Ensemble methods for measuring dynamics of limit order books

Jian Wang

wangjian790@gmail.com

Financial math Ph.D Candidate
Florida State University

©©©©

7th high frequency data conference

Table of contents

1 Brief summary

2 High frequency trading

3 DataSet

4 Methodology

5 Model fit

6 Trading strategy and PnL results

7 Future work

8 Questions

Contents

1. Brief summary
2. High frequency trading
3. DataSet
4. Methodology
5. Model fit
6. Trading strategy and PnL results
7. Future work
8. Questions

Brief summary

- Our main goal is to use ensemble machine learning methods to predict the limit order book spread crossing opportunities.
- Use high frequency data to predict relatively long time future price changing trend(eg. 5 seconds later) to prevent illegal actions.
- 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.
- Design a simple trading strategy and demonstrate out of sample Profit and Loss(PnL)

Contents

1 Brief summary

2 High frequency trading

3 DataSet

4 Methodology

5 Model fit

6 Trading strategy and PnL results

7 Future work

8 Questions

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:

Passive: use limit order book

Market Making

Allow the market maker to purchase a company's securities and,at the same time, the market maker is also acted as an underwriter of the securities in a secondary public offering.

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

Aggressive: use market book

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.

Order anticipate

Detection trading which confirms the existence of large institutional buys or sellers in the marketplace and then trade ahead of these buyers or sellers in anticipation that their large orders will move market prices

Market Manipulaiton(illegal):

According to Dodd-Frank Wall Street Reform and Consumer Protection Act of 2010 (“Dodd-Frank Act”)

Spoofing

Bidding or offering with the intent to cancel the bid or offer before execution. The line between spoofing and momentum ignition is ambiguous

Front running

Trading securities in personal account based on the knowledge of advance knowledge of pending orders from its customers.The line between front running and order anticipate is ambiguous

May be more safe to use passive trading strategies in HFT in the future. We pay more attention to statistical arbitrage methods.

Contents

1. Brief summary
2. High frequency trading
3. DataSet
4. Methodology
5. Model fit
6. Trading strategy and PnL results
7. Future work
8. Questions

Dataset

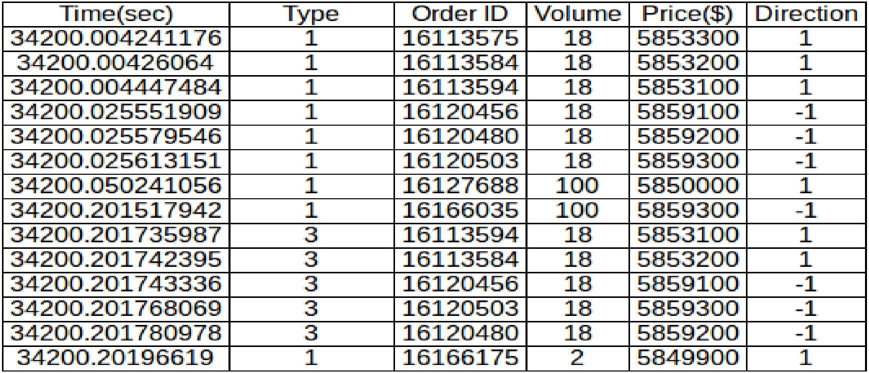
Limit order book data

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 10^{-4} dollar and each tick is one cent, 5 Event type, such as execution, cancellation and so on, 2 Direction ask and bid.

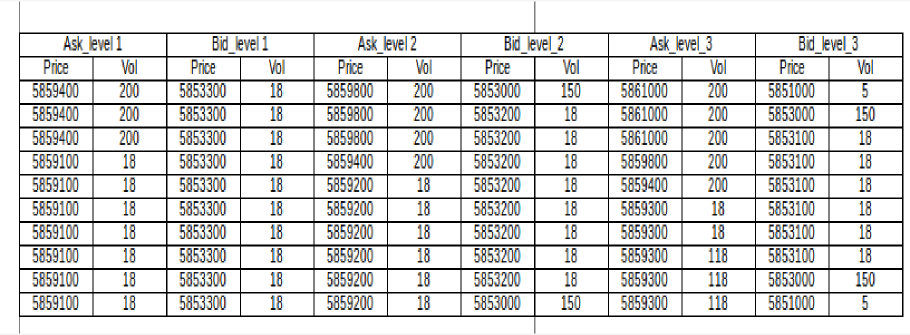
Order book types:

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. Price is in 10^{-4} dollar.

