**E:\1Polyu\year4sem1\LE3\polyu-logo.pngE:\1Polyu\year4sem1\LE3\polyu-comp-logo.png**

**COMP4433 Data mining and Data warehousing**

**Group Project report**

**Utilization of mining and modeling skills in stock market**

**Instructor: Dr. Korris Chung**

**Group members:**

**WANG Jianxun 12130815D**

**ZHENG Haoyu 13102597D**

**GENG Xu 12132031D**

**2016/4/9**

**Abstract:**

Stock market is a typical occasion to apply data analytic skills for profitable decision making. More and more organizations are developing their own auto-stock trading applications to make money based on those analytic skills. This project aims at starting some basic explorations in this field.

Data mining and machine learning are two overlapping and effective methodologies in knowledge discovery, especially in the era of big data. Data mining algorithms come up with interesting and useful rules and patterns. Machine learning, as a sub-topic of data mining, focus on providing models for existing data. Lots of works has been done in those fields for discovering the secret behind stock market.

In part 1 of this report, we used several data mining algorithms to come up with interesting trend patterns from the stock data.

In part 2 of this report, some machine learning technics are utilized to provide models for stock data based on different tuples and features.

**Part 1. Mining Interesting Trend Pattern from Stock Data**

Contents

[1. Introduction 3](#_Toc447898845)

[2. General Data Preprocessing 3](#_Toc447898846)

[3. Flow mining 3](#_Toc447898847)

[3.1 General Description 3](#_Toc447898848)

[3.2 Data preprocessing 4](#_Toc447898849)

[3.3.1 First Trial 4](#_Toc447898850)

[3.3.2 Second Trial 5](#_Toc447898851)

[3.3.3 Third Trial 6](#_Toc447898852)

[4. Price Rate of Change (ROC) mining – ARM & C5.0 7](#_Toc447898853)

[4.1 General Description 7](#_Toc447898854)

[4.2 Data Preprocessing 8](#_Toc447898855)

[4.3 Inter-Stock Mining targeting ROC and stock flow 8](#_Toc447898856)

[4.4 Intra-Stock Mining targeting ROC and stock flow 9](#_Toc447898857)

[4.5 Intra-Stock Mining targeting ROC 9](#_Toc447898858)

[4.5.1 ARM mining of ROC flow using previous binned ROC 10](#_Toc447898859)

[4.5.2 ARM mining of binned ROC using previous binned ROC 10](#_Toc447898860)

[4.5.3 C5.0 mining of ROC flow using previous ROC real value 11](#_Toc447898861)

[5. Close Price deviation mining - ARM 11](#_Toc447898862)

[5.1 General Description 11](#_Toc447898863)

[5.2 Data Preprocessing 11](#_Toc447898864)

[5.3 Parameters setting & Interesting Findings 12](#_Toc447898865)

[5.4 Possible Improvement 12](#_Toc447898866)

[6. Direct Close Price mining – C5.0 13](#_Toc447898867)

[6.1 General Description 13](#_Toc447898868)

[6.2 Data Preprocessing 13](#_Toc447898869)

[6.3 Findings in Inter-Stock Mining 13](#_Toc447898870)

[6.4 Findings in Intra-Stock Mining 14](#_Toc447898871)

## 1. Introduction

The project focus on the data processing, analyzing and mining on the stock market from 2000 to 2007. Using several mechanisms and algorithms, we aim to find some rule or principles that can help us better understand and depict the stock market data.

Our team utilized the IBM SPSS Modeler as the main analyzing tools. The algorithms and mechanisms we used are pre-built functions provided by the software.

In the project, we dig deeply inside a single stock and also the whole stock market, which contains several stocks. Under the seemingly messy data, we have got a lot of useful information. The following paragraphs will discuss these in different aspects.

## 2. General Data Preprocessing

The first general Data Preprocessing consists of noisy filtering. The definition of noise in this project is those records of trading days with each stock has zero trading volume. Such records may indicate the suspension of trading on those days and can have significant impact on the later mining result. In order to keep the significance of each stock and their variation in price, the stock prices are not normalized.

In addition, since there are several missing records at the beginning of stock 857, the missing records are all filled with value 0. The reason of adopting this method is because it is probably that the stock 857 was not open for trading at those days and it was only available after April 7th, 2000. Hence, in order to indicate such phenomenon, the value 0 is chosen to fill in the blank.

Finally, there are several columns left blank in order to visually separate the data records. These columns are disregarded at the beginning for the purpose of simplicity.

## 3. Flow mining

### 3.1 General Description

In the process of data processing, we included an important factor named flow. Flow is an indicator that illustrates whether today’s close price is higher than yesterday’s. Flow is a categorical feature.

For stock buyers and sellers, flow, to some extent, could be a helpful feature. If one knows today’s possible flow before today’s market begins, then he or she may have a basic idea of the stock price and is more likely to make a wise decision on buying or selling.

### 3.2 Data preprocessing

The value of flow is decided as follow:

Flow = up if today’s close > yesterday’s close;

Flow = down if today’s close < yesterday’s close;

Flow = level if today’s close = yesterday’s close;

In this trial, we generate each day’s flow.

