Assignment 2 : Predicting Price Movement of the Futures Market

Yen-Chuan Liu
yel001@ucsd.edu

Jun 2, 2015

1. Dataset Exploratory Analysis

Data mining is a field that has just set out its foot and begun to sprout. Data analytics is particularly important in the financial field. For this assignment, I will perform analyze 1-minute-interval trading data of from 2000/1/1 to 2015/04/01 of TXF (symbol/futures name) from TAIFEX Futures Exchange.

This is also known as the bar chart data. Each bar indicates the prices of trades that happened in the time interval (1-minute in this case). Example is shown in fig. 1. (shown as candle bar).

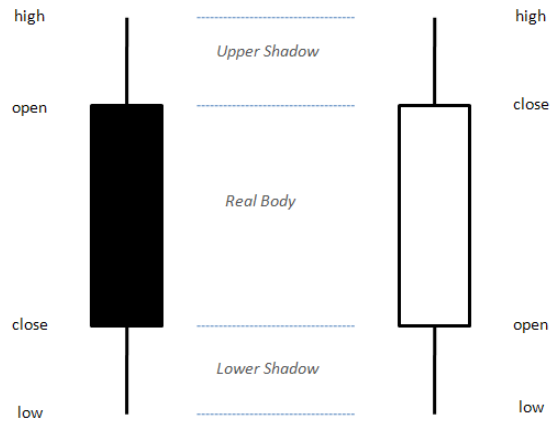


Fig 1. Candle bar example

Each bar (data point) contain the opening price, highest price, lowest price, closing price, and volume of that 1-minute interval.

Example data point:

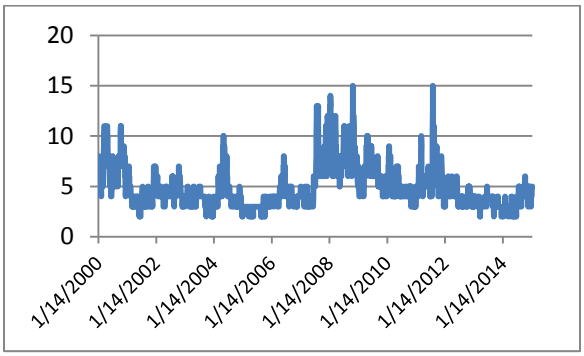
"Date","Time","Open","High","Low","Close","TotalVolume"

1/2/2014,08:45:00,8644,8648,8644,8646,378

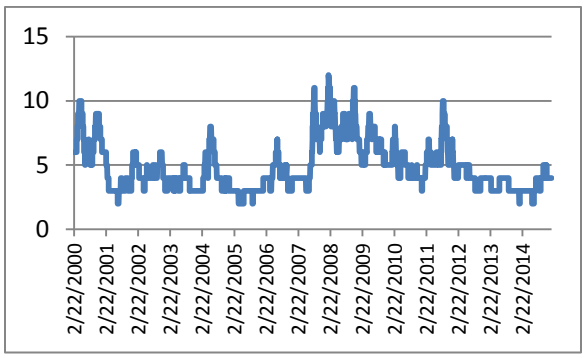
This is the trading data from TAIFEX and TAIFEX futures market is open from 8:45AM to 13:45PM (5 hours) during weekday. Therefore, each day there are 5*60 = 300 1-minute trading data. The dataset contains 1,100,003 datapoints. The unit of the dataset is considered "point". If the price rose from 9000 to 9001, it is considered risen 1 point. Here I perform some exploratory analysis on the dataset in several aspects.

First, I analyze the data in aspects of bar length, this is to understand the how "active" the price movement is. The average bar length of all data (highest-lowest): 4.8. The average bar length of all data (open-close): 2.62. Open-close average is less than highest-lowest average is expected because open-close is in the range of highest-lowest. Here, I continue the analysis with moving average.

