PROJECT MINESTOCK

Progress Report

Written by TONG Man Kin (08502621D)
Supervised by Prof. CHAN Chun Chung Keith, The Hong Kong Polytechnic University

January 13, 2011

Project MineStock
-
Mid-term Check Point Progress Report

Written by TONG Man Kin (08502621d),
Supervised by Prof. CHAN Chun Chung Keith,
The Hong Kong Polytechnic University

January 13, 2011

Contents

1 Abstract

The financial tsunami in 2008 has turned the world into a turmoil. Its destructive effect has lasted until today, causing people to reevaluate the risk of their derivative investments, and reconsider their current investment strategy. One important lesson we learnt from the collapse of Lehman Brothers is that the derivative instruments are only effective in transferring a type of risk (i.e. market risk) to another (i.e. counterparty risk). They have no use, ironically, to reduce the total amount of risks that one investor is exposing in, which is very different from what we usually perceived. Clearly, it would be much wiser for an inidivual investor to adopt the classical technique - portfolio diversification, instead of putting a vain hope, again, on those derivatives.

The writer believes a sophisticated portfolio optimization tool would satisfy the need of investors nowadays and help the promotion of rational investing. This tool is therefore proposed to be built. By applying data mining techniques, assisted by theories of financial mathematics, this scientific tool is considered able to suggest both the optimal stock selection and their optimal weighting in portfolios, and thus the investors are able to diversify the risk of their investments.

In this progress report, the writer will stress on the related studies and the programming works done. For more detailed business background and problem specifications, readers are suggested to refer to the proposal of Project Mine-Stock. According to the set project schedule, this project is slightly delayed. However, it is still expected to be completed on time without changes in major specifications.

2 Objectives

2.1 Project Objectives

This project is targeted to research on existing data mining algorithms, design and implement a sophisticated software tool which can assist its users when they make investment decision. Users would enjoy the benefit of risk aversion when they study on, and then apply the stock combination advised by the tool. The study on data mining techniques will also benefit other software developers when they extend the capabilities of this tool or apply the same technique to achieve related goals. Last but not least, as one of the portfolio optimization tools available in the future, it intends to contribute in promoting the practice of diversification and rational investing to the general public.

2.2 Project Scope Description

To accomplish the objectives stated in section 2.1, the following project specifications, scope and boundary are suggested:

- Software specifications

  - Contains an executable command-line component for collection of end-of-day (EOD) stock data from Yahoo! Finance HK. This component will accept input to specify the time frame of data to be collected, and save the data in user desired location.
  - Contains an executable command-line component which analyze the stock data collected by the above process. This component will have several data mining algorithm built-in and allow users with relevant knowledge to select their preferred algorithm.
  - Contains a Graphical User Interface (GUI) for user interaction. This component will visualize the analysis result, suggest additional stocks for the user's portfolio and calculate the optimal contribution of each stock in it. Statistical calculation of risk carried by the portfolio, and return per risk, will also be illustrated to user.

- Project scope
  - All stocks listed in main board or growth enterprise market (GEM) in Hong Kong, will be analyzed by the software tool.
  - Price data of these stocks, since their listing, or since the earliest date that the writer could acquire their EOD data, will be analyzed by the software tool.

3 Studies Conducted

3.1 Studies on Clustering Techniques

As stated in the proposal, the clustering techniques are the key pillars of this project. An effective algorithm is very important for the software users to find out the stocks which are able to effect the greatest diversification on their current portfolio. In this section, the writer is going to discuss the clustering algorithm studied so far.

3.1.1 Classical K-means Technique

K-means technique is one of the classical clustering algorithm, it focuses on measuring similarity and dissimilarity of data sets, and after that, grouping these data sets into different clusters. Data sets in the same cluster have similar data, where data sets across different clusters will be more distinct.

One of the most common methods for measuring similarity and dissimilarity is euclidean distance. According to Gan, Ma and Wu [1], the formula of calculating euclidean distance is:

$$d_{euc}\left(x,y\right)=\left[\sum_{j=1}^{d}\left(x_j-y_j\right)^2\right]^{\frac{1}{2}}$$

In the formula, x_j and y_j represents the j-th attribute of data set x and y, respectively. The greater the euclidean distance, the higher the dissimilarity between the two data sets.

After we know how the distance between data sets is defined, we need to know how we can group different data sets into clusters. K-means clustering method suggests that we could initially create those clusters by randomly selecting a data set for each of the clusters, and then for the rest of the data sets, we could compare the distance between them and the mean of each clusters, and finally putting the data sets into the cluster where they have the shortest distance.

For its application on the stock data, the software algorithm has followed the below steps:

- Main algorithm
    1. According to the given parameter of number of clusters, randomly select one stock for each of the clusters.
    2. For each stock S randomly selected above, assign its set of return $S = \{R_{s1}, R_{s2,}, \ldots, R_{sT}\}$ to represent the center of the cluster.
    3. For each stock S,
        (a) Calculate the euclidean distance between its set of return and the center of each cluster C, $d_S = \{d(S, C_1), d(S, C_2), \ldots, d(S, C_n)\}$
        (b) Find the minimum element among the set of distance calculated, assign stock S as an element of the cluster C which they have the smallest distance.
    4. For each cluster C,
        (a) Sum up the set of return of stock S inside $C = \{S_{c1}, S_{c2}, \ldots, S_{cN}\}$ for each time t, after that, take an average of them. Formally, $C_{center} = \{\sum_i^n R_{t1}/n, \sum_i^n R_{t2}/n, \ldots, \sum_i^n R_T/n\}$
    5. If there is any changes in the element of clusters, redo the steps starting from step 2.

K-means and euclidean distance are the easiest and most popular way to perform clustering on data sets. However, in the writer's analysis, this algorithm may not be the best way to process time series data. The followings show the result of a trial of k-means algorithm, processing 1302 stocks which are having 3 months of EOD price data.