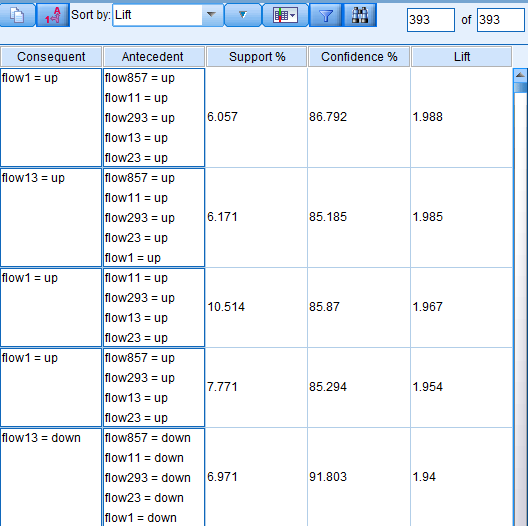
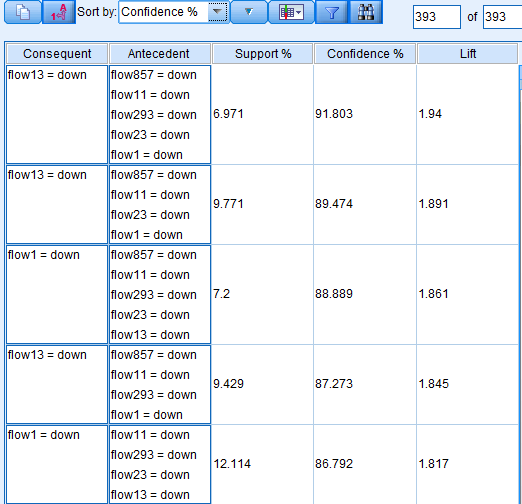
In flow mining, the main mechanism we took is ARM Associate Rule Mining. We planned to utilize ARM to see whether the flows we obtained can give us some indications on today’s flow.

### 3.3.1 First Trial

The first trial is to discover the relation of different stock’s flow in the same day.

The SPSS working flow file is “inter\_data\_ARM\_flow\_1day\_rel”. The source file is stock\_all.csv.

The flows of each stack are both input and target of the ARM mechanism.

Setting the threshold at 5% support and 60% confidence, we found 393 rules meeting the criteria.

Among those rules, there are several that worth special attention. They have a relatively high confidence (usually more than 80%) and relatively high lift.

We can see that for these rules, when the flows of other 4 or 5 stocks (antecedent) are up/down, it is of great possibility that the consequent flow is up/down. This may imply some inner correlation between the different stocks.

*We may make a hypothesis that the stock market has a general “environment”. When the “environment” tends to be “up” in terms of flow, then the flow of a single stock may be more likely to be “up”, and vice versa. Flow of others stocks forming the “environment” could be seen as a predictor of one stocks’ flow.*

Having found the rules of relations of different stocks’ flow in one day, once we have the information of other’s stocks situation for one day, we may predict the aiming stock’s flow.

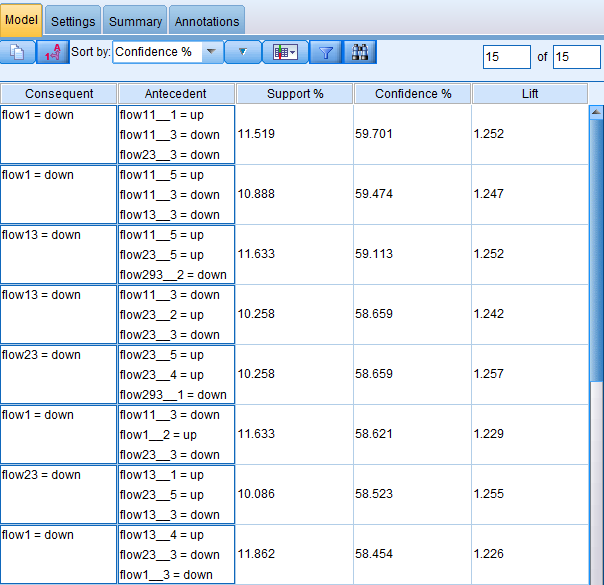
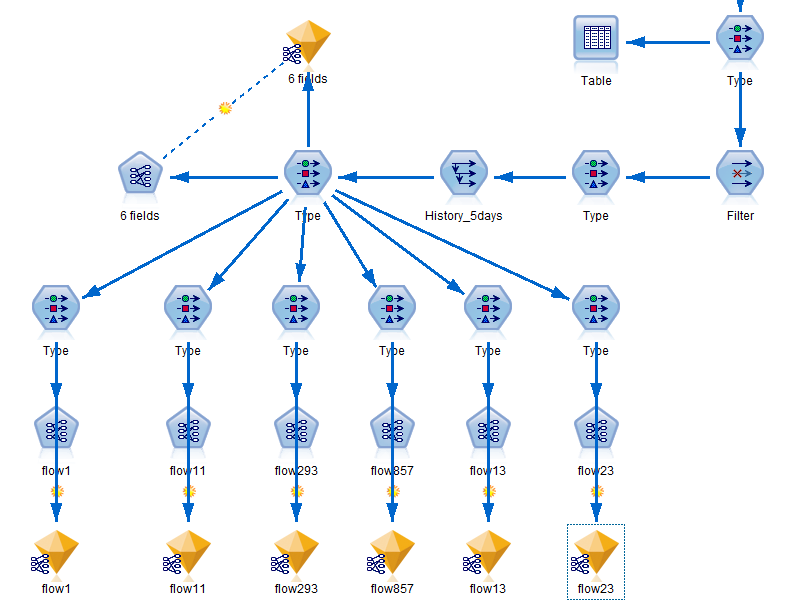
### 3.3.2 Second Trial

The second trial is to use the past 5 days’ flow to check if there are any rules that points to todays’ flow.