Average bar length (of range 5 days) :

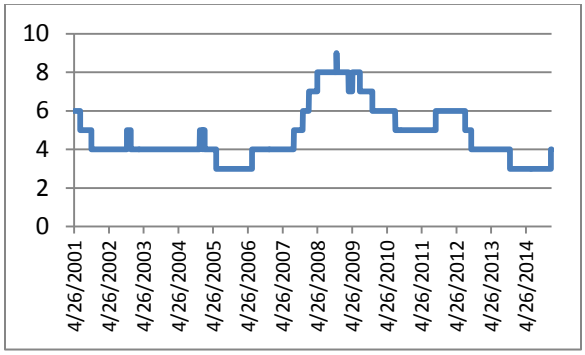


Average bar length (of range 20 days) :



The highest point is less than the average bar length of range 5 days is expected because the average is less influenced by sudden burst of price movement.

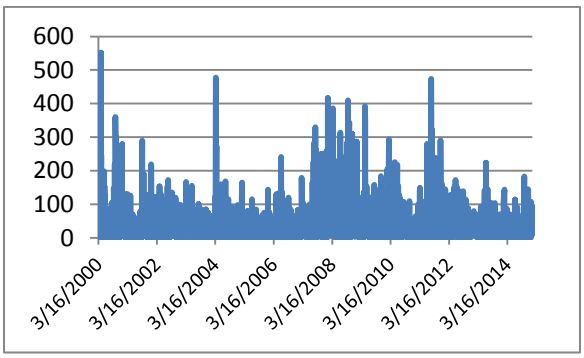
Average bar length (of range 240 days):



It's noticeable that bar length during or around 2000, 2004, 2008, 2011 have much higher bar length, this implies higher price movement, which is not surprising because there were financial crisis during years.

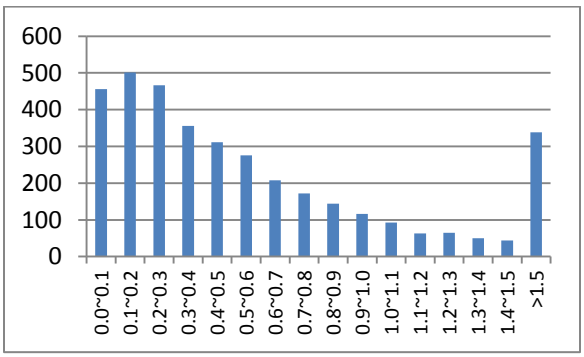
Opening gap is the difference of opening price today and the closing price yesterday. Here are some analytics in aspects of opening gap: the average of opening gap of all data (with absolute value) is 41.59, -0.07 without absolute value. This means that on average, downward opening gap is bigger than upward opening gap. Here, I continue the analysis with moving average.

The average of opening gap (of range 60 days) :



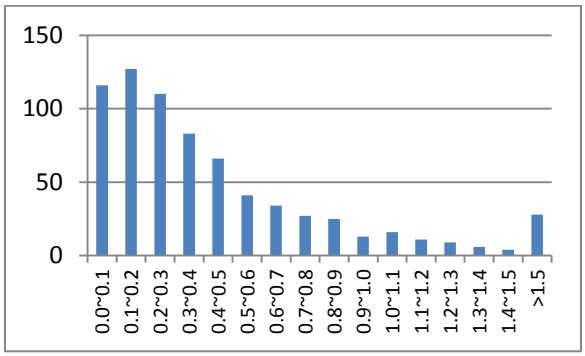
The average of opening gap of all data is around 42 yet it looks like there are many opening gaps >100 points. This means that most opening gaps are small but there are few big ones. We can make assumption about the distribution of opening gaps based on this observation. It is also noticeable that the opening gap chart shape looks a lot like average bar length chart. This explains that the opening gap is very highly correlated with bar length. For example, if there's a big opening gap, the bar lengths around that time tend to be larger also.

