Stock Price Moving Pattern Classification and Recognition Project Overview and Plan

Author: Yuyang Liu
Student ID: 10660111

Supervised by Dr. Xiaojun Zeng

School of Computer Science
University of Manchester

Signature _____

Date _____ / _____ / _____

Contents

Chapter 1

Introduction

1.1 Overview

As one of the most popular financial products, stock takes high risks for high rewards. Its price is affected by innumerous uncertain factors such as policies of local government, breaking of diseases, etc. [40]. To find strategies that can maximize profits and reduce risks, stock investors may need to extract the latent trading and price moving patterns of stock market [20]. However, due to the randomness, complexity and uncertainty existing in stock market, those patterns are elusive and hard to be discovered [26]. [5] states that even though researchers made massive efforts to teaching machine to recognize data patterns automatically in the past decades, humans are still the best pattern recognizers in most cases so far. Based on whether expert derived patterns exists (and being used) or not, pattern discovery methods can be classified into two categories: supervised methods and unsupervised methods [18]. In terms of stock price, supervised methods require fluctuation patterns pre-defined by experts, while unsupervised methods do not need prior knowledge of interesting structures. Consider that stock market is dynamic and rapidly developing and stock price series shows no periodic trajectory (at least in the short term) [36], the pre-defined patterns may need to be updated frequently (relies heavily on experts). Increasing stock price data volume makes manually defining patterns much more challenging [5]. These problems make stock researchers such as [42] focus more on unsupervised pattern discovery methods.

1.2 Motivation

The majority of existing stock-related researches aim to predict the stock price or moving trending. However, [36] already shown that stock pirce is unpredictable at least in the short term. Instead of directly predicting the stock pirce, we hope to retrieve the latent stock price moving patterns to help decision making. As mentioned in section 1.1 supervised methods require domain knowledge, therefore, we seek help from unsupervised methods. Most previous works such as [18, 42] mainly try to modify clustering algorithms to get better performance. They use the original data (or part of the data) as

the input vector, ignore the usage of representation learning. In this project, we will adapt some novel time-series data representation methods, especially the path signature feature (PSF) used in [7, 9, 28], combined with traditional clustering algorithms to better reveal the hidden fluctuation patterns of stock pirce.

1.3 Aims and Objectives

The general goal of this project is to categorize different stock price trajectories into several classes. It can be detailed as follows:

1. Finding a proper representation method that can reduce the data dimension and better reveal the feature of a stock price trajectory.
2. Grouping those generated feature vectors to get the final set of hidden patterns based on clustering algorithms.

To achieve the goal, following works are need to be done:

1. Review the recent works related to time-series data representation and clustering algorithms.
2. Collect stock price data from open-source websites.
3. Preprocess data. In this process, we will apply the common numerical data processing pipeline to our data set.
4. Apply data representation algorithms to our data, analyze the effectiveness of those transformation in different aspects. This may require numerical analysis and visualization.
5. Apply multiple clustering algorithms to the generated representation.
6. Evaluate and compare the performance of those clustering algorithms based on certain metrics.
7. Visualize the patterns generated by clustering algorithms.
8. Integrate pattern matching algorithm into this project. Apply it to test set for the final evaluation.

1.4 Description of the Work

Project scope:

In this project, we will analyze some of the US-based stocks trading on the NYSE, NASDAQ, and NYSE.

In our dataset, the ealierest records could date back to 1970s, and it's hard to analyze the whole series. Instead, we will split their historical series into smaller fragments. To get more stable patterns, we mainly focus on the long-term price fluctuation, meaning that each smaller fragment will cover more than 6 months data. Data compression and representation algorithms will be applied to each fragment to generate the feature vectors of them. Then those feature vectors will be grouped by clustering algorithms, the centroid of each group is regarded as a unique pattern. Finally, the value of the generated patterns will be examined by using pattern matching methods to checking whether similar patterns can be found in test data.

The final deliverables of this project will be:

1. The data set used in this project
2. The code implementation of this project
3. The dissertation with experimental results, analysis and visualization of this project

1.5 Report Structure

The remaining chapters are organised as follows: Chapter 2 briefly introduces works related to time-series data representation and clustering. Chapter 3 shows the detailed methodology used in this project. Chapter 4 includes the ethical consideration of this project. Chapter 5 evaluates the potential risks of this project. Chapter 6 introduces some metrics that can be used to evaluate the performance of the algorithms used in this project. Chapter 7 includes our progress and future work.