Contents

1 Brief summary

2 High frequency trading

3 DataSet

4 Methodology

5 Model fit

6 Trading strategy and PnL results

7 Future work

8 Questions

Methodology

Six machine learning algorithm candidates:

Basic methods:logistic regression with L1 penalty, logistic regression with L2 penalty,support vector machine, decision tree method(simply described)

Ensemble methods:Ada-boosting method, random forest method(mainly described).

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

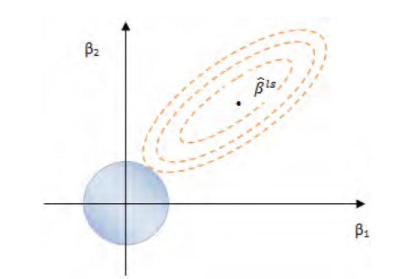
Ridge regression

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

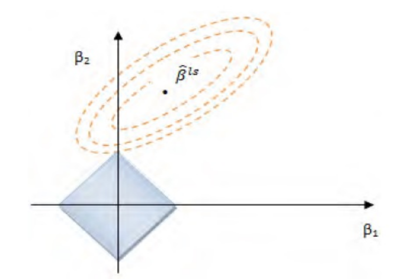
Lasso regression

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

Coefficients:



Coefficients:



Methodology

Comparison of L1 and L2 Penalized Model

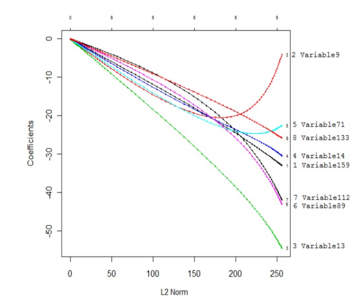
Ridge regression

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

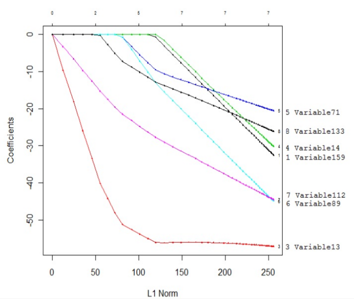
Lasso regression

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

Path::



Path::



Methodology

Support vector machine

- Maximize the margin among support vectors
- Use kernel function to project data from low dimension to high dimension with the same inner product, which is helpful to solve non-linear problem.

Try to maximize the margin:

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

primal form:

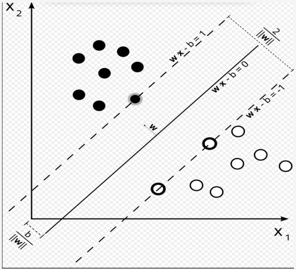
$$\max_{W,b} \; r = 1/||W||$$

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

Dual form:

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

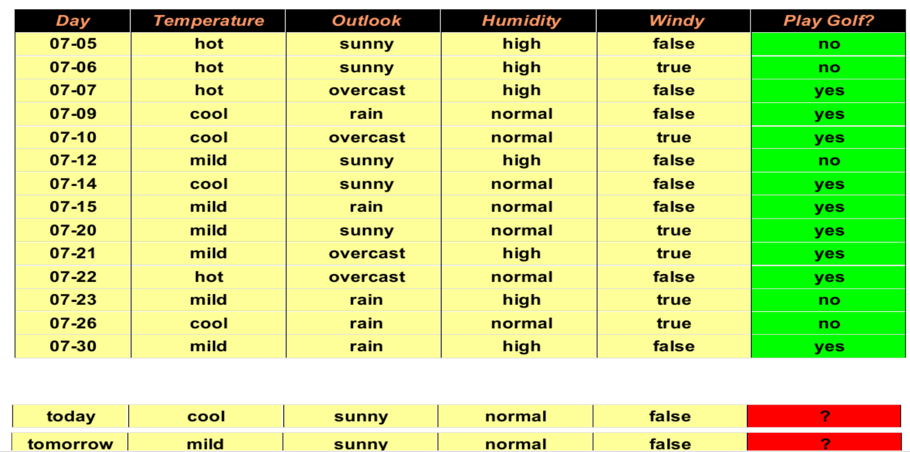
$$\text{s.t.} \alpha_l \geq 0, \sum_{l=1}^{M} \alpha_l y_l = 0$$