The SPSS working flow file is “inter\_data\_ARM\_flow\_5day\_predict”. The source file is stock\_all.csv.

In this trial, the prediction features used as input include the flows of the past 5 days. The “flow num1\_num2” means the flow of stock ID num1 in the day num2.

We generate the past 5 days’ flow for every day. We designed 7 sub-streams in this trial, which is taking all the flows as input and output and taking 6 different single stocks respectively. So in summary we got 7 output set with a lot of rules found.



In the general “all-include” sub-stream, we set the threshold at 10% supports and 58% confidence and finally got 15 rules.

In these rules we can find that there is some frequent set that occurs a lot in the whole data source. Flows of other stocks in the past 5 days can be also used to predict the flow of the aiming stock today.

These rules are comparatively more direct for the buyers and sellers as they can see the possible flow of one stock from the existing information (flows for the past 5 days). To paraphrase, the flows of other stocks in the past 5 days may instruct them.

However, in the rules we found, the maximum confidence is no more than 60, which means that the confidence is still not very satisfying. Also, most lifts of the rules are approximately 1.2-1.3, which means that the relation of the antecedents and consequent are not so strong.

Besides the output set for the “all-include” sub-stream, we still have 6 sets that take the past 5 days’ flows of the stock itself as the input. In these sub-streams we set the threshold at 5% support and 55% confidence.

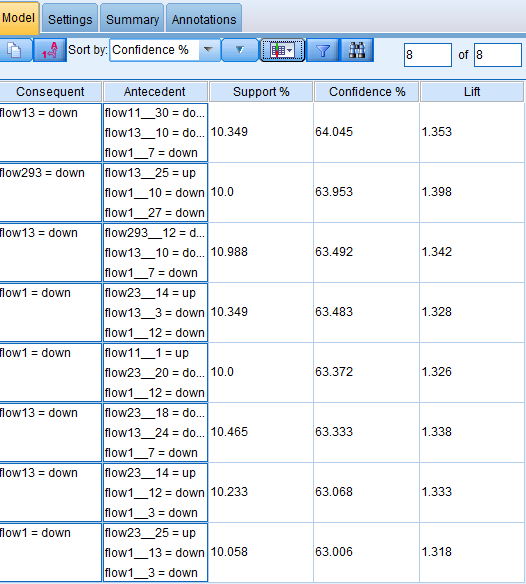
For each output set, we have no more than 10 rules generated. For stock ID 11, we have no rules found with 5% supports and 55% confidence so we lower the confidence criterion to 50% and found 8 rule.

We can conclude from these 6 output set that using the flows of a single stock to predict the flow of it is more difficult. The support and confidence are lower and the rules are fewer. But still we can found some rules that indicates the inner regularity of a stock.

*We may assume that 5 days is too small a time window to observe the variations and trends of the stocks. Rules found with 5 days’ flow could be used as reference, but using them alone to judge the stock market may be risky.*

### 3.3.3 Third Trial

The third trial is to use the past 30 days’ flow to check if there are any rules that points to todays’ flow.

The SPSS working flow file is “inter\_data\_ARM\_flow\_30day\_predict”. The source file is stock\_all.csv. The “flow num1\_num2” means the flow of stock ID num1 in the day num2.

In this trial we set the threshold at 10% support and 63% confidence.

Under such threshold we can dig out 8 rules. The lift is between 1.3-1.4, which is also higher than the “5 days” trial. The maximum confidence we get is 64%.

When we lower the confidence threshold to 50%, we get more than 240000 rules.

30 days is a larger time window compared to 5 days. The rules found with 30 days’ flow may be a better description to the stock market.

**4. Price Rate of Change (ROC) mining – ARM & C5.0**

4.1 General Description

The price rate of change is an indicator that measures the percentage change between a stock’s most recent price and its price several periods ago. Its mathematical representation is as follow:

**ROC = (close – close\_N)/close\_N**

* close: today’s close price of a stock
* close\_N: the close price of stock N days ago

Due to the reason that ROC is usually considered as a price momentum indicator or a velocity indicator and it measures the strength of change in prices, ROC is possible to be a strong predictor of price change on a single day.

This part is firstly dedicated to find the potential pattern between ROC and flow of stock price. Both inter-stock mining and intra-stock mining are conducted in order to find the significance of ROC on predicting a single stock price change as well as possible relation between different stocks in this manner. In addition, the relation between a single stock’s ROC and its previous ROC is also examined. Furthermore, in order to evaluate the influence of ROC’s time gap, two different versions of ROC are used which are ROC5 having time gap as 5 days and ROC30 having time gap as 30 days. For the pattern between a stock’s ROC and its previous ROC, only the intra-stock mining is conducted since the relation of different stocks’ correlation can be better explained by more intermediate factors like their flow.

**The code for this part of mining is *inter\_data\_ARM\_PROC5\_var\_predict* and *inter\_data\_ARM\_PROC30\_var\_predict*.**

4.2 Data Preprocessing

The calculation of ROC will be based on the equation in the previous section. The results are a real value and will be discretized using the **binning method**. The binning is based on the mean and standard deviation of those values in order to get statistical meaning in the later mining. For example, if the real value exceeds the mean value by 2 times of standard deviation, it means the ROC indicates not only momentum in price increase but also great severity in increasing rate. In the later report, when “ROC” is mentioned, it is referring to the binned ROC using this method, if there is no other explanation.