Cluster #	Count of stocks
1	2
2	638
3	14
4	628
5	20
Total	1302

Table 1: Count of stocks in each clusters, algorithm: k-means, timespan: 3 months EOD, no. of clusters: 5

We can observe that most of the stocks have gone into clusters 2 and 4. The writer's studies on the clustering results found that most of the stocks, if they appeared to have a long term trend, either upside or downside, they will congregate into one or two clusters, resulting the other clusters contains only the outliers. For example:

Stock code	Mode daily return	Highest return	Highest return %
642.HK	+0	+0.19	+190%
8298.HK	+0	+0.15	+100%

Table 2: Detail of cluster 1, algorithm: k-means, timespan: 3 months EOD, no. of clusters: 5

The reason that most of the stocks go into one or two clusters is related to how the algorithm deal with the specialities of time series data. In the calculation of euclidean distance, the return of each day is simply treated as another attribute of the stock. Therefore, difference in distribution of short term fluctuations may end up offsetting each other. Like the case shown in the following figure:

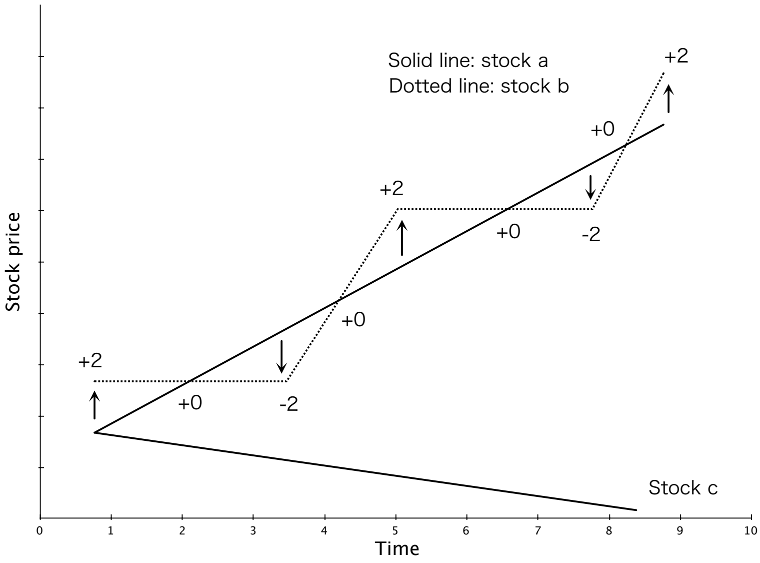


Figure 1: Although stock (a) and stock (b) have a very different short term fluctuation pattern, their euclidean distance is 0 as the fluctuations offset each other.

In the above example, stock a and b will be assigned into the same cluster. Only stock c, which has a very high distance with stock a and b, is likely to be assigned into another seperate cluster.

The studies above shows that k-means may be a effective way to locate outlier of the market. However, it is clearly not the best option to cluster the stocks evenly into groups that meets our application needs.

3.1.2 Identical Sequence Extraction

To handle the speciality of sequencial data, previous works done by Han and Kamber [2], Ma and Chan [3], suggests that the sequencial data can be divided into a number of subsequences by sliding a window with a predefined width, across the whole sequence. After that, analysis can be performed on those subsequences.

To apply this in historical price analysis, we can define the window by a desired width w and apply this to the price or price change series of stock S:

$$S = \{P_1, P_2, \ldots, P_n\}$$

After that, a set of subsequence, with a count of $n - w + 1$, will be produced as follows:

$$S_{sub} = \begin{bmatrix} S_1 = \{P_1, P_2, \ldots, P_w\} \\ S_2 = \{P_{w+1}, P_{w+2}, \ldots, P_{2w}\} \\ \vdots \\ S_{n-w+1} = \{P_{n-w+1}, P_{n-w+2}, P_n\} \end{bmatrix}$$

Once every stocks have their historical price data converted into subsequences, we can apply different data mining techniques to discover the interesting patterns among them. This includes clustering and seperating different stocks into groups according to the difference of their occurance on each sequence.

As either the stock price or price change data are in continuous numeric form, a very low matching rate would be observed if the numeric sets are just compared directly. Therefore, certain discretization method on the numeric data are used to divide these continuous data into certain discrete interval, in effect providing an approximated result during the comparison. Readers can refer to section 3.1.4 for more details.

For example, we can discretize the price change number into 3 intervals, they are up, U, down, D, and no change, N. Given this rule, we will be able to produce a discretized price change sequence, like the following:

$$S = \{U, N, N, D, D, D, U, D, N, U, U, N, U, D, D, D, \ldots\}$$

Now we are able to perform the above discussed window sliding steps and count the occurances of each possible sequence. As a result, if the window has a width of 5, a matrix like the one below could be formed.

$$O_S = \begin{bmatrix} \{N,N,N,N,N\} & 101 \\ \{N,N,N,N,U\} & 87 \\ \{N,N,N,N,D\} & 81 \\ \vdots & \vdots \\ \{U,U,U,U,N\} & 7 \\ \{U,U,U,U,D\} & 13 \\ \vdots & \vdots \\ \{D,D,D,D,N\} & 11 \\ \{D,D,D,D,D\} & 21 \end{bmatrix}$$

Every possible sequences can act as an attribute in the clustering process. When all stocks have their subsequence occurances counted, we can apply the k-means algorithm on the set of occurances of each stock. In sum, the software deliverable will perform the following steps for this algorithm:

- Data preprocessing
  1. Discretize the price data of every stocks.
- Main algorithm
  1. According to the given parameter of width, construct a list with all possible sequence.
  2. For each stock S, slide a window with the defined width, count the occurances of each possible sequence and form the matrix O_S.
  3. Perform k-means algorithm on the set of O_S acquired.

As the sequencial nature of data is handled by introducing the concept of sequence in the algorithm, we can expect that it can produce a more effective clustering result. The followings show the result of a trial of this sequence extraction algorithm, processing 769 stocks which are having 5 years of EOD price data.

Cluster #	HSI constituent	Count of stocks
1	True	0
	False	38
2	True	0
	False	149
3	True	0
	False	255
4	True	37
	False	218
5	True	0
	False	72
Total		769

Table 3: Count of stocks in each clusters, algorithm: sequence extraction, discretize: price change - interval 5, timespan: 5 years EOD, no. of clusters: 5

Figure 2 shows the differences of secleted sequence occurances between the center of clusters, calculated with the identical settings mentioned above. It is obvious that in the result of this trial, most of the actively traded stocks as well as all the HSI constituent stocks are grouped into cluster 4. Stocks which are not actively traded and always stay in its price level, are grouped into cluster 1, 2 and 5. Stocks which have not apparent patterns are grouped into cluster 3.