Methodology

Decision tree: Use **entropy and information gain** to define the root and parent nodes, split the data into different classes.

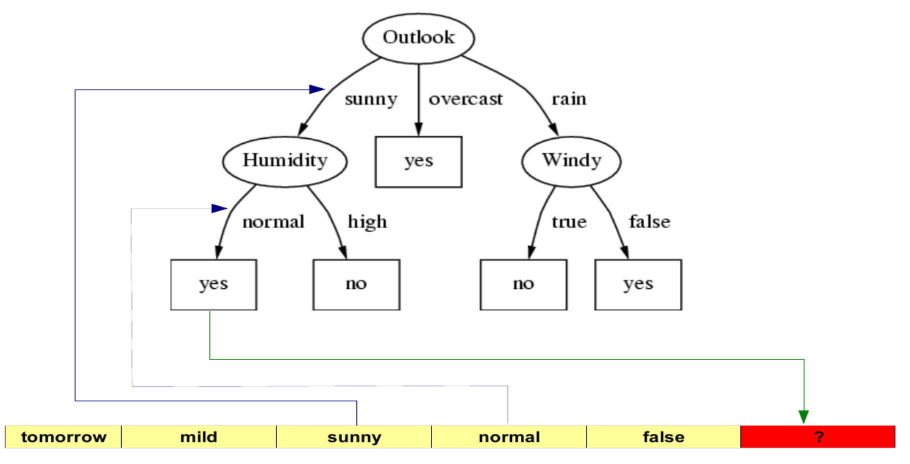
Example: Know the history of playing golf or not, given new data, make prediction, use as basic classifier



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



tomorrow	mild	sunny	normal	false	?

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:

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

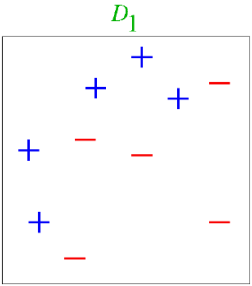
Methodology

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

AdaBoost example: TOY example:

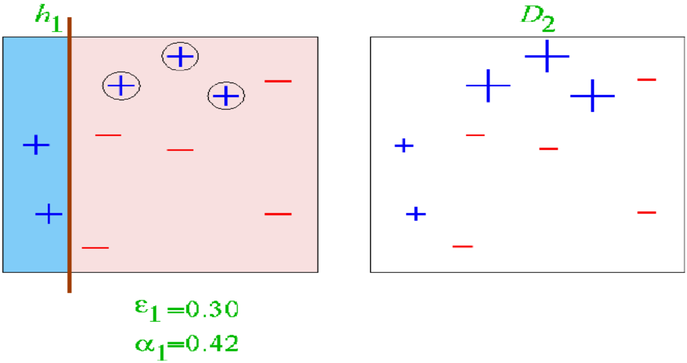


(taken from Verma & Thrun, Slides to CALD Course CMU 15-781, Machine Learning, Fall 2000)

Methodology

Round 1:

