Research Paper

Volume : 4 | Issue : 5 | May 2015 • ISSN No 2277 - 8179

Frequent Sequential Pattern Mining With Weighted Regular Expression and Length Constraint

Computer Science

KEYWORDS : Data Mining, Frequent pattern mining, constraint based pattern mining, regular expression constraint, weight constraint, length constraint

Dipak R. Kawade Sangola College, Sangola, Dist-Solapur (MS)

Dr. Kavita S. Oza Department of Computer Science, Shivaji University, Kolhapur (MS)

ABSTRACT

In this internet world, huge amount of data is collected in digital form. These data contains useful hidden knowledge. To discover hidden knowledge from huge data is time consuming, tedious and challenging task. Data mining's frequent pattern mining technique is useful for such purpose. Lots of algorithms were already tackling this issue but main problem related to frequent pattern mining is efficiency and effectiveness of algorithm. To solve this problem many researchers were used constraint based pattern mining technique. For efficiency and effective execution of algorithm present study uses regular expression constraint which is inexpensive in time and memory. Many of the researcher's uses support count technique. Main problem related with support count technique is to find ideal support value. To solve this issue present work uses weight constraint. Length of pattern is also important criteria. Present work also uses length constraint. Present work uses three constraints namely regular expressing which focus user interest, weight which concentrate on importance of item and length which restrict length of pattern.

INTRODUCTION

In this digital world, huge amount of data is available in digital form. This large data contain significant and non-significant patterns. Main challenge is finding interesting patterns which are helpful for decision making process.To find patterns from this huge data is very tedious and time consuming task. So there is need some automated technology which may do this job efficiently and effectively. Frequent pattern mining technique is useful for this purpose.Frequent pattern mining was first proposed by Agrawal and Srikant.[12] Since from that various researchers presented different frequent pattern mining algorithms. These algorithms categorized as closed sequential pattern mining, constraint-based sequential pattern mining, biological pattern mining etc.[1] Sequential pattern mining has become an essential task with broad applications, such as analyzing web access patterns, customer purchase patterns, DNA sequences etc[2], prediction of diseases [7].

Main issue related with frequent pattern mining technique was efficiency and effectiveness of algorithms. To mine user interested patterns, constraint based pattern mining techniques are used. Application point of view, there are seven constraints are defined namely Item, Length, Super Pattern, Aggregate, Regular expression, Duration and time [3].

Most of pattern mining algorithms were used support count technique. But support count technique has two main problems as 1. If support count is low, it will generate huge useless patterns and 2. If support count is high, it remains undercover some important patterns.

In DNA or biomedical analysis some genes are causes particular disease and are important for fighting with diseases [4]. In this case importance of genes is helpful.

So, minimum support base technique is not enough to find frequent useful patterns. To tackle this issue, many researchers have presented weighted pattern mining algorithms [4-6] to discover useful patterns according to importance.

In weighted pattern mining technique, weights are assigned to each item according to importance. Value must be positive integer and lies between 0 and 1.

Length of pattern is also one of important issue related to frequent pattern mining technique. It is important that long sequential patterns having min support may be important whereas short patterns having high support may not be useful. Therefore long important patterns may be easily pruned whereas unnecessary short patterns may be generated as frequent patterns. To solve this problem we use length constraint.

To overcome these drawbacks, present algorithm uses Regular Expression Constraint with weights and length constraint.

Basic Concept and Terminology

In this work we introduce some important terminology discussed as follows.

Length of sequence:-Let I={i1,i2,...,in} be a set of all items. A S={s1,s2,...,sn} is ordered list of items. |S| is size of sequence S i.e. total number of items in sequence.

Weights of an Item: -The weight of an item is non-negative real value lies between 0 and 1.

Weights of a Sequence: - A weight of sequence is nothing but average value of items in a sequence denoted by SeqW. Average can be taken by using simple statistical mean.

Frequent weighted Sequence:- Frequent weighted sequence is nothing but Weights of a Sequence(SeqW) which satisfy min_weight i.e. SeqW≤min_weight.

Min_length:- which represent minimum length of sequence. Min_length contains positive integer value greater than 0. min_length>0.

Max_length:- which represent maximum length of sequence. Max_length contains positive integer value greater than 0 and must be higher than min_length. i.e. max_length>min_length.

Present algorithm discovers frequent sequential patterns which satisfy length constraint and weight constraint.

Literature review

Garofalakis, Rastogi and Shim [8] introduced "Regular Expression (RE)" constraint which is flexible constraint and used to focus on user defined patterns. Author developed framework or set of algorithm known as "SPIRIT (Sequential Pattern mIning with Regular expressIonconsTraint)" which support user defined RE constraint. RE constraint is useful in a situation where anti-monotonicity like properties is not useful. Experimental study shows that SPIRIT algorithm's efficiency and effectiveness is very good and its speed is also good as compare to other algorithms.