Besides the binning method, the real value of ROC is also reclassified in order to indicate the flow of ROC. The reclassification using the following equation:

up, if ROC>0

**ROC\_flow** = level, if ROC = 0

down, if ROC<0

The reason of such additional reclassification is that the binning method may not be able to reflect the flow of ROC in certain cases, due to the mathematical characteristics of mean & standard deviation binning method. It is possible that the ROC actually increases or decreases but still within the moderate level which both phenomenon will be labeled as “level”.

The preprocessing to acquire flow of each stock is the same as the flow mining, hence it is omitted here.

4.3 Inter-Stock Mining targeting ROC and stock flow

Considering inter-stock mining, the consequents are different stocks’ flow in a single day and the antecedents are different stocks’ ROC on the previous day. ARM is used to conduct mining task in which each records of days after preprocessing is treated as a customer. The reason of doing so is that such formulation can better reflect the relation between different stocks. The minimum confidence is set to be 60% in order to get strong rules. In addition, the minimum support is set to be 1% in order to allow the appearance of records with strongly deviated ROC value, since the ROC value is binned using mean & standard deviation method. Another reason for such low support is to include the appearance of rare phenomenon but with considerable value.

For mining using ROC5, there are 287 rules mined. The maximum confidence is 77.78%. In addition, the minimum lift is 1.258 and maximum lift is 1.823. According to the mined rules, each stock’s price flow is variously caused by different stocks’ previous ROC. The antecedent ROC can indicate different momentum with the consequent flow, in which case representing a negative correlation of price between the antecedent stock and consequent stock.

For mining using ROC30, there are 293 rules mined with maximum support being 78.94%, minimum lift being 1.255 and maximum lift being 1.863. The mined rules also show similar variation with the mining using ROC5. There is certain degree of similarity between the rules mined using ROC5 and ROC30. For example, both mined rules containing the following ones:

ROC\_13=down & ROC\_23 = moderate-down 🡪 flow\_293 = up

*Such consistency actually strengthens the possibility of relation between these stocks*. But due to the high volume of mined rules, it is not guaranteed that all mined rules shows such consistency.

From the performance perspective, the mining using ROC5 and mining using ROC30 do not show significant difference. Mining using ROC30 may show slight advantage from the perspective of mined rule number and maximum confidence. However, such weak difference cannot be used as evidence to claim that ROC30 is better than ROC5 when conducting mining between ROC and stock price flow.

**The result of this part mining can be viewed by clicking the output named *inter\_stock ROC & flow* under corresponding source code file.**

4.4 Intra-Stock Mining targeting ROC and stock flow

When doing intra-stock mining regarding ROC and stock flow, the consequents are still the flow of each single stock flow and the antecedents are the ROC of previous 5 days. ARM is used to conduct the mining, treating each day as a customer. The reason of using this method instead of using sequential ARM is to control the time gap between antecedents and consequents. The minimum confidence is set to be 60% and the minimum support is set to be 1%, due to the same reason in inter-stock mining.

The results of mining on each stock show different interesting patterns of past ROC upon stock flow, with confidence from 60% to 80%. The lifts of mined rules are all above 1.2, which indicates strong correlation between historical ROC and current stock flow. The strong statistical support implies that ROC being the hypothesized predictor of current stock flow is possibility true.

There are several other interesting findings. Firstly, using ROC5 and ROC30 does not show significant difference in the performance of such rule mining. But *this does not necessarily mean that the time gap of ROC has no influence upon stock flow prediction*. Further experiment regarding such issue is required. In addition, increasing the number of historical ROC used as features does increase the maximum confidence and number of mined rules. However, the new mined rules all include some of the features from the current mining method, which are the previous five days of ROC. Such finding implies that *the influence of ROC is possibly stronger if the ROC is closer to the current day*.

**The result of this part mining can be viewed by clicking the output name *flow<stock#>* under corresponding source code file.**

4.5 Intra-Stock Mining targeting ROC

This part of mining mainly targets to find the possible patterns in ROC. Both ARM and C5.0 are used to conduct mining task. For ARM mining, the consequents are the ROC of current day and the antecedents are the ROC of previous 5 days. The consequent ROC is represented in two ways which are the direct ROC flow and the mean-standard-deviation binning of ROC. The reason of using the representation is to avoid the situation that there is change in ROC yet the change is not significant, in which case the ROC consequent will have the value of “level”. For C5.0, the classification condition is the real value of ROC and the final label is the flow of current ROC, in order to satisfy the condition of applying decision tree model. The cross-validation with 10 folds is adopted to estimate the accuracy of generated decision tree.

4.5.1 ARM mining of ROC flow using previous binned ROC

Firstly, talking about the ARM mining of ROC flow using binned ROC as antecedent, the performance is considerable. Both mining using ROC5 and ROC30 will produce several hundred rules and many of them have very significant confidence higher than 90%. Most of mined rules with such high confidence have a similar pattern as follow:

ROC of previous N day = up/down 🡪 ROC flow up/down

* “up” in antecedent: ROC is higher than mean by 2 times of standard deviation
* “down” in antecedent: ROC is lower than mean by 2 times of standard deviation

Such rules imply that if the previous ROC is strongly deviated from the mean either positively or negatively, it is highly possible that ROC today will also indicate an increase or decrease in price.