To confirm the assumption about opening gap, here are charts of the distribution of opening gap. The analytics are made in units of percent to more accurately capture the meaning of opening gap. For example, if the opening gap is +100 (today's open:9100, yesterday's close: 9000), it is 100/9100 = 1.09%. The distribution of opening gap (in percentage) of all data:

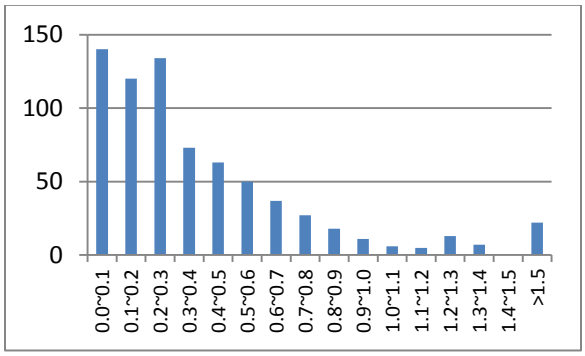


To see if opening gap distribution is different from time to time:

From 2004/1/1~2007/1/1:



From 2012/1/1~2015/1/1:

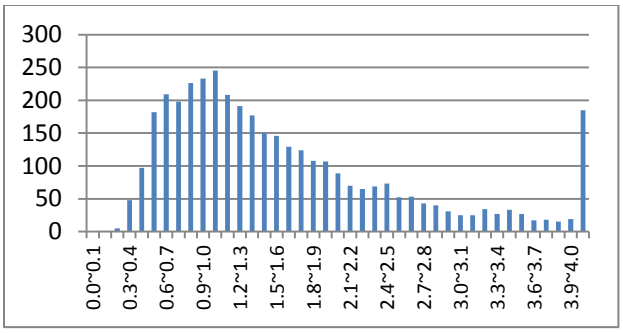


The comparison of the two charts is to tell if the opening gap distribution changes with time. The results indicate that the distribution shape is pretty much the same, with 80% of opening gaps within 0.6%.

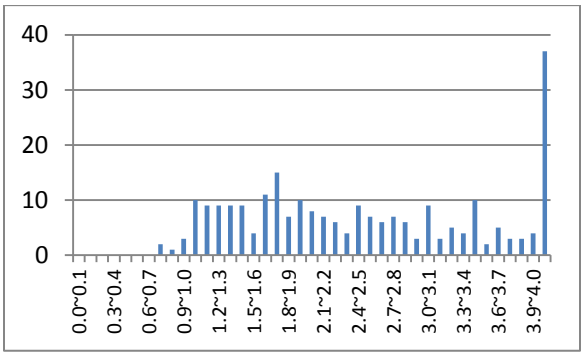
Next, I perform some analytics in aspects of range per day (highest price daily-lowest price daily). Average of range per day for all data is

Average of range per day (of range 5 days)(of range 20 days)(of range 60 days) for all data is 110.67 points, or 1.67% in percentage.

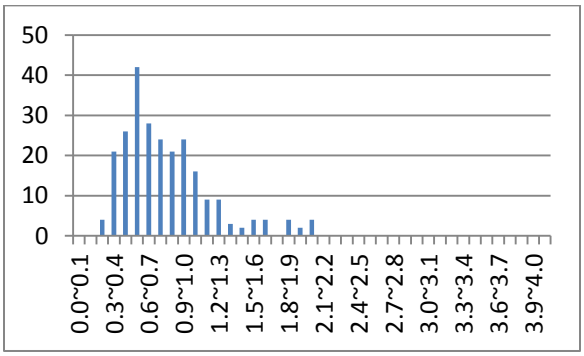
Distribution of range per day of all data: (in percentage)



The year of 2008 is a year of financial crisis. As the charts have shown above, the bar length and opening gap around that year is much larger than others. Here I use the distribution of range per day to affirm that show that the market, or price, is much more active during that year.