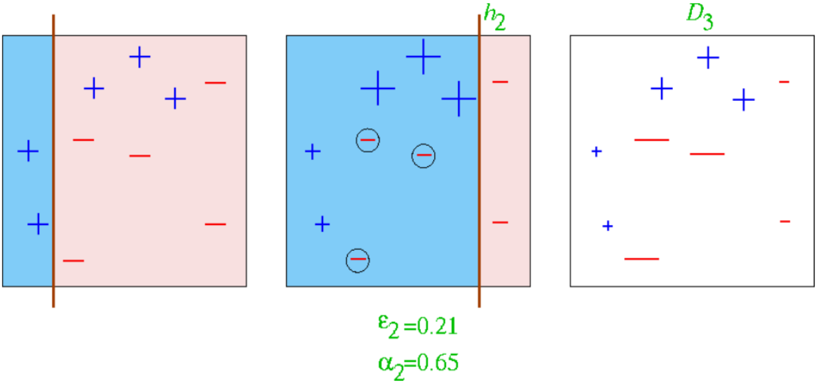
AdaBoost example: TOY example:



Methodology

Round 2:

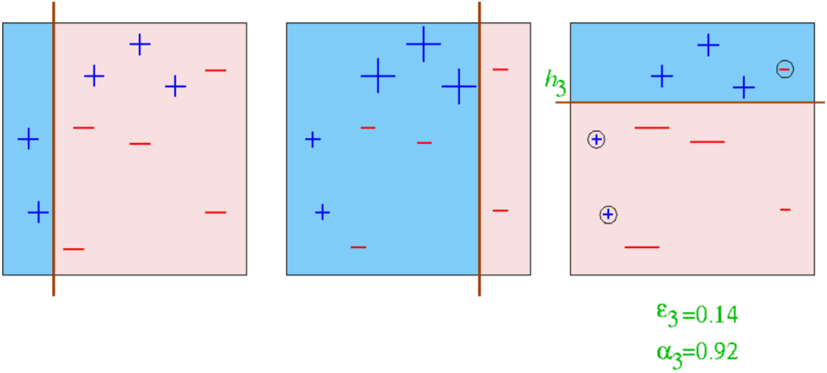
AdaBoost example: TOY example:



Methodology

Round 3:

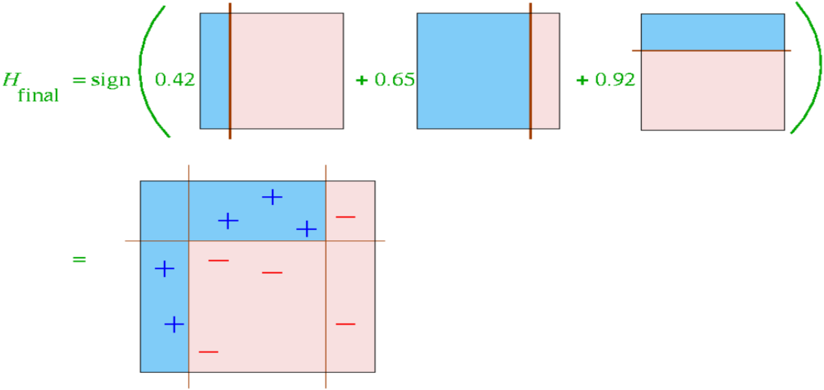
AdaBoost example: TOY example:



Methodology

Final round:

AdaBoost example: TOY example:


$H_{\text{final}} = \text{sign}\left(0.42 \quad + 0.65 \quad + 0.92 \right)$
=
+
+
+
+
+
−
−
−
−
−

Methodology

Second ensemble method: Random forest

choose N samples(bootstrap) and M attributes (features) each time

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:

```
Initialize the observation weights $\omega_i$=1/$N$,$i$=1,2,...,$N$;
for $m$=1 to $M$ do
    Fit a classifer $G_m(x)$ to the training data using weights $\omega_i$;
    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 $\omega_i \leftarrow \omega_i \cdot exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, \ldots, N$ ;
Output $G(x) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$
```