3.1.3 Similar Motif Discovery

While the previous algorithm divides the subsequence of price series evenly by the same width. The principle of motifs [4] are trying to locate exact or approximate matches with longest length possible, among the sets of time series data.

Given a 3-interval setting of discretizing the price change series, we will be able to produce some sets of stock sequences, like the following examples of stock a and b:

$$S_a = \{U, N, N, D, D, D, U, D, N, U, U, N, U, D\}$$

$$S_b = \{N, N, D, D, D, U, D, N, N, U, U, N, U, D\}$$

We can then identify that the longest motif, or the longest consecutive subsequence between S_a and S_b, is $\{N, U, U, N, U, D\}$, which is the tailing subset in the sequence. However, we can also notice a great similarity in the starting part of the sequence, it is not considered just because there is an irrelevant ticks, or noise in between. Therefore, intead of identifying the the longest consecutive subsequence, identical sequences which is not linked together should also be considered, as follows:

$$S_a = \{U, \{N, N, D, D, D, U, D\}, \{N, U, U, N, U, D\}\}$$

$$S_b = \{\{N, N, D, D, D, U, D\}, N, \{N, U, U, N, U, D\}\}$$

The above result shows that the two stocks have a similar motif with timespan equals 13, instead of 6. This value is also known as the length of the longest common subsequence.

$$lcs(S_a, S_b) = 13$$

In our application, the algorithm compares each stock with the others using this method, and then build up a similarity matrix to store the length of longest common subsequences between different stocks. This matrix will have a size of n^2, where n denotes the number of stocks processed by the algorithm.

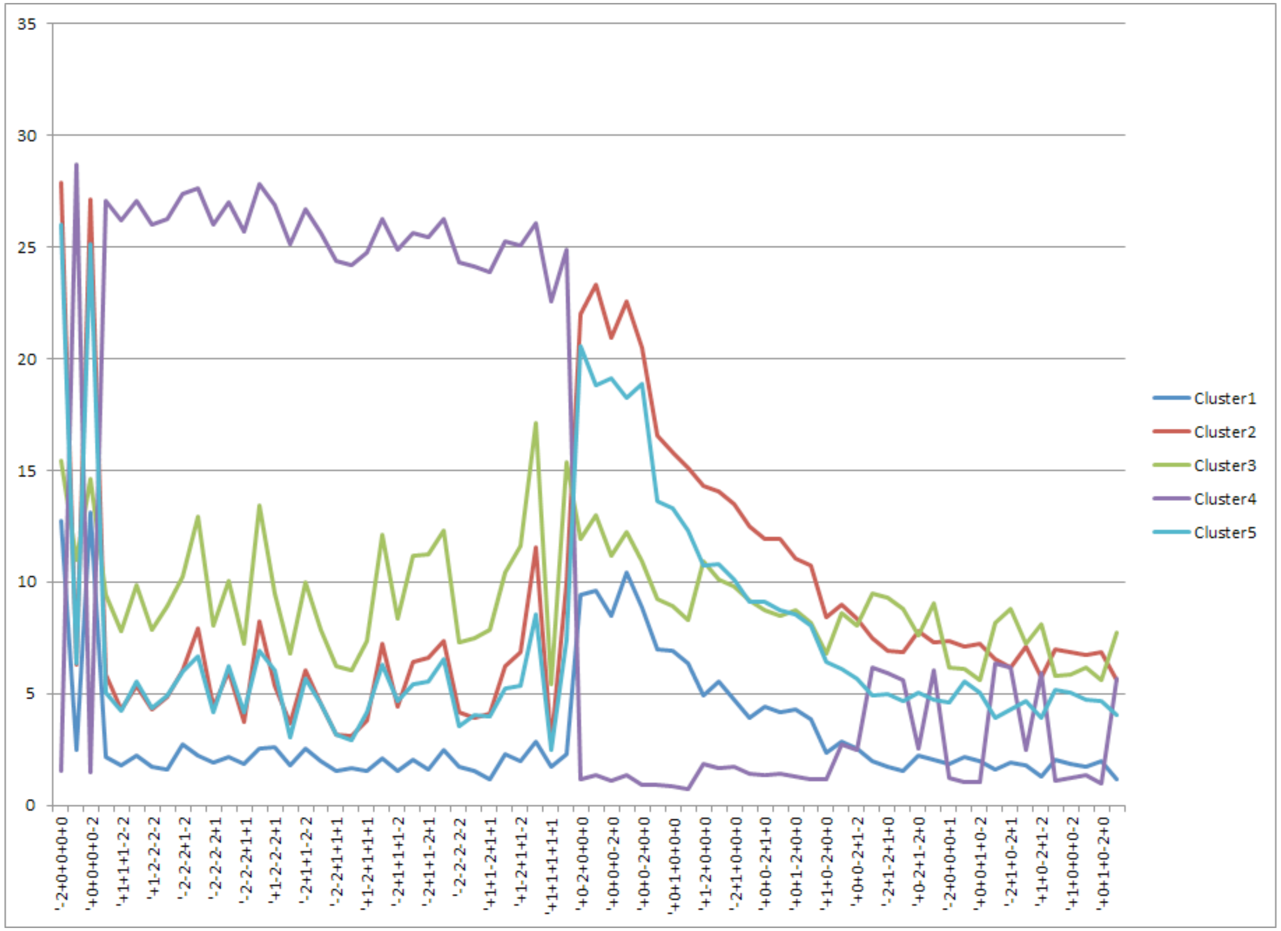


Figure 2: Difference between clusters, algorithm: sequence extraction, discretize: price change - interval 5, timespan: 5 years EOD, no. of clusters: 5

$$Similarity = \begin{bmatrix} sim(S_1, S_1) & sim(S_1, S_2) & \cdots & sim(S_1, S_n) \\ sim(S_2, S_1) & sim(S_2, S_2) & & \vdots \\ \vdots & & \ddots & \\ sim(S_n, S_1) & \ldots & & sim(S_n, S_n) \end{bmatrix}$$

The sim function denoted above is an extension of lcs calculations, it includes a similarity multiper, m, which is used to magnify the degree of simlarities among the stocks, as follows:

$$sim(S_a, S_b) = lcs(S_a, S_b)^m$$

After adding this parameter, a much greater similarity will be observed if two stocks have a very similar or even identical fluctuation patterns. In contrast, a very low similarity will be result from stocks which they have nothing in common at all. The similarity multiper can be defined by the users.

