International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 1 (2019) pp. 79-84

Discovery of Temporal Association Rules in Multivariate Time Series

Dinh Thuan Nguyen
Department of Information Systems,
University of Information Technology, VNU-HCM
HoChiMinh city, Vietnam.

Ba Duy Lam Khuat
Department of Information Systems,
University of Information Technology, VNU-HCM
HoChiMinh city, Vietnam.

Abstract

This paper presents a method for discovering association rules in multivariable time series, proposes two efficient algorithms based on the basic ideas of the Apriori algorithm to discover the frequent patterns from single time series and multivariate time series. The advantage of the two algorithms is that it avoids multiple scans on the time series and can be extended to any number of time series.

Keywords: Association rules, temporal association rules, multivariate time series

I. INTRODUCTION

Nowadays, along with the development of science and technology, digital data has become ever more popular, data collection takes place daily, part of which is time series data. A time series is a sequence of data points arranged chronologically [1].

A single time series consists only of single observations (scalar: described only by a magnitude value) recorded over time points. Discovering association rule on univariate time series rule to find the relationship between the patterns only on a single time series. In fact, a number of different time series may be related, discovering the association rule to find relationships between the patterns on many different time series is called discovery association rule on multivariate time series [2].

In this paper, we will focus on the method of discovering temporal association rules on multivariate time series. In discovering temporal association rules, due to the chronological nature of time series data [3], the time between the left and right sides of the rule needs to be considered.

A work on the discovery of association rules of multivariate time series is proposed in [4] by Xue Ruidong et al. However, the algorithm only deals with the number of time series, which can not be expands the number of time series to more than 3.

The work in this paper resolves the remaining problems in [4], proposing two corresponding algorithms for the discovery of frequency patterns on unvariate time series and multivariate time series. Both algorithms for efficiency are demonstrated through experimental setup. Furthermore, the algorithm can extend the handle for any number of time series.

The reaearch is fund by Vietnam National University HoChiMinh city (VNU-HCM) under grant number C2017-26-11.

II. THEORY AND RELATED WORKS

A. Data Preprocessing

Min-max normalization: Follow the following formula, min_v and max_v is the maximum value and the minimum value of the dataset, v is current value and v' is a new value, new value between 0 and 1, the formulation is shown below :

$$v' = \frac{v - min_v}{max_v - min_v} \quad (1)$$

Piecewise aggregate approximation (PAA): A time string is expression x = x_1,……, Xian, n is the length of the original time series. Let N be the length of segmentation. To compress the time series from n dimensions to N dimensions, apply the formulation is shown below [6]:

$$\overline{x_i} = \frac{N}{n} \times \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n_i}{N}} x_j \quad (2)$$

Monotonicity features extraction: The difference value between two continuous data points v_0, v_1 is compared with a levelThreshold parameter to set symbolic for each segment. Three symbols are applied to represent the time series [2]:

Table I. Monotonicity Feature Extraction

Symbol	Description	Definition
u	Up	$v_1 - v_0 >$ levelThreshold
d	Down	$v_1 - v_0 <$ -levelThreshold
l	Level	$\|v_1 - v_0\| \leq$ levelThreshold

B. Temporal association rules

A temporal association rule that is related to the time factor between the left and right sides of the rule [5]. Meaning if the item X once happens then Y will happens within time T:

$$X \xrightarrow{T} Y \quad (3)$$

C. Intra pattern

Intra pattern: An intra pattern is a character string found in a single time series, denoted by P = (s_1, s_2, s_3, ..., s_k), k is its size.

Occurrence position list: Each occurrence of an intra pattern has its position start and posittion end. All positions of the intra pattern are saved in the position list (PL).

Frequent intra pattern: An intra pattern is called a frequent intra pattern if $|PL_p| \geq$ minsup. Frequent intra patterns are saved in the frequent patterns list (FP).

D. Inter pattern

Inter pattern: An inter pattern is a sequential combination of intra patterns, each intra pattern in an inter-pattern called each pattern block, pattern blocks that may come from different time series. An inter pattern is denoted by P = (b_1, b_2, b_3, ..., b_k), b_i is a pattern block, k is its size.

Occurrence position list: Each occurrence of an inter pattern has its position start and position end. Start is the starting position of the first pattern block, end is the starting position of the last pattern block. All inter pattern positions are saved in the position list (PL).

Frequent inter pattern: An inter pattern is called a frequent inter pattern if $|PL_p| \geq$ minsup. Frequent inter pattern patterns are stored in the inter frequent pattern (FP).

III. IMPLEMENTATION

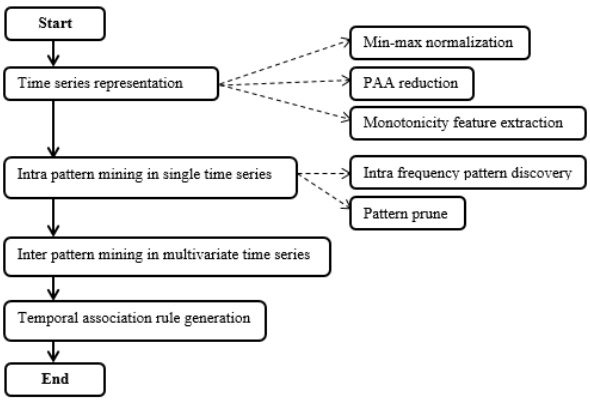

Figure 1. The structure of the work [2]

A. Platform and language

The methods are installed in the R language, running on Windows 7 32 bit operating system, Intel Core i3, CPU 2.50 Ghz, Ram 4G.

B. Dataset

The application is based on a dataset of 15 Vietnamese stock stored in csv file from 2003 to 2018, including properties such as Date, Stock Price, Transaction Volume.

C. Data preprocessing

Data normalization: Values can be stored in different units in different time series, min-max normalization method to unify the unit of measure between multiple time series. Applying the method, the time series has a value between [0,1].

Data reduction: The magnitude of the time series is reduced through the PAA method of formula (2), reducing the data to fit the trend analysis in each segment and can reduce the amount of processing data.

Data representation: Time series should be discrete into symbolic representations, convenient for pattern discovery with the association rule [7]. This paper uses the monotonicity feature extraction method described in section 2.1.

D. Patterns discovery

Based on the idea of the Apriori algorithm: If a frequent pattern, all of its sub patterns are also frequent. According to that idea, the list was created to save the frequent patterns found in each algorithm, the following frequent patterns being generated based on the combination of the previously saved frequent patterns, minimizing be space-searching [2].

E. Intra pattern discovery

Step 1: Scanning each time series for the first time to save the frequent intra pattern one character (k = 1), save in two lists FP and PL.

Step 2: Run algorithm algorimth_1 to find frequent intra pattern with k ≥ 2.

```
Input: minsup, max_character_size, FP, PL
Output: FP, PL
For k = 2 : max_character_size
{
    Each Pa in FP(k-1)
    {
       Each Pb in FP(k-1)
       {
            Assign pa_suffix =
                    substring(Pa,2,k-1)
            Assign pb_prefix =
                    substring(Pb,1,k-2)
            If pa_suffix=pb_prefix
            {
              Assign p_new =
                      concat(Pa[1],Pb)
               pl_a = [PL.pattern = Pa]
               pl_b = [PL.pattern = Pb]
               pl_new = {pl_a join pl_b on
              ((pl_a.start+1)=pl_b.start),
               pl_new.pattern=p_new,
               pl_new.start=pl_a.start,
               pl_new.end=pl_b.end}
               If count(pl_new) ≥ minsup
               {
              Add p_new to FP
                  Add pl_new to PL
               }
            }
       }
    }
}
```

F. Pattern prune

In the frequent intra patterns found, there are many patterns with little useful information. Therefore, two pruning conditions are proposed:

The max support count condition, denoted by max_s. This condition is used to remove the frequently encountered patterns has high support but less useful information. The pattern must have minsup ≤ max_s.

