SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 3, ISSUE 5
ISSN 2054-7390

Temporal Association Rule Mining: With Application to US Stock Market

Ting-Feng Tan, Qing-Guo Wang, Tian-He Phang, Xian Li and Jiangshuai Huang
Department of Electrical & Computer Engineering, National University of Singapore
a0123784@nus.edu.sg; elewqg@nus.edu.sg; lixian@nus.edu.sg; jshuang@nus.edu.sg

ABSTRACT

A modified framework, that applies temporal association rule mining to financial time series, is proposed in this paper. The top four components stocks (stock price time series, in USD) of Dow Jones Industrial Average (DJIA) in terms of highest daily volume and DJIA (index time series, expressed in points) are used to form the time-series database (TSDB) from 1994 to 2007. The main goal is to generate profitable trades by uncovering hidden knowledge from the TSDB. This hidden knowledge refers to temporal association rules, which represent the repeated relationships between events of the financial time series with time-parameter constraints: sliding time windows. Following an approach similar to Knowledge Discovery in Databases (KDD), the basic idea is to use frequent events to discover significant rules. Then, we propose the Multi-level Intensive Subset Learning (MIST) algorithm and use it to unveil the finer rules within the subset of the corresponding significant rules. Hypothesis testing is later applied to remove rules that are deemed to occur by chance. After which, multi-period portfolio optimization is done to demonstrate the practicality of using the rules in the real world.

Keywords: Temporal data mining, financial time series, Knowledge discovery, events, DJIA, hypothesis testing, multi-period portfolio optimization.

1 Introduction

With regards to financial market predictions or pattern recognitions, many economists believe in the Efficient Market Hypothesis. The Efficient Market Hypothesis states that the security markets are extremely efficient in reflecting information about the individual stocks or about the stock market as a whole [1]. The Efficient Market Hypothesis even implies that neither Fundamental Analysis nor Technical Analysis will give an investor a return higher than that obtained from holding a randomly selected portfolio of individual stocks. The Random Walk Hypothesis is a very famous hypothesis which is closely related to the Efficient Market Hypothesis. The Random Walk Hypothesis indicates that the stock prices are always unpredictable and random. It states that the price changes in the future are independent of the prices and the news today or in the past. For example, the price changes tomorrow will reflect only the news tomorrow and will be independent of the price changes and news today [2]. Since the news in the future is unpredictable, the price changes in the future will be unpredictable and random as well. By putting these two famous hypotheses together, the stock market can easily be illustrated as a perfect market, where the news are reflected in the market instantly and none of the analysis or rules will help investors

DOI: 10.14738/tmlai.35.1051
Publication Date: 29th September, 2015
URL: http://dx.doi.org/10.14738/tmlai.35.1051

to gain greater return without taking greater risk. Hence, all the news and the prices are completely random and every security in the market will always be presented by its true value.

However, by studying historical events, the markets are not always perfect. Famous and recent examples can easily be named, such as the 1997–2000 Internet Bubble and the 2001 - 2006 United States Housing Bubble. These were caused by over-valuation of many companies as well as assets in the marketplaces. If the markets are always perfect, the share prices of the companies should not be grossly over-valued and should be reflecting their true values all the time. Thus, history shows that the financial markets are often imperfect, especially in the short term. Some financial economists believe that the financial market maybe a voting mechanism in the short run; in the long run, it is a weighing mechanism [1]. The detection of the short term trends or patterns may be done by using either traditional financial analysis or more complex techniques, such as Knowledge Discovery in Databases (KDD).

KDD is a non-trivial process for identifying valid, previously unknown, potentially useful and ultimately understandable information from the data [3, 4]. It consists of five steps and they are data selection, data preprocessing, data transformation, data mining and interpretation. Data Mining is the fourth step of KDD, where the interested knowledge, rules, or patterns are recognized. A number of famous Data Mining techniques are actively used in analyzing the financial market. They include neutral networks, decision tree, factor analysis, nearest neighbor techniques, clustering, association rules, and inductive logic programming [5]. In particular, association rules are used in our proposed framework.

Associating rule mining [8] is a popular tool used in many applications such as bioinformatics, web usage mining, continuous production, intrusion detection. However, association rules that discover concurrent relationships between items within the same period, are insufficient for financial market prediction. Nevertheless, the following type of association rules might interest analysts: If the prices of Intel and General Electric shares showed an increasing trend over the past 1 month, there is a 60% likelihood that the price of Microsoft shares will show a positive trend in the next 1 month. These association rules with an added time dimension are referred as temporal association rules or time-series association rules, and will be the focus of this study.

The issues of discovering and acquiring the hidden knowledge from the data have been frequently reviewed in the literatures of KDD and Data Mining. For example, various techniques and applications were reviewed and considered in [9, 10]. [11] illustrated a Data Mining approach for time series databases, which emphasized on the preprocessing stage of KDD. The temporal relationships between items were then extracted based on the transformed database. Based on the study done in [11], the time series databases (TSDBs) could be transformed into transactional databases with an extra time dimension. Classical association rule learning algorithms, such as APRIORI and ECLAT, could then be applied to extract the frequent patterns or relationship between the items from the transformed TSDBs. On that basis, [12] demonstrated Data Mining algorithm, called MINEPI, which focused explicitly on the time parameter. Based on the previous study, [13] presented the Representative Episodal Association Rules (REAR) algorithm, which further would convert the TSDBs into discrete representations for the generation of association rules. This algorithm used the concept of episodes as defined in [12] and further reduced the dimension by converting episodes into discrete symbols with the aid of sliding window. Temporal association rules were then generated from the frequent episodes detected with the sweep of sliding window in the database. After introducing the REAR algorithm, [14] presented Minimum Occurrence With Constraints And Time Lags (MOWCATL) algorithm. This is an enhanced algorithm based on REAR. The

MOWCATL further classifies episodes into antecedents and consequents. The temporal association rules are then learnt with a time lag between the antecedents and consequents. The REAR and MOWCATL are the results of improvements from the Gen-FCE, Gen-REAR in [13], and MINEPI in [12].

Furthermore, [15] modified the MINEPI to MINEPI+ and EMMA for mining the frequent episodes in TSDBs. ARMADA, an algorithm capable of mining for temporal rules in databases based on interval data, was introduced in [16]. Recently, [6] introduced a heuristic methodology for learning association rules from the financial market TSDBs. This heuristic methodology is mainly based on the MOWCATL algorithm but ECLAT algorithm is used for the Data Mining process. It is mainly supported by the toolkit which they developed, namely CONOTOOL. Some of the studies were done for financial markets, such as Madrid Stock Exchange (IGBM), Spain, by [6], The Stock Exchange of Thailand (SET), Thailand, by [17], and Singapore Exchange (SGX) by [18].

The state of art temporal association rule mining techniques are able to unveil the underlying association rules from the financial TSDBs. However, curse of dimensionality could arise when dealing with high-dimensional data. To overcome this, we propose the Multi-level Intensive Subset-based Training (MIST) algorithm and use it to extract the subset data of the significant rules in a recursive fashion. Hence, this breadth-first association rules searching mechanism enables efficient discovery of finer rules as the complexity and the processing time are greatly reduced by removing unrelated elements in the subset. Another shortcoming of the current rule mining techniques is that some rules might be useful but are ignored if they do not satisfy the support threshold as they occur rarely. To resolve this, we suggest using hypothesis testing to retain rules if the occurrences are sufficiently large compared to chance occurrences. These two major additions will be further explained in the modified framework after we have described the general framework.

The rest of the paper is organized in the following manner. The general framework, based largely on the heuristic methodology for temporal association rule mining by Dante et al. [6], is introduced in Section 2. The limitations are discussed and will provide motivation for improvements. By making some modifications to the general framework, our proposed modified framework is elaborated in Section 3 and 4. Here, Multi-level Intensive Subset Learning (MIST) algorithm and hypothesis testing are introduced to overcome the limitations. In Section 5, the top four component stocks in terms of the highest daily volume from Dow Jones Industrial Average (DJIA) and DJIA itself are used as a case study. This section also analyses the rules and simulation results. Section 6 presents the usefulness of the rules in the real world using multi-period portfolio optimization. Section 7 concludes the paper.

2 Temporal Association Rule Mining

Let p_i where i = 1, 2, …, n be the time-series data which is used to form a time-series database where sliding time windows are introduced as the timing constraints. The time window for antecedents will be referred to as Input Window and window width is given by IW, while the time window for consequents will be referred to as Output Window and window width is given by OW. The definitions for antecedents and consequents will be elaborated later in this section. Sliding time windows are shown in Figure 1.

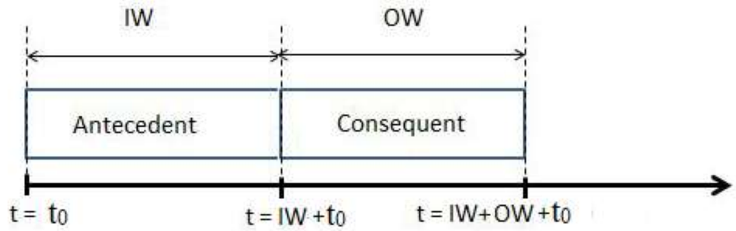

Figure 1 Illustration of sliding time windows

where unit of t is in days and t0 refers to a particular day.

p_i can be classified according to the respective range for each item. The Symbolic Aggregate Approximation (SAX) algorithm [17, 19] is chosen and used for this purpose because one of its key advantages is the ability to classify time series data based on its percentile position in the overall distribution.

To elaborate on SAX further, it is the first symbolic representation for time series that allows data points reduction and indexing with a lower-bounding distance measure. SAX is as well-known as representations such as Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT), except that it requires less storage space. One of the advantages of the SAX algorithm is its ability to convert time-series data of N size into time-series data of n size, where $n \le N$ and $\frac{N}{n} \in Z^+$, by applying the Piecewise Aggregate Approximation (PAA) technique. Z^+ refers to the set of positive integers. The idea of PAA technique is to divide time-series data of N size into n segments with equal length and the average value of each segment is used. The equation of the PAA technique to convert a time-series data, C, of the length of N into length n, in which the ith element of C, is calculated by the following equation [20].

$$\tilde{p}_i = \frac{k}{N} \sum_{j=\frac{N}{k}(i-1)+1}^{\frac{N}{k}i} C_j \tag{1}$$

Figure 2 shows a time-series C that is represented by PAA (by the mean values of equal segments). In the example above, data points are reduced from N=60 to k=6.

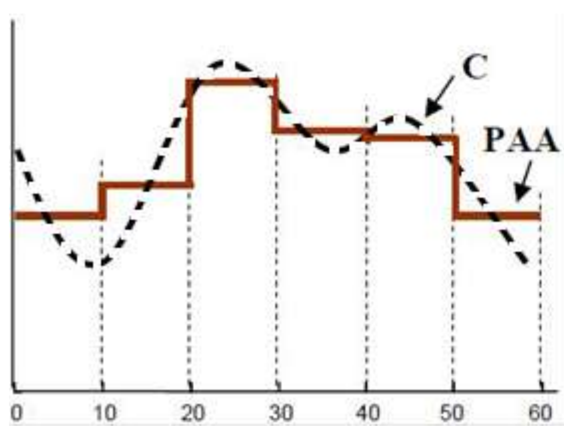

Figure 2 Illustration of PAA

In order to obtain the string representation after a time-series data is transformed into the PAA representation, symbolization region should be determined. According to [19], by empirically testing more than 50 datasets, it is defined that normalized subsequences have distribution that highly resembles the Gaussian distribution. Therefore, by calculating the mean and the variance of the subsequences PAA data and symbolizing each of the PAA data based on its percentile value, the SAX algorithm is then able to convert the whole time series data into symbolized string representations. This quantization process transforms $\tilde{p}_i$ to $\bar{p}_i$. The implementation of the SAX algorithm is contributed by the researchers [19]. Their implementation is chosen because it is widely used in the related research field and is also recommended by the SAX main research website to interested researchers; hence, the correctness of the implementation is ensured. Figure 3 gives the illustration of SAX.

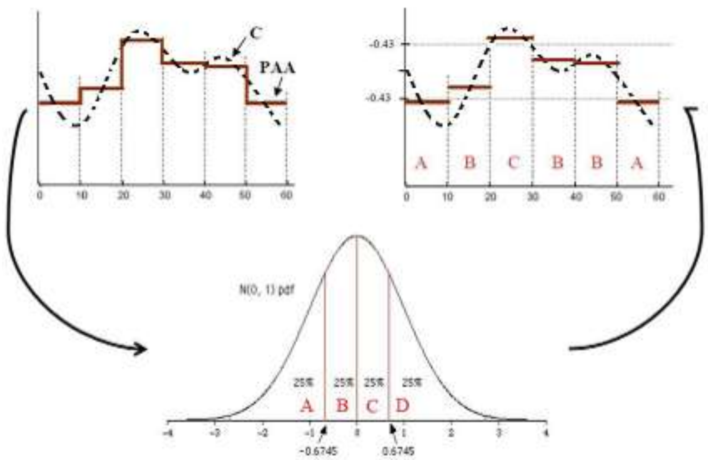

Figure 3 Illustration of SAX: The N-size time series data was first converted into k-size data using PAA technique. The k-size data are then normalized and symbolized by mapping it in the Gaussian distribution.

The SAX program requires four parameters as shown in Table 1.

Table 1 Explanation of the parameters of the SAX program

Parameter	Description
Data_Raw	The data to be processed by SAX algorithm.
N	The number of entries in the provided Data_Raw.
n	The number of output segments
alphabet_size	Number of discrete symbols used to represent the string.

Hence, the corresponding inputs and outputs can be given by $X_j = \left[\bar{p}_i, \bar{p}_{i+1}, \bar{p}_{i+2}, \ldots \right]$ and $Y_j = \left[\bar{p}_i \right]$ where i = 1, 2, …, n and j = 1,2, …, m. The inputs are known as antecedents and the outputs are known as consequents. A typical association rule can be written as

$$X \Rightarrow Y \tag{2}$$

The “support” $S(X \Rightarrow Y)$ of the rule is defined as the fraction of observations where the rule holds and the formula is as follows

$$S(X \Rightarrow Y) = \frac{N(X,Y)}{N(T)} \quad (3)$$

where N(X,Y) is the number of times when both the antecedent and consequent are observed and N(T) is the number of transactions in the database. The "confidence" $C(X \Rightarrow Y)$ of the rule is its support divided by the support of the antecedent.

$$C(X \Rightarrow Y) = \frac{S(X \Rightarrow Y)}{S(X)} \quad (4)$$

This can also be viewed as the conditional probability that Y occurs, given that X occurs [7]. We can then generate all possible rules based on the below constraints

$$S(X \Rightarrow Y) \geq s \quad and \quad C(X \Rightarrow Y) \geq c \quad (5)$$

where "s" is the support threshold and "c" is the confidence threshold.

For the temporal association rules extraction algorithm, we have chosen to use MOWCALT algorithm by Harms et al [14] as they provide a clear explanation with pseudo code for their algorithms and the implementation is relatively simple. The rules extracted in this manner are called significant rules. See Algorithm 1.

Algorithm 1

1) Generate Antecedent Target Episodes of length 1 which we denote as ATE_1;
2) Generate Consequent Target Episodes of length 1 which we denote as CTE_1;
3) Record occurrences of ATE_1 and CTE_1 episodes;
4) Prune unsupported episodes from ATE_1 and CTE_1 based on minimum support threshold;
5) k=1;
6) while ($ATE_k \neq \emptyset$) do
7) Generate Antecedent Target Episodes ATE_{k+1} from ATE_k;
8) Record each occurrence of the episodes;
9) Prune the unsupported episodes from ATE_{k+1};
10) k++;
11) Repeat Steps 5 – 11 for consequent episodes using CTE_{k+1};
12) Generate combination episodes from antecedent episodes and consequent episodes;
13) Record each occurrence of the combination episodes;
14) Return the supported combination episodes that satisfy the minimum confidence threshold and these are the relevant rules;

To evaluate the usefulness of the rules, we need to run simulation for test data and measure the performance against some benchmarks.

Refer to below for a recap on the general framework as described in this section.

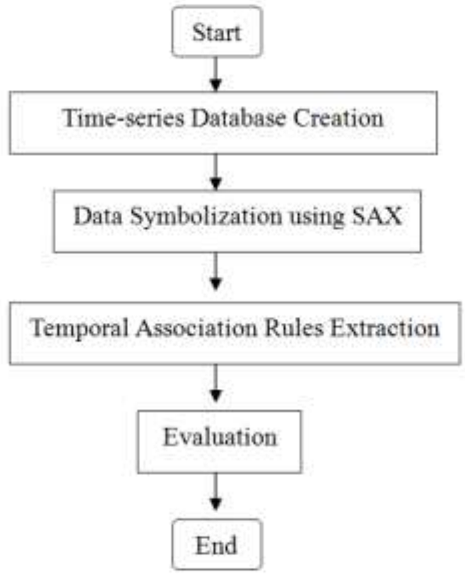

Figure 4 Flowchart of general framework

3 MIST Algorithm

The number of possible temporal association rules, H, can be easily formulated as the following.

$$H = \sum_{j=2}^{Q} \left(M^{j} \times \binom{Q}{j} \right) \tag{6}$$

Where M is the number of symbols, Q is the number of items, and j is the total number of both Antecedents' and Consequent's episodes in the corresponding rules. It can be observed that the number of possible temporal association rules is growing exponentially with the number of symbols and stocks. If there are only fifteen items to be symbolized with just five symbols, H will be 4.7018×10^{11}. Therefore, huge memory resources and processing time are required even for small values of M and Q. If items are required to be quantized with more symbols to achieve higher resolution, the hardware requirements might be a huge issue.

To tackle the problem of heavy computation costs, we propose the Multi-level Intensive Subset-Training (MIST) Algorithm as an efficient way to discover the finer temporal association rules with less symbols (breadth searching) first, and only perform a deeper searching with more symbols (depth searching) for the relevant rules recursively. This algorithm will continue to perform depth searching until there is insufficient data size or no more significant rules are needed to be diagnosed. The processing time and memory required will be greatly reduced because only the subset of the data is discretized with more symbols. The idea behind MIST algorithm is mainly based on the downward closure lemma, which states that the subsets of a frequent pattern are also frequent. See Algorithm 2 for the pseudo code and refer to Figure 5 for illustration.

Algorithm 2

1) Check for rules where the numbers of occurrences satisfy the minimum data size;

2) For each of the relevant rule, discretize with additional symbols;
3) Create new time-transaction database using the occurrences of the rule
4) Use Algorithm 1 to discover new finer rules;
5) Repeat Steps 1-4 as long as there are rules to be analyzed;
6) Return all new finer rules.

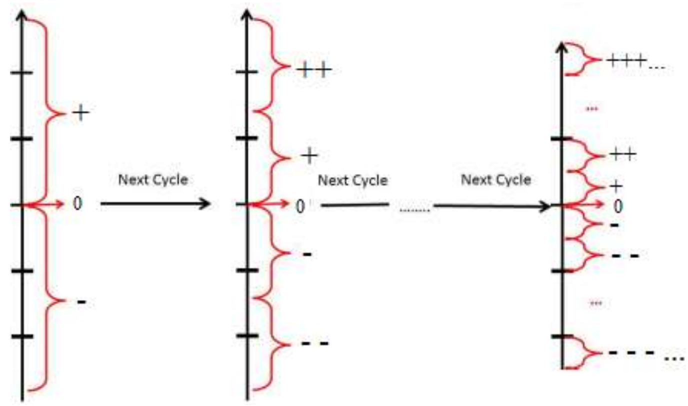

Figure 5 Overall view of MIST Algorithm: '+' and '-' are used as symbols for illustration purpose

4 Hypothesis Testing

Some temporal association rules might have low occurrences and, hence are rejected by the support threshold. However, these could be useful rules if their occurrences are not by chance.

To overcome this limitation, we need to check if the occurrences are sufficiently large enough for us to believe that the rules do not take place by chance. To find this out, the idea is to perform hypothesis testing where null hypothesis is set up as H0: $O_{rule} = O_{random}$ and the alternative hypothesis is formulated as H1: $O_{rule} > O_{random}$ at the a% significance level. O_{random} is the number of times that a particular rule gets triggered successfully in the random walk case and O_{rule} is the recorded occurrence of the rule as observed during learning from the training data. We assume that distribution is normal since we will use a sample size that is large enough [29]. So, Z-test is used. If H0 is accepted, it means the rule occurs by chance. Alternatively, if H0 is rejected, it means the rule does not occur by chance Table 2 shows the steps and Algorithm 3 shows the pseudo code.

Table 2 Explanation of the steps for hypothesis testing

#	Description
Step 1	H0: $O_{rule} = O_{random}$ H1: $O_{rule} > O_{random}$
Step 2	Select a significance level a%
Step 3	Use Z-test
Step 4	Compare the computed test statistic with critical value. If the computed value is within the rejection region, the null hypothesis will be rejected. Otherwise, the null hypothesis will not be rejected.

Algorithm 3

1) Generate normally-distributed pseudorandom numbers using randn in Matlab;
2) Use results from Step 1 to create time-series database to mimic random walk;
3) Follow the steps in Algorithm 1 to get the occurrences O_{random};
4) Repeat, say k times for Steps 1 – 3 to get different samples of O_{random};
5) Set the recorded occurrence of the rule we are interested in as O_{rule};

6) Conduct Z-test where H_0: $O_{rule} = O_{random}$ and H_1: $O_{rule} > O_{random}$ at the a% significance level
7) Repeat Steps 5 – 6 for the rest of the rules
8) Return the rules that pass the Z-test;

By adding MIST algorithm and hypothesis testing, we get the modified framework in the below figure.

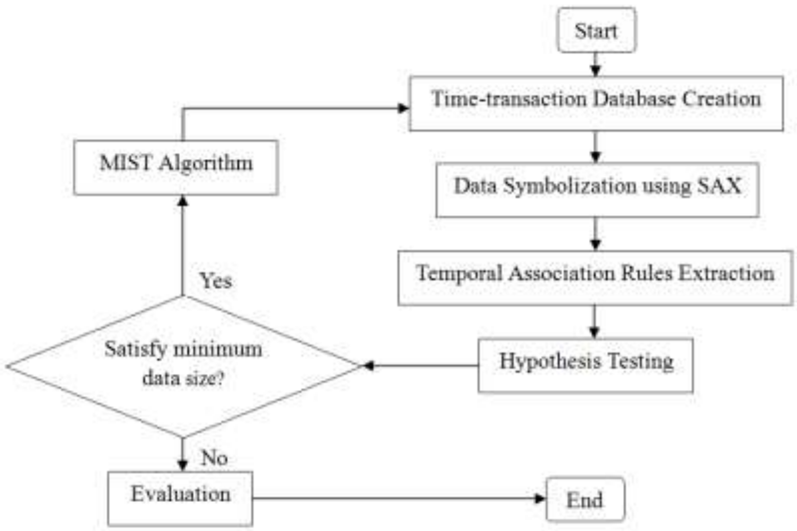

Figure 6 Flowchart of modified framework

5 Case Study for US Stocks

We will use the top four component stocks in terms of the highest daily volume from Dow Jones Industrial Average (DJIA) and DJIA itself will be used as a case study. Financial data are configured with 5 financial time-series: 4 series of stocks (daily adjusted close price in USD) and 1 series related to DJIA (daily close index measured in points). Data is taken for January 1994 to December 2013 from Yahoo Finance [21]. Data for January 1994 to December 2007 is used as training data while data for January 2008 to December 2013 is used as test data. Note that the adjusted close prices take into considerations all corporate actions, such as stock splits, dividends/distributions and rights offerings.

Table 3 Composition of database

Asset Name	Acronym
Cisco Systems, Inc.	CSCO
General Electric Company	GE
Intel Corporation	INTC
Microsoft Corporation	MSFT
Dow Jones Industrial Average	DJIA

Instead of using p_i as per Section 2, we use r_i which is rate of return (ROR) and formula is given below.

$$r_i = \frac{(P_t - P_{t+W})}{P_t} \times 100\% \tag{7}$$

where P_t represents the price at time t and P_{t+W} refers to the price after W number of days from time t. Here, we use W = 20 and IW = 3 and OW = 1.

After the stock prices database is converted into ROR database, the data can then be classified according to the range for ROR of each stock. One of the options is to classify ROR with a fixed range. For instance, a fixed range from 0% to 1% can be classified as weak positive trend with symbol "+", while 1% to 3% as semi-strong positive trend with symbol "++" for all the stocks. However, this method does not work well because every stock should have its own range due to its nature of business or certain unique characteristics. Hence, the SAX algorithm is used. The time series ROR data is first separated into three main categories, which are (negative gain) "-", (positive gain) "+", and (no gain) "0". Classification is then carried out by the SAX algorithm. Notice that the (no gain) category does not require to be symbolized again because its ROR is zero. This process is illustrated by Figure 7.

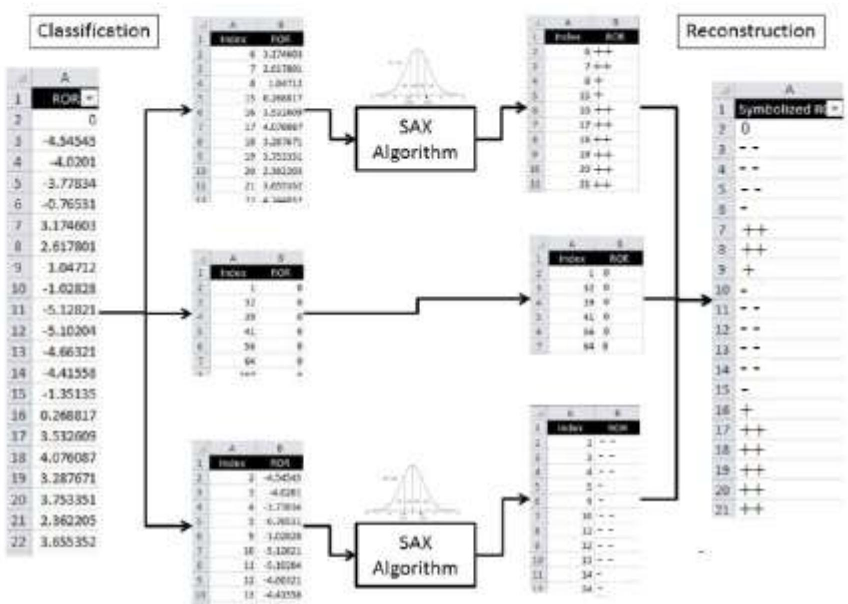

Figure 7 An overview of the classification process using SAX

To discover significant rules, we use MOWCALT algorithm, and select support threshold to be 10% and confidence threshold to be 50%. Also, the consequent will have two sets of measurements: one for when the rule holds and the other for when the rule fails. Let us define TM to be the mean for ROR in the first case and FM to be the mean for the ROR in the second case. So, the expected gain (EG) of the rule can be calculated using the following formula.

$$EG = C \times TM + (1 - C) \times FM \tag{8}$$

where C is the confidence of the rule as per defined in equation (4).

Next, we implement the algorithm described in Section 4 with k = 10 and a = 1% to carry out hypothesis testing. To get finer rules from the significant rules discovered using MOWCALT algorithm, we use the MIST algorithm and set data minimum size as 40. Lastly, we use multi-period portfolio optimization approach to simulate trading in the real world. We shall call this 'Rule Learning' model and will compare this against standard mean-variance model and equally-weighted model in terms of measurements such as terminal return and Sharpe ratio. This approach enables us to evaluate the usefulness of the rules.

Our trading strategy and some assumptions are as below.

- Trade according to the rule's consequent when all the antecedents are triggered
- Entry price is tomorrow's open price
- Close the position by the end of Output Window
- No limitation on short sell holding period
- No target profit (TP) and no stop loss (SL)
- Dividends, stock split and other corporate actions are ignored
- Trading costs (TC) are taken to be 0.18% of the value of the position

Table 4

Rule	Antecedents	Consequent	Support (%)	Confidence (%)	Expected Gain (%)	Total Trades	Success Rate (%)	Average Return per Trade (%)
1	{CSCO2+,DJA3+,INTC1+}	{CSCO+}	0.73	96	3.02	21	62	**-0.13**
2	{DJA2+,INTC1+,MSFT3+}	{CSCO+}	0.61	95	3.57	32	**47**	**-0.16**
3	{CSCO2+,GE1+,MSFT3+}	{CSCO+}	0.58	95	3.30	16	69	1.27
4	{CSCO2+,MSFT1+,MSFT3+}	{CSCO+}	0.58	95	2.69	42	60	0.43
5	{CSCO1+++,CSCO3+,MSFT2+}	{GE+}	0.58	95	1.57	5	**40**	**-2.95**
6	{DJA3+,GE1+,GE2+}	{GE+}	0.64	92	1.35	9	**44**	**-0.23**
7	{CSCO3+,GE2+,MSFT1+}	{INTC+}	0.67	96	2.60	10	70	1.89
8	{CSCO3+,DJA2+,MSFT1+}	{INTC+}	0.61	95	2.60	19	84	2.23
9	{CSCO3+,GE2+,MSFT1+}	{DJA+}	0.73	96	0.92	12	75	0.49
10	{GE2+,MSFT1+,MSFT3+}	{DJA+}	0.70	96	1.06	23	87	1.60
11	{DJA3+,GE1+,GE2+}	{DJA+}	0.70	96	1.11	10	50	**-0.39**
12	{DJA3+,GE1+,MSFT2+}	{DJA+}	0.64	96	1.06	15	53	**-0.27**
13	{DJA2+,MSFT1+,MSFT3+}	{DJA+}	0.61	95	1.31	15	93	0.95
14	{CSCO2+,DJA3+,GE1+}	{DJA+}	0.61	95	1.07	12	83	1.08
15	{DJA3+,GE1+,GE2+}	{DJA+}	0.61	95	1.17	11	55	**0.00**
16	{DJA3+,GE2+,MSFT1+}	{DJA+}	0.58	95	1.09	19	68	0.56
17	{CSCO2+,DJA3+,MSFT1+}	{DJA+}	0.58	95	1.06	35	86	1.35
18	{DJA3+,GE1+,MSFT2++}	{DJA+}	0.58	95	1.27	6	67	0.92

In addition, we will only select rules, which have antecedents that contain episodes for each of the three time steps. This will allow the price trends to be sufficiently described before the possible occurrence of the consequent. From here on, we will focus only on the finer rules and will refer to them simply as rules. For ease of presentation, we take the top rules in terms of confidence levels from each of the assets and focus on those that satisfy a certain support threshold. This gives us 18 rules and the simulation results are presented in Table 4.

The rules in Table 4 can be interpreted in the following manner. For example, rule 18: {DJA3+, GE1+, MSFT2++}$\Rightarrow${DJA+} means that when DJA3 in time step k+2 and GE1 in time step k show "+" positive trends, and MSFT2 in time step k+1 shows "++" positive trend, DJA in time step k+3 will show "+" positive trend with support value of 0.58% and confidence value of 95%. The average "Success Rate" here is 66% compared to the average "Confidence" value of 95%. This is expected because the time period used for testing is different from the time period used to discover rules. Using entry price as tomorrow's open price has also contributed to this phenomenon. Also, it is observed that the number of times that each rule is triggered is small compared to the number of occurrences for a typical significant rule. This is reasonable as these rules in Table 4 were generated using MIST algorithm, which discovers finer rules with more symbols from the subset of the corresponding significant rules.

6 Multi-period Portfolio Optimization

6.1 Introduction

In a single-period portfolio management, the aim is to optimize investment decisions for a fixed planning horizon with no flexibility to vary the portfolio composition. This means that the investor allocates his capital among various securities at the start of the investment period and observes the final outcome at the end of the investment period. One famous single-period model is the mean-variance framework suggested by Markowitz [22, 23]. In this model, risk is defined as the variance of a portfolio returns. Although the Markowitz model was first introduced over 50 years ago, it is still very popular among many portfolio managers. Also, it has been updated with refinements over time.

Let us now consider the multi-period case whereby investors can rebalance their portfolio at the end of each period until the terminal date is reached. Rebalancing can be done at only fixed times in a discrete-time model while investors in a continuous-time model can reallocate at any time. New information such as economic data and significant world events is available to investors for each period. Hence, investors are required to consider the new information and respond appropriately based on their portfolio objectives and constraints. In [24], Merton pioneered portfolio optimization for a multi-period continuous-time model, where stochastic control theory and dynamic programming are used for analyzing the appropriate partial differential equation of Hamilton-Jacobi-Bellman. For multi-period discrete-time model, the variance of wealth is non-separable in the sense of dynamic programming as noted in [25]. Since dynamic programming is not able to tackle the particular cost function in [25], an embedding technique is proposed to allow the problem to be transformed to an auxiliary problem, which can be solved to retrieve explicit, optimal solutions by dynamic programming. Later, it was shown in [26] that the problem can be solved by using convex analysis and no dynamic programming is involved.

We will proceed to describe the setting under which we will implement multi-period discrete-time model with the rules discovered from the proposed modified framework described earlier. In literature, the number of risky assets is typically fixed. We will allow this number to vary depending on which rules get triggered. Also, there are usually no limitations for weights in most of the current works. Here, we will state the limitation for each risky asset. The goal is to maximize the expected log of geometric growth rate of the portfolio, subjected to the weight constraints.

6.2 Log-optimal Portfolio

The general expression for wealth V_n at the end of n periods is given by the following.

$$V_n = R_n R_{n-1} \dots R_{k+1} R_k R_{k-1} \dots R_2 R_1 V_o \tag{9}$$

where V_0 is the initial wealth and R_k is a random return variable for k = 1,2…,n. Taking logarithm of both sides will give us the below.

$$\begin{aligned} \ln V_n &= \ln V_o + \ln R_n + \ln R_{n-1} \dots + \ln R_{k+1} + \ln R_k + \ln R_{k-1} \dots + \ln R_2 + \ln R_1 \\ &= \ln V_o + \sum_{k=1}^{n} \ln R_k \end{aligned} \tag{10}$$

Hence, we will select $U(V_n) = \ln V_n$ as the utility function. By maximizing the same utility function at each step, maximization of the expected final utility is ensured. This is because the separation property holds and the multi-period case will simplify to a series of single-period cases [27].

Each ith risky asset has price p_i governed by the standard geometric Brownian motion equation [27].

$$\frac{dp_i}{p_i} = \mu_i dt + dz_i \tag{11}$$

where μ_i is the drift term and z_i refers to a Wiener process with variance parameter σ_i^2. Thus, choosing the suitable weights for the risky assets to maximize the overall growth rate is equivalent to solving the problem below.

$$\max\left[\left(1-\sum_{i=1}^{m} w_i\right) r_f + \sum_{i=1}^{m}\left(\mu_i w_i - \frac{1}{2}\sum_{j=1}^{m} w_i \sigma_{ij} w_j\right)\right] \tag{12}$$

where r_f is the constant risk-free rate and σ_{ij} is the individual component in the covariance matrix. To solve this, set the derivative with respect to w_i equal to zero and we obtain the following.

$$\mu_i - r_f - \sum_{j=1}^{m} \sigma_{ij} w_j = 0 \tag{13}$$

Hence, when there is a risk-free asset, the log-optimal portfolio has weights for the risky assets that satisfy the below.

$$\sum_{j=1}^{m} \sigma_{ij} w_j = \mu_i - r_f, \ \textit{for i=1,2,…,m} \tag{14}$$

This is a system of m linear equations and m weights can be found easily.

6.3 Simulation to Evaluate Usefulness of Rules

The buy and sell position will be for any of the assets stated in Table 3, so the number of risky assets are 5. We will apply the following constraints: $20\% \le |w_i| \le 50\%$, where w_i is the weight for the ith asset, and for the risk-free asset, we will allow a weight of -100% to 100%. Here, m can be any number from 0 to 5 depending on the rules that get triggered. Expected gain of the rule will be the relevant μ_i. Covariance matrix will be calculated based on a look-back period of 220 days. The risk-free rate r_f will be calculated based on the historical data of 1 month US treasury bill [28].

Applying a confidence threshold of 70% to the pool of rules discovered in Section 5, we have 7,985 rules. For comparison purpose, we will also run simulation using the standard mean-variance model with covariance and returns estimated based on a look-back period of 220 days on a rolling basis. A portfolio with equal weightings will be simulated as well. Trading costs are taken 0.04%. The simulations will be conducted using data from January 2008 to December 2013. Results are presented in Table 5.

Table 5

Model	Terminal Return	Average Monthly Return (%)	Standard Deviation of Monthly Returns	Monthly Sharpe Ratio
Rule Learning	1.6804	0.9534	0.0721	0.1323
Standard Mean-Variance	1.1186	0.4023	0.0697	0.0577
Equally-weighted	1.0113	0.6819	0.1158	0.0589

It is observed that Rule Learning model gives the highest 'Terminal Return', 'Average Monthly Return' and 'Monthly Sharpe Ratio' compared to the other two models. This means that the rules are applicable in portfolio management in the real world.

7 Conclusion

We have proposed a modified framework that applies temporal data mining technique to financial time series. After using the MOWCATL algorithm to discover significant rules, we found finer rules within the subset of the respective significant rules by applying our proposed MIST algorithm. Later, we use hypothesis testing to filter for rules that do not occur by chance. From the simulation results, the rules are found to be useful in trading individual positions and for portfolio management. The right mixture of rules would enable investors to enhance portfolio performance as shown in our multi-period portfolio optimization model.

REFERENCES

[1]. Burton G Malkiel. The efficient market hypothesis and its critics. The Journal of Economic Perspectives, 17(1):59–82, 2003.

[2]. Burton Gordon Malkiel. A random walk down Wall Street: including a lifecycle guide to personal investing. WW Norton & Company, 1999.

[3]. Herbert A Edelstein. Introduction to data mining and knowledge discovery.Two Crows Corporation, 2, 1999.

[4]. Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. AI magazine, 17(3):37, 1996.

[5]. Boris Kovalerchuk and Evgenii Vityaev. Data mining in finance: advances in relational and hybrid methods. Springer, 2000.

[6]. Conti Dante and J Martinez De Pison Francisco. Finding temporal associative rules in financial time-series: a case of study in madrid stock exchange (igbm). In Proceedings of the 9th WSEAS international conference on computational intelligence, man-machine systems and cybernetics, pages 60–68. World Scientific and Engineering Academy and Society (WSEAS).

[7]. Trevor Hastie, Robert Tibshirani and Jerome Friedman. The Elements of Statistical Learning: Data mining, Inference and Prediction, Second Edition. In chapter 14, pages 490.

[8]. http://en.wikipedia.org/wiki/Association_rule_learning

[9]. Jiawei Han, Micheline Kamber, and Jian Pei. Data mining: concepts and techniques. Morgan Kaufmann, 2006.

[10]. David J Hand. Principles of data mining. Drug safety, 30(7):621–622, 2007.

[11]. Mark Last, Horst Bunke, and Abraham Kandel. Data mining in time series database. 2004.

[12]. Heikki Mannila, Hannu Toivonen, and A Inkeri Verkamo. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1(3):259–289, 1997.

[13]. Sherri K Harms, Jitender Deogun, Jamil Saquer, and Tsegaye Tadesse. Discovering representative episodal association rules from event sequences using frequent closed episode sets and event constraints. In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, pages 603–606. IEEE.

[14]. Sherri K Harms, Jitender Deogun, and Tsegaye Tadesse. Discovering sequential association rules with constraints and time lags in multiple sequences, pages 432–441. Springer, 2002.

[15]. Kuo-Yu Huang and Chia-Hui Chang. Efficient mining of frequent episodes from complex sequences. Information Systems, 33(1):96–114, 2008.

[16]. Edi Winarko and John F Roddick. Armadaâ˘A ¸San algorithm for discovering richer relative temporal association rules from interval-based data. Data & Knowledge Engineering, 63(1):76–90, 2007.

[17]. Kittipong Warasup and Chakarida Nukoolkit. Discovery association rules in time series data. 29(4):447–462, 2012.

[18]. Hongjun Lu, Jiawei Han, and Ling Feng. Stock movement prediction and n-dimensional inter-transaction association rules. In Proc. of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, page 12.

[19]. Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pages 2–11. ACM.

[20]. Battuguldur Lkhagva, Yu Suzuki, and Kyoji Kawagoe. Extended sax: extension of symbolic aggregate approximation for financial time series data representation. DEWS2006 4A-i8, 7, 2006.

[21]. https://sg.finance.yahoo.com/q?s=%5EDJI

[22]. Markowitz, H. M. (1952), Portfolio selection, Journal of Finance 7, 77–91.

[23]. Markowitz, H. M. (1959), Portfolio Selection: Efficient Diversification of Investments, Wiley, New York, NY.

[24]. R. Merton, Optimum Consumption and Portfolio Rules in a Continuous Time Model, Journal of Economic Theory, 3(4), 373–413, 1971.

[25]. D. Li, and W. Ng, Optimal Dynamic Portfolio Selection: Multi-period Mean-Variance Formulation, Mathematical Finance, 10(3), 387–406, 2000.

[26]. Jianping He, Qing-Guo Wang, Peng Cheng, Jiming Chen and Youxian Sun, Multi-period Mean-variance Portfolio Optimization with High-order Coupled Asset Dynamics, 2013

[27]. David G. Luenberger, Investment Science, 1998

[28]. http://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yield

[29]. http://www.investopedia.com/terms/c/central_limit_theorem.asp