The algorithm will perform the steps of k-means on the similarity matrix after the matrix has been acquired. This process is meaningful because, if one stock is similar with another stock, their similarity metrics to the rest of the stocks must be similar as well. In sum, the software deliverable will perform the following steps for this algorithm:

- Data preprocessing
    1. Discretize the price data of every stocks.
- Main algorithm
    1. For each stock S,
        (a) Calculate the similarity with all stocks
        (b) Fill the similarity matrix with calculation results
    2. Perform k-means algorithm on the similarity matrix

The followings show the result of a trial of this motif discovery algorithm, processing 1170 stocks which are having 1 years of EOD price data. A relatively even distribution of stocks among the 5 clusters can be observed.

Cluster #	HSI constituent	Count of stocks
1	True	0
	False	205
2	True	0
	False	65
3	True	1
	False	314
4	True	44
	False	419
5	True	0
	False	122
Total		1170

Table 4: Count of stocks in each clusters, algorithm: motif discovery - multiper 2, discretize: price change - 5 interval, timespan: 1 years EOD, no. of clusters: 5

Figure 3 shows the similarity metrics between the center of clusters and the HSI constituent stocks, with the same settings used for table 4. We can see that the clusters are quite distinct that they have a large distance to each other.

3.1.4 Discretization of Price Data

There are currently two discretization method implemented in the software deliverable, they include:

- Discretize by price change
  - By using this option, the countinuous numeric data of price change can be discretize into a number of intervals defined by users, to represent different degrees of upside or downside of the ticks of price change.
- Discretize by price level
  - By using this option, the countinuous numeric data of price can be discretize into a number of intervals defined by users, in order to represent different levels of price compared with the average price of that particular stock. As this option will become very ineffective in processing prices with a very long timespan due to the long term growth rate of the stock, another parameter named as referencing period is provided for users. Users can define the length of this period, after that, the price ticks will be compared with the average of their corresponding referencing period, instead of comparing with the all-time average.

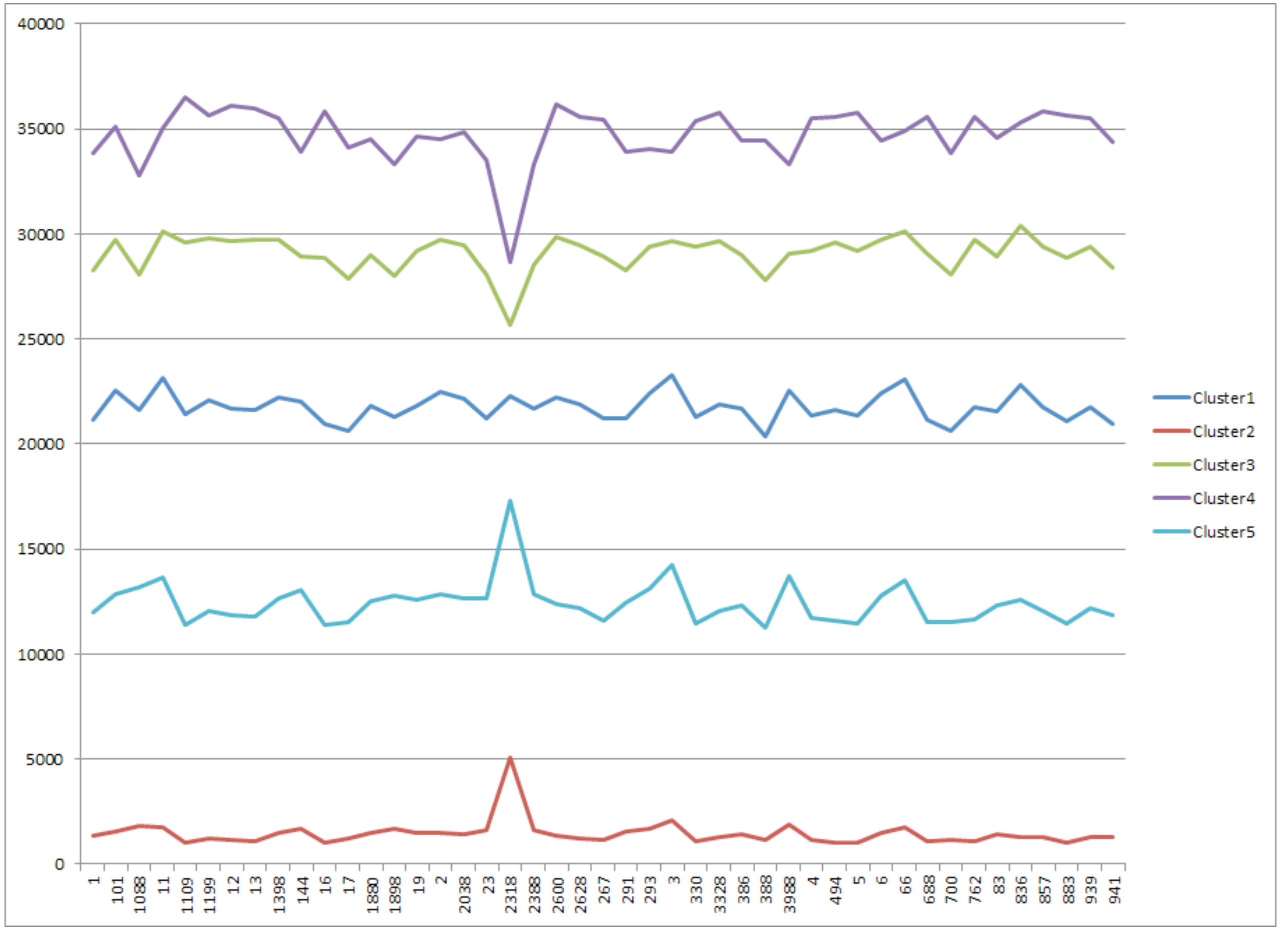


Figure 3: Difference between clusters, algorithm: motif discovery - multiper 2, discretize: price change - 5 interval, timespan: 1 years EOD, no. of clusters: 5

