CHAPTER 6

TEMPORAL ASSOCIATION RULE MINING BASED RECOMMENDER SYSTEMS FOR STOCK TRADING

6.1 Introduction

It was observed in Chapter 5 that the classifier based approach was successful in overcoming the issue of requiring expert intervention in generation of the trading rule by incorporating the trading decision as the part of the prediction process. It was also observed that the system was able to outperform the B&H strategy. In this chapter, two novel stock trading recommender systems based on mining of temporal associations in the stock price data are presented. The proposed recommender systems offer an improvement over the classifier based approach by allowing more profitable trades to take place thus increasing the overall profits. Association rule mining (ARM) is used for identifying relationships between items in a transaction database. Typically, temporal information is not considered while mining association rules. The recommender systems presented in this chapter are also significant in that they present a technique that allows for the extension of one of the most popular ARM algorithms, the Apriori algorithm, to mining temporal associations.

There have been a few studies on application of ARM techniques for financial applications. In (Paranjape-Voditel and Deshpande, 2013), a system which recommended a portfolio of stocks based on ARM is proposed. In (Srisawat, 2011), ARM is used to discover relationships between individual stocks in the Thailand stock market. In (Ting, Fu and Chung, 2006), ARM is used to identify frequently occurring patterns in a stock time series and to find interrelationships between stock price movements of pairs of stocks. However, the economic performance of the system is not considered and the temporal information content in the time series (such as the day of the week or the date) is not preserved. In (Kumar and Kalia, 2011), an ARM based system to find similarity between stocks traded on Indian stock markets, similar to the one proposed in (Ting, Fu and Chung, 2006), is presented. However, no trading strategy is proposed and the temporal information content is not preserved.
The recommender systems proposed in this chapter are capable of generating the transaction database from stock time series itself and can incorporate the temporal information into the items of the transaction database, thereby helping identify the temporal behavior of the stock time series. HP filter (Hodrick and Prescott, 1997) is used in the present study to separate the trend and cyclic components of the time series under consideration. From the cyclic components, the duration of the dominant cycle is identified and this information is used to select the training and testing datasets. SAX (Lin et al., 2007) is then employed to convert the time series into its equivalent symbolic representation. Association rules are mined from the resulting items. Parameters of the system are then optimized using evolutionary algorithms to optimize the in-sample performance. The optimum set of parameters obtained is used for testing the out-of-sample performance. The proposed system is validated on four different stocks belonging to two different stock markets (UK and India), over three different time frames.

The rest of the chapter is divided into following sections: Sections 6.2 and 6.3 present the design and implementation of the two ARM based stock trading recommender systems. Section 6.4 presents the conclusions.

6.2 A temporal ARM based recommender system incorporating day-of-the-week information

6.2.1 System design

6.2.1.1 Selection of training and testing datasets

First step in the design of the proposed stock trading recommender system is the selection of training and testing datasets. The selection process as proposed in (Nair et al., 2013), (Nair and Mohandas, 2014) and described in section 3.2 is followed. It can be observed in this case as well, that training sets consisting of number of samples that are multiples of the dominant cycle duration resulted in good out-of-sample performance. A line search is used to identify the multiple for each time frame that generated the best in-sample performance. The optimal test data duration is taken as length of the dominant cycle.

6.2.1.2 SAX based conversion of time series to symbols

Once a training dataset is selected, the dataset is then converted to symbols with the help of SAX algorithm (Lin et al., 2007). The SAX algorithm works by first converting the series,
say $Y = \{ y(t), y(t-1), \ldots, y(t-N-1) \}$, consisting of $N$ samples, into piecewise aggregate approximation (PAA) representation and then converting them into symbols. The algorithm requires two parameters to be specified at the outset: the window size $D_w$ (the number of samples that are considered together and converted into a single symbol) and the number of symbols $S_s$. As the first step, the data is normalized. Normalization does not change the original shape of the series. After normalization, PAA is used to reduce the dimensionality of the dataset. In PAA, the total number of samples are first divided into $D_w$ equal sized bins (windows) with each bin consisting of $M$ samples, such that $M * D_w = N$. Then each bin $b_i$ is represented by a single value using the equation in (6.1) below:

$$b_i = \frac{D_w}{N} \sum_{j=\frac{N}{D_w}(i-1)+1}^{\frac{N}{D_w}i} y_j \quad i = 1, 2, \ldots, D_w$$

(6.1)

The resulting series $B$ will have $D_w$ number of samples, ie $B = \{ b_1, b_2, \ldots, b_{D_w} \}$.

In order to obtain a symbolic representation of the series, the amplitude of the series is divided into $S_s$ intervals, with each interval assigned a unique symbol. It is assumed that, the samples in the series $Y$, when normalized, follow normal distribution, $N(0,1)$. Hence, for $S_s$ symbols, a total of $S_s - 1$ breakpoints are selected on the normal distribution curve such that equiprobable intervals are produced. After the breakpoints have been fixed, PAA levels are fixed and each segment is assigned with a symbol. The entire process is illustrated in Figure 6.1 (Lin et al., 2007).

![Figure 6.1 SAX representation of a series](Lin, Keogh, Wei, & Lonardi, 2007)
In Figure 6.1, the original series consisted of 128 samples i.e. \( N = 128 \). Also \( D_w = 8 \) and \( S_s = 3 \). The symbols were \{a, b, c\}. As can be seen, the normalized series was divided into 8 bins with each bin being represented by its mean value (the PAA representation). Since \( S_s = 3 \), the series is divided into three segments with each segment represented by a symbol. Now in the final SAX representation, the initial series with \( N = 128 \) has now been reduced to a sequence of 8 symbols: baabcabc.

Since the stock time series data available is on a daily basis, and as the proposed recommender system attempts to recommend one-day-ahead trading decision, the window size is chosen as 1, i.e. \( D_w = N \). It can also be seen from the description above that the selection of the number of symbols is subjective in nature. In the present study, optimization algorithms are employed to identify the optimal \( S_s \).

### 6.2.1.3 Mining temporal association rules

The temporal ARM based stock trading recommender system proposed in this section incorporates the day-of-the-week (DoW) information along with the symbolic representation of the stock price time series to generate the itemsets that are then utilized for mining temporal association rules. The stocks considered in the present study trade on exchanges that allow trades from Monday to Friday every week, except on holidays as declared by the respective stock exchanges. The day and date information is available along with the daily stock data (opening price, closing price, day’s high price, day’s low price, traded volume) for all the stocks in this study from (Yahoo!, 2014).

Considering one trading week as one transaction (half of the minimum intermediate trend duration), the DoW information is added to the corresponding string representation of the time series (daily closing prices). For the week ‘i’, the transaction is then represented as \( d_{week_i} \), where \(|d_{week_i}| \leq 5 \). This is due to the fact that in case of holidays falling in the week, there will be less than 5 items in the transaction \( d_{week_i} \).

Also, if the time series spans ‘w’ weeks, the transaction database can be represented as

\[
D_{\text{temporal}} = \{ d_{\text{week}_1}, d_{\text{week}_2}, \ldots, d_{\text{week}_w} \} \tag{6.2}
\]

and \( \sum_{i=1}^{w} |d_{\text{week}_i}| = N \) \tag{6.3}
A typical example:
\[ d_{\text{week},i} = \{(I_1, \text{Monday}), (I_2, \text{Tuesday}), (I_2, \text{Wednesday}), (I_1, \text{Thursday}), (I_3, \text{Friday})\} \] (6.4)

It must be noted that each symbol-weekday pair is considered to be a single item.

Hence, if the alphabet size = \( a \), then the total number of unique attributes, on incorporating the temporal (DoW) information = \( a \times 5 \).

For an itemset \( I_j \):
\[
\text{support} (I_j) = \left( \frac{\sum_{i=1}^{w} K_i}{w} \right)
\]

where
\[
K_i = \begin{cases} 
1 & \text{if } d_{\text{week},i} \cap I_j = I_j, \\
0 & \text{otherwise}
\end{cases}
\]

If \( \text{support} (I_j) \geq \text{minimum support} \), then the item \( I_j \) is selected as a frequent 1-itemset.

Now the frequent 1-itemsets are used to generate candidate 2-itemsets. Those 2-itemsets that satisfy the minimum support requirement are selected as the frequent 2-itemsets.

From the identified frequent 2-itemsets, the association rules are identified with the help of Apriori algorithm (Agrawal and Ramakrishnan, 1994). The Apriori algorithm detects associations between items in a transaction database. Let \( I = \{I_1, I_2, \ldots, I_m\} \) be the ‘m’ unique items in the transaction database. An association rule then will be of the type \( I_A \Rightarrow I_B \) (ie \( I_A \) ‘implies’ \( I_B \)) where \( I_A, I_B \subseteq I \) and \( I_A \cap I_B = \{\} \). In Apriori (Agrawal and Ramakrishnan, 1994) association rule mining algorithm, the support for this association rule is given by

\[
\text{support} (I_A \Rightarrow I_B) = P(I_A \cup I_B)
\] (6.6)

The confidence is expressed as
\[
\text{confidence}(I_A \Rightarrow I_B) = P(I_B \mid I_A)
\] (6.7)

As can be seen from the above expressions, the traditional Apriori algorithm does not consider temporal information as a parameter for mining association rules from the dataset.

In the recommender system proposed in this section, temporal information is incorporated into the ARM process by adding the DoW information to the itemsets.

### 6.2.2 Trading strategy

Trades are recommended based on the rules obtained from the selected frequent 2-itemsets.

The frequent 2-itemsets selected are of three types:
(i) The symbols for both the items are same, say, \{I_2, Tuesday\}, \{I_2, Wednesday\}). This implies that there is no significant price change from Tuesday to Wednesday of the week. This is a situation where trading will not result in profit. Hence no trade should be executed.

(ii) The symbol for item on the preceding day corresponds to a price range which is lower than that indicated by the symbol on the succeeding day, say, \{I_1, Thursday\}, \{I_2, Wednesday\}). Considering that the price range indicated by I_2 is greater than price range indicated by I_1, it implies that price levels on Thursday will be lower than price levels on Wednesday. Hence, no trade should be executed.

(iii) The symbol for item on the preceding day corresponds to a price range which is higher than that indicated by the symbol on the succeeding day, say, \{I_2, Tuesday\}, \{I_3, Friday\}). Considering that the price range indicated by I_2 is less than price range indicated by I_3, it implies that price levels on Tuesday will be lower than price levels on Friday i.e. price levels will go up from Tuesday to Friday. Hence, stock should be bought at the opening of the next day of the first 1-itemset (Wednesday morning) and sold just before the close on the day indicated in the second 1-itemset (Friday evening). Only these types of 2-itemsets are considered as trading rules and are stored along with their respective support values.

Assuming that the rules have been generated using the training set, the trading system works based on the steps given below.

1. At the end of the trading session, the closing price of the stock is converted to its symbolic form using SAX and the temporal information is added to it.
2. The resulting 1-itemset is compared to the selected 2-itemsets. Only those 2-itemsets which have one item matching with the 1-itemset generated at the day’s closing, are selected as candidate rules.
3. Check if the second item in the candidate 2-itemsets has the DoW later than the 1-itemset. Only the rules that satisfy this criterion are selected. This is to ensure that only those rules, using which trading is possible, are selected.
4. If more than one 2-itemsets are selected in step 3, the supports are compared and the 2-itemset with the highest support is taken as the trading rule. In case no 2-itemset is selected in step 3, no trading signal is considered to have been generated.
5. If a trading rule has been selected, buy the stock at the opening price on the next day and hold till the day indicated in the second item of the rule. Sell the stock at the closing of the indicated day.
Minimum support and the number of symbols for each dataset is optimized with the help of optimization algorithms. Two common evolutionary algorithms were initially considered, namely GA and ABC (Karaboga and Basturk, 2007), with the objective function being the maximization of PF. ABC was used for this system since only two parameters need to be specified at the outset. The ABC used in the present study has colony size: 40 and the maximum number of iterations to check for improvement in the food source: 100. For GA, the initial population size is 40. Two individuals with highest fitness are directly selected for the next generation. 80% of the remaining individuals are subjected to crossover. In the crossover process, initially a random binary vector with number of bits equivalent to number of genes in each chromosome is generated. For each bit position of the binary vector containing ‘1’, the corresponding gene is taken from one of the parents and for each ‘0’, corresponding gene is taken from the other parent. Similarly, for another off-spring, a new random vector is generated. Remaining individuals in the population are subjected to mutation. The Gaussian mutation technique is used. The GA stops if the value of the objective function value does not improve in 10 consecutive generations.

Block diagram of the proposed system is presented in Figure 6.2
Figure 6.2 Block diagram of the Temporal ARM based stock trading recommender system incorporating DoW information
6.2.3 Results
The proposed trading recommender system is validated on four different stocks, namely, Cipla, HUL, GSK and RBS, with Cipla and HUL drawn from the emerging market- India (specifically, the BSE) and GSK and RBS from the mature market-UK (the LSE). Three different time frames are considered for each of these stocks to demonstrate the efficacy of the proposed system under different market conditions such as uptrend, downtrend and no trend in the stock price movements. Hence the proposed system is validated on a total of 12 different datasets. In all the figures presented below, the profits for Cipla and HUL are in Rs. while for GSK and RBS, the units are GBp. Time frames are selected by first identifying the dominant cycle duration using the HP-filter –DFT based technique described in previous chapters and then selecting the samples for training set as the multiples of the dominant cycle duration that result in highest in-sample profits. Number of samples in the test set is taken as $L$ assuming that the current trend continues for at least one more cycle. The time frames selected are presented in Table 6.1.

<table>
<thead>
<tr>
<th>Stock</th>
<th>Timeframe 1</th>
<th>Timeframe 2</th>
<th>Timeframe 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In</td>
<td>Out</td>
<td>In</td>
</tr>
</tbody>
</table>

The datasets represent the three commonly occurring trends in stock price movements, namely: uptrend, downtrend and no specific trend, as can be seen clearly from the Figure 6.3- 6.5.
Figure 6.3 Cipla time frame-1 daily closing price indicatinguptrend

Figure 6.4 RBS time frame-2 daily closing price indicating down trend
A transaction cost of 0.5% per trade was considered for all the stocks over all time frames. The results are presented in Appendix D. The results presented in the tables are separated into ‘training’ or in-sample results, (In) and ‘testing’ or out-sample results (Out). It must be noted that for Cipla and HUL, the currency unit is Indian Rupees (Rs) while for GSK and RBS, the currency unit is Great Britain pence (GBp). Initially, two variants of the recommender system are evaluated, with GA and ABC as the optimization algorithms respectively while the objective function was common- the PF. Performance of the two recommender systems is evaluated on eight different performance measures, as suggested in (Brabazon and O’Neill, 2006). The total in-sample and out-of-sample profits generated are presented in Figure 6.6. For HUL and Cipla, the profits are in Rs. while for GSK and RBS, it is in GBp. It can be seen from Figure 6.6 that the ABC-optimized variant generates better out-of-sample performance than the GA-optimized variant for eight of the twelve datasets considered.
Figure 6.6 In and out-sample profits for temporal ARM based recommender system incorporating DoW information optimized using GA & ABC with PF as the obj. funs.

As the maximizing the profits generated from each trade is a desirable feature of a stock trading recommender system, another objective function was also considered- the profit per successful trade (P/ST). Recommender system performance is evaluated on the eight performance measures, similar to the recommender systems evaluated earlier in the chapter. The results are presented in Appendix D. It is observed from the results that use of P/ST as the objective function results in better out-of-sample profits when compared to PF. The in-sample and out-of-sample profits for the recommender system optimized using ABC and with P/ST and PF as the objective functions are presented in Figure 6.7.
The effect of combining GA with P/ST as the objective function is also studied, the performance evaluated on the eight measures used to evaluate other recommender systems discussed in this chapter and the results are presented in Appendix D. The GA-P/ST based recommender generated in much better out-of-sample performance when compared to GA-PF based system. A comparison of the overall in-sample and out-sample profits generated by the GA-P/ST based recommender system and the GA-PF based recommender system is presented in Figure 6.8.
It is also observed from the results (see Appendix D) that the overall profits generated by the temporal association rule mining-based recommender systems tend to significantly outperform the traditional B&H strategy. For comparison purposes, the in-sample and out-sample total profits for temporal ARM-based recommender system with DoW information optimized using ABC with P/ST as the objective function for all the stocks considered over all the three time frames is presented alongside the profits generated by the B&H strategy, in Figure 6.9.

Figure 6.8 In and out-sample profits for temporal ARM based recommender system with DoW information optimized using GA with PF and P/ST obj. fns.
Figure 6.9 In & out-sample profits for temporal ARM based recommender system with DoW information optimized using ABC with P/ST obj. fns. compared to B&H
6.3 A Temporal ARM based one-day-ahead trading recommender system

The second recommender system proposed in this chapter mines temporal association rules from the stock time series data to generate one-day-ahead trading recommendations. The block diagram of the proposed system is presented in Figure 6.10.

![Figure 6.10 Temporal ARM based one-day-ahead recommender system block diagram](image)

6.3.1 System Design

In this temporal ARM based recommender system, the number of symbols to be considered (while converting the time series to symbols using SAX) for each dataset are determined based on the minimum and maximum value of the data (here the data is the closing value of the stock every day). As a modification to the SAX algorithm, in this study, the breakpoints are not kept fixed, as proposed by (Lin et al., 2007), but dynamically vary with each dataset. The algorithm used for finding the bounds is presented in the algorithm SAX Bounds below. The identification of optimal breakpoints is accomplished with the help of GA, which finds the optimal breakpoint between each of the two consecutive bounds that maximize the given objective function. Hence, if there are a total of \( \beta \) bounds \{Bounds(1), Bounds(2), \ldots, Bounds(\beta)\}, the GA will find \( \beta-1 \) optimal breakpoints with the upper and lower bounds for each breakpoint in the GA given by the set of tuples \{ (Bounds(1), Bounds(2)), (Bounds(2),
Bounds(3)), …, (Bounds(β-1), Bounds(β))}. Detailed description of the GA parameters and the objective functions used is presented in the section: GA based system optimization.

**Algorithm: SAX Bounds**

**Input:** max, min  
// max: maximum closing price, min: minimum closing price in the given dataset  
**Output:** Bounds  

**Begin**

1. Temp ← min  
2. Bounds(1) ← min  
3. i ← 1  
4. **While** Temp ≤ max **do**  
   Temp ← Temp + 1.02* Temp  
   i ← i + 1  
   Bounds (i) ← Temp  

**End While**

**End**

**6.3.2 Mining one-day-ahead temporal association rules**

The temporal association rule mining system proposed for generating one-day-ahead recommendations, works as follows:

Time series represented by \( Y = \{y(t), y(t-1), \ldots, y(t-N-1)\} \) is converted into its string representation using SAX (Lin et al., 2007) algorithm, with the cut-points being determined using GA as described previously. Then the set of symbols can now be represented as \( S = \{S_1, S_2, \ldots, S_\beta\} \). Representing the time series \( Y \) using its symbolic representation, \( I = \{I_1, I_2, \ldots, I_n\}, \forall I \in S \), the symbols are then used to form the transaction database taking two symbols at a time, ie. two- itemsets \( d_1 = \{I_1, I_2\}, d_2 = \{I_2, I_3\}, \ldots, d_{n-1} = \{I_{n-1}, I_n\} \).

The transaction database \( D_{\text{temporal}} = \{d_1, d_2, \ldots, d_{n-1}\} \).

For the itemset \( d_j \):

\[
\text{support}(d_j) = \left( \frac{\sum_{j=1}^{n-1} K_i}{n-1} \right) 
\]

(6.8)

where \( K_i = 1 \) if \( d_j \cap K_i = d_j \),

0 otherwise
6.3.3 Trading strategy

Trades are recommended based on the rules obtained from the selected frequent 2-itemsets. The rules obtained are of three types:

(i) The symbols for both the items are same, say, \( S_3 \Rightarrow S_3 \). This implies that there is no significant price change from the first day to the next day. This is a situation where trading will not result in profit. Hence no trade should be executed.

(ii) The symbol for item on the preceding day corresponds to a price range which is higher than that indicated by the symbol on the succeeding day, say, \( S_2 \Rightarrow S_1 \). Considering that the price range indicated by \( S_2 \) is greater than price range indicated by \( S_1 \), it implies that price levels on the second day will be lower than price levels on the first day. Hence, no trade should be executed.

(iii) The symbol for item on the preceding day corresponds to a price range which is lower than that indicated by the symbol on the succeeding day, say, \( S_4 \Rightarrow S_5 \). Considering that the price range indicated by \( S_4 \) is less than price range indicated by \( S_5 \), it implies that price levels on the next day will be higher than price levels on the preceding day. Hence, stock should be bought at the opening of the next day of the first 1-itemset and sold just before the close on the day indicated in the second 1-itemset. Only this type of 2-itemsets are considered as trading rules and are stored along with their respective support values.
The trading system works based on the steps below.

1. At the end of the trading session, the closing price of the stock is converted to its symbolic form using SAX and the temporal information is added to it.
2. The resulting 1-itemset is compared to the selected 2-itemsets. Only those 2-itemsets which have one item matching with the 1-itemset generated at the day’s closing, are selected as candidate rules.
3. It is checked if the second item in the candidate 2-itemsets has the DoW later than the 1-itemset. Only the rules that satisfy this criterion are selected. This is to ensure that only those rules, using which trading is possible, are selected.
4. In case of more than one 2-itemsets being selected in step 3, the supports are compared and the 2-itemset with the highest support is taken as the trading rule. In case no 2-itemset is selected in step 3, no trading signal is considered to have been generated.
5. In case a trading rule has been selected, buy the stock at the opening price on the next day and hold till the day indicated in the second item of the rule. Sell the stock at the closing of the indicated day.

6.3.4 Results

In the present study, GA is used to identify the optimal breakpoints in SAX. Objective functions used are PF and P/ST. All the GA parameters are identical to the one described in section 6.2. Datasets considered for the recommender systems proposed in section 6.2 have been considered for the present study as well. Performances of both the variants of the proposed system are validated on eight different parameters as suggested in (Brabazon and O’Neill, 2006). The results are presented in Appendix D. It was observed that employing P/ST as the objective function resulted in much higher out-of-sample profits when compared to PF being used as the objective function in ten of the twelve datasets considered. Comparison of overall in-sample and out-of-sample profits for both the is presented in Figure 6.10.
Figure 6.11 In & out-sample profits for temporal ARM based one-day-ahead stock trading recommender optimized using GA with PF and P/ST obj. fns. compared to B&H
6.4 Conclusions

In this chapter, two stock trading recommendation systems based on mining of temporal association rules in stock prices were presented. The first recommender system incorporated SAX and the DoW information into the association rule mining process of the Apriori ARM algorithm, allowing it to identify temporal association rules from the stock price time series. GA and ABC algorithms were then employed to optimize the recommender system performance. The second system proposed was a temporal ARM based recommender system for generating one-day-ahead stock trading recommendations. This recommender employed GA to optimize the system performance. In both the types of recommender systems, two different objective functions (PF and P/ST) for the optimization algorithms considered, were evaluated. The recommender systems were validated on four different stocks belonging to two different economies – two stocks from the emerging economy: India and two from the mature economy: UK. For each stock, the system was validated over three time frames. From the results obtained, it can be seen that the proposed recommender systems outperform the traditional benchmark, B&H strategy. It is also observed that for both categories of temporal ARM based recommender systems proposed in this chapter, optimization with P/ST as the objective function generates higher out-of-sample profits. A comparison of the profits generated by the best-performing variants of the two categories of recommenders to the B&H returns along with the corresponding number of trades for all stocks over all time frames considered is presented in Figure 6.11. In Figure 6.11, R1 represents the temporal ARM based recommender system incorporating the DoW information optimized using ABC and objective function P/ST, while R2 represents the one-day ahead recommender system optimized using GA and P/ST as the objective function. The total number of trades executed in the time frame considered for recommenders is represented as TT-R1 and TT-R2 respectively. Profits are represented in the figure using bar chart while the total trades are represented using line chart.
Figure 6.12 Profit comparison for best performing ARM based recommenders with B&H and the corresponding TT

It is observed from the results that temporal ARM based one-day-ahead trading recommender generates the highest out-of-sample profits at the cost of an increase in the number of trades that need to be executed in the same time-frame. However, both the recommenders are seen to be capable of outperforming the traditional B&H strategy. Hence, it can be said that the proposed temporal ARM based recommender systems can be successfully used to generate stock trading recommendations that can help a layperson trade successfully in the stock markets.