Chapter 2

Background and Related Works

2.1 Literature Review

For decades, researchers attempt to reveal the rules behind stock price's changing. Their works can be categorized into two basic groups: (1) directly predicting stock price, including using traditional "technical analysis", neural network and fuzzy time-series, etc.; (2) extracting interesting patterns/turning points for decision making, including using clustering algorithms and charts, etc. In this project, we focus on works that mainly utilizes clustering algorithms to find the similarity in stock trends. Similar to other time-series data, stock price data are long sequences with varying length, therefore, they cannot be directly analyzed. To make the analysis tractable, [18] proposed a pattern discovery framework for stock time series, which includes three phases: (1) segmentation phase: split intractable long sequences into smaller fragments; (2) clustering phase: group the fragments into clusters, the centroids of clusters are regarded as unique patterns; (3) matching phase: use pattern matching algorithms to find matches of the generated patterns in test data. To mitigate the time complex problem brought by the length of patterns, they used a perceptually important point (PIP) identification algorithm to compress the data. [4] used chaotic map clustering (CMC) algorithm to identify the temporal patterns of the stock prices, each company/stock is assigned to a map, and similarity between stocks/companies were measured by the coupling strengths between maps, their work proved that co-movement of stock price exists in the same industrial branch. [27] studied on the association between stocks and used K-means algorithm to group stocks, they proved that similar stock price fluctuation can happen within geographic regions. [10] manually defined a stock price pattern called PIP bull-flag based on the pattern defined by [24, 25], proposed two matching algorithms to find pre-defined patterns (a timing on the stock series), and used template matching technique to help investment strategy making. [30] examined the performance of K-means, SOM and Fuzzy C-means clustering algorithms on stock data with 9 evaluation criteria, and found that the clusters generated by K-means were the most compact. [1] argued that previous one-phase approaches can not deep into smaller granularity, and proposed a three-phase clustering method for stock clustering: (1) approximate

clustering; (2) purifying and summarization; (3) merging. [23] used the trajectories before stock price jumps rather than the whole data series to classify stocks, and found that the patterns around turning points are more guiding.

2.2 Related Works

This section gives a brief review of techniques used in this project, including time-series data compression, path signature, and clustering algorithms.

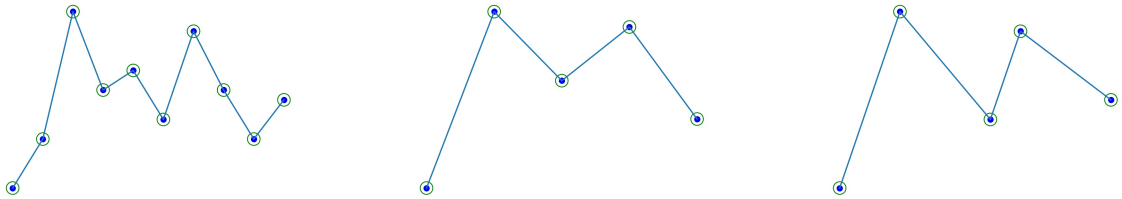
2.2.1 Time-series data compression

Stock price records are typical time-series data, and they have natures such as large data volume, high dimensionality, etc. Analyzing the raw data may need huge computing resource, therefore, in most algorithms, the data are compressed. There are several methods to reduce the dimension of the original data, simple approaches include: (1) simpling [2], which randomly selects n points from a time series (n is the dimension after compression); (2) piecewise aggregate approximation (PAA) [21, 39], which segments the time series into n sub trajectories, and uses the numerical mean of each trajectory to represent the whole time series.

More promising methods aim to use the perceptually important points (PIP) to represent the time series [17]. The PIP identification algorithm is first proposed by [11] and then used wildly in time series data analysis. Given a time series $P = \{P_1, P_2, \cdots, P_k\}$, where $P_t \in P$ is a data point, PIP identification process will calculate the importance of all the data points, and the first n points with higher importance will be used to represent the whole series. This process works as follows: the start point P_1 and the end point P_2 in P are the first two PIPs. The third point will be the one with maximum distance to the first two PIPs. The order of remaining PIPs will be decided by their vertical distance to the line crossing its two adjacent PIPs. This process continues until n PIPs are found (n is the reduced dimension) or all points in P are ordered. Figure 2.1(b) and 2.1(c) show the compression results of PPA and PIP algorithms respectively.