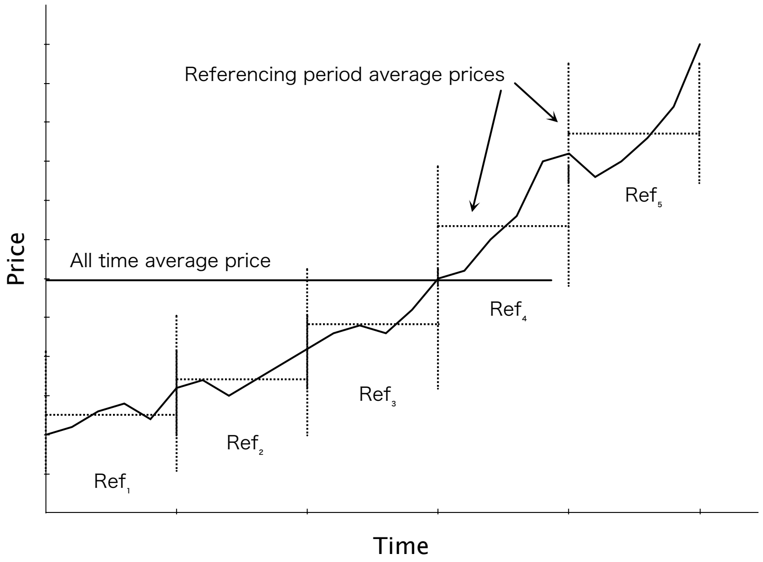


Figure 4: Illustrated average price calculation by seperating the whole period into referencing periods

3.2 Studies on Portfolio Optimization

After some desirable stocks have been shortlisted, there are certain metrics for us to review and then select the best one among the stocks. Also, there are methods for us optimize the proportion of each stock in the portfolio, so that we can achieve a weighting which has the lowest risk but the highest return among the possible options. In this section, the writer is going to discuss the related techniques studied so far.

3.2.1 Modern Portfolio Theory

Developed by Markowitz [5, 6], Modern Portfolio Theory (MPT) is the first and also the most important foundation of mathematical techniques in portfolio optimization. It introduces the analysis of investment portfolios by considering the expected return and underlying risk of each individual assets and, crucially, the interrelationship of these assets in the portfolio. It provides a mathematical framework for quantifying risk and return, enables comparison between portfolios using these quantitative measurements. Before this, investors can only examine their investments one by one through fundamental or technical analysis, and then build up portfolios of their favored stocks without the concern of their relationship in between. Markowitz's contribution is a breakthrough in both the mathematical and finance areas at that time.

In general, risk and return are positively proportional. This means investments which have higher risk are expected to have a higher return. For example, investing on the real estate market would have a higher return than stock market, as the relatively lower market liquidity creates extra uncertainty, or risk, which requires a higher return to compensate. As a result, if there are two portfolios that offer the same expected return, there is no reason for investors to select the more risky one. With the belief on this condition that investors are risk averse, MPT calculates and compares the risk per return of given portfolios, so that investors can review these statistics and select a better portfolio.

The theory considers a portfolio as a weighted combination of its assets, and thus the return of a portfolio is the weighted return of its assets. Therefore, the expected return of portfolio can be expressed as the following formula:

$$E(R_p) = \sum_i w_i E(R_i)$$

R_p represents the return of the portfolio, R_i represents the return of individual stock, or investment i, w_i represents the the proportion of asset i contributed to the portfolio.

The theory also models the return of an investment as normally distributed and the risk of it as the standard deviation of return, and the variance of portfolio return is be expressed as:

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij}$$

The second part of the formula is considered as a model of the interrelationship of two stocks and how this relationship affects the return of the portfolio, where ρ_{ij} is the correlation efficient between the return of the two stocks, i and j. And the standard deviation, or the risk of the portfolio, can be simply calculated by taking a square root of the variance:

$$\sigma_p = \sqrt{\sigma_p^2}$$

If an investor is provided with various portfolios for him to choose, he or she can apply the formulae above to calculate the return and risk of each portfolio and plot the following graph:

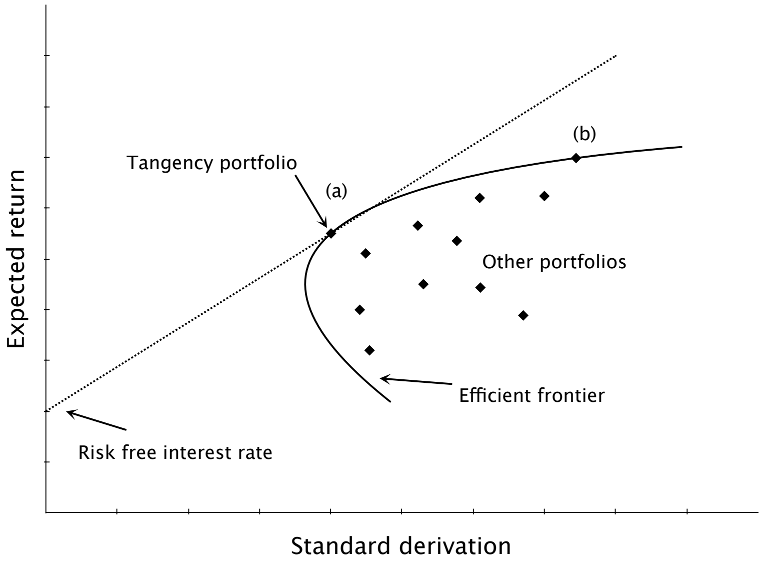


Figure 5: Illustrated efficient frontier. Portfolio (a) has the greatest return per risk among others, portfolio (b) has the same stocks as (a), but differs in each stock's weighting.

The curve which intercept with the straight line started from risk free interest rate, is called the efficient frontier. It is a portfolio which can offer the greatest return per risk, at point (a) of the graph, when certain weighting is applied on its stocks. That particular portfolio and weighting of stocks is the best option available for the investor. He or she can simply reduce the holdings of this optimal portfolio and purchase risk free assets if lower risk desired, or simply borrow risk free assets to increase the holdings of this portfolio if higher risk and return is desired.

Instead of plotting out figure 5, the optimal portfolio can also be found by finding the maximum of return per risk of all available portfolios, the return per risk of a certain portfolio can also be calculated by the following formula:

$$S = \frac{E(R) - E(R_f)}{\sigma}$$

This is also called the Sharpe ratio. $E(R_f)$ indicates the expected return offered by the risk free asset.

