Ensemble methods for capturing dynamics of limit order books

Jian Wang

wangjian790@gmail.com

Financial math Ph.D. Candidate
Florida State University

Summer 2017 Ph.D. defense

Table of contents

Contents

Brief summary

- Our main goal is to use ensemble machine learning methods to predict the limit order book price cross over opportunities.
- Use high frequency data to predict relatively long time future price changing trend(eg. 5 seconds later) to prevent illegal actions.
- Deal with relatively large dataset. Each stock contains hundred thousand data samples
- Features selection: choose what kind of data as our independent variables(choose x_i s) and compute feature importances.
- 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)
- Build limit order book python package and the codes are portable.

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:

Passive: use limit order book

Market Making

Provides liquidity by matching buyer and seller orders or by buying and selling through its own securities inventories. Earn liquidity rebates and bid ask spread.

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 buyers 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

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

AMZN as example:

Time(sec)	Event Type	Quantity	Price	Side
34200.017459617	5	1	2238200	-1
34200.18960767	1	21	2238100	1
34200.18960767	1	20	2239600	-1
34200.18960767	1	100	2237500	1
34200.18960767	1	13	2240000	-1
34200.18960767	1	2	2236500	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:

Table : Limit book example of stock AMZN, a sample on 2012-06-21

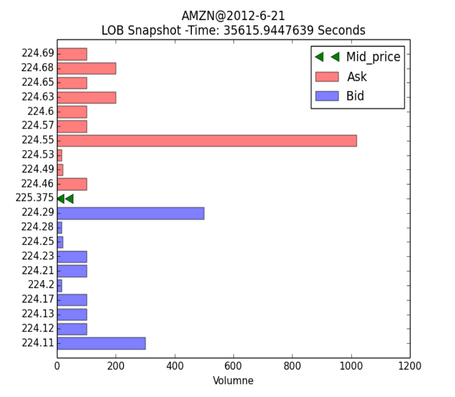
Level 1				Level 2				...
Ask		Bid		Ask		Bid		...
Price	Quantity	Price	Quantity	Price	Quantity	Price	Quantity	
2239500	100	2231800	100	2239900	100	2230700	200	...
2239500	100	2238100	21	2239900	100	2231800	100	...
2239500	100	2238100	21	2239600	20	2231800	100	...
2239500	100	2238100	21	2239600	20	2237500	100	...
2239500	100	2238100	21	2239600	20	2237500	100	...
2239500	100	2238100	21	2239600	20	2237500	100	...

From level 1 to level 10, where the first level is the best bid and ask. Price is in 10^{-4} dollar.

Contents

Problem

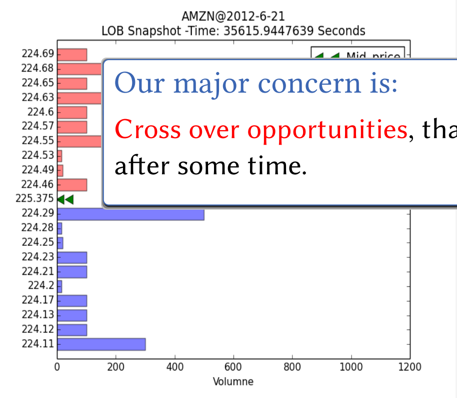
Predict arbitrage opportunities of high frequency data based on fixed future time



- 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)

Problem

Predict arbitrage opportunities of high frequency data based on fixed future time



- At Time t: $P_t^A > P_t^B$, no arbitrage

three

enote as

Our major concern is:

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

•$P^B_{t+\Delta t} > P^A_t$: bid higher, denote as -1 in our model

•otherwise(implies that no direction change)

Problem

Ask low example(5 seconds future):

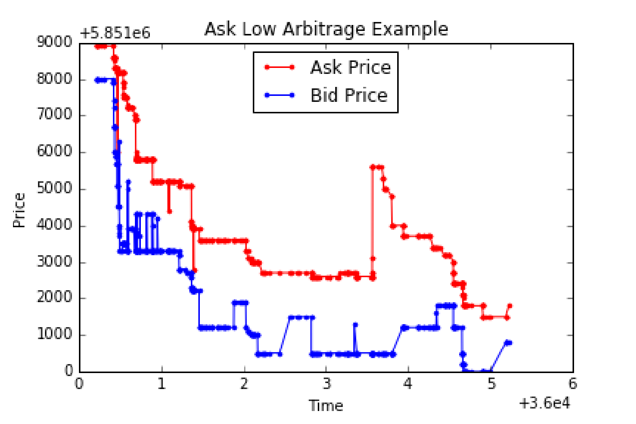


Figure : Ask low arbitrage example

Bid high example(5 seconds future):

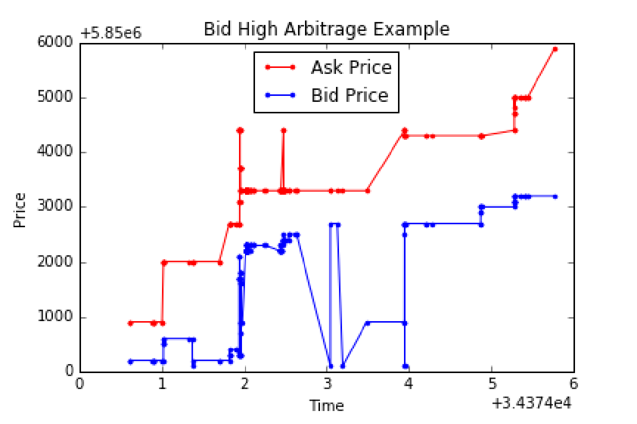


Figure : Bid high arbitrage example

Problem

No arbitrage example(5 seconds future):