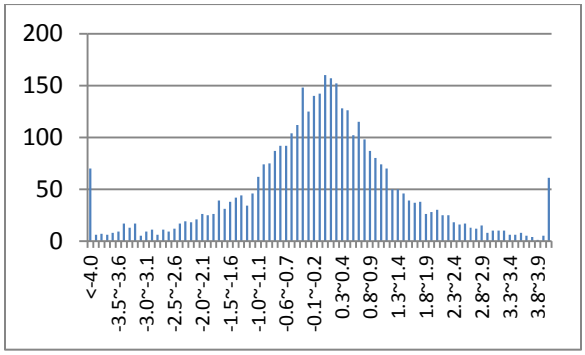
And here's the distribution of range per day for 2014:



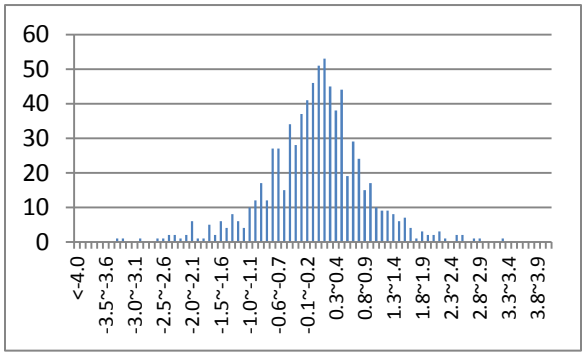
It is clear that the distribution of range per day greatly differs from these two years. With this distribution difference, I have confirmed that the market, or price movement, does change over time.

Next, I am interested in the rise/fall (close price – open price) per day. The average of rise/fall per day (with absolute value) of all data is 59.68 points, or 0.9% in percentage. Notice that this is roughly half of average of range per day. Another way of calculating rise/fall per day is to take opening gap into consideration, then the rise/fall is today's close price – yesterday's close price. Calculated in this way, the average (in absolute value) of all data is 74.52 points, or 1.12% in percentage.

The distribution of rise/fall (considering opening gap) of all data:



The distribution of rise/fall per day from 2012/1/1 to 2015/1/1 :



From the above analytics, we can tell that the market of 2012~2015 "less active" than some of those financial crisis years. As a result, this distribution difference indicates the same thing. The distribution of 2012~2015 is more centered and less ranged. 90% of times, the rise/fall per day is within -1.0% to 1.0 %.

Next, I would like to know if the rise/fall today is related to, or influenced by, the rise/fall yesterday. Disregarding opening gap, the average of rise/fall from 2012~2015 is 38.59 points, or 0.48% in percentage. Since most of the days, the rise/fall is within -1.0% to 1.0%, let's consider days >1.0% or <-1.0% a big rise/fall day. From 2012~2015, I averaged the rise/fall of days after a big rise/fall day, the result is 30.82 points, or 0.39% in percentage. This is 38.59-30.82 = 7.77, or 20% less than average. This result indicates that the rise/fall of yesterday have an impact on today's rise/fall.

Next, I want to know if price movement is more active during a specific time of a day. I split the 300-minute day by 100 minute, with 8:45am-10:25am being the first time zone, 10:25am-12:05pm being the second, and 12:05pm-13:45pm being the third. I calculate the range of price within each time zone. From 2012/1/1 to 2015/4/1, the price range for time zone 1 is 59.49, 51.10 for time zone 2, and 53.25 for time zone 3. This is reasonable because we can expect that price being more active during the opening and closing of market.

2. Predictive Task:

Stock and futures market are very complicated in many ways, mainly because there are too many factors that affect the market. Analytics in this field can go very far. However, for this assignment, I want to predict the rise/fall in day. In other words, for each minute in a day, I want to know whether the price later in today will rise or fall. This is to simulate day-trading, trades that don't hold overnight. Each datapoint has 3 possible labels (1: will rise later today, -1: will

fall later today, 0: neither). To be classified as "will rise later today", for the rest of time in the same day of the datapoint, it has to rise over 40 points. It is hardcoded, manually-determined as 40 points being the threshold floor to simplify the predictive task. To find out about the label for each datapoint, here's an example. If the current data point is {time: 2014/4/1 09:00, price: 9000}, scan from the next datapoint to the last datapoint in the same day, check each datapoint: if there's a datapoint whose price is >9040 than current label is 1, or if there's a datapoint whose price is <8960, then current label is -1, if neither then current label is 0. It is possible that it will both rise and fall sometime in the same day. For each datapoint, it's label is just set to the first condition met. For example, if within a day, it first rises and then falls. This is what could happen. At daily opening, 08:45, the price is 9000 with label "1", because at 09:30, the price is 9040. Since 9040-9000>=40, the datapoint of 08:45 is set to label 1 (meaning "will rise later today"). Then, at 09:30, the label is -1 because at 10:20, the price will drop back to 9000. Since 9000-9040<=-40, so the datapoint of time 09:30 is set to label -1 (meaning "will fall later").

It is more likely toward the end of a day that the label will be 0 because it is less likely that it can rise/fall >40 points in a short time.

To begin the predictive task, I first create the features vector for each datapoint. There are limitless ways to create features vector for each datapoint. However, I will build the features vector based on some of the factors I believe possibly have an impact on what the prediction goal (whether it will rise or fall later today). I will approach the prediction task with two models. First, I will use linear regression to train a model. This is to find out which features are more important. Second, I will use cosine similarity to predict. For each datapoint to predict, I will find the datapoint in the past with the max cosine similarity, and simply "copy it label". This is based on my assumption that history will repeat itself. So by finding the datapoint in the past that is most similar to the current condition, the label should likely be the same.

There are many models in data dimining techniques. However, not all of them could be "appropriate" for this prediction. Essentially, those models could still be used in the process of analysis, perhaps to tell us some information about the data, but might not be suitable to be used for prediction. For example, I can come up with a large set of features and run some reduction on algorithms on the vector. This process only tells me which parts of the features are more important but the reduction algorithms are not making prediction for me.

For this assignment, I have chosen linear regression and cosine similarity for prediction. Although linear regression is mainly used to predict continuous-value labels, I used it simply to find out which features are more important. For these two different approaches, I will build two different vectors. The information I want to capture in the vector is the same. It is just the representation that is different.

For linear regression, each [ ] is essentially a condition check, a Boolean, which will resolve to 0 or 1. The reason I set it up this way instead of continuous value is that I believe that the values have categorical meaning instead of continues value meaning. For example, for a feature to capture the rise/fall of yesterday, instead of using the value of the actual rise/fall yesterday, I made it into a condition check [yesterday rise<0.2%] [yesterday rise>0.2%]… Another reason of doing this is to have all the features weight the same when train with a regressor.

For cosine similarity, as oppose to features vector for linear regression, I will build

the features vector using continuous values instead of 0,1.

The features vector I built contains of 18 features. Since for linear regression, I made the continuous value feature into categorical features, it becomes 68 features. The features I selected are what I believe have an impact on the rise/fall. From the exploratory analysis, we know that rise/fall is a normal distribution with most of the times within range 1%. Just by this information, if at sometime during a day the price has already fallen 1%, it is unlikely that it will fall another 40 points later today; in other words, the label at that time is unlikely to be -1. Although features representations are different, the features vectors contain the same information. For example, some of the features are yesterday's rise/fall, yesterday's rise/fall (including opening gap), yesterday's range, current time, current rise/fall, current rise/fall (including opening gap), current range, today's opening gap.

For this assignment, I will use the last 35000 datapoints, which is roughly 2014/11/1 to 2015/4/1 (half year) as test set.

Since the exploratory analysis shows that the market changes with time, it is unnecessary to have the training set include all the historical data. To predict the half-year testset, I have chosen 2 years of training data.