MPT is considered effective to select an optimal portfolio among a few options, or deciding an optimal proportion for the given stocks inside the portfolio. However, it becomes very difficult to decide whether a stock should be included in a portfolio to achieve the maximum Sharpe ratio. Like Markowitz described [6], this theory is focus on the choice of portfolio, instead of the observation on the stocks.

3.2.2 Capital Asset Pricing Model

The Capital Asset Pricing Model (CAPM), introduces another risk metrics, Beta (β), which is also commonly used nowadays [7]. It measures the responsiveness of a stock to movements of the market portfolio, which is considered effectively diversified and consist of only the systematic risk of the market.

Beta is a representation of both the risk and return of an asset, so if a stock is considered to have a high beta, not only its risk is high, its expected return, according to the model, is high too. An asset's beta can be calculated by the following formula:

$$\beta = \frac{Cov(R, R_M)}{\sigma^2(R_M)}$$

In this formula, R represents the return of the asset and R_M represents the return of the market portfolio. We can consider the Heng Seng Index as the market portfolio as this project is mainly concerned about the stocks in Hong Kong.

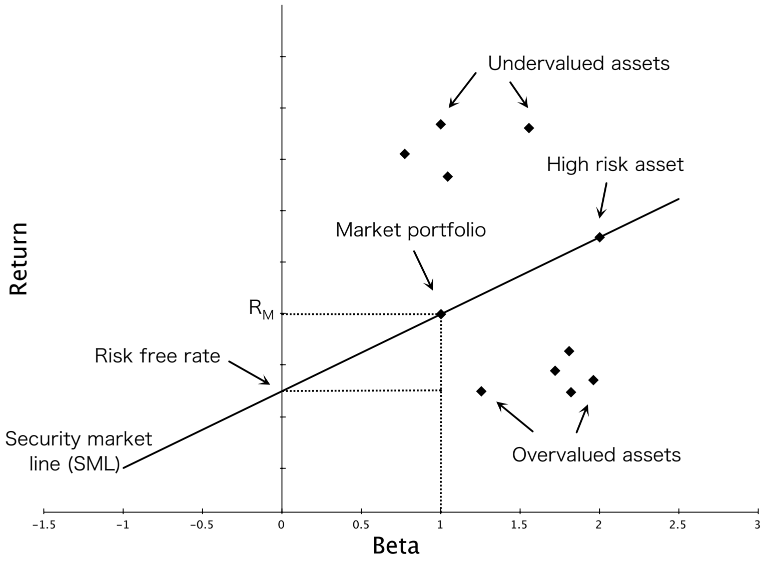


Figure 6: Illustrated graph of CAPM. The beta of a market portfolio is defined as 1. Stocks with its beta higher than 1 indicates that it has a higher risk as well as higher return, historically, compared with the market portfolio.

According to the CAPM model, the characteristic line which connects the point of risk free asset and market portfolio is the Security Market Line (SML). The stocks which lay on this line is correctly priced at the moment. However, for the stocks which are not laying on SML, they are either undervalued or overvalued. Stocks which is currently undervalued is quite likely worth investing because their expected return higher than others which have the same amount of risk.

By providing these metrics, users of MineStock Workbench can evaluate the stocks found by the data mining algorithms very easily. They are, therefore, allowed to compare and select their own preferred stocks for portfolio diversification.

3.2.3 Value at Risk

Value at Risk (VaR) is a popularly used method to measure the market risk of the financial assets. It is considered as a threshold value such that the mark-to-market (MTM) loss of one asset over the given time period, in a given probability level, exceeds this value.

For example, if a stock has a one day 5% VaR of $1,000, there is a 5% probability that the specific stock will fall by value by more than $1,000 over a one day period, assuming the market is normal and there is no trading on the portfolio within the day.

According to Choudhry [8], there are three ways to evaluate the VaR of an asset, as follows:

- Delta normal method
  - Also known as the parametric method. It assumes the distribution of return is normal, and uses standard deviation as a parameter to calculate VaR.
- Historical method
  - It uses the actual return distribution from the historical data, and measure the amount of loss with the given percentage of occurance.
- Monte carlo simulation
  - It uses a radom walk function to simulate stock price, and then measure the amount of loss with the given percentage of occurance, using the simulated set of data.

Although all the three methods are widely used in the financial industry, in a recent study performed by the the writer and Siu, et al [9], indicates that their calculation results on one asset could be very different in some exceptional cases, especially when the stock has a long term upward trend in the referencing period. In these cases, even the standard deviation of the stock return is high, resulting in the parametric VaR is high, it overstate the possible amount of loss compared with the historical method, given the same time interval and confidence level.

In this project, after the clustering of stocks has been done, the users can review their current portfolio and consider adding stocks from other clusters in order to enjoy the effect of diversification. The software deliverable of this project will provide historical VaR of each stock for user reference, such that a stock with less occurance of downside or big loss could be easier to locate.

4 Developmental Works Conducted

4.1 Design

As stated in the project proposal, the deliverable of this project will be completely built by Microsoft C# and .NET Framework 4. The writer has chosen the Microsoft Visual Studio 2010 for its development.

For the sake of easy devolopment and maintenance. The MineStock Workbench is extensivly modularize into different components, described as follows:

- Common
  - A Dynamic-link Library (DLL) project, contains utility classes and entity classes needed for the whole project.
- GetStockData
  - A console application project, contains the functions to fetch the historial stock data from the internet.
- GetTbillData
  - A console application project, contains the functions to fetch the spot rate of U.S. treasury securities from the internet.
- FilterStockData
  - A console application project, contains the functions to filter the invalid stock data collected by the above modules.[1]
- CategorizeStockData
  - A console application project, contains the algorithms to perform discretization on the stock data.
- ClusterStockData
  - A console application project, contains the algorithms to perform clustering on the stock data.
- GUI_v1
  - A windows form application project which integrates all the above modules and provides a easily accessible control for those modules.

Please note that the development of MineStock Workbench is still in progess and the above information are subject to changes.

[1] This is a countermeasure to the occasional incorrectness of data provided by Yahoo! Finance HK, please refer to section 5 for more information.

4.2 Code Metrics

The maintainability of the source codes is always concerned by the writer, as such, they have been kept in an acceptable level. Table 5 shows a list of code metrics of the modules, calculated by using the Microsoft Minimum Recommended Rules set, offered in Microsoft Visual Studio 2010.

