Boosting method for capturing high frequency limit order book dynamics

Jian Wang jw09r@my.fsu.edu Florida State University

Contribution

More than half of the markets in today's fast-paced financial world now use a limit order book(LOB) to facilitate trade(Rosu 2009). Therefore, capturing the dynamics of high frequency LOBs become a challenging task for our financial engineering researchers. We propose boosting machine learning methods to predict future spread crossing opportunities. Our main contributions are concluded as follows:

- Use boosting methods to improve the performance for weaker classifier.
- Predict the future spread crossing opportunities based on fixed time interval.
- Compare the accuracy rate and computation time among different machine learning methods

Definition

Definition 1 *An order* $x = (p_x, \omega_x, t_x)$ *submitted at time* t_x *with price* p_x *and size* $\omega_x > 0$ *(respectively,* $\omega_x < 0$*) is a commitment to sell (respectively,buy) up to* $|\omega_x|$ *units of the traded asset at a price no less than (respectively, no greater than)* p_x*.*

Definition 2 *An LOB* $\mathcal{L}(t)$ *is the set of all active orders in a market at time t.*

Definition 3 *The bid price at time t is the highest stated price among active buy orders at time t,*

$$b(t) := \max_{x \in \mathcal{B}(t)} p_x$$

The ask price at time t is the lowest stated price among active sell orders at time t,

$$a(t) := \min_{x \in \mathcal{A}(t)} p_x$$

Definition 4 *The bid-ask spread at time t is* $s(t) := a(t) - b(t)$*. The mid price at time t is* $m(t) := [a(t) + b(t)]/2$

Definition 5 *Bid higher spread crossing at time t after* Δt *time interval is* $P^{Bid}_{t+\Delta t} > P^{Ask}_t$*, ask lower spread crossing at time t after* Δt *time interval is* $P^{Ask}_{t+\Delta t} < P^{Bid}_t$*, no spread crossing at time t after* Δt *time interval is* $P^{Bid}_{t+\Delta t} <= P^{Ask}_t$ *and* $P^{Ask}_{t+\Delta t} >= P^{Bid}_t$

Data Set

Our dataset contains limit order prices on 2012-06-21 of four stocks from NASDAQ (Apple,Amazon,Intel and Microsoft). For each stock, it divided into two major components: the message book and the order book.

- Message book: contains time when the order book came, prices,volumes, event types and directions.
- Order book: contains price levels, prices and volumes in each level for every event.

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

Snapshots of LOBS

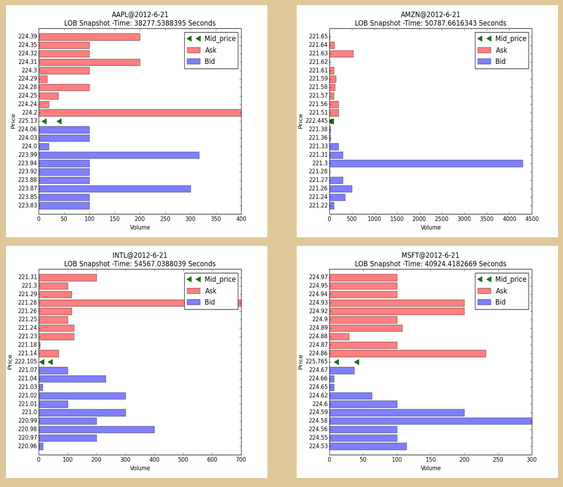

Figure 1 shows the snapshot of 10 level ask and bid prices of four stocks. The red bars show the ask prices, the blue bars represent the bid prices and the green triangle is the mid price.

Conclusion and future works

1.Boosting methods performance better than other machine learning models in this case
2.Can use high performance method such as parallel computing to improve the speed
3.Can test the Profit and Loss(PNL) which traders mainly concern

Boosting methods

Boosting method is a procedure that combines the outputs of many "weak" classifiers to produce a powerful "committee", it was originally designed for classification problems and later extended to regression.

AdaBoost

AdaBoost, formulated by Yoav Freund and Robert Schapire, is the most popular boosting algorithm, it puts more weights on the false classification data:

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

Gradient boosting

Gradient boosting uses gradient descent methods to minimize the residual in each step:

Algorithm:

1 Initialize $f_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$;
2 **for** m=1 to M **do**
  for i=1,2,...,N **do**

$$r_{im} = -[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}]_{f=f_{m-1}}$$

  Fit a regression tree to the targets r_{im} giving terminal regions R_{jm}, j=1,2,...,J_m;
  for j=1,2,...,J_m **do**

$$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$

  Update
  $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$;
3 Output $\hat{f}(x) = f_M(x)$

Numerical results for AAPL

We try to predict the future 5 seconds spread crossing situation. Training samples are 8000 and test samples 2000. The parameter for adaboosting and gradient boosting are: depth =3 and iterations =500. Computer is 8G memory and Intel Xeon E3 processor(4 cores), the results are listed in the following table

Methods	Accuracy rate	CPU time
Logistic	57.15%	0.06
Ridge(alpha=1)	59.83%	0.01
Lasso(alpha=0.001)	60.75%	16.54
SVM	60.15%	4.04
Decision tree	51.40%	0.24
Ada Boosting Tree	72.90%	30.19
Gradient Boosting Tree	71.95%	14.53

Table 1 shows the predicting accuracy and computation time for AAPL data under different machine learning methods.

Numerical results for AMZN

We applied the same parameters that used in AAPL to test AMZM data

Methods	Accuracy rate	CPU time
Logistic	66.97%	1.75
Ridge(alpha=1)	64.32%	0.01
Lasso(alpha=0.001)	65.60%	16.60
SVM	64.54%	6.19
Decision tree	56.01%	0.26
Ada Boosting Tree	74.03%	29.40
Gradient Boosting Tree	77.23%	14.4

Table 2 shows the predicting accuracy and computation time for AMZM data under different machine learning methods. We can also see that the performances for AMZN are better than for AAPL to some extent.

References

[1] Alec N.Kercheval,Yuan Zhang Modeling high-frequency limit order book dynamics with support vector machines In *Quantitative finance 2014*
[2] Rosu,I., A dynamic model of the limit order book. In *Rev.Financ.Stud.,2009,22,4601-4641.*
[3] Trevor Hastie, Robert Tibshirani, Jerome Friedman The Elements of Statistical Learning: Data Mining, Inference, and Prediction,Second Edition