Predicting whether it will rise/fall later in the same day is to simulate real-life day-trading. To test for validity of the prediction model, one way is the compare the predicted labels and the actual labels. If the predicted label is 1, but the actual label is 0, I will consider it as half-wrong. If the predicted label is 1, but the actual label is -1, I will consider it as total-wrong. Otherwise, it is considered predicted correctly. As a result, there will be two accuracies, 1-accuray(based on correct), 2-accuray(based on correct + halfwrong considered correct)

However, simply comparing labels might not be the best way to test for validity. For example, if the situation shown in the following image happens, althogh some of the labels are falsely predicted. It is profiting in reality. As a result, I will also test for validity in a way that simulate practical trading. If predicted as 1, buy; if -1, sellshort, if 0, hold current position. Each point profitted is $6.5 USD. For example, if predicted 1 at price 9000, and predicted -1 at price 9001, I have bought at 9000 and sellshorted at 9001. For this trade, I will profit 9001-9000 = 1 point, which is $6.5 USD.

The process to obtain the features to build the features vector is the same as the exploratory analysis described above. For rise/fall is calculated by (close-open). Rise/fall (including opening gap) is calculated by (close – close of the day before).

I will compare my prediction model to two baselines. First, find out what's the mode (most frequently appear) of the labels, and always predict the mode. Second, randomly predict the labels.

3. Related Literature

Predicting whether the market will rise or fall is essentially what every traders wants to do, or is doing. This is simply because when you buy, you are essentially predicting that the price will rise, and when you sell or sellshort, you are predicting that the price will fall. While most individuals trade manually, by looking at charts and studying history, practicing and finding patterns. Most or almost all funds hire data analyst, traders, programmer, and strategist to work on program trading. They design trading

strategies and code the strategies, then optimize and test the data on historical trading data to validate how well the strategies perform. This field is known as quantitative research or quantitative trading.

To obtain the dataset that I used, I exported it from stock market software. I can also download this data from the TAIFEX futures exchange governmental official website. It is not considered an existing dataset that people use to "study", it is simply easily-obtainable trading historical data. However, I build features on the data to analyze the data. I have about a few years of experiences in the field of program trading. Similar datasets that I have studies include futures from many different countries and stock market data. For this assignment, I used the basic historical trading data, containing time, price, and volume. I did not incorporate volume into the research for this assignment to avoid getting too complicated.

There are other data associated to the stock and futures market that people are able to obtain, such as chips data. An example of chips data would be what positions the investment funds are holding. In other words, depending on symbol and countries, we might be able to obtain data about what position investment funds are holding, and this could have a high correlation to the market movement since investment funds could be very good at market prediction.

About the methods currently employed to study this type of data is as described above. Generally, people, usually traders, since they would "know" market patterns better, come up with trading strategies. Then they code, optimize, test the strategies. Trading strategies are essentially set of rules to enter and exit the market.

I have approached market data with a different approach for this assignment. I utilize data-mining strategies to analyze the data in a more data-driven way. This is different from human-designed trading strategies approach. As for comparing existing work with my own findings, it is difficult to do so. This is because trading strategies can vary, there could be strategies that profit a lot as well as strategies that doesn't profit. And I do not how the strategies of others perform, therefore I couldn't compare my findings with others.

4. Results

Cosine similarity runs slowly because there are many datapoints in the training set to each be computed. To keep it more practical, too-slow computation is unacceptable because by the time result comes out, the market price would possibly have moved a lot. To resolve this, since from the exploratory analysis I know that the market is most active around the time of opening, I cut down both the test set and training set to only contain datapoints before 10:00am. To keep all model fair with comparison, each model’s training set and test set is the same, all reduced. Another reason of doing this is that since as time move toward the closing of market every day, it is more and more likely that the label will be 0 since there’s not much time for it to rise or fall later in the day. So since there are lots of 0s toward the end of every day, I do not want this to interfere with training of model. Rather, I would prefer that the labels are of roughly the same numbers of 0s, 1s, and -1s. This also justifies reducing training and testing sets to datapoints before 10:00am.