4.3 Screenshot

Figure 7 is a preview screenshot of the current developmental version of MineStock Workbench. You can see that the user interface of collecting stock data, performing clustering, etc, have already been implemented. Since the development of software is still in progess, the outlook and design of the user interfaces may change very often, and therefore the detailed description of each form and user guide will only be provided in the final report.

5 Difficulties Encountered

The followings are two major problems which the writer faced in this term, extra effort is allocated to deal with these problems and thus resulting a delay of the total progress. The problems have been addressed accordingly.

- Inaccuracy of stock data collected
  - The stock data collected from Yahoo! Finance HK is occasionally incorrect. Historical price data of several stocks consist of price ticks which are observed with price changes when the dates are actually Hong Kong holidays. These extra ticks will definately affect the result of clustering algorithms and, therefore, unexpected debugging and programming efforts are paid to locate and resolve the issue. The software is now enabled to filter the incorrect ticks of a stock, by referencing ticks of another stock and index.
- Underestimated workload on algorithm design and implementation
  - The design and implementation of algorithms have taken longer time than originally expected. As a result, some planned work of this term cannot be completed on time.

6 Schedule

The following table shows the schedule of the first half of Project MineStock, from October to December. This project is slightly delayed. However, it is still expected to be completed on time without changes in major specifications.

Module	Maintainability Index[a]	Cyclomatic complexity[b]	Depth of inheritance[c]	Class coupling[d]	Lines of code
CategorizeStockData	62	62	1	19	155
ClusterStockData	66	188	1	34	500
Common	86	72	2	28	163
FilterStockData	54	23	1	13	77
GetStockData	46	49	1	22	132
GetTbillData	63	11	1	16	35
GUI_v1	56	395	7	115	2931

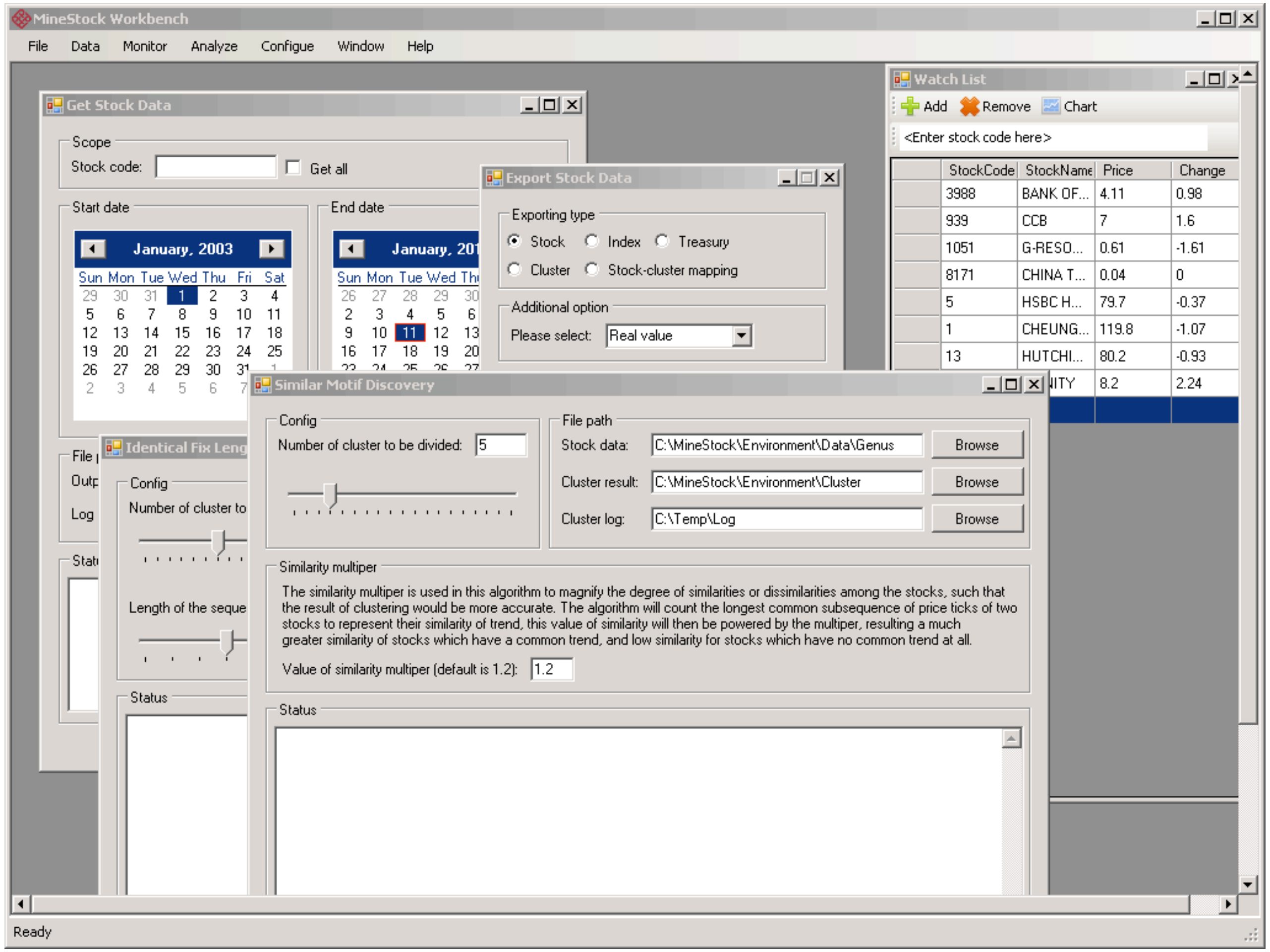
Table 5: Code metrics of the project modules

[a] Measures ease of code maintenance. Higher values are better.

[b] Measures number of branches. Lower values are better.

[c] Measures length of object inheritance hierarchy. Lower values are better.

[d] Measures number of classes that are referenced. Lower values are better.