source:math description of algorithm from Elementary of statistical learning

Random forest algorithm:

```
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.
Output the ensemble of trees $\{T_b\}_1^B$
To make a prediction at a new point x:
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:math description of algorithm from Elementary of statistical learning

Contents

1 Brief summary

2 High frequency trading

3 DataSet

4 Methodology

5 Model fit

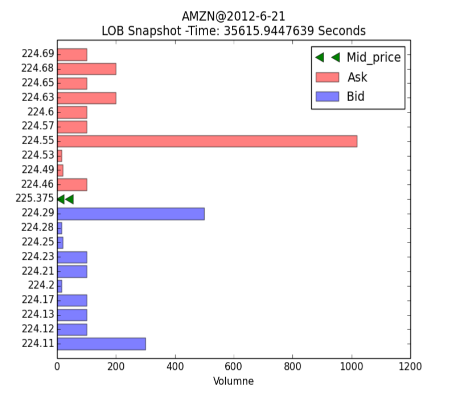
6 Trading strategy and PnL results

7 Future work

8 Questions

Model fit

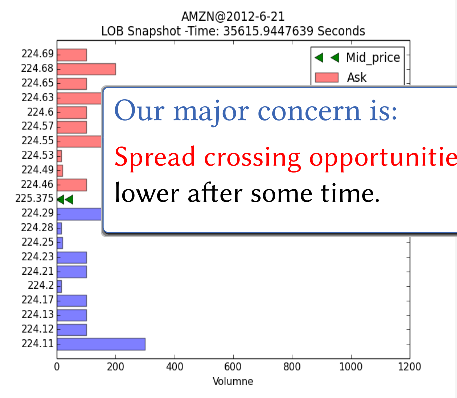
order book snapshot:



- 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 that no 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 that no direction change)

Our major concern is:

Spread crossing opportunities, that is bid higher or ask lower after some time.

Model fit

Ask low example(5 seconds future):

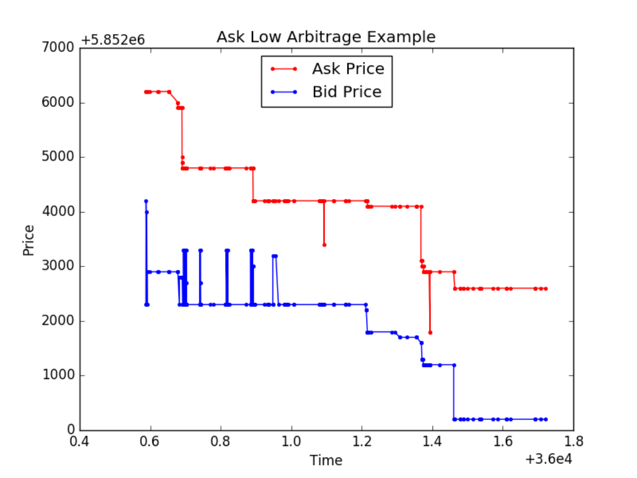


Figure : Ask low arbitrage example

Bid high example(5 seconds future):

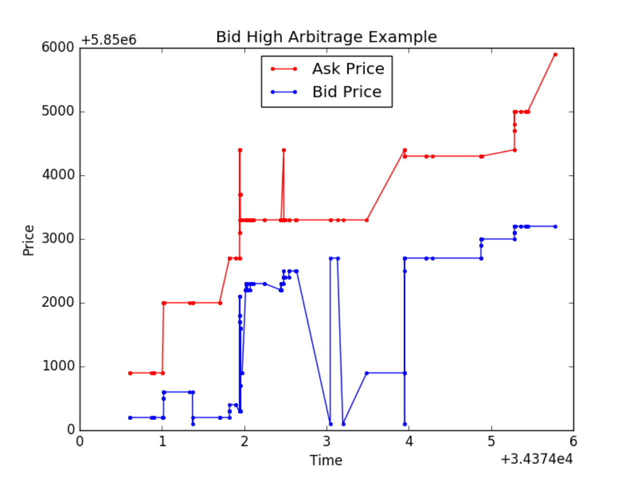


Figure : Bid high arbitrage example

Model fit

No arbitrage example(5 seconds future):

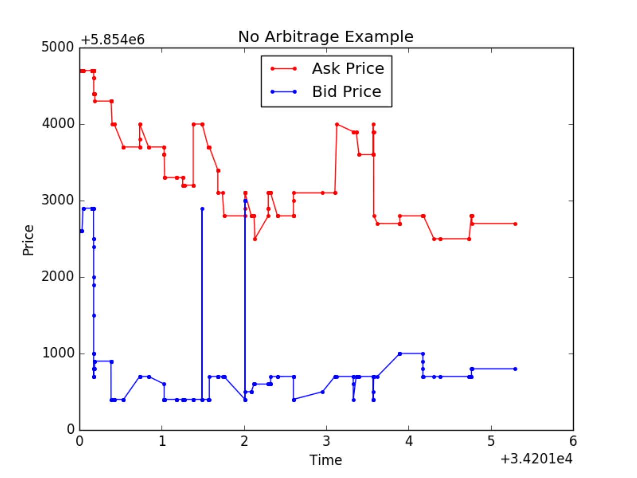
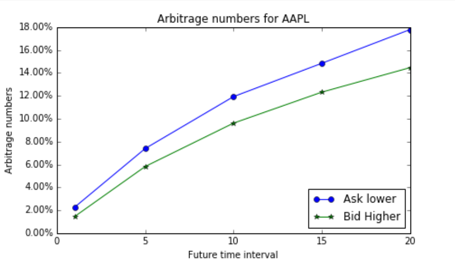


Figure : No arbitrage example

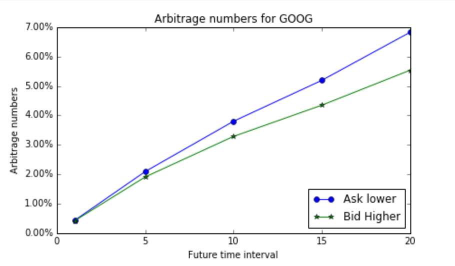
Model fit

Arbitrage opportunities based on future time: imblanced data