However, not all rules satisfy such pattern. For example:

ROC5\_293\_5 = moderate-up & ROC5\_293\_1 = moderate-down 🡪 ROC5\_293\_flow = down

This rule also has the confidence of 100%, which indicates a very strong and yet interesting rule.

For the performance difference between mining using ROC5 and ROC30, the mining using ROC 30 will produce a significant improvement in result, considering the number of rules mined. It is possibly because *the change rate represented by ROC30 is more salient than ROC5, which leads to the increase of rule mined*.

**The result of this part mining can be viewed by clicking *ROC\_<stock#>\_flow* in the corresponding source code file.**

4.5.2 ARM mining of binned ROC using previous binned ROC

This part of mining using binned ROC as both antecedents and consequent. Due to the reason that it is possible for ROC to increase/decrease but still within the range of standard deviation to mean, many of mined rules has “level” as the consequent value. In fact, when using ROC5 to conduct such mining, most of the consequent will have the value “level” (Hence, this part of mining is removed from the source code of ROC5). However, the mining using ROC30 still produce considerable results that has value other than “level. For the mined result using ROC30, the consequents generally follow the same trend with its antecedents. For example, if the consequent has the value “up”, the antecedents will usually have the value of “up”, “moderate-up” and/or “level”. Such phenomenon is possibly due to *the long term trend representation of ROC30 and long term trend tends to have the same direction in changing*. However, this indicates that *people should pay more attention to the change of direction in ROC, which may imply a future fluctuation in stock price*.

**The result of this part mining can be viewed by clicking *ROC30\_<stock#>\_SDBIN\_Reclassify3* in the corresponding source code file.**

4.5.3 C5.0 mining of ROC flow using previous ROC real value

This part of mining is using the previous ROC real value to predict the current ROC flow label, using a decision tree model. There is a significant similarity between the trained model of different stocks using either ROC5 or ROC30, which is that all of them using the ROC of 1 day before current day as the first layer attribute for partitioning. This phenomenon is consistent with the previous hypothesis that *the influence of ROC is possibly stronger if the ROC is closer to the current day*.

Comparing the performance difference between using ROC5 and ROC30, the decision tree constructed using ROC30 has a higher average mean accuracy reported by cross-validation for each stock. For ROC5, the mean accuracy reported by cross-validation is around 70%. On the other hand, decision trees of ROC30 have the mean accuracy around 90%. Such phenomenon is consistent with the hypothesis that *the long term trend represented by ROC30 makes is a more stable predictor on itself*.

**The result of this part mining can be viewed by clicking *ROC\_<stock#>\_flow* in the corresponding source code file.**

**5. Close Price deviation mining - ARM**

5.1 General Description

Based on the theory of mean reversion, which indicates that the stock price will ultimately return to its historical mean, it is hypothesized that the deviation of close price of previous days can be potential predictors for the close price of current day. However, it is also mentioned in the theory that it is impossible to predict when the return phase will begin. Hence, it is understandable if the finding does not consistent with the theory itself.

This part of mining generally focuses on using ARM to find the potential pattern between the deviation of previous close price and flow of current day. In this part, only the intra-stock mining is conducted due to the lack of evidence indicating the relation of close price variance between different stocks.

**The code for this part of mining is *inter\_data\_ARM\_mean\_reversion*.**

5.2 Data Preprocessing

The close price of each stock will be binned using mean-&-standard-deviation method. Such method will directly indicate the location of such close price in the normal distribution of overall historical close price. The 2-time of standard deviation method is adopted since it best reflects the distribution of close price comparing to the 1-time version and 3-time version, which the 1-time version is too general and 3-time version has labels with incredibly small support.

The generation of flow is the same as Flow mining, which is omitted here.

5.3 Parameters setting & Interesting Findings

ARM is adopted attempting to find potential trend patterns. The consequents are the stock flow of current day and the antecedents are the binned close price of previous 15 days. The minimum confidence is set to 60% in order to find strong rules and the minimum support is set to 1% in order to allow the appearance of close price with deviation larger than 2-time standard deviation.

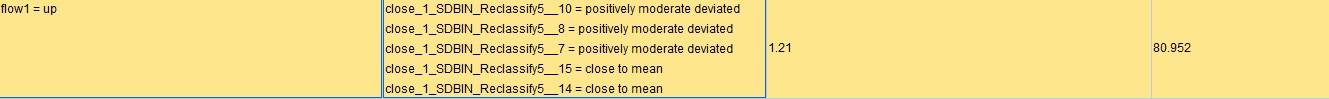
Talking the performance of such rule mining, the mining on all stocks can produce several hundreds of rules, except stock 857. Most of the rules mined for stock 857 contains the consequent value as “level”. This phenomenon is possibly introduced by *the inclusion of period that stock 857 has not yet been available for trading*. After exclusion of these rules, the number of rules are 47. Such low number of rules mined is also possibly influence by the period mentioned above, which *severely decrease the overall historical close price mean of stock 857*.

The mining on other stocks generates rules with maximum performance from 70% to 80%, maximum lift from 1.7 to 1.8 and minimum lift above 1.2. Disregarding the validation of hypothesis mentioned in the general description, *the deviation of previous close price can be considered useful predictor of stock flow on current day*.

Considering the rules mined, they show great variation with no definite support for the mean reversion theory. There are rules consistent with the claim from mean reversion theory, for example:

P:\COMP4433\Report\mean_reverse_1.jpg

However, there are also rules indicate the reverse of such theory, for example:



As mentioned, it is hard to predict the occurrence of reversion phase. Hence, such phenomenon actually does not disapprove the validity of mean reversion theory.