Stock code: □ Get all

Exporting type: ● Stock ○ Index ○ Treasury ○ Cluster ○ Stock-cluster mapping

Please select: Real value

<Enter stock code here>

StockCode	StockName	Price	Change
3988	BANK OF...	4.11	0.98
939	CCB	7	1.6
1051	G-RESO...	0.61	-1.61
8171	CHINA T...	0.04	0
5	HSBC H...	79.7	-0.37
1	CHEUNG...	119.8	-1.07
13	HUTCHI...	80.2	-0.93
...ITY		8.2	2.24

Number of cluster to be divided: 5

Stock data: C:\MineStock\Environment\Data\Genus

Cluster result: C:\MineStock\Environment\Cluster

Cluster log: C:\Temp\Log

The similarity multiper is used in this algorithm to magnify the degree of similarities or dissimilarities among the stocks, such that the result of clustering would be more accurate. The algorithm will count the longest common subsequence of price ticks of two stocks to represent their similarity of trend, this value of similarity will then be powered by the multiper, resulting a much greater similarity of stocks which have a common trend, and low similarity for stocks which have no common trend at all.

Value of similarity multiper (default is 1.2): 1.2

Figure 7: A preview screenshot of MineStock Workbench

#	Start Date	Duration	Task	Progress
A1	October 7, 2010	1 week	Final amendment on project specifications and approval	100%
A2	October 7, 2010	3 weeks	Further study on data mining algorithms	90%
A3	October 14, 2010	1 week	System design	90%
A4	October 21, 2010	1 week	Implementation of CMD: data collection module	100%
A5	October 28, 2010	1 week	Implementation of CMD: k-means clustering module	100%
A6	November 4, 2010	3 weeks	Implementation of CMD: other clusting modules	90%
A7	November 25, 2010	1 week	Design of GUI: data collection, clustering, view clustering result	100%
A8	December 2, 2010	1 week	Implementation of GUI: data collection, clustering	90%
A9	December 9, 2010	2 weeks	Implementation of GUI: charting library integration	0%
A10	December 23, 2010	1 week	Implementation of GUI: view clustering result	0%
A11	December 30, 2010	2 weeks	Preparation of mid-term progress report	100%
		17 weeks	Total	72%

Table 6: Project schedule and progress (term 1)

7 Miscellaneous

During the project progression, the software deliverable has been named as 'MineStock Workbench', in order to create a clearer image of its purpose. Also, the project has been renamed as 'Project MineStock' accordingly.

References

[1] G. Gan, C. Ma, J. Wu, *Data Clustering: Theorey, Algorithms, and Applications.* Philadelphia: Society for Industrial and Applied Mathematics, 2007, pp.71 & 161.

[2] J. Han, M. Kamber, *Data Mining - Concepts and Techniques*, 2nd Edition. Sao Francisco: Elsevier, 2006, pp. 493-497.

[3] C. H. Ma, C. C. Chan, "UPSEC: An Algorithm for Classifying Unaligned Protein Sequence into Functional Families." *Journal of Computional Biology*, 15, 4, pp. 431-443, May 2008.

[4] A. Mueen, E. Keogh, Q. Zhu, S. Cash, B. Westover, "Exact Discovery of Time Series Motifs." University of California, Feb 23, 2009. [Online]. Available: http://www.siam.org/proceedings/datamining/2009/dm09_045_mueena.pdf. [Accessed: Feb 14, 2010]

[5] H. M. Makowitz, "The Early History of Portfolio Theory: 1600-1980," in *Harry Markowitz: Selected Works*, H. M. Makowitz, Ed. New Jersey: World Scientific Publishing Company, pp. 5-16, 2009.

[6] H. M. Makowitz, "Portfolio Selection," *The Journal of Finance*, **7**, 1, pp. 77-91, Mar 1952.

[7] D. McNulty, "Bettering Your Portfolio With Alpha And Beta," *Investopedia*, Jan 11, 2011. [Online]. Available: http://www.investopedia.com/articles/07/alphabeta.asp. [Accessed: Jan 11, 2011]

[8] M. Choudhry, *An Introduction to Value-at-risk*, 4th Edition. West Sussex: Wiley, 2006, pp. 36-37.

[9] S. H. Siu, S. M. Mei, Y. C. Tang, M. K. Tong, "Value-at-risk (Simulated Loss of a Portfolio)." Course Project, Computational Finance, The Hong Kong Polytechnic University, Hong Kong, 2010.

Cover Sheet for Mid-Term Check Point Progress Report

Final Year Project
Department of Computing

[√] **BSc Scheme in Computing (61031)**
[ ] **BAC –PTE (61025)**

(Please √ as appropriate)

Student Name: TONG Man Kin	Student Number: 08502621D
Supervisor Name: Prof. CHAN Chun Chung Keith	Date: 12th January, 2011

Attach this page, with answers to the following questions, as the ***cover sheet*** to your mid-term check point progress report and have it signed by your supervisor. Then, submit **ONE** copy to General Office (Room: PQ806) by the deadline specified in the schedule. Please follow the format describe in the guideline.

◆ **have you been meeting your supervisor regularly**

No regular meeting has been made, but I will contact my supervisor whenever there is problem which cannot be solved. Also, status reporting is done by E-mails.

◆ **roughly, how much time have you spent on your project up till now**

Around 200 hours.

◆ **do your supervisor and coexammer suggest any modification to your project proposal and have you revised your proposal to their satisfaction**

No modification to the project proposal has been made.

◆ **are there any major difficulties encountered**

Yes, they include the occasional inaccuracy of data source, underestimated workload on algorithm design and implementation. Please refer to the report for more detail.

◆ **how are these difficulties, if any, handled**

The prior problem is handled by extra programming effort. Please refer to the report for more detail.

◆ **are there any outstanding difficulties that you need to seek help (from supervisor, project coordinator, technicians, etc)**

I have sought advice from my supervisor and algorithm design and implementation.

◆ **are there any major deviation from your original problem definition and objectives**

Currently no deviation is proposed.

◆ **what are the justifications of the deviation, if any**

N/A

◆ **are you on schedule according to your own project plan**

This project is slightly delayed. However, it is still expected to be completed on time without changes in major specifications.

◆ **if you are behind the schedule, explain why**

I have switched the scheme of study from full computing degree to a major and minor option in this year. Therefore, I am studying a greater number of credits, and thus facing a higher amount of workload compared with other final year students.

◆ **any other things which you want to report**

N/A

Responses from supervisor (please check):

i. totally agree with the content of the report [✓]
ii. partially agree with the content of the report [ ]
iii. totally disagree with the content of the report [ ]

Supervisor Comments (please briefly comment on the performance of the student and in case of response ii. and iii, state why you disagree)

Supervisor Name: Keith Chan Signature: