cases} y - c & y >= \alpha, \textit{buy action} \\ -y - c & y <= -\alpha, \textit{sell short action} \\ 0 & \textit{otherwise} \end{cases}$$

where y is the net capital gain from the transaction, α is the significant level and c is the trading cost.

Trading strategy

Naive trading strategy:

```
initialize: PnL=0
for i =1 to length(test_set) do
    input test_set[i] into model and get Predict[i]
    if Predict[i]==1(Ask low) then
        Sell short at bid price
        Clear the short option Δt seconds later
        PnL+=Bid_price_t − Ask_price_{t+Δt}
    else if Predicted[i]==-1(Bid high) then
        Buy at ask price
        Sell at bid price Δt seconds later
        PnL+=Bid_price_{t+Δt} − Ask_price_t
    else
        Take no action
    return PnL
```

Contents

Future work

- Compare the results in spark machine learning package.Can deal with big data problem
- Add more meaningful features and calculate the interaction.
- Conduct cross validation or bagging methods.
- Test the Profit and Loss(PNL) which traders mainly concern.
- Neural network and deep learning. AlphaGo Google deepmind?

Reference

Alec N.Kercheval,Yuan Zhang
Modeling high-frequency limit order book dynamics with support vector machines
In *Quantitative finance 2014*

Rosu,I.,
A dynamic model of the limit order book.
In *Rev.Financ.Stud.,2009,22,4601-4641.*

Trevor Hastie, Robert Tibshirani, Jerome Friedman
The Elements of Statistical Learning: Data Mining, Inference, and Prediction,Second Edition

Contents

QA

Thanks a lot and Questions