2.2.2 Path signature

Path signature is the core part in rough path theory, it was first studied by [9]. As a novel vector representation for sequential data, it has already be used in fields such as financial data analysis [19],



(a) Original time series (b) PPA Result (c) PIP Result

Figure 2.1: Compression Results

handwritten text recognition [37], human action recognition [38], etc. The brief introduction could be: given a sequence (continuous or discrete) $P = (P_1, P_2, \cdots, P_T) \in \mathbb{R}^{T \times D}$, where T is number of points in the sequence, D is the dimension of each data point, let P_t^i denotes the i-th attribute of point P_t $(1 \leq t \leq T, 1 \leq i \leq D)$, the whole sequence can be represented by n-fold iterated integral over the sequence (n could be infinite). The simplest case is that when $D = 1$, the 1-fold representation of P is a real value defined as:

$$S(P)_{1,T}^{1} = \int_{1<t\leq T} dP_t^1 = P_T^1 - P_1^1 \tag{2.1}$$

Similarly, the 2-fold representation is also a real value:

$$S(P)_{1,T}^{11} = \int_{1<t\leq T} S(P)_{1,T}^{1} dP_t^1 = \frac{1}{2}(P_T^1 - P_1^1)^2 \tag{2.2}$$

The k-fold representation is:

$$S(P)_{1,T}^{11\cdots 1} = \frac{1}{k!}(P_T^1 - P_1^1)^k \tag{2.3}$$

The final signature representation of sequence P is the vector $(S(P)_{1,T}^{1}, S(P)_{1,T}^{11} S(P)_{1,T}^{11\cdots 1}) \in \mathbb{R}^k$, whose dimension k is the number of fold wanted by users.

When D = 2, 1-fold representation has 2 elements defined as follows:

$$\begin{aligned} S(P)_{1,T}^{1} &= \int_{1<t\leq T} dP_t^1 = P_T^1 - P_1^1 \\ S(P)_{1,T}^{2} &= \int_{1<t\leq T} dP_t^1 = P_T^2 - P_1^2 \end{aligned} \tag{2.4}$$

2-fold representation has $D^2 = 4$ elements:

$$\begin{aligned} S(P)^{11}_{1,T} &= \int_{1<t\leq T} S(P)^1_{1,T} dP^1_t = \frac{1}{2}(P^1_T - P^1_1)^2 \\ S(P)^{22}_{1,T} &= \int_{1<t\leq T} S(P)^1_{1,T} dP^1_t = \frac{1}{2}(P^2_T - P^2_1)^2 \\ S(P)^{12}_{1,T} &= \int_{1<t_1\leq T} \int_{1<t_2\leq T} dP_{t_1} dP_{t_2} \\ S(P)^{21}_{1,T} &= \int_{1<t_2\leq T} \int_{1<t_1\leq T} dP_{t_2} dP_{t_1} \end{aligned} \tag{2.5}$$

In general, when D = d, the i-th element of k-fold representation can be seen in eqaution 2.6, where $(n_1, n_2, \cdots, n_k) \in \{1, \cdots, k\}$:

$$S(P)^{n_1,n_2,\cdots,n_k}_{1,T} = \int_{1<t_k\leq T} \cdots \int_{1<t_3\leq t_4} \int_{1<t_2\leq t_3} dP^{n_1}_{t_1} dP^{n_2}_{t_2} \cdots dP^{n_k}_{t_k} \tag{2.6}$$

The final signature of the whole path P is the collection of all elements in every fold defined as follows:

$$\begin{aligned} S(P)_{1,T} = (1, &S(P)^1_{1,T}, \cdots, S(P)^D_{1,T}), \\ &S(P)^{1,1}_{1,T}, \cdots, S(P)^{2,1}_{1,T}, \cdots, S(P)^{D,1}_{1,T}, \cdots, S(P)^{D,D}_{1,T}, \\ &S(P)^{1,1,\cdots,1}_{1,T}, \cdots, S(P)^{n_1,n_2,\cdots,n_k}_{1,T}, \cdots, S(P)^{D,D,\cdots,D}_{1,T}, \cdots) \end{aligned} \tag{2.7}$$

Conventionally, the first term is set to 1. With the increment of fold k, the collection could be infinite. However, in practice, there is no need of using a large k, users can decide the k based the the dimension formula of the final vector: $\omega(D, k) = (D^{k+1} - 1)(D - 1)^{-1}$. The truncated elements are regarded as the path signature feature of the original sequence, and then can be used in further analysis.

2.2.3 Clustering algorithms

Clustering is a classical and important task in unsupervised learning. We will use it to find the stcok price moving patterns. Traditional clustering methods can be summarized into following 9 categorizes. Since the main focus of this project is not about clustering algorithms, their technical details are not discussed.