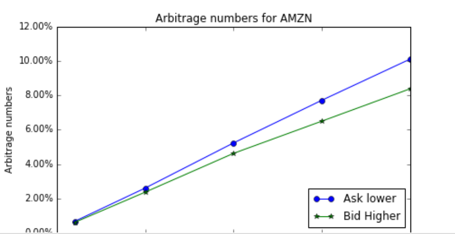
AAPL:



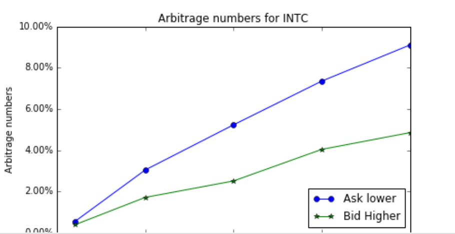
GOOG:



AMZN:



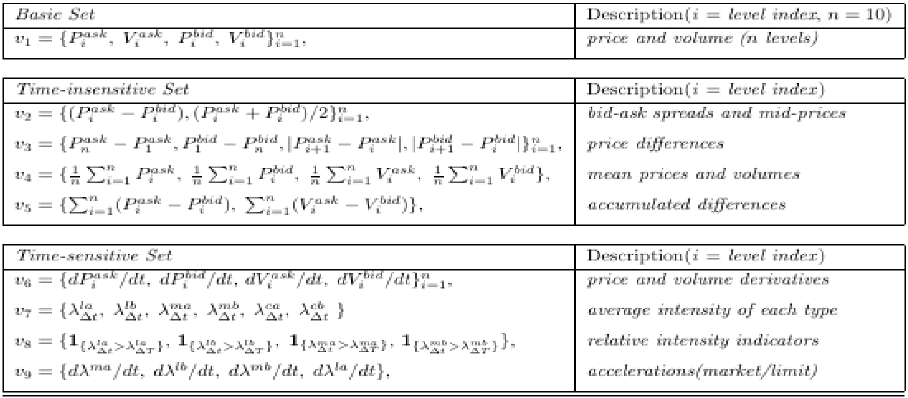
INTC:



Model fit

Build features:

According to Dr.Kercheval and Yuan Zhang(2015) in quantitative finance



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*
$v_8 = \{\mathbf{1}_{\{\lambda_{\Delta t}^{la} > \lambda_{\Delta T}^{la}\}},\ \mathbf{1}_{\{\lambda_{\Delta t}^{lb} > \lambda_{\Delta T}^{lb}\}},\ \mathbf{1}_{\{\lambda_{\Delta t}^{ma} > \lambda_{\Delta T}^{ma}\}},\ \mathbf{1}_{\{\lambda_{\Delta t}^{mb} > \lambda_{\Delta T}^{mb}\}}\},$	*relative intensity indicators*
$v_9 = \{d\lambda^{ma}/dt,\ d\lambda^{lb}/dt,\ d\lambda^{mb}/dt,\ d\lambda^{la}/dt\},$	*accelerations(market/limit)*

- 9 dataset 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? Pay more attention to rare events

Precision

Precision is the probability that a detected event is really arbitrage.

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

Recall

Recall is the probability that an aritrage opportunity is detected.

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

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:

AMZN ask low predict(5 seconds):

train to test ratio is: 9:1

Table : AAPL Accuracy rate and CPU time

Model	Training time(s)	Training accuracy	Test accuracy	Test f1 score
Logistic(Lasso penalty)	538	97.71%	98.45%	8.28%