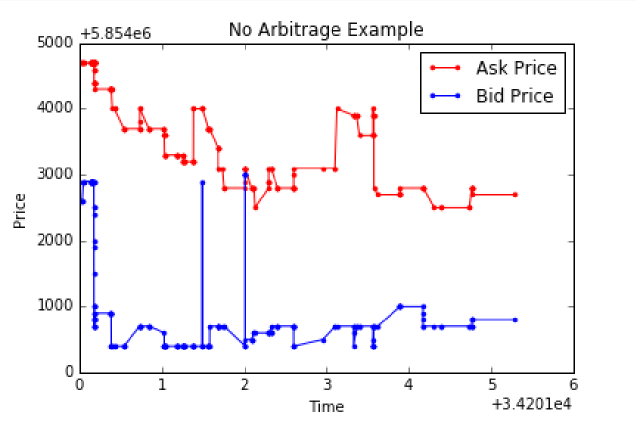


Figure : No arbitrage example

Why shall we predict arbitrages on future time interval instead of future events,like most past papers did? Easy to design a trading strategy.

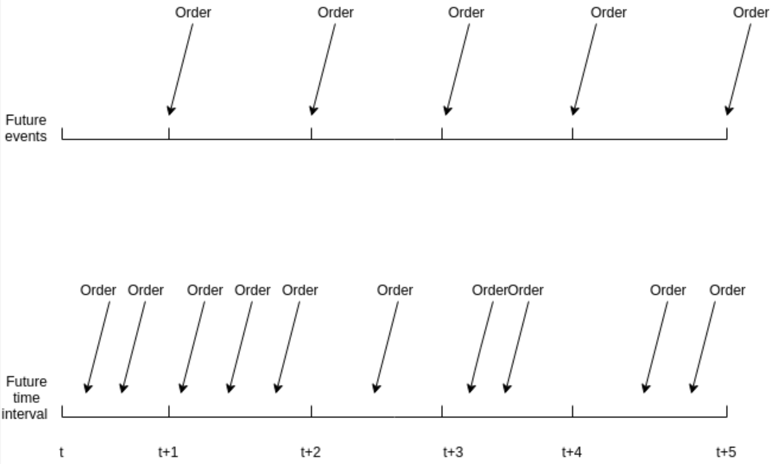
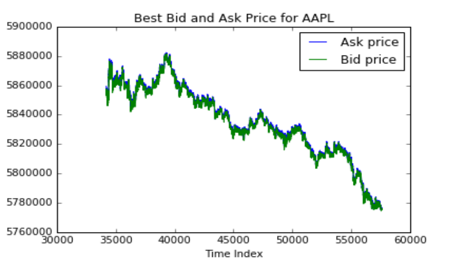


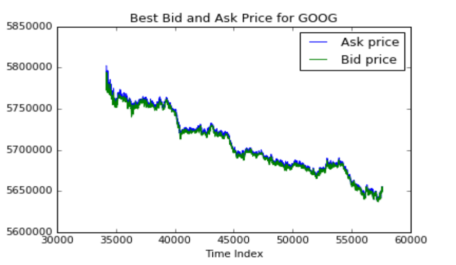
Figure : Future events Vs. future time interval

Stock Price

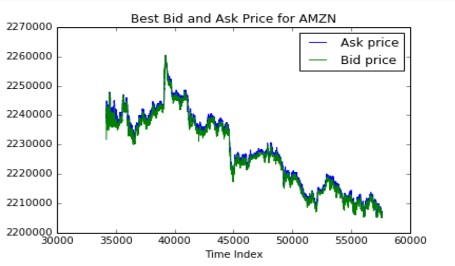
AAPL:



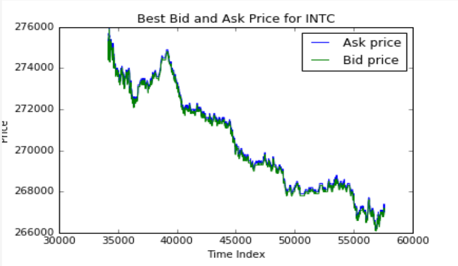
GOOG:



AMZN:



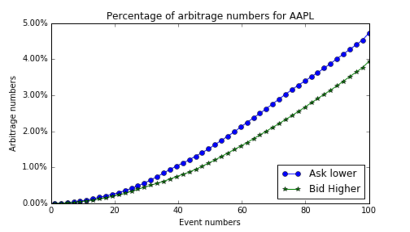
INTC:



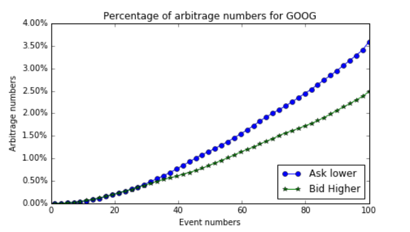
Arbitrage opportunities

Arbitrage opportunities based on future event

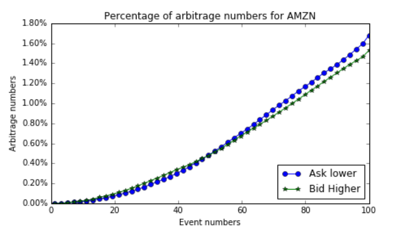
AAPL:



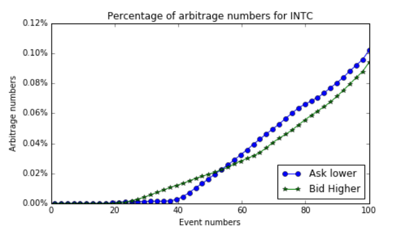
GOOG:



AMZN:



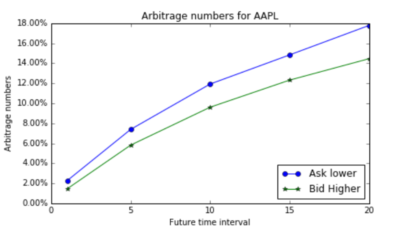
INTC:



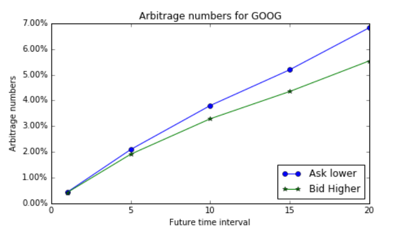
Arbitrage opportunities

Arbitrage opportunities based on future time

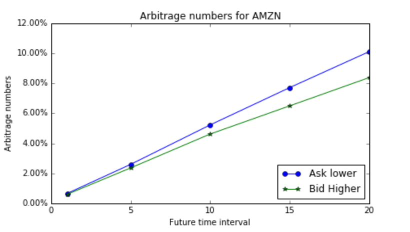
AAPL:



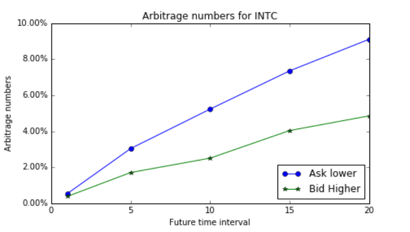
GOOG:



AMZN:



INTC:



Contents

Methodology

Classification Problem:

$$Y = f(X)$$

Where Y is category responses and X is feature vectors. Y in our case corresponds to occurrence of arbitrages. f is the model that maps the features into categories. Therefore, building meaningful features and choosing suitable model schemes are critical.

Methodology

Build features:

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

Basic Set	Description(i=level index, n=10)
$v_1 = \{P_i^{ask}, V_i^{ask}, P_i^{bid}, V_i^{bid}\}_{i=1}^{n}$ (40)	price and volume(n levels)
Time-insensitive Set	Description(i=level index)
$v_2 = \{(P_i^{ask} - P_i^{bid}), (P_i^{ask} + P_i^{bid})/2\}_{i=1}^{n}$ (20)	bid ask spreads and mid prices(n levels)
$v_3 = \{\lvert P_{i+1}^{ask} - P_1^{ask}\rvert, \lvert P_{i+1}^{bid} - P_1^{bid}\rvert, \lvert P_{i+1}^{ask} - P_i^{ask}\rvert, \lvert P_{i+1}^{bid} - P_i^{bid}\rvert\}_{i=1}^{n-1}$ (36)	price difference
$v_4 = \{\frac{1}{n}\sum_{i=1}^{n} P_i^{ask}, \frac{1}{n}\sum_{i=1}^{n} P_i^{bid}, \frac{1}{n}\sum_{i=1}^{n} V_i^{ask}, \frac{1}{n}\sum_{i=1}^{n} V_i^{bid}\}$ (4)	mean prices and volumes
$v_5 = \{\sum_{i=1}^{n}(P_i^{ask} - P_i^{bid}), \sum_{i=1}^{n}(V_i^{ask} - V_i^{bid})\}$ (2)	accumulated difference
Time-sensitive Set	Description(i=level index)
$v_6 = \{\partial P_i^{ask}/\partial t, \partial P_i^{bid}/\partial t, \partial V_i^{ask}/\partial t, \partial V_i^{bid}/\partial t\}_{i=1}^{n}$ (40) $v_7 = \{\lambda_{\Delta_t}^{la}, \lambda_{\Delta_t}^{lb}, \lambda_{\Delta_t}^{ma}, \lambda_{\Delta_t}^{mb}, \lambda_{\Delta_t}^{ca}, \lambda_{\Delta_t}^{cb}\}$ (6) $v_8 = \{1_{\lambda_{\Delta_t}^{la} > \lambda_{\Delta_T}^{la}}, 1_{\lambda_{\Delta_t}^{lb} > \lambda_{\Delta_T}^{lb}}, 1_{\lambda_{\Delta_t}^{ma} > \lambda_{\Delta_T}^{ma}}, 1_{\lambda_{\Delta_t}^{mb} > \lambda_{\Delta_T}^{mb}}\}$ (4) $v_9 = \{\partial\lambda^{ma}/\partial t, \partial\lambda^{lb}/\partial t, \partial\lambda^{mb}/\partial t, \partial\lambda^{la}/\partial t\}$ (4)	price and volume derivatives average intensity of each type relative intensity indicators accelerations(/limit)

- contain price,volume, bid ask spread, price difference and volume difference for each level, mean of price and volume.
- total 156 variables, can be treated as high dimensional problems.