1. Partition based: clustering algorithms that assume that the center of data points is the center of the corresponding cluster. Typical algorithms include K-means [29] and K-medoids [31].

2. Hierarchy based: clustering algorithms that generate clusters based on the hierarchical relationship among data points, either in a bottom-up or top-down way. Typical algorithms include BIRCH [41].

3. Fuzzy theory based: clustering algorithms that use possibility to represent belonging relationship among objects rather than binary status 0 or 1. One data points could belongs to several clusters at the same time. Typical algorithms include FCM [6], FCS [13].

4. Distribution based: clustering algorithms that assume there were multiple distributions in the original data, and data points sampled from same distribution belong to a same cluster. Typical algorithms include GMM [32].

5. Density based: clustering algorithms that assume data points in high density region belong to the same cluster. Typical algorithms include DBSCAN [15] and Mean-shift [12].

6. Graph theory based: clustering algorithms that regard data points as nodes and their relationship as edges in a graph. Typical algorithms include CLICK [34].

7. Grid based: clustering algorithms that transform original data space into a grid structure with fixed size. Typical algorithms include STING [35].

8. Fractal theory based: clustering algorithms that attempt to divide data points into multiple groups that share some common characters with the original data. Typical algorithms include FC [3].

9. Model based: clustering algorithms that pre-define a model for each cluster and mathcing data points best fitting for that model. Typical algorithms include COBWEB] [16] and SOM [22].

Most previous works try to improve the quality of extracted patterns by improving the performance of clustering algorithms. However, we doubt that with a proper data representation method, a simple clustering algorithm can generate acceptable results. Therefore, in this project, we seek to improve the quality of extracted patterns by finding a better representation method of stock price records.

Chapter 3

Research Methodology

Problem Definition:

The main task of this project is to find the latent patterns of stock price changing. In this project, we define the stock price data as $X = (X_1, \cdots, X_N) \in \mathbb{R}^{N \times T \times D}$, where $X_m = (x_{m_1}, \cdots, x_{m_T}) \in \mathbb{R}^{T \times D}$ represents all historical series of a single stock, N is the number of stocks, T is the number of time slices, D is the dimension of a single data. We further divide the task into 6 sub-tasks: (1) time-series data preprocessing; (2) feature extraction; (3) pattern discovery; (4) pattern matching; (5) evaluation.

3.1 Preprocessing

Data are the core part of data mining tasks, high quality data can improve the performance of models and algorithms. The aim this phase is processing the raw data we collect into more structured data, main steps include:

1. Filling missing values: the collected data set may have incomplete data, this common problem is caused by various factors such as recordding error, market suspension, etc. In practice, there are several methods than can be used to fill missing values, including: (1) nearest neighborhood substitution; (2) mean value substitution; (3) regression, etc. In this project, we will use the second method. The missing value of a data point will be filled by the average of its nearest two neighbours.

2. Normalization: normalization is a standard process existing in most data mining tasks. It aims to change the values of all data features to a common scale while reserving the differences in the ranges of values. In terms of stock price data, it mainly helps to uniform monetary unit.

3. Segmentation: time-series data is numerical and continuous, it's crucial to split them into discrete pieces, especially in trend analysis [17]. Common discretization methods include: (1) sliding window; (2) PIP based segmentation; (3) minimum message length (MML), etc. In this project, we aim to find the long-term pattern of stock price fluctuation, and hence will use sliding window method with the window size more than 6 months.

3.2 Feature Extraction

Feature engineering attempts to find the latent features of original data that can improve the performance of machine learning algorithms. We will use it in this project for two purposes:

1. Data compression: as stated by [18], the large size of time series data could cause time complexity problem. They found that with the increment of the length of patterns, the run time of pattern discovery process grows exponentially. One method to mitigate this problem is compressing the sequence. In this project, we will use pecewise aggre-gate approximation (PAA) and perceptually important points (PIP) for data compression.

2. Data representation: each segment of our divided data is a matrix rather than a single vector. To fed them into clustering models, representation/transformation step is required. In this project, we will use the path signature as the representation method, and examine its effect in stock price analysis.

It is worth noting that using data compression and representation algorithms may cause information loss. The effect will be examined in our experiment.

3.3 Pattern Discovery

In this phase, we will follow the common approach in unsupervised pattern discovery tasks: using clustering algorithms to find interesting groups. In detail, we will apply several clustering algorithms to the generated features of all segments, each algorithm will produce a set of groups. The centre (numerical average) of each group will be treated as a unique pattern.

