High frequency data trading

Jian Wang

wangjian790@gmail.com

Financial math Ph.D Candidate
Florida State University

©©©©

High Frequency data Conference data Conference 2015

Table of contents

Contents

- Main goal is to use boosting machine learning method to predict the limit order book price cross over opportunity.

- Use the high frequency data to predict relatively long time future price changing trend.

- Compare the accuracy rate and calculation time among different machine learning methods, and show that the boosting method can improve the predicting performance to some extent.

Contents

Dataset

Limit order book data

The dataset contains limit order book prices of specific stock 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.

More details can be found in the following two charts.

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

Time is in sec and minimum time change is nanosecond, Price is in dollars and each tick is one cent, 7 Event type, such as execution, cancellation and so on, 2 Direction ask and bid.

Order Book:

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	...

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

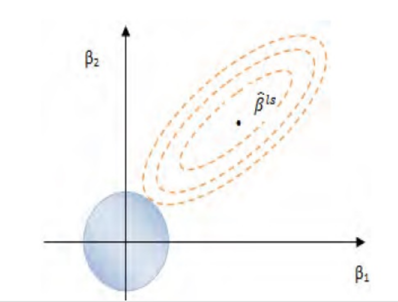
Ridge regression

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

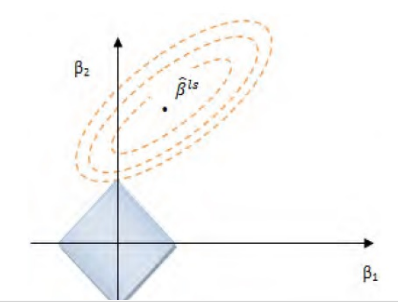
Lasso regression

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

Coefficients:



Coefficients:



Methodology

Comparison of L1 and L2 Penalized Model

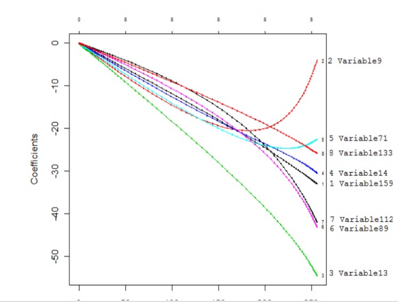
Ridge regression

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

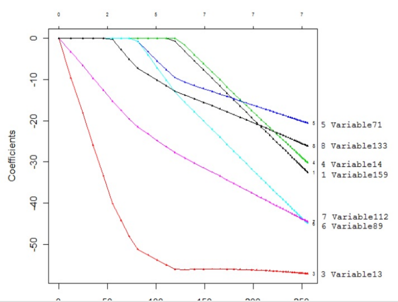
Lasso regression

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

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:

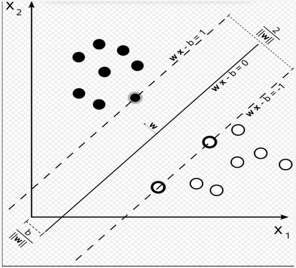
$$\max_{W,b} \ r = 1/||W||$$

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

Dual form:

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

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



Methodology

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 polinomial

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

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): $k(x, y) = exp(-\gamma||x - y||^2)$

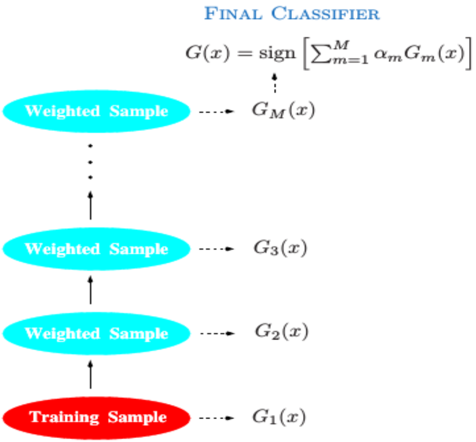
Methodology

Boosting methods

- 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"

Methodology

Boosting methods



Methodology

Adaboosting algorithm:

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\mathrm{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \mathrm{sign}\left[\sum_{m=1}^M \alpha_m G_m(x)\right]$.

source:ESL

- Put more weights on the false classification data

Gradient tree boosting algorithm:

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^N L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute
   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm},\ j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute
   $$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$
   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

source:ESL

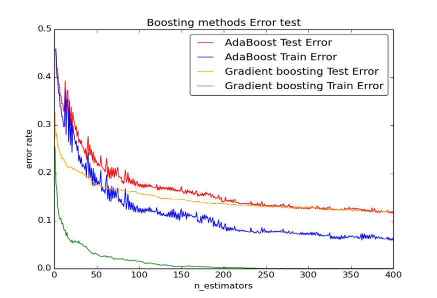
- Use gradient descent methods to minimize the residual in each step

Boosting methods error rate evolution

Boosting method can dramatically increase the performance of even a very weak classifier.We further implement the figure 10.2 in the ESL for example. Suppose features $X_1, X_2, \ldots, X_10$ are standard independent Gaussian, and the deterministic target Y is defined by:

$$Y = \begin{cases} 1 & \mathit{if} \sum_{j=1}^{10} X_j^2 > \chi_{10}^2(0.5) \\ -1 & \mathit{otherwise} \end{cases}$$

where $\chi_{10}^2(0.5)$ is the median of a chisquare random variable with 10 degrees of freedom.



- Boosting methods can reduce the prediction error rate to around one third of the original.
- In this case, Gradient boosting method performance better.

Methodology

Glasso:

Suppose we have N multivariate normal observations of dimension p , with mean μ and covariacne Σ. Let $\Theta = \Sigma^{-1}$ and S be the empirical covariance matrix, the problem is to maximize the log- likelihood

$$lnP(X|u,\Sigma) = -\frac{N}{2} ln|\Sigma| - \frac{1}{2}\sum(x_n - u)^T\Sigma^{-1}(x_n - u)$$

combined with the L_1 penalty

$$ln|\Theta| - tr(S\Theta)) - \lambda||\Theta||_1$$

Algorithm

Many algorithms for this problem, The following might be the oldest and simple one by Meinshausen and Buhlmann(2006)

- Estimate a sparse graphical model by fitting a lasso model to each variable, using others as predictors
- Set Σ_{ij}^{-1} to be non zero, if either the estimated coefficient of variable i

Methodology

Glasso:

Suppose we have N multivariate normal observations of dimension p , with mean μ and covariacne Σ. Let $\Theta = \Sigma^{-1}$ and S be the empirical covariance matrix, the problem is to maximize the log- likelihood

$$lnP(X|u,\Sigma) = -\frac{N}{2} ln|\Sigma| - \frac{1}{2}\sum(x_n - u)^T \Sigma^{-1}(x_n - u)$$

combined with the L_1 penalty

$$ln|\Theta| - tr(S\Theta)) - \lambda||\Theta||_1$$

Algorithm

Many algorithms for this problem, The following might be the oldest and simple one by Meinshausen and Buhlmann(2006)

- Estimate a sparse graphical model by fitting a lasso model to each variable, using others as predictors
- Set Σ_{ij}^{-1} to be non zero, if either the estimated coefficient of variable i

Bayesian-Glasso model

For the high dimensional problem, it is not very easy to built the Bayesian network due to its exponentially increasing complexity.
Our idea is to first use the Glasso model to conduct the model selection and then use Bayesian network structure learning process to define the network structure.

Algorithm

- Use Glasso algorithm to find the edges among variables
- Use greedy search methods to change the direction only on those existed edges
- Choose the direction which has the optimal BIC score
- Finish when all the edges are reached or attain the maximum iteration numbers

Bayesian-Glasso model

For the high dimensional problem, it is not very easy to built the Bayesian network due to its exponentially increasing complexity.
Our idea is to first use the Glasso model to conduct the model selection and then use Bayesian network structure learning process to define the network structure.

Algorithm

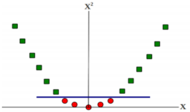
- Use Glasso algorithm to find the edges among variables
- Use greedy search methods to change the direction only on those existed edges
- Choose the direction which has the optimal BIC score
- Finish when all the edges are reached or attain the maximum iteration numbers

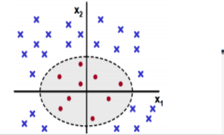
Methodology

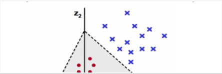
Higher dimensional situations

sometimes, in lower dimension we can not separate the data properly, so we need to project the data to the high dimensions

Examples







Methodology

Reference

- Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". Journal of the Royal Statistical Society, Series B 58 (1): 267–288. JSTOR 2346178
- Hoerl, A.E. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12: 55-67
- Vapnik, V. (1995). "Support-vector networks". Machine Learning 20 (3): 273. doi:10.1007/BF00994018

Packages

- R packages: glm, glmnet, e1071,bnlearn,huge
- Python packages: sklearn (svm, ridge, lasso, logistic)

Methodology

Reference

- Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". Journal of the Royal Statistical Society, Series B 58 (1): 267–288. JSTOR 2346178
- Hoerl, A.E. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12: 55-67
- Vapnik, V. (1995). "Support-vector networks". Machine Learning 20 (3): 273. doi:10.1007/BF00994018

Packages

- R packages: glm, glmnet, e1071,bnlearn,huge
- Python packages: sklearn (svm, ridge, lasso, logistic)

Contents

Numerical results

- Same day stock return series analysis:
  we use the same day stock return series to build the machine learning models. For the Bayesian network, we only use the R package. For the logistic regression, ridge regression, lasso and svm, we used different languages(R and Python) and also compare the CPU time. To test the accuracy rate of model, we choose the first 1000 data as training data and the last remaining 257 data as testing. GS as response(discretized as 1 and -1) and the other 453 stocks as predictors.
- Predict the stock data:
  we used the last one day, two day,... to last five day stock returns as the predictors and today's GS return series as response to see if our model can be used to predict the stockdata.
  Still use the first 1000 data as training and the remaining 252 data as testing. GS as the response and the other 2270 variables as predictors.

Numerical results

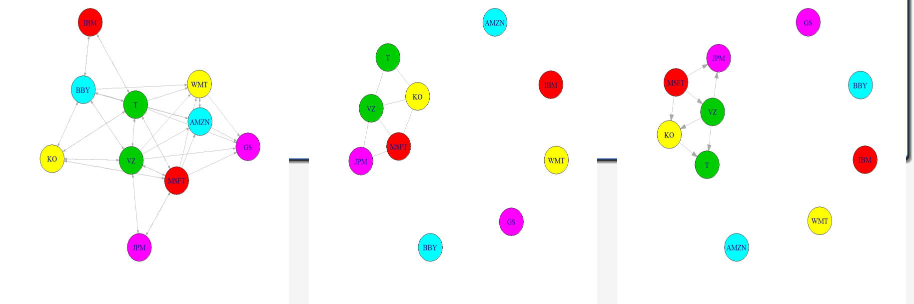
Bayesian network

Table : 10 companies

Stock code	Industry	Company name
GS	Financials	Goldman Sachs Group
JPM	Financials	JPMorgan Chase & Co.
MSFT	Information Technology	Microsoft Corp.
IBM	Information Technology	International Bus. Machines
T	Telecommunications Services	AT&T Inc
VZ	Telecommunications Services	Verizon Communications
WMT	Consumer Staples	Wal-Mart Stores
KO	Consumer Staples	Coca Cola Co.
AMZN	Consumer Discretionary	Amazon.com Inc
BBY	Consumer Discretionary	Best Buy Co. Inc.

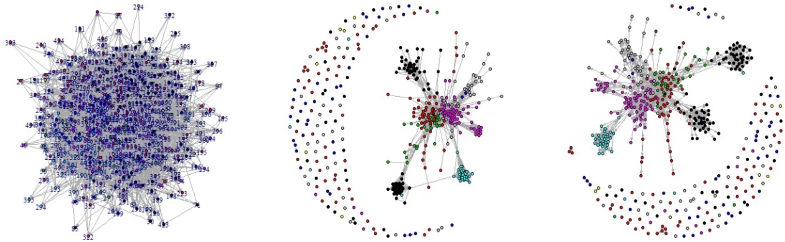
Numerical results

Results for simple cases



Numerical results

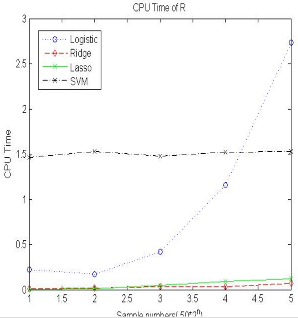
Results for total stocks



Numerical results

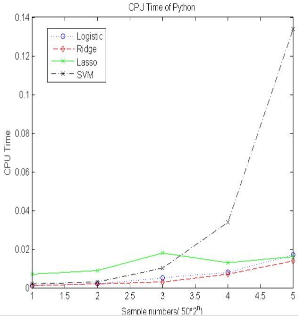
CPU Time for R

We changed the number of samples from 50 to 800, doubled each time to test the running time for the different machine learning methods:



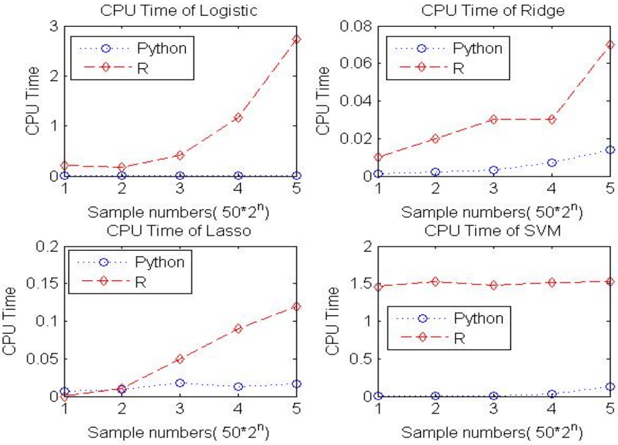
CPU Time for Python

We changed the number of samples from 50 to 800, doubled each time to test the running time for the different machine learning methods:



Numerical results

Comparison of CPU time for python and R



Numerical results:

Accuracy rate:

Table : Accuracy rate

Methods	Python	R
Logistic	69.8%	68.4%
Ridge(λ=1)	73.9%	77.4%
Lasso(λ=0.01)	78.6%	79.0%
svm(linear)	72.4%	71.8%
svm(poly)	74.3%	71.2%
svm(rbf)	75.1%	72.8%

Results:

- lasso (λ=0.01) and svm (rbf) performed good for both two languages.
- Python performed better in most cases but the difference is not

Numerical results:

Predicted

$$R_t^{GS} = \sum_{i=1:5} \sum_{j=1:454} \beta_{i,j} R_{t-i}^{j} \tag{1}$$

Predict:

Table : Accuracy rate and CPU time

Methods	Accuracy rate	CPU time
Logistic	51.2%	0.1210
Ridge(λ=1)	54.0%	0.1230
Lasso(λ=0.01)	49.2%	0.0940
svm(linear)	52.8%	1.1931
svm(poly)	45.6%	1.2800
svm(rbf)	47.2%	1.2921
Bayesian Glasso	51.6%	around two hours

Contents

Future work

- Compare with the time series model, such as Garch(machine learning methods can consider the whole economic environment while time series cannot)
- Deal with the high frequency data instead of daily data

Contents

QA

Thanks a lot and Questions

Contents

Title

Each frame should have a title.

Without title somethink is missing.

A frame

Some text

A frame

Some te

A block hovering above the slide

I am visible on slide two.

Contents

unnumbered lists

- Introduction to LaTeX
- Course 2
- Termpapers and presentations with LaTeX
- Beamer class

lists with pause

- Introduction to LaTeX

lists with pause

- Introduction to LaTeX
- Course 2

lists with pause

- Introduction to LaTeX
- Course 2
- Termpapers and presentations with LaTeX

lists with pause

- Introduction to LaTeX
- Course 2
- Termpapers and presentations with LaTeX
- Beamer class

numbered lists

1. Introduction to $\LaTeX$
2. Course 2
3. Termpapers and presentations with $\LaTeX$
4. Beamer class

numbered lists with pause

1. Introduction to LaTeX

numbered lists with pause

1. Introduction to LaTeX
2. Course 2

numbered lists with pause

1. Introduction to LaTeX
2. Course 2
3. Termpapers and presentations with LaTeX

numbered lists with pause

1. Introduction to LaTeX
2. Course 2
3. Termpapers and presentations with LaTeX
4. Beamer class

Contents

Tables

Date	Instructor	Title
WS 04/05	Sascha Frank	First steps with LaTeX
SS 05	Sascha Frank	LaTeX Course serial

Tables with pause

A	B	C

Tables with pause

A	B	C
1	2	3

Tables with pause

A	B	C
1	2	3
A	B	C

Contents

blocs

title of the bloc

bloc text

title of the bloc

bloc text

title of the bloc

bloc text