Methodology

Models:

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) are used as benchmarks

Ensemble methods: AdaBoosting method, random forest method(mainly described).

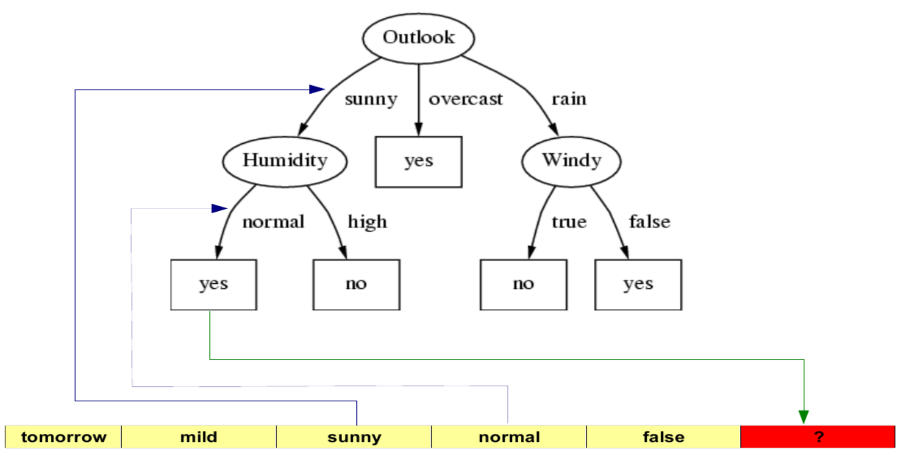
Methodology

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

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

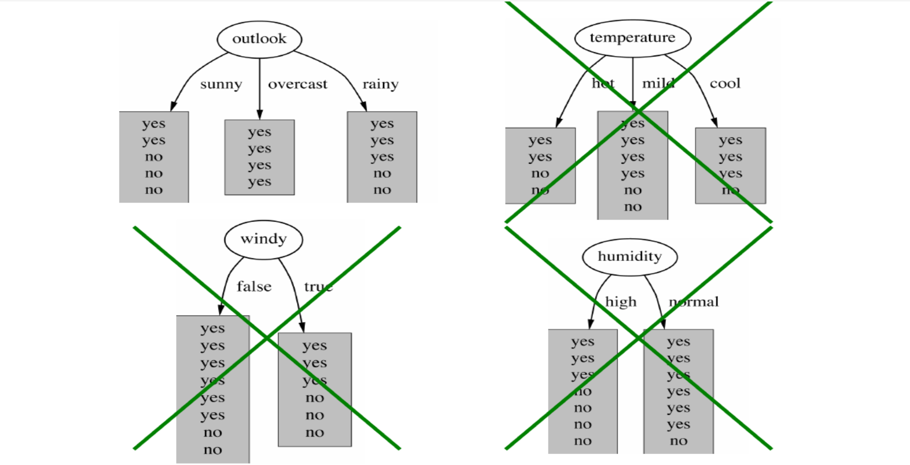
today	cool	sunny	normal	false	?
tomorrow	mild	sunny	normal	false	?

Methodology



Methodology

Which attribute to select as the root?



Methodology

Entropy:

Entropy is a measure for un-orderedness

$$E(s) = -\sum_{i=1}^{n} p_i log_2 p_i$$

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 = overcast: 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:

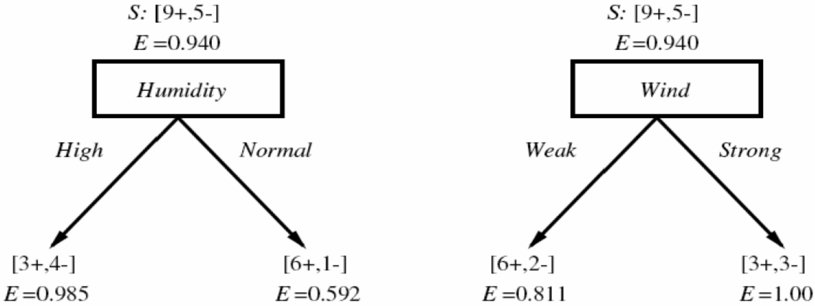
When an attribute A splits the set S into subsets S_i