3.4 Pattern Matching

To examine whether the generated patterns have practical value, we then will apply pattern matching techniques to the test data. In detail, we will apply the same pre-processing pipeline to the test data. Since the generated patterns and each segment are vectors, we then will use common vector similarity measures methods such as pearson correlation coefficient and cosine similarity, with a pre-defined threshold, to check whether similar patterns can be find in test set.

3.5 Evaluation

The evaluation will be conducted in all 4 phases mentioned above. Differences are that the evaluation in phase 3.1 and 3.2 are mainly in the form of visualization, while that in phase 3.3 and 3.4 are mainly based on numerical comparison with objective criteria, details can be found in chapter 6.

Chapter 4

Ethics and Professional Considerations

Our project is about stock market analysis, the follwing statements prove that this project will not cause ethic issues:

1. We have read the University of Manchester Code of Research Conduct and Research Ethics
2. The stock price data used in this project are open-sourced, we collected them from Kaggle
3. The data can be accessed and downloaded by every researcher
4. None of these data will be displayed in the outcome of this project. And none of these data will be used beyond this project
5. The use of this secondary data or archive has no risk of disclosure of the identity of individuals
6. We have complied with the data access requirements of the supplier
7. This project does not involve participation of people
8. This project does not involve access to filed sites and animals

Chapter 5

Risk Consideration

The research part of this project does not contain potential risks, however, similar to other engineering projects, accidents could happen on the project management level. To avoid accidents and ensure the project progresses smoothly, periodically reviewing on the whole project is required. Measures include:

1. Periodically backing up project files in case of physical damage on devices.
2. Consulting with supervisor immediately once emergency issue happens.
3. Encrypting project files in case of unauthorized access.

Chapter 6

Project Evaluation

This chapter introduces the evaluation methods and criteria used in this project. As mentioned in section 3.5, we will examine each pahse and the final outcome separately. Details can be found below.

6.1 Preprocessing Phase Evaluation

The evaluation of this phase is relatively intuitive since it is mainly about data review. We will compare the original data and the processed data with plots.

6.2 Feature Extraction Phase Evaluation

In this phase, we will mainly examine the effect of different data compression algorithms. Similar to preprocessing phase, the evaluation will be conducted in the form of diagram since there is no numerical method that can directly score them. The deeper examination of both data compression and path signature will be conducted in pattern discovery phase, where objective numerical standards can be applied.

6.3 Pattern Discovery Phase Evaluation

In this phase, we will experiment with different combination of data representation methods (path signature and simple concatenation), compression algorithms (PPA and PIP) and clustering algorithms (to be decided). The assessment criteria are the common metrics used for clustering algorithms without the requirement of ground truth class assignments, including:

1. Silhouette Coefficient [33]: for each data point, its silhouette coefficient is composed of two scores:
    - **a:** the average distance between a data point and all other data points in the same cluster
    - **b:** the average distance between a data point and all other data points in the next nearest cluster

The Silhouette Coefficient s for a single data point is defined as follows, where s ranges from -1 to +1:

$$s = \frac{b - a}{\max(a, b)} \tag{6.1}$$

The final Silhouette Coefficient is defined as the mean of the Silhouette Coefficient of each data point. Higher value means better clustering results.

2. Calinski-Harabasz Index [8]: given a data set E of size n_E, and the number of grouped clusters k, the Calinski-Harabasz score s is defined as follows:

$$s = \frac{tr(B_k)}{tr_(W_k)} \times \frac{n_E - k}{k - 1} \tag{6.2}$$

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T \tag{6.3}$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E) c_q - c_E)^T \tag{6.4}$$

Where $tr(B_k)$ and $tr(W_k)$ are trace of the between group dispersion matrix and within-cluster dispersion matrix respectively, C_q represents the points in cluster q, c_q represents the centre of q, c_E represents the centre of E and n_q represents the number of points in q. Higher value means better clustering results.

3. Davies-Bouldin Index [14]: given two cluster C_i and C_j generated from the same data set without overlapping, their similarity R_{ij} is defined as follows:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{6.5}$$

Where s_i is the average distance between each point in cluster i and the centroid of i, d_{ij} is the distance between centroids of clusters i and j. The Davies-Bouldin index is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij} \tag{6.6}$$

k is the number of clusters. Lower Davies-Bouldin index means better clustering results.

6.4 Pattern Matching Phase Evaluation

As mentioned in section 3.4, the evaluation of this phase is mainly about vector similarity measurement. Given two pattern vectors X and Y, the possible indexs we may use are:

1. Euclidean distance:
$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{6.7}$$

2. Cosine value:
$$cos(X,Y) = \frac{X \cdot Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{6.8}$$

3. Pearson correlation coefficient:
$$PC(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{X}) \times (y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{X})^2} \times \sqrt{\sum_{i=1}^{n}(y_i - \bar{Y})^2}} \tag{6.9}$$

We then will examine the practical value of our method based on those indexs.

Chapter 7

Project Plan

This chapter covers the progress made so far, and the remaining tasks needed to be done. Figure 7.1 shows the gantt chart of our project. Diamonds with orange color are milestones of the project.

7.1 Progress

The first phase of our project is carried out smoothly, followings are progress we made so far:

1. Literature review: we searched and read more than 40 papaer related to our project, have a basic sense of which techniques may be required in this Project.

2. Data collection: we searched and downloaded a stock price dataset called **Huge Stock Market Dataset**, which contains full historical daily price and volume data for all US-based stocks and ETFs trading on the NYSE, NASDAQ, and NYSE MKT.

3. POP writing: we wrote the POP based on the marking scheme.

4. Tool selection: we decided to use Python as the main programming language since it has various data structures that are easy to manipulate.

5. Code implementation (partial): we wrote some sketch of functions that will be used in our implementation.

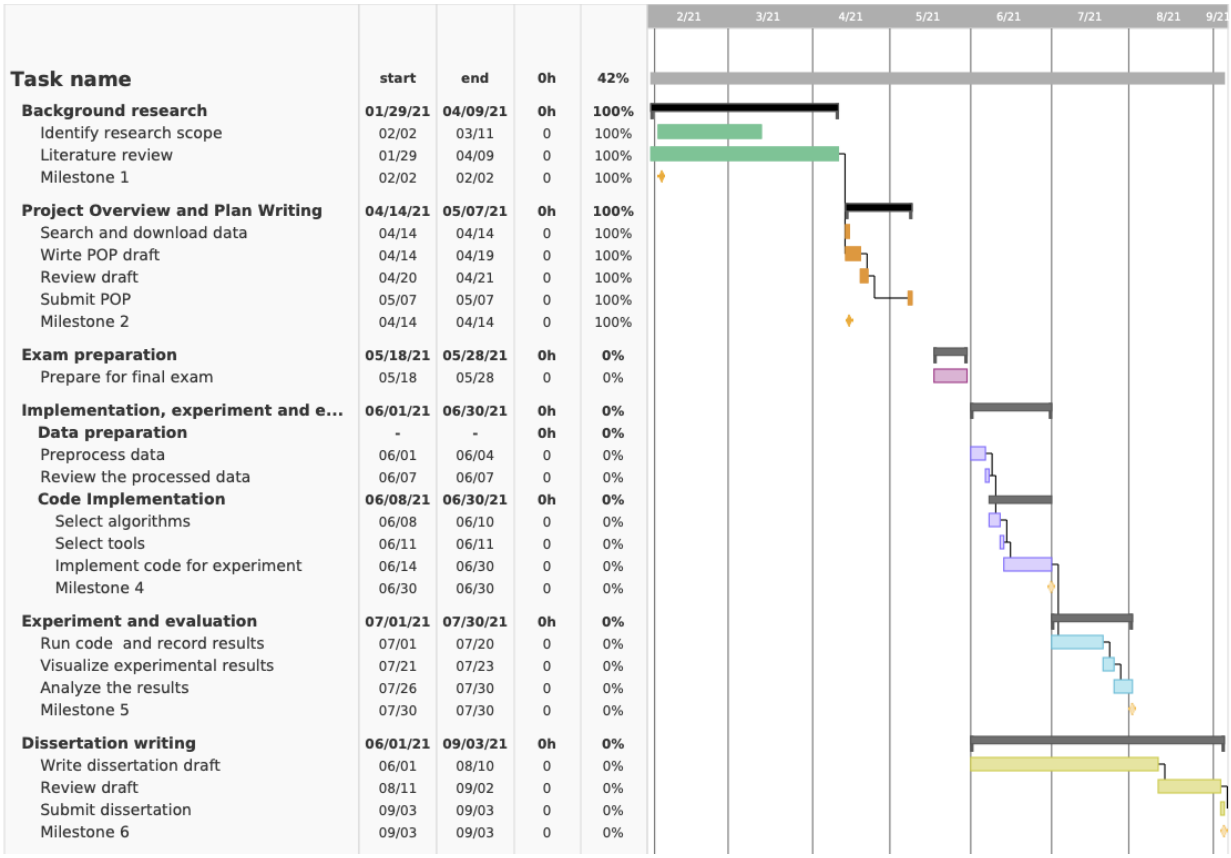
7.2 Future Work

Generally, the remaining tasks of this project include:

1. Completion of code implementation, including (1) select proper algorithms; (2) decide programming language and libraries that will be used in this project; (3) add or re-implement functions based on our requirements; (4) design experiment methods and implement functions for evaluation.

2. Experiment and evaluation, including (1) fed data to our models and record the output; (2) visualize the output; (3) evaluate the experimental results based on pre-defined metrics.

3. Dissertation writing.



Task name	start	end	0h	42%
Background research	**01/29/21**	**04/09/21**	**0h**	**100%**
Identify research scope	02/02	03/11	0	100%
Literature review	01/29	04/09	0	100%
Milestone 1	02/02	02/02	0	100%
Project Overview and Plan Writing	**04/14/21**	**05/07/21**	**0h**	**100%**
Search and download data	04/14	04/14	0	100%
Wirte POP draft	04/14	04/19	0	100%
Review draft	04/20	04/21	0	100%
Submit POP	05/07	05/07	0	100%
Milestone 2	04/14	04/14	0	100%
Exam preparation	**05/18/21**	**05/28/21**	**0h**	**0%**
Prepare for final exam	05/18	05/28	0	0%
Implementation, experiment and e...	**06/01/21**	**06/30/21**	**0h**	**0%**
Data preparation	**-**	**-**	**0h**	**0%**
Preprocess data	06/01	06/04	0	0%
Review the processed data	06/07	06/07	0	0%
Code Implementation	**06/08/21**	**06/30/21**	**0h**	**0%**
Select algorithms	06/08	06/10	0	0%
Select tools	06/11	06/11	0	0%
Implement code for experiment	06/14	06/30	0	0%
Milestone 4	06/30	06/30	0	0%
Experiment and evaluation	**07/01/21**	**07/30/21**	**0h**	**0%**
Run code and record results	07/01	07/20	0	0%
Visualize experimental results	07/21	07/23	0	0%
Analyze the results	07/26	07/30	0	0%
Milestone 5	07/30	07/30	0	0%
Dissertation writing	**06/01/21**	**09/03/21**	**0h**	**0%**
Write dissertation draft	06/01	08/10	0	0%
Review draft	08/11	09/02	0	0%
Submit dissertation	09/03	09/03	0	0%
Milestone 6	09/03	09/03	0	0%

Figure 7.1: Gantt Chart

Bibliography

[1] Aghabozorgi, S., and Teh, Y. W. Stock market co-movement assessment using a three-phase clustering method. *Expert Systems with Applications 41*, 4 (2014), 1301–1314.

[2] Åström, K. J. On the choice of sampling rates in parametric identification of time series. *Information Sciences 1*, 3 (1969), 273–278.

[3] Barbará, D., and Chen, P. Using the fractal dimension to cluster datasets. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (2000), pp. 260–264.

[4] Basalto, N., Bellotti, R., De Carlo, F., Facchi, P., and Pascazio, S. Clustering stock market companies via chaotic map synchronization. *Physica A: Statistical Mechanics and its Applications 345*, 1-2 (2005), 196–206.

[5] Basu, J. K., Bhattacharyya, D., and Kim, T.-h. Use of artificial neural network in pattern recognition. *International journal of software engineering and its applications 4*, 2 (2010).

[6] Bezdek, J. C., Ehrlich, R., and Full, W. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences 10*, 2-3 (1984), 191–203.

[7] Boedihardjo, H., Geng, X., Lyons, T., and Yang, D. The signature of a rough path: uniqueness. *Advances in Mathematics 293* (2016), 720–737.

[8] Caliński, T., and Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods 3*, 1 (1974), 1–27.

[9] Chen, K.-T. Integration of paths–a faithful representation of paths by noncommutative formal power series. *Transactions of the American Mathematical Society 89*, 2 (1958), 395–407.

[10] Chen, T.-l., and Chen, F.-y. An intelligent pattern recognition model for supporting investment decisions in stock market. *Information Sciences 346* (2016), 261–274.

[11] Chung, F.-L., Fu, T.-C., Luk, R., Ng, V., et al. Flexible time series pattern matching based on perceptually important points.

[12] Comaniciu, D., and Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence 24*, 5 (2002), 603–619.

[13] Dave, R. N., and Bhaswan, K. Adaptive fuzzy c-shells clustering and detection of ellipses. *IEEE Transactions on Neural Networks 3*, 5 (1992), 643–662.