5.4 Possible Improvement

The performance of using close price deviation can be improved by using the Moving Average instead of using overall historical mean. The reason is that it is highly unlikely that a company’s stock price will always has the same mean. When a company’s performance increases, it is understandable that they can have a higher stock price in the future. By applying Moving Average to calculate the close price deviation, the antecedents can have a better simulation upon the current trend of overall stock price and thus becoming a more efficient predictor. Similar indicator of stock price currently adopted are Percentage Price Oscillator (PPO) and Moving Average Convergence Divergence (MACD).

**6. Direct Close Price mining – C5.0**

6.1 General Description

In 1965, Paul Samuelson proposed that stock market is most likely martingales, which means the best prediction of stock price on next day is the stock price on current day. Based on this theory, it means the most efficient predictor of close price on current day are possibly the features of 1 day before this day. By adopting this hypothesis, this section using the open price, the highest price, the lowest price, close price and volume of previous day to predict the close price of current day. Both inter-stock mining and intra-stock mining have been implemented to exam the efficiency of using such features as predictors.

**The code for this part of mining is *inter\_data\_class\_martingle*.**

6.2 Data Preprocessing

Since C5.0 is used to conduct mining, all previous day’s features are kept as real value in order to keep the originality and prevent removing the significance of real value’s boundary. For the classification label as the close price of current day, it is binned using mean-&-standard-deviation method to get its location in the normal distribution. Such method can mostly keep the originality of data and make it possible to conduct decision tree construction.

6.3 Findings in Inter-Stock Mining

The statistical attribute of constructed decision tree is as follow:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stock # | 1 | 11 | 293 | 857 | 13 | 23 |
| Tree depth | 7 | 10 | 3 | 6 | 7 | 2 |
| Mean accuracy | 95.3% | 93.8% | 96.8% | 97.5% | 96.9% | 98.1% |

As indicated from the table, the mean accuracy from 10-fold cross-validation are all quite high, indicating that the features from previous day can be efficient predictors. One interesting findings is that all 6 constructed decision trees choose the close price of previous day as attribute in the first layer to conduct further partitioning. The importance of such factor varies from 0.67 to 0.97. This finding actually consistent with empirical experience that *there is rarely strong fluctuation of stock market between two days*. In addition, *the importance of previous day’s close price in the constructed decision tree may imply the stability of stock price in a long term view*.

6.4 Findings in Intra-Stock Mining

The statistical attribute of constructed decision tree is as follow:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stock # | 1 | 11 | 293 | 857 | 13 | 23 |
| Tree depth | 4 | 6 | 3 | 4 | 4 | 4 |
| Mean accuracy | 95.3% | 93.0% | 96.2% | 98.1% | 96.6% | 97.7% |

Comparing the findings in intra-stock mining and inter-stock mining, it can be seen that the difference in accuracy is limited. *Some of the difference in accuracy seems to be the result of cross-validation which using different fold for training and testing each time*. For example, the constructed trees for stock 293 in inter-stock mining and intra-stock mining are exactly the same. The inclusion of other stocks’ feature can increase the mined tree performance, like stock 23, or decrease the performance, like stock 1. From the constructed decision tree, it can only be concluded that *some stock has stronger influence on itself while others may be affected more by other stocks*. The reason of such phenomenon can be discovered by examining the relation between the companies of these stocks.

**PART 2. Utilization of machine learning models in stock data**

**1. Introduction:**

This project aims at carrying out stock market analysis using various machine learning methods in order to help investors maximize their profit during stock investment.

First of all, Hidden Markov Model (HMM) is utilized as a representative of ‘pure computer science’ method that relies only on close data of each day, without heuristics from financial or business industry. After fitting a discrete HMM or continuous HMM with Gaussian Mixture Model (HMM-GMM), we can do 2-class classification(up/down) or regression(prediction on close price), correspondingly.

Secondly, Support Vector Machine(SVM) is used as classifiers that employ industrial heuristics – stock indicators.

**2. Data Preprocessing:**

Both methods use the data source that been preprocessed in following way:

1. Calculate all indicators (as listed in SVM method section) from original data.

2. Normalize each tuple (open, close, volumn, indicator values etc…) of each stock to be mean = 0 and standard deviation = 1 to eliminate inter-stock, inter-tuple and inter-indicator variance.

**3. HMM:**

Under the circumstance of time-sequence modeling, HMM hold assumption that p(n+1) is dependent on p(n), which determines a markov chain, to simulate a time-sequence model. Compared with assumption made by bayes model, which grant the independence between p(n+1) and p(n) (under the circumstance of time-sequence modeling), I want to make the hypothesis that the dependence assumption by HMM is more reliable, according to my intuition. So I prefer HMM (discrete and continuous) rather than bayesian classification for fitting time-series stock data.

**3.1 Basic operation:**

3.1.1 HMM train:

In order to fit an HMM model(no matter discrete or continuous), a series of data need to be provided.

Parameter:

1. O: observation, number of features, in this case, o=1 stands for only use close value.

2. S: number of hidden state, tunable

3. T: window size, tunable

4. Prior probability vector,Transition matrix and emission(observation) matrix are randomly initialized using o and s.

Continuous HMM train:

Apart from parameters above, emission matrix can not be initialized because it follows the Gaussian mixture distribution(GMD). Additional parameter need to be provided for fitting such a distribution:

M: number of Gaussian mixture

So that the GMM will be modeled and further be used to model continuous HMM.