- 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



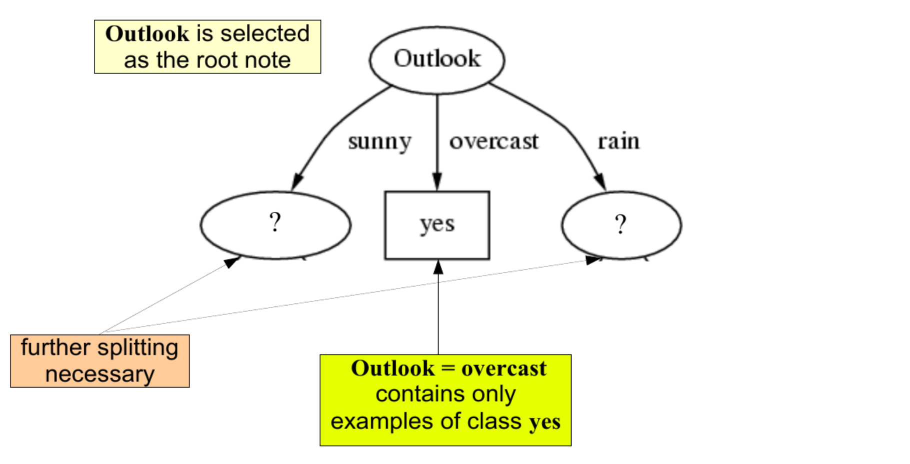
Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

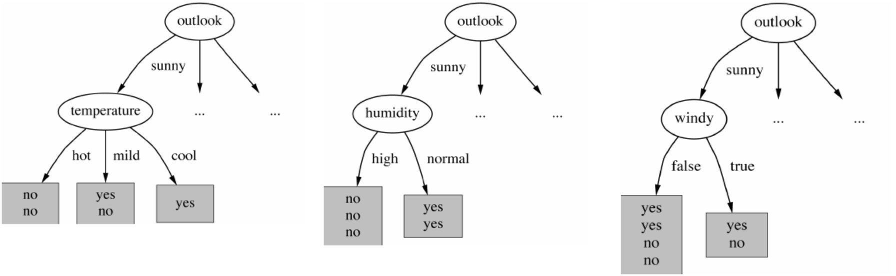
$Gain(S, Outlook) = 0.246$

$Gain(S, Temperature) = 0.029$

Methodology



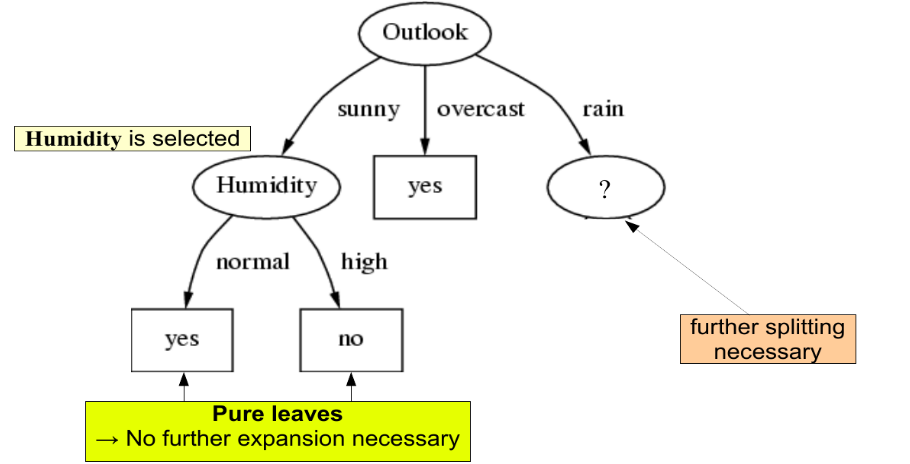
Methodology



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

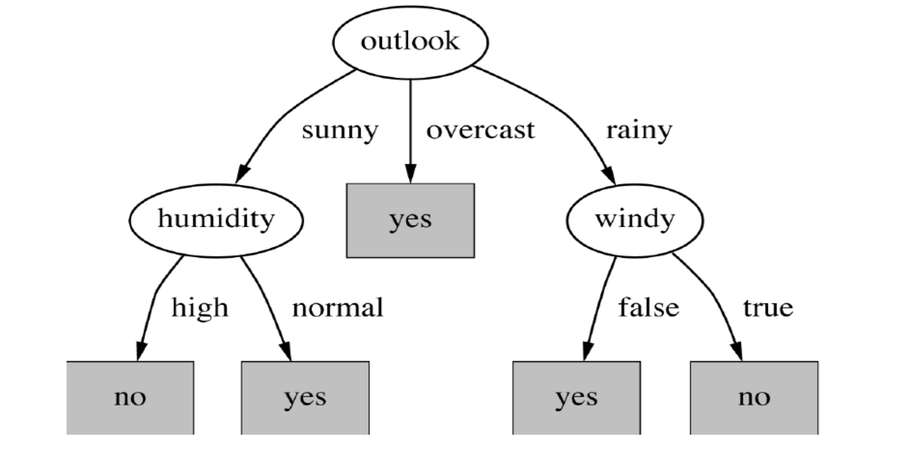
Humidity is selected

Methodology



Methodology

Final structure:



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 identical and relatively 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:

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

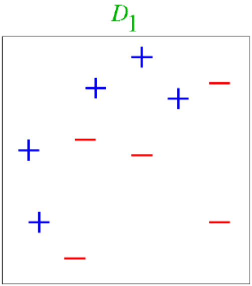
Methodology

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:

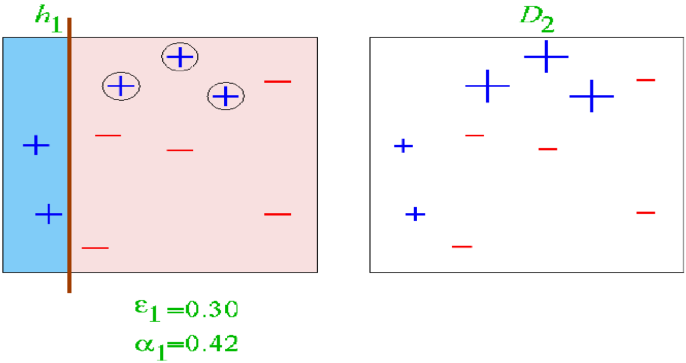


(taken from Verma & Thrun, Slides to CALD Course CMU 15-781, Machine Learning, Fall 2000)

Methodology

Round 1:

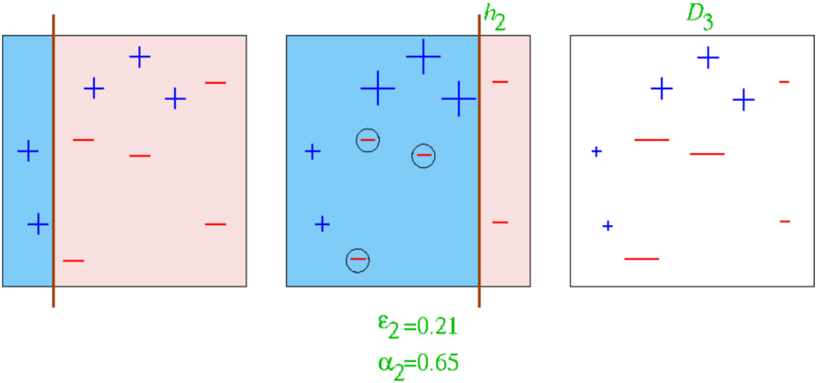
AdaBoost example: TOY example:



Methodology

Round 2:

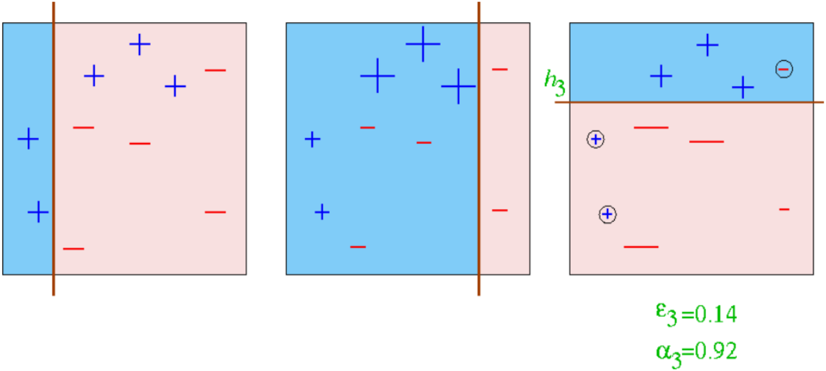
AdaBoost example: TOY example:



Methodology

Round 3:

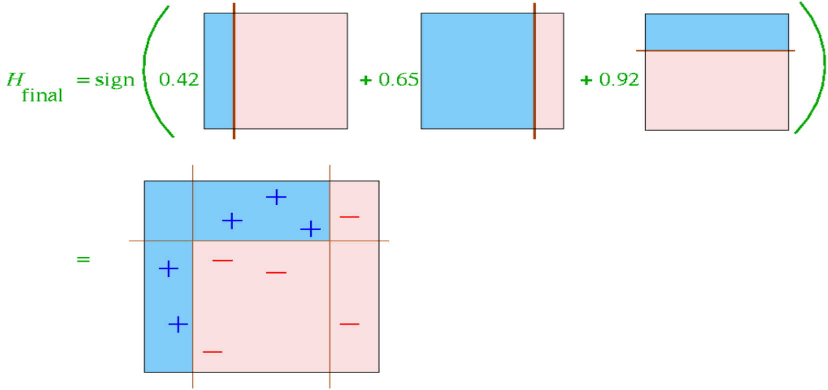
AdaBoost example: TOY example:



Methodology

Final round:

AdaBoost example: TOY example:



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

Numerical results: Measurement

Criteria: Only consider accuaracy? Imbalanced data?

Precision

Precision is the probability that a (randomly selected) detected arbitrage opportunty is real arbitrage opportunites.

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

Recall

Recall is the probability that a (randomly selected) real arbitrage opportunity is detected by our model.

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

F1 score

A measure that combines precision and recall is the harmonic mean of precision and recall.β is usually chosen as 0.5

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

Numerical results: Binary case

AMZN ask low predict(5 seconds):

Train to test ratio is: 9:1

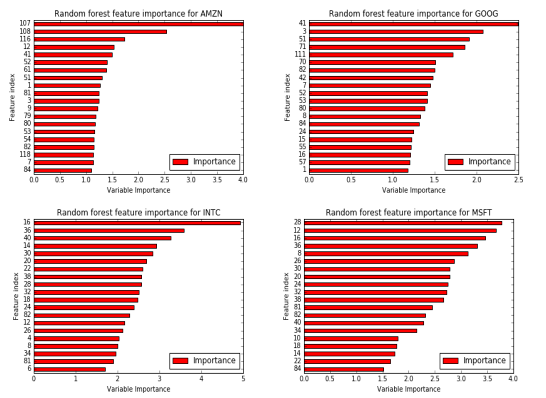
Table : Binary prediction results of stock AMZN

Model	Training time(s)	Training F1 score	Test time(s)	Test Recall	Test Precision	Test F1 score
Logistic regression(Lasso penalty)	260.7	8.8 %	0.002	2.9%	75.0%	5.6%
Logistic regression(Ridge penalty)	7.2	8.8 %	0.01	2.9%	75.0%	5.6%
SVM(Poly 2 kernal, 5000 estimator)	75.7	61.5 %	4.3	29.1%	96.8%	44.8%
Decision Tree(no pruning)	3.9	61.8 %	0.003	30.1%	91.2%	45.3%
AdaBoost(number of estimate=100)	30.0	96.5 %	0.04	73.8 %	92.7%	82.2%
Random forest(number of estimate=100)	37.5	99.1 %	0.11	72.8 %	96.2%	82.9%

Remark: training samples 90000 and test samples 10000. The estimation number for AdaBoost and random forest is 100.Computer is 8G memory and Intel Xeon E3 processor(4 cores)

Numerical results: Binary case

Feature Importance:



For stock AMZN, the first three important features are 107, 108 and 116 which represents $\partial P_5^{ask}/\partial t$, $\partial P_6^{ask}/\partial t$, and $\partial P_4^{bid}/\partial t$ respectively. The first 3 import stock for GOOS are 41, 3, 51 which represent $P_1^{ask} - P_1^{bid}$, P_3^{ask}, and $P_1^{ask} + P_1^{bid}$. For stock INTC, they are 16, 36, 40 which represents V_6^{ask}, V_6^{bid}, and V_{10}^{bid}. For stock MSFT, they are 28, 12 and 16 which represent P_8^{bid}, V_2^{ask}, and V_6^{ask}.

Feature importance: Random forest

Table : Ratio of features on bid and ask side

Side	AMZN	GOOG	INTC	MSFT
Ask side features	18	17	10	10
Bid side features	7	10	10	10
Ratio of ask to bid	2.6	1.7	1	1

For mid prices and bid ask spread features, we count on both bid and ask side. From the results of these four stocks, we can see that features on ask side play a more important role.

Feature importance: Random forest

Table : The most frequent occurrence features among four stocks

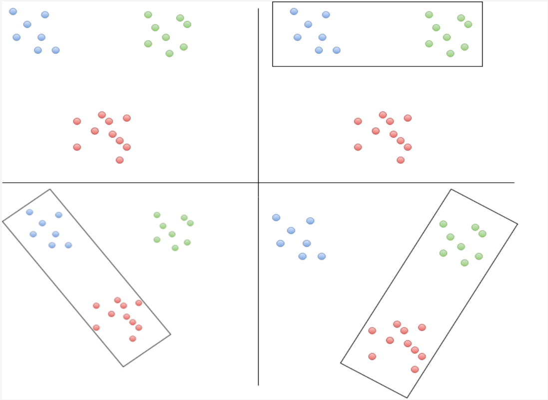
Index	Feature	Number of occurrence
82	$\lvert P_5^{ask} - P_4^{ask} \rvert$	4
84	$\lvert P_7^{ask} - P_6^{ask} \rvert$	3
81	$\lvert P_4^{ask} - P_3^{ask} \rvert$	3
24	P_4^{bid}	3
16	V_6^{ask}	3
12	V_2^{ask}	3
8	P_8^{ask}	3

Price differences between adjacent price level are important.

Multi-classes schemes:

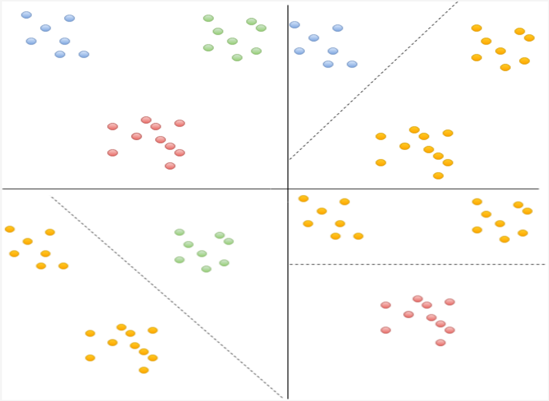
Multi-class classification results:ask-low as 1, bid-high as -1, and no arbitrage as 0

One against one:



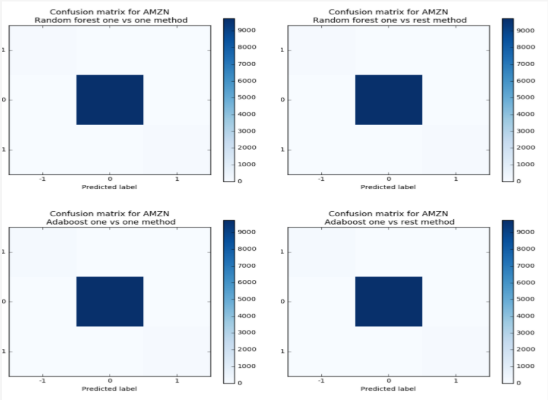
Multi-classes schemes:

Multi-class classification:ask-low as 1, bid-high as -1, and no arbitrage as 0
One against rest:



Numerical results: Multi-classes

Classification matrix for multi-class classification:ask-low as 1, bid-high as -1, and no arbitrage as 0



X-axis is predicted labels and Y-axis is true labels

Numerical results: Multi-classes

Classification matrix for multi-class classification results:ask-low as 1, bid-high as -1, and no arbitrage as 0

One against One

Random forest:

$$\begin{bmatrix} 109 & 27 & 0 \\ 0 & 9736 & 3 \\ 0 & 33 & 92 \end{bmatrix}$$

AdaBoost:

$$\begin{bmatrix} 128 & 8 & 0 \\ 2 & 9726 & 11 \\ 0 & 18 & 107 \end{bmatrix}$$

One against Rest

Random forest:

$$\begin{bmatrix} 109 & 27 & 0 \\ 0 & 9737 & 2 \\ 0 & 38 & 87 \end{bmatrix}$$

AdaBoost:

$$\begin{bmatrix} 120 & 16 & 0 \\ 0 & 9736 & 3 \\ 0 & 27 & 98 \end{bmatrix}$$

Contents

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:

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

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

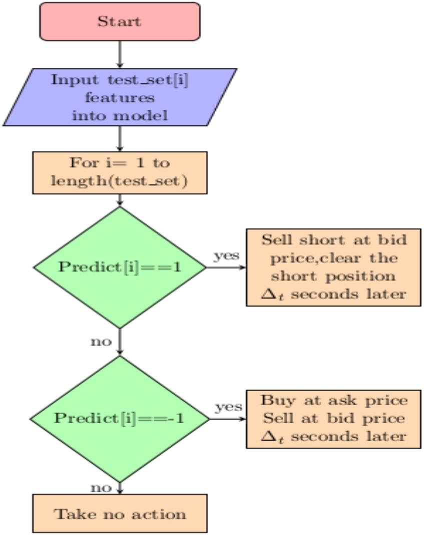
Trading strategy

Naive trading strategy:

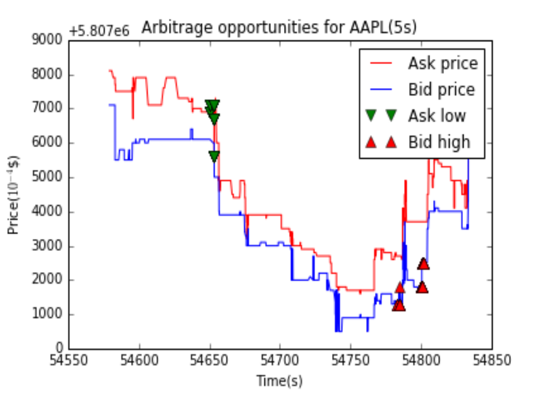
Assume: $\alpha = 0$ and $c = 0.02$

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 Δt seconds later
        PnL+=$Bid\_price_t - Ask\_price_{t+\Delta t} - cost$
    else if *Predicted[i]==-1(Bid high)* **then**
        Buy at ask price
        Sell at bid price Δt seconds later
        PnL+=$Bid\_price_{t+\Delta t} - Ask\_price_t - cost$
    else
        Take no action
    return PnL

Strategy framework:



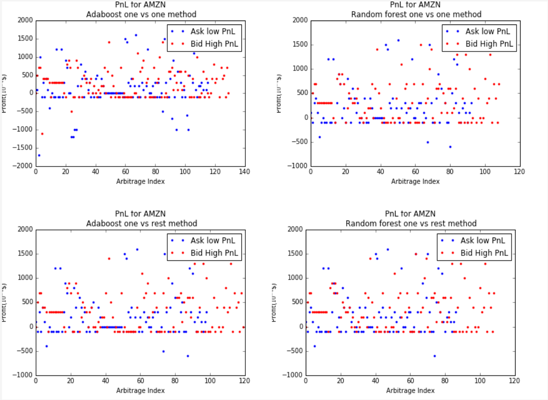
Trading strategies:



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

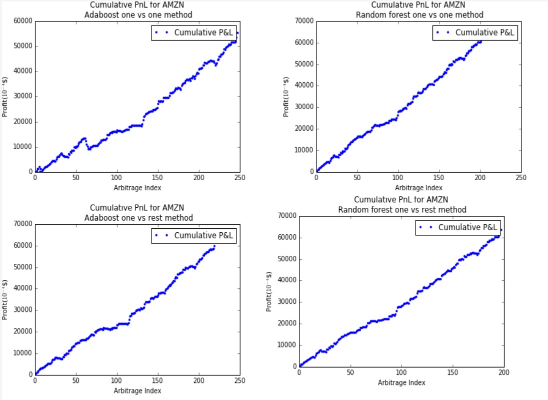
Each PnL result:

For simplicity, assume significant level $\alpha = 0$ and trading cost c equal to \$0.02.



Cumulative PnL result:

For simplicity, assume significant level $\alpha = 0$ and trading cost c equal to \$0.02.



Contents

Future work

- Apply our frameworks to spark system. Can deal with bigger data problem
- Add more meaningful features and calculate the interaction.
- Try other powerful machine learning tools such as reinforcement learning or deep learning.
- Extend the similar frameworks into other financial markets such as exchanges or options

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