Here are the initial results:

	baseline-1	baseline-2	Linear regression	Cosine simi
Accuracy-1	0.7990	0.3387	0.7928	0.3972
Accuracy-2	1.0	0.8339	0.9936	0.8547
profit	0	-225USD	310 USD	300 USD

Notes:
baseline-1 (predicting all with the mode)
baseline-2 (randomly predicting labels)
accuracy-1(based on correct only)
accuracy-2(halfwrong considered correct also)
since the data is cut down to only predict before 10:00am, profit calculation is based on exiting the market at 10:00am. In order words, there will be no trades from 10:00am to the closing (13:45pm).

For baseline-1, since the mode (most frequently appeared label in training set) is 0, it always predict 0. Therefore, the profit is 0 (never enters the market). However, since a lot of the actual labels are 0, that explains why it has high accuracies.

Linear regression did not work at first because it almost always predict zero. That's why it's accuracy-2 is very close to 1 since accuracy-2 considers half wrong as correct. And since a lot of the actual labels are 0, by almost always predicting 0 gains the highest accuracy-1. It's profit number being 310USD is not significantly because it only has 5 trades. I believe that this is due to overfitting. After training the regressor, I used a loop to find out what the best determining values are to separate the labels. However, this looping optimization will likely to make the best values very marginal so that it will almost always predict 0 to minimize error. To resolve this, I changed the error calculation to be as long as the predicted label is different from the actual label, error+=1, regardless of the whether it's half wrong or total wrong. This fixed the problem and here's the updated accuracies.

	Linear regression (updated)
Accuracy-1	0.2682
Accuracy-2	0.7571
profit	-735 USD

Still, the linear regression approach does not work very well. It has a low accuracy-1 and a negative profit number. Possibilities that cause this include overfitting, or un-effective feature selection. Since the linear regression approach results in a very low accuracy, I did not consider the fitted parameters meaningful.

For the cosine similarity model, I also did an update to increase its performance. I changed it to a moving training set, which means that after a datapoint is predicted, it is moved from test set to training set, and the oldest datapoint in the training set (farthest away datapoint) is removed from training set. This way, training set does not change in size but its datapoints are changing as we move along when predicting.

This is the updated performance.

	Cosine similarity (updated)
Accuracy-1	0.4256
Accuracy-2	0.8742
profit	420 USD

There is an improvement comparing with the original cosine similarity model. However, the result accuracies are still not very high, and the profit number is also low. The result indicates that by finding the most similar datapoint in the past and copy its label does not work very well. This tells us either the features I selected were not very effective, or it could also be that even with very similar condition in the futures market, it does not mean the price movement later on will be the same.

There are still many ways to optimize my prediction model. First off, I could use other features to build the features vector and the result should be very different. Another thought is to perform reduction before predicting. For example, some of the features in the features vector could have high correlation with each other. This way, the weight of "actual" individual feature could be unbalanced, because there are essentially many features that represent the same information. As a result, I think I can run PCA or other reduction algorithms to reduce the features vector. I optimize for the features used so that I avoid overfitting because of adding too many features. Another thought is that choosing the max cosine similarity of the datapoints in the past might not be the best decision. Another way to do it is to save the cosine similarities of the datapoints in the past, then choose the top 5 (this number can be optimized) and find the average of them. Prediction then will be based on this average.

Lastly, even if there exists models to predict very accurately of the futures market, it needs to predict it “fast.” Otherwise, in real-time trading, it might do a lot worse because of the prediction delay since it might takes many seconds to predict for each datapoint.

Sources:

Fig.1 – Candlestick_Chart1.png

“http://www.independent-stock-investing.com/images/Candlestick_Chart1.png”