The min entropy condition, denoted by min_e. This condition is used as an additional condition for filtering less informative patterns. The pattern must have e (p) ≥ min_e. An intra pattern denoted by p = (s_1, s_2, s_3, ... s_i, ..., s_k) where s_i is a character in the pattern p, k is its size. The entropy of a pattern is calculated by the expression:

$$e(p) = -\sum_{i=1}^{k} P_{s_i} \times \log P_{s_i}, \; P_{s_i} = \frac{|s_{i==}|}{k} \quad (4)$$

G. Inter pattern discovery

```
Input: minsup, max_block_size, FP, PL, minT, maxT
Output: FP, PL
For k = 2 : max_block_size
{
  Each Pa in FP(k – 1)
  {
    Each Pb in FP(k – 1)
    {
      Assign pa_suffix = Pa[2,k-1]
      Assign pb_prefix = Pa[1,k-2]
      If pa_suffix=pb_prefix
   {
     pl_a = [PL.pattern = Pa]
        pl_b = [PL.pattern = Pb]
        If k = 2
        {
        Assign p_new=concat(Pa,Pb)
           pl_new = {pl_a join pl_b on
          ((pl_a.start+minT) ≤ pl_b.start
        and pl_b.start≤(pl_a.start+maxT)),
            pl_new.pattern=p_new ,
            pl_new.start=pl_a.start,
            pl_new.end=pl_b.start,
            pl_new.join=pl_a.start}
        }
        Else
            {
             Assign p_new=concat(Pa[1],Pb)
            pl_new = {pl_a join pl_b on
            (pl_a.end=pl_b.join),
            pl_new.pattern=p_new,
            pl_new.start=pl_a.start,
              pl_new.end=pl_b.join,
              pl_new.end=pl_b.end}
            }
            If count(pl_new) ≥ minsup
            {
              Add p_new to FP
              Add pl_new to PL
            }
        }
      }
    }
  }
```

Algorimth_2 to find frequent inter pattern. The two parameters minT, maxT given in algorithm_2 for constraining the position of the pattern block combination.

H. Temporal association rule generation

Rule generation : Each of frequent inter pattern, denoted by P = (B_1, B_2, B_i ,..., B_k) can generate k-1 rules, as shown below:

Table II. Temporal association rule generation

Pattern	P = (B_1,B_2,B_i,...,B_k)
Possible Rules	B_1 => B_2,B_i,...,B_k
	B_1,B_2 => B_i,...,B_k
	B_1,B_2,B_i => B_{i+1},...,B_k

Rule confidence : A temporal association rule the same as (3), rule $P_a \xrightarrow{T} P_b$,with P_a include pattern blocks (B_{a1},…,B_{a2},$B_{a|Pa|}$), P_b include patterns blocks (B_{b1},…,B_{b2},$B_{b|Pb|}$) then P_c denoted by P_c = (B_{a1},…,B_{a2},$B_{a|Pa|}$,B_{b1},…,B_{b2},$B_{b|Pb|}$). Rule's confidence calculated by the equation below :

$$conf = \frac{PL_{Pc}}{PL_{Pa}} \quad (5)$$

IV. RESULT AND EVALUATION

Apply time series discovery method on multivariate time series to find the relationship between price trend or trading volume among many stocks. Based on historical time series of stocks. With rules found, used to detect events of rising or falling trend, may occur at a future time limit.

In practical application, use a set of 15 stock time series, each containing 3812 data instances. Figure 2 shows normalized data.