Finally, HMM is trained using EM (Baum Welch) algorithm.

3.1.2 HMM classification/regression

HMM don’t provide directly provide classification/regression result. Instead, it provide loglike probability for each testing sequence. We can do classification according to its argmax (max likelihood).

For example, under a test case of window size w=5, known sequence s=[1 2 3 4] and regression value domain [1,10], classification label domain[1,2,3,..,9,10].

Classification analysis should be:

Loglik[1] = loglik(DHMM,[s 1])

Loglik[2] = loglik(DHMM,[s 2])

…

Loglik[10] = loglik(DHMM,[s 10])

result = argmax(loglik)

Regression is derived from classification, with a fixed discretizer for to transfer continuous data to categorical (discrete) data.

Regression analysis should be:

Discretizer = 100;

Testvalues = discretize(value domain,discretizer)

For all testvalues

Loglik[n]=loglik(MHMM,[s testvalues[n]])

result = argmax(loglik)

**3.2 Preprocessing**

Apart from preprocessing specified before for normalization, we also need to split training/testing dataset and prepare data for classification/regression.

3.2.1 Split training/test set:

Split dataset to training set (80%) and test set (20%) at 80% percentile of date.

3.2.2 Training/test accuracy:

For both training and test data, we will classify or regress the corresponding attribute of each day, assuming the previous t days records are known (from training and testing set), which is identical to practical situation. The detailed algorithm is to adapt HMM classification/regression on a moving window with step size 1 on corresponding dataset.

**3.3 Experiment 1: Discrete HMM, binary classifcation.**

Input: binary sequence which stands for stock going up/down compared with close price of previous day.

Observation: O = 2 (binary, 0 and 1)

Tunable parameter:

Window size: T = 5

Number of state: S = 15

Accuracy:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stock | Stock1 | Stock11 | Stock13 | Stock23 | Stock293 | Stock857 |
| acc\_train | 0. 5828 | 0.6003 | 0.5763 | 0.5681 | 0.5893 | 0.5849 |
| acc\_test | 0.4919 | 0.5368 | 0.5259 | 0.5014 | 0.5177 | 0.5226 |

**3.4 Experiment 2: Discrete HMM, discretized classification to simulate regression.**

Input: discretized earn value:

Earn(t) = (close(t)-close(t-1))/close(t-1)

Reason of failure:

According to statistical plotting, we can find that the input sequence value roughly forllows gaussian distribution. So that the frequency of the label stands for the mean of the input data is dominating. So the classification result will be highly biased that the average label will be dominant. So that all our regression result using discrete HMM will be the average earn value.

**3.5 Experiment 3: Continunous HMM, sequence regression**

Parameters:

Number of observation O=1, only the close value.

Tunable parameters:

To get the best combination of [s,m,t] ([number of hidden state, number of mixture in GMM, window size]), a number of combinations are tested under the close value of stock 1 and evaluation methods below being carried out. Finally I pick up few combinations:

S = 35(fast)/ 50(accurate)

M = 15

T = 3

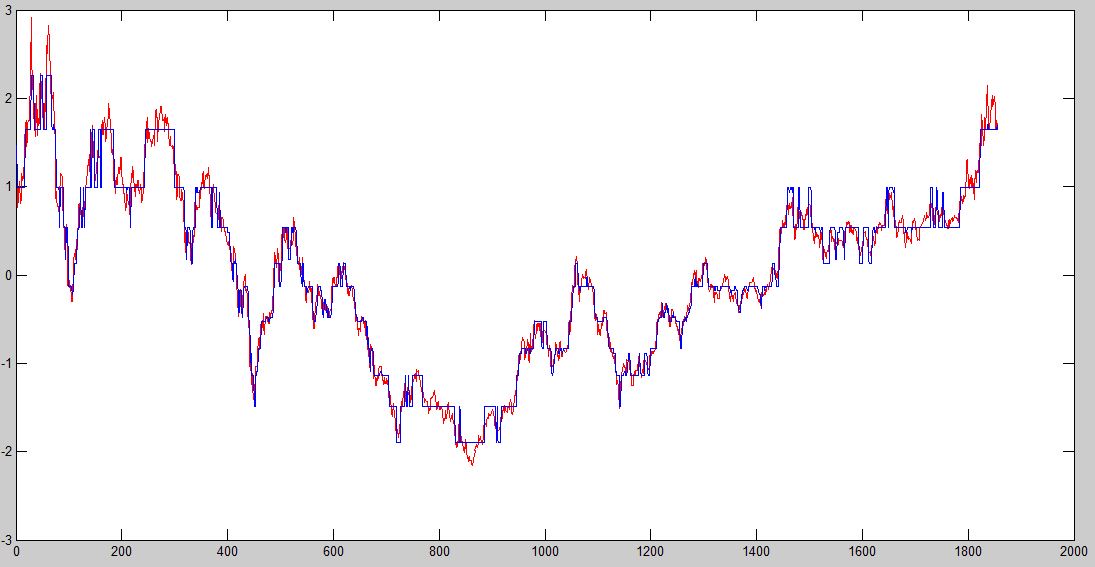
Discretizer on prediction = 100 (100 partition of domain value)

For T=3, prediction is based on previous 2 days.