Logistic(Ridge penalty)	7	97.71%	98.45%	8.28%
SVM(Poly 2 kernal,5000 estimator)	72	98.70%	99.00%	54.95%
Decision Tree(no maximum depth)	3.76	98.67%	98.95%	51.61%
Ada boosting(100 estimator)	365	99.99%	99.56%	84.05%
Random forest(100 estimator)	31	99.80%	97.15%	81.04%

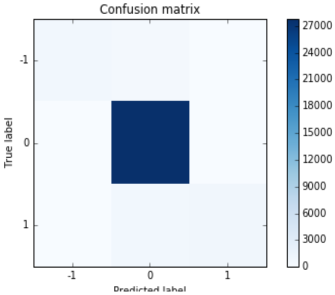
remark: training samples 90000 and test samples 10000. Computer is 8G memory and Intel Xeon E3 processor(4 cores), Logistic solve linear problem, svm and decision tree solve non linear problem, ensemble methods improve the performance

Methodology

Classification matrix for multi-class classification results:one against one and one against all

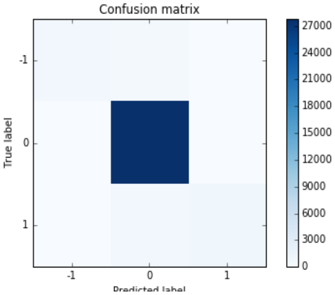
One against One

$$\begin{bmatrix} 118 & 58 & 0 \\ 2 & 9661 & 2 \\ 0 & 51 & 108 \end{bmatrix}$$



One against Rest

$$\begin{bmatrix} 113 & 61 & 0 \\ 4 & 9661 & 3 \\ 0 & 55 & 103 \end{bmatrix}$$



Contents

1 Brief summary

2 High frequency trading

3 DataSet

4 Methodology

5 Model fit

6 Trading strategy and PnL results

7 Future work

8 Questions

PnL

According to Nan Zhou, Wen Cheng, Yichen Qin Zongcheng Yin(2015) in quantitative finance.
PnL is the profit and loss through 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 transaction, α is significant level and c is trading cost.

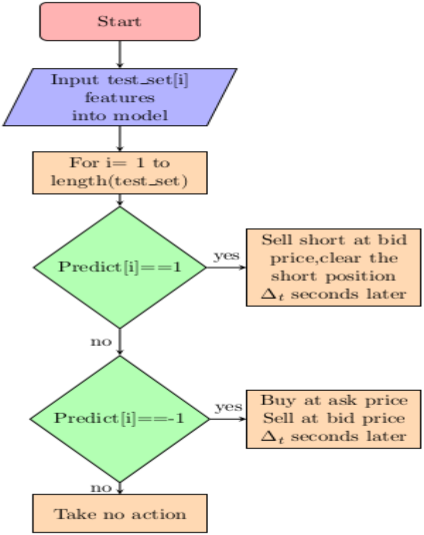
Trading strategy

Naive trading strategy:

Assume: α and c equal to 0