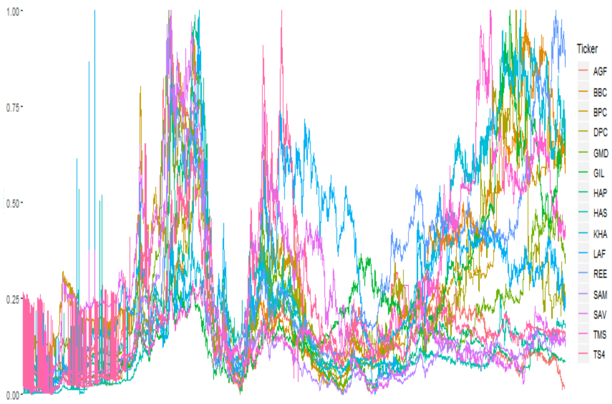

Figure 2. Normalized data for 3812 data instances

The data instances for each time series is processed to decrease 15 times ($\frac{n}{15}$ = N), $\frac{3812}{15} \approx 254$. Observed data are reduced, but the overall trend is unchanged. Figure 3 show normalized data for 254 data instances.

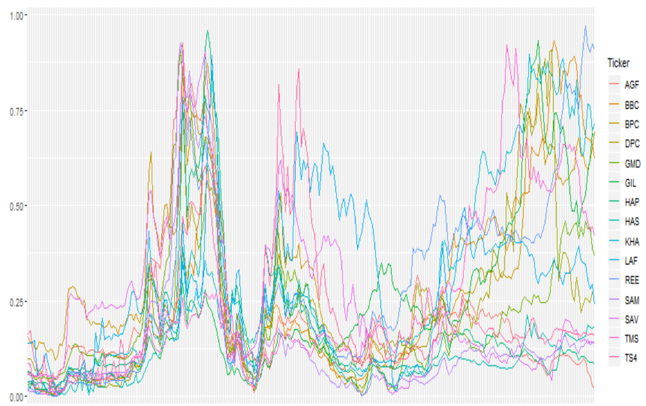

Figure 3. Normalized data for 254 data instances

In the symbolic representation method, the value of levelThreshhold is set to 0.001 to ensure the smallest change detection. Figure 4 shows the symbolic presentation of a time series of stock code AGF.

```
udlludududluduudduuddddlluldludlllullududduuuududdduuuddddduuuduuddudlduudddududdluduu
dudududddddddduuuuluudduuuuddddduduuddududdddududdduluuddluludddduuuduuududllllllllllud
uudddududuuududdddddllluudddlduludluduuddlududdduldllludulduludludduddlluldludududldd
```

Figure 4. Symbolic presentation of a time series

Two prune conditions with min entropy set to 0.1 and max sup set to 100 to remove the intra patterns with less useful information. Experimental results were 114 of patterns, if not pruned were 218 patterns, so the method of pruning removes 104 patterns. This method significantly reduces the number of patterns, thus reducing processing time in the following steps.

Table 3 shows the number of prunning patterns for each time series.

TABLE III. Number of patterns in each time series after pruning

Time series	Pattern number (retained / removed)	Time series	Pattern number (retained / removed)
AGF	14/7	SAM	14/7
BBC	14/7	SAV	15/7
GIL	15/7	TMS	15/7
GMD	14/7	TS4	15/7
HAS	15/7	BPC	14/7
KHA	15/7	DPC	15/7
LAF	13/6	HAP	16/7
REE	14/7		
Total (retained/ removed)		**218/104**	

The min confidence is set very high at 0.9 to ensure that the most powerful and useful rules are found. In addition, some Cosine, Jaccard, all confidence, and Kulc measures are used to further evaluate the rule. The table 4 summarizes the set parameters and the results of this experiment.