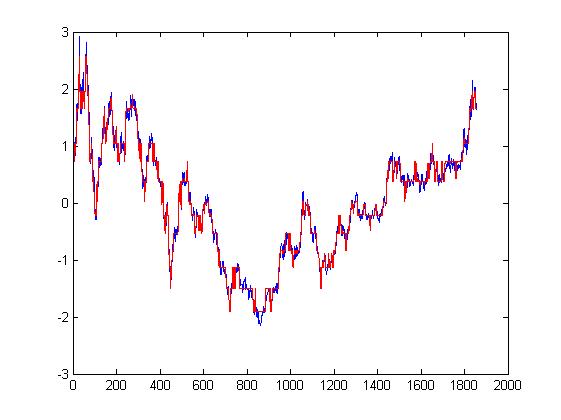
Result by MSE:

S=35;M=15,T=3 (result.mat)

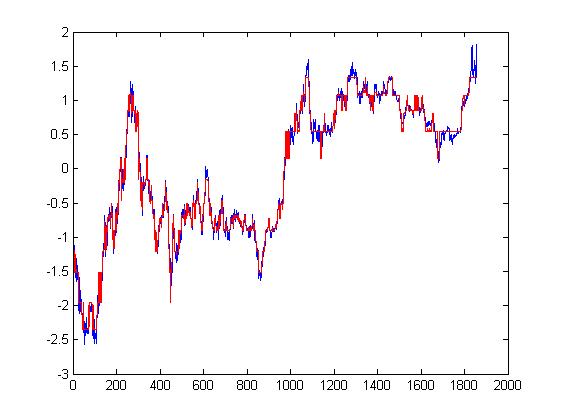
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stock | Stock1 | Stock11 | Stock13 | Stock23 | Stock293 |
| acc\_train | 0. 2227 | 0.1689 | 0.1502 | 0.0853 | 0.2059 |
| acc\_test | 0.1257 | 0.1985 | 0.1207 | 1.5197 | 0.6257 |

stock 1 S=50, M=15, T=3 (result.mat)

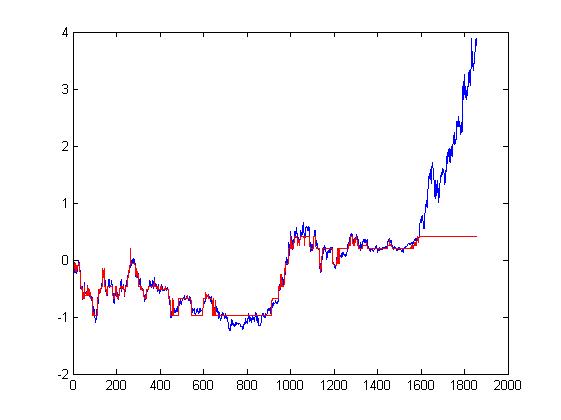
stock 1 S=50, M=10, T=3 (res5010.mat) err\_train:0.1345 err\_test:0.1138 (result.mat)



stock 11 S=50, M=10, T=3 (res5010.mat) err\_train: 0.1253 err\_test:0.1138 (result.mat)



stock 23 S=50, M=10, T=3 (res5010.mat) err\_train:0.0885 err\_test:1.5197 (result.mat)



For more test cases and results, please refer to part2code/HMMall.

Program source code:

experiment 1 and 2: stock\_dhmm.m

experiment3: stockmhmm\_trainer.m and stock\_mhmm.m

**4. SVM:**

**4.1 Introduction:**

The mechanism of SVM is basically to separate different classes using hyperplanes, which are determined most likely by support vectors that on the borders on classes. SVM need a input of n-dimensional features and class labels to determine the inter-class boundaries.

In this SVM experiment, I tried to import some industrial heuristics. I made the assumption that some indicators, as been widely used by stock analytics, can help a lot to enhance the performance of the classifier.

**4.2 Features:**

The features of SVM training and testing are indicators that been calculated from raw data (open, close, high, low, volume). There are totally 16 features and they are listed below:

WILLR, ROCR3/12, MOM1/3, EMA6/12, MACD, OBV, RSI6/12, ATR14, MFI14, CCI12/20, TRIX

(For detail calculation, please refer to part2code/stock-allindicator.xlsx)

**4.3 Labels:**

The label is binary (0 and 1) which indicates the trend of close value of next transaction day, compared with close value of today:

The label ensures that, in real case, the features(indicators) of transaction day p is calculated from previous days but label(p) is unseen.

**4.4 Training/Test data seperation**

After calculating features and labels, all records of each transaction day are considered as samples, rather than time series. So the sepration of training and test set is a randomsplit of original data source.

**4.5 Dimension reduction – PCA**

Since there are totally 16 features with only no more than 1850 records available, the SVM won’t converge in limited iterations practically, which requires reduction on feature dimensions.

In this experiment, we choose first 8 eigenvectors for dimension reduction to represent at least 90% of original feature space. Correspondingly they are:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stock | Stock1 | Stock11 | Stock13 | Stock23 | Stock293 | Stock857 |
| Variance | 0.9063 | 0.9027 | 0.9118 | 0.9124 | 0.9035 | 0.8996 |

**4.6 Testing accuracy:**

Five experiments, totally different training/testing split and svm models, are done for each stock data to eliminate random cases, the averaged accuracy on test set shows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stock | Stock1 | Stock11 | Stock13 | Stock23 | Stock293 | Stock857 |
| accuracy | 0.5285 | 0.6860 | 0.7115 | 0.6851 | 0.6813 | 0.6918 |