```
initialize: PnL=0
for i =1 to length(test_set) do
    input test_set[i] features into model and get result of Predict[i]
    if Predict[i]==1(Ask low) then
        Sell short at bid price
        Clear the short option $\Delta t$ seconds later
        PnL+=$Bid\_price_t - Ask\_price_{t+\Delta t}$
    else if Predicted[i]==-1(Bid high) then
        Buy at ask price
        Sell at bid price $\Delta t$ seconds later
        PnL+=$Bid\_price_{t+\Delta t} - Ask\_price_t$
    else
        Take no action
    return PnL
```

Strategy framework:



Start

Input test_set[i] features into model

For i= 1 to length(test_set)

Predict[i]==1

yes

Sell short at bid price,clear the short position Δ_t seconds later

no

Predict[i]==-1

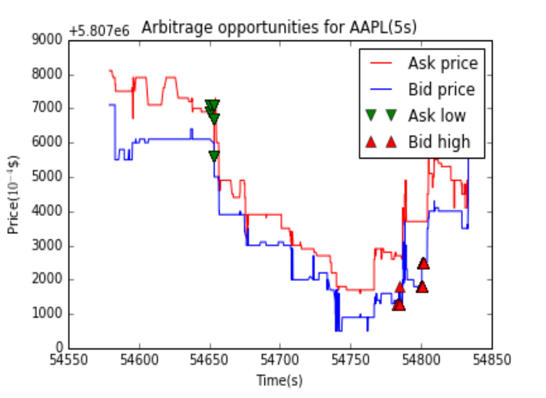
yes

Buy at ask price Sell at bid price Δ_t seconds later

no

Take no action

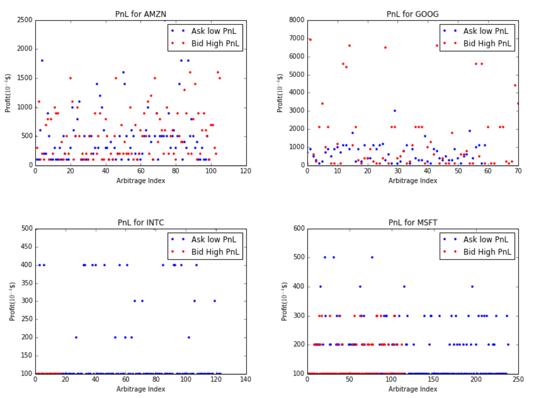
Trading strategies:



Ask low occurs: sell short current bid price. Bid high occurs: buy at current ask price

Each PnL result:

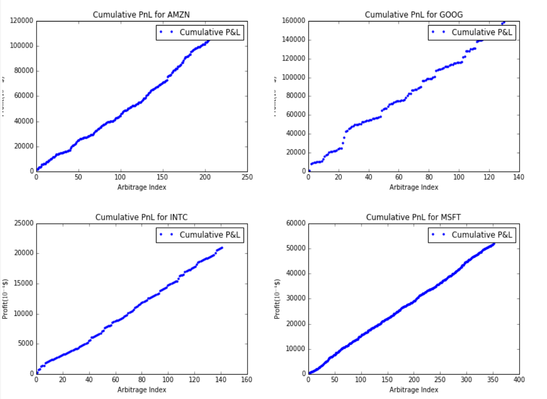
one against rest example:
For simplicity, assume both significant level α and trading cost c equal to 0.



Cumulative PnL result:

one against rest example:

For simplicity, assume both significant level α and trading cost c equal to 0.



reason: high precision!

Contents

1 Brief summary

2 High frequency trading

3 DataSet

4 Methodology

5 Model fit

6 Trading strategy and PnL results

7 Future work

8 Questions

Future work

- Compare the results in spark machine learning package.Can deal with big data problem
- Add more meaningful features and calculate the interaction.
- Neural network and deep learning. AlphaGo Google deepmind?
- Submit on journal of high frequency or quantitative finance

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

1 Brief summary

2 High frequency trading

3 DataSet

4 Methodology

5 Model fit

6 Trading strategy and PnL results

7 Future work

8 Questions

QA

Thanks a lot and Questions