[14] Davies, D. L., and Bouldin, D. W. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, 2 (1979), 224–227.

[15] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (1996), vol. 96, pp. 226–231.

[16] Fisher, D. H. Knowledge acquisition via incremental conceptual clustering. *Machine learning 2*, 2 (1987), 139–172.

[17] Fu, T.-c. A review on time series data mining. *Engineering Applications of Artificial Intelligence 24*, 1 (2011), 164–181.

[18] Fu, T.-c., Chung, F.-l., Ng, V., and Luk, R. Pattern discovery from stock time series using self-organizing maps. In *Workshop Notes of KDD2001 Workshop on Temporal Data Mining* (2001), vol. 1, Citeseer.

[19] Gyurkó, L. G., Lyons, T., Kontkowski, M., and Field, J. Extracting information from the signature of a financial data stream. *arXiv preprint arXiv:1307.7244* (2013).

[20] Hu, Y., Feng, B., Zhang, X., Ngai, E., and Liu, M. Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications 42*, 1 (2015), 212–222.

[21] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems 3*, 3 (2001), 263–286.

[22] Kohonen, T. The self-organizing map. *Proceedings of the IEEE 78*, 9 (1990), 1464–1480.

[23] Kong, A., Azencott, R., and Zhu, H. Pattern recognition in trading behaviors before stock price jumps: new method based on multivariate time series classification. *arXiv preprint arXiv:2011.04939* (2020).

[24] Leigh, W., Modani, N., Purvis, R., and Roberts, T. Stock market trading rule discovery using technical charting heuristics. *Expert Systems with Applications 23*, 2 (2002), 155–159.

[25] Leigh, W., Purvis, R., and Ragusa, J. M. Forecasting the nyse composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. *Decision support systems 32*, 4 (2002), 361–377.

[26] Li, H., Shen, Y., and Zhu, Y. Stock price prediction using attention-based multi-input lstm. In *Asian Conference on Machine Learning* (2018), PMLR, pp. 454–469.

[27] Liao, S.-H., Ho, H.-h., and Lin, H.-w. Mining stock category association and cluster on taiwan stock market. *Expert Systems with Applications 35*, 1-2 (2008), 19–29.

[28] Lyons, T. Rough paths, signatures and the modelling of functions on streams. *arXiv preprint arXiv:1405.4537* (2014).

[29] MacQueen, J., et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, Oakland, CA, USA, pp. 281–297.

[30] Nanda, S., Mahanty, B., and Tiwari, M. Clustering indian stock market data for portfolio management. *Expert Systems with Applications 37*, 12 (2010), 8793–8798.

[31] Park, H.-S., and Jun, C.-H. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications 36*, 2 (2009), 3336–3341.

[32] Rasmussen, C. E., et al. The infinite gaussian mixture model. In *NIPS* (1999), vol. 12, pp. 554–560.

[33] Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics 20* (1987), 53–65.

[34] Sharan, R., and Shamir, R. Click: a clustering algorithm with applications to gene expression analysis. In *Proc Int Conf Intell Syst Mol Biol* (2000), vol. 8, p. 16.

[35] Wang, W., Yang, J., Muntz, R., et al. Sting: A statistical information grid approach to spatial data mining. In *VLDB* (1997), vol. 97, pp. 186–195.

[36] Wei, J., and Huang, J. An exotic long-term pattern in stock price dynamics. *PLoS One 7*, 12 (2012), e51666.

[37] Xie, Z., Sun, Z., Jin, L., Feng, Z., and Zhang, S. Fully convolutional recurrent network for handwritten chinese text recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (2016), IEEE, pp. 4011–4016.

[38] Yang, W., Lyons, T., Ni, H., Schmid, C., and Jin, L. Developing the path signature methodology and its application to landmark-based human action recognition. *arXiv preprint arXiv:1707.03993* (2017).

[39] Yi, B.-K., and Faloutsos, C. Fast time sequence indexing for arbitrary lp norms.

[40] Zhang, L., Aggarwal, C., and Qi, G.-J. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (2017), pp. 2141–2149.

[41] Zhang, T., Ramakrishnan, R., and Livny, M. Birch: an efficient data clustering method for very large databases. *ACM sigmod record 25*, 2 (1996), 103–114.

[42] Zhang, X., Liu, J., Du, Y., and Lv, T. A novel clustering method on time series data. *Expert Systems with Applications 38*, 9 (2011), 11891–11900.