Table IV. Result experimental

Resources	Value
Data preprocessing	
Number of time series	15
Data instances	254
LevelThreshold	0.001
Intra pattern	
Min support	5% ≈ 12
Size	3
Entropy prune	0.1
Max support prune	100
Total pattern	114
Operation Time	9.13 secs
Inter pattern	
Min support	5% ≈ 12
Size	3
Min time	5
Max time	8
Pattern size 2	4,762
Pattern size 3	56,566
Operation Time	23.2 hours
Rules generation	
Min confidence	90%
Number of rules	**830**
Operation Time	**1.52 hours**
Number of rules detected	**93**
Total time	**≈ 25 hours**

The table 5 summarizes the results of the operation time on different numbers of time series, the unit is in seconds.

Table V. Operation time

Num Time Series	Pre-process	Prune	Intra Patterns	Inter Patterns	Rules	Total Time
3	4.66	0.04	2.1	178.2	6.15	191.15
5	9.55	0.06	2.36	1047.6	27.26	1086.83
7	14.07	0.07	2.54	4824	120.6	4960.74
10	16.2	0.13	4.19	16020	443.4	16483.92
15	24.65	0,15	6.57	83520	5472	89023.37

On the rules found, scan the list positions of the pattern on the left side of the rule, detecting the position of the right-hand pattern occurring in the future. In this experiment, we give an example of a highly confidence rule of 0.93, REE:udd,GMD:du => LAF:ud. Scan the position lists, pattern REE: ud, GMD: duo has the start position and the end position is (244,250), so the posittion of the pattern LAF:ud detected can appear in the time interval [250 + 5,250 + 8] = [255,258], figure 5 illustrates this step, table 6 is the result of the detection rules and the number of entities returned on different numbers of time series.

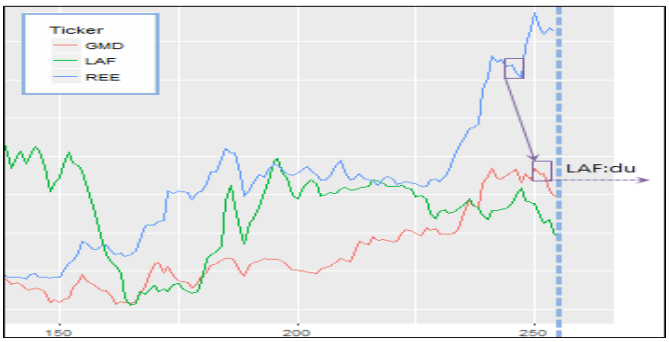

Figure 5. Detecting the position of a pattern occurring in the future

Table VI. The total number of items

Num Time Series	Intra Patterns	Inter Patterns (Size 2)	Inter Patterns (Size 3)	Total Inter	Rules	Rules Detect
3	22	173	343	516	5	1
5	37	480	1539	2019	16	2
7	52	1043	5033	6076	53	5
10	74	1922	11071	12993	138	23
15	114	4762	56566	61328	830	93

The total processing time is approximately 25 hours to handle over 15 time series. Compared to work in [4], as reported by the article, it handles only 3 time series with each time series, including 4319 data instances, processing time to find rules consists of 2 patterns = 935590ms ≈ 0.25 hours, processing time to find rules consists of 3 patterns = 54651748ms ≈ 15 hours. We also installed the two proposed algorithms in [4] for our dataset and compare it with our work. Experimental results are shown as table 7 and table 8.

Table VII. Experiment with our method

Resources	Value
Input parameters	
Number of time series	3
Data instances	254
Intra pattern	
Min support	5% ≈ 12
Size	3
Entropy prune	0.1
Max support prune	100
Total pattern	**22**
Operation Time	**1.54 secs**
Inter pattern	
Min support	3
Size	3
Min time	4
Max time	4
Pattern size 2	**245**
Pattern size 3	**45**
Operation Time	**5.36 mins**
Rules generation	
Min confidence	90%
Number of rules	**3**
Operation Time	**3.1 secs**
Total Time	**5.43 mins**

Table VIII. Experiment with the proposed method in [4]

Resources	Value
Input parameters	
Number of time series	3
Data instances	254
Pattern size	3
Find frequent patterns	
Total pattern	**49**
Operation Time	**0.33 secs**
Find rules	
Min support pattern	5% ≈ 12
Min support rule	3
Time	4
Min confidence	90%
Total rules	**22**
Operation Time	**24.55 mins**
Total Time	**24.55 mins**

From the experimental results between the two methods, notice the pattern discovery process in method [4], a little faster than our method. This is because of our method there are pruning steps. When adjusting data instances larger is 3812, minsup 5% of number data instances, the total operation time of the method of [4] takes more than 7h to complete. Our method takes 21.42 mins to complete.

To evaluate the model, the dataset is divided into two parts, with 90% = of the data for rule discovery, 10% of the remaining data for the test. The results as shown in the table 9:

Table IX. Evaluate the detection model with minconf is 0.9

Num Time Series	Rules Found	Detected Rules	Correct Rules	Incorrect Rules	Correct Ratio
3	3	1	1	0	100%
5	8	2	1	1	50%
7	35	5	2	3	40%
10	88	12	8	4	67%
15	619	77	42	35	55%
Total	**753**	**97**	**54**	**43**	**56%**

Configuring minconf = 0.7 to extend the rules table increases the probability of detection, but accuracy can be reduced. The results as shown in the table 10 :

Table X. Evaluate the detection model with minconf is 0.7

Number of Time Series	Number of Rules Found	Number of Detected Rules	Correct Rules	Incorrect Rules	Correct Ratio
3	93	30	15	15	50%
5	279	105	39	66	37%
7	813	338	163	175	48%
10	1922	836	346	490	41%
15	12492	6011	2541	3470	42%
Total	**15589**	**7320**	**3104**	**4216**	**42%**

V. CONCLUSION

The discovering temporal association rule is relatively complex due to the large size and chronological order of the time series data. A number of algorithms have been proposed to explore the temporal association rules [4]. But the algorithm cannot be expanded because it cannot be processed to find association rules that have more than three time series, so the efficiency is not high.

The two algorithms proposed in this article are based on the idea of the Apriori algorithm: If a pattern is Frequent, all its sub pattern are frequent too. According to the idea, the two lists were created to save the frequent patterns found in each step, the following frequent patterns being generated based on combinations of the previously saved frequent patterns, so minimizing be space-searching. In addition, both algorithms can extend the handle for any number of time series.

The pruning condition is given to reduce the number of patterns found in each single time series. This will reduce the execution time for the pattern discovery step in the multivariate time series.

Confidence evaluation is a common choice for association rule mining. It is a very high threshold setting for discovering rules that are truly useful. In addition, a number of other reliable methods of measurement are also used to evaluate the rule in full.

ACKNOWLEDGMENTS

We are very grateful to the Information Systems Lab, University of Information Technology (UIT), VNU-HCM for providing the best facilities and spiritual support.

REFERENCES

[1] Brillinger, D. R. (1975). “Time series: Data analysis and theory”. New York: Holt, Rinehart. & Winston.

[2] Yi Zhao. (2017) “Discovery of temporal association rules in multivariate time series” Master’s Thesis, Department of Information System and Technology (IST)

[3] Tak-chung Fu, “A review on time series data mining”, Engineering Applications of Artificial Intelligence 24 (2011) 164–181

[4] Xue, Ruidong, et al. (2016) “Sensor time series association rule discovery based on modified discretization method”. Computer Communication and the Internet (ICCCI), 2016 IEEE International Conference on. IEEE

[5] Das, Gautam, et al. "Rule Discovery from Time Series". KDD. Vol. 98. No. 1. 1998.

[6] Eamonn Keogh1, Kaushik Chakrabarti2, Michael Pazzani1, Sharad Mehrotra. (2001) “Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases”. Knowledge and Information Systems 3: 263–286

[7] Tim Schluter and Stefan Conrad “About the Analysis of Time Series with Temporal Association Rule Mining”, 978-1-4244-9927-4/11/$26.00 ©2011 IEEE