Some of the constraints used in sequential pattern mining don't satisfy regular nor context free property of automata so it is not

possible to develop equivalent DFA or PDA. For implantation of formal language tool in data mining, author uses regular expression constraint for sequential pattern mining in his paper [9]. Author developed new algorithm i.e. "Pattern Mining Using Formal Language Tools (PMFLT)" for sequential pattern mining. By using formal language tools and regular expression in pattern mining gains it can reduce search space and discovered only user interested patterns. Author construct Turing machine for this purpose.

This paper [11] discusses about sequential pattern mining technique for identification of persons in global environment. This algorithm works in three steps 1. Identify frequent patterns 2. Assign weights to mined pattern and 3. Classify the pattern. For classification purpose author uses classification algorithms and suggest that episode weight technique is more reliable than frequency based technique. Experimental evaluation also shows that, longer pattern will get higher accuracy than smaller size pattern.

YUN and RYU [10] use weight and length decreasing support constraint where weight constraint considers importance of the item in the sequence. Generally long patterns have low support but pattern are important. Algorithm pushes weight and length decreasing support constraint deep into mining process and uses prefix projection growth method. Weighted Smallest Valid expression property is used for pruning search space. Experimental study shows that algorithm outperformance SPAM

Frequent Sequential Pattern mining with Weighted Regular Expression and Length Constraint Algorithm

Present algorithm uses weights, length and RE for discovering frequent patterns. This algorithm takes input as weight of each item, regular expression, and min_length and max_length.This algorithm scan database only once and find frequent sequential patterns which satisfy given min_weight, min_length and max_length. Present algorithm discovered user interested patterns with consideration of importance of an item as well as consider length of an item.

Algorithm works in following steps

1. Algorithm scan database only once and discovered patterns which satisfy give regular expression constraint.

2. Repeated patterns were deleted and recorded the count of patterns.

3. Find average weight of each sequence (SeqW) and prune sequences which don't satisfy given min_weight.

4. Identifies sequences which satisfy min_length and max_length and delete other sequences which don't satisfy length constraint. Select those sequence who satisfy weight constraint.

5. Display frequent sequential patterns which satisfy given regular expression, given weight and give length constraint.

Algorithm

Input: Database Sequence, Regular Expression, weights of each item, Minimum Sequence weight and length constraint i.e. min_length and Max_length

Output: Frequent sequential patterns.

Step 1:-Extract list of patterns from DNA sequence which satisfy given Regular Expression RE.

Matchcoll=RE.matches(S1)

Where S1 is Sequence and RE is regular expression.

PatternList=FindPattern(S1,RegularExpression)

For i=0 to total_pattern

Ptternlist[i]=matchcoll[i]

Step 2:- distinct patterns can be selected.

DistinctPatterns=IsPresent(PatternList,Pattern)

For i=o to patternlist.count

If pattern=patternlist[i] then

Add to distinctpattern

Delete repeated patterns.

Step 3:- Count occurrence of each pattern

For I =0 to patternlist.count

If pattern=patternlist[i]

Countpattern[j]++

Step 4:- find Sequence weight for each sequence

For i = 0 to pattern_length

avgwt = avgwt + w.weight;

SeqW=avgwt/pattern.length.

Step 5:-Select those pattern who satisfy min_length and Max_length

Step 6:-Select those patterns which satisfy minimum weight. And display those patterns with sequence length, its frequencies and SeqW.

FrequentPattern=SelectedPattern(PatternList,DistinctPattern,MinimumSupport)

For (i=0;i<distinctpattern.count;i++)

//compare with patternlist

Experimental Result

All experiments are carried out with Intel Core i3, 3.3 GHz processer with 2 GB RAM and Windows XP operating system. This algorithm is implemented in Visual Studio 2008 and for development of application C# language is used. For experimental result, real life DNA sequence is used which is downloaded from the web site of National Centre for Biotechnology Information [4].

For experimental purpose DNA sequence having length 11585 is taken. The part of DNA sequence is shown as follows.

This algorithm is implemented in visual studio 2008 using C# language. Algorithm accepts DNA sequence, Regular Expression, weights of each element, minimum average weight of sequence and min_length and max_length. In this experiment weights of each item inputted as A=0.9, C=0.4, G=0.3 and T=0.6. Remember that weights values are lies between 0 and 1. This value represents importance of an element. High value means high importance and low value means low importance.

Average weight of sequence is 0.6 and length lies between 6 and

Volume : 4 | Issue : 5 | May 2015 • ISSN No 2277 - 8179

8. When program gets executed and find frequent sequential pattern which are shown in table. This table also shows frequency i.e. number of times that pattern appear, length of sequence and average weight of each sequence. For this experiment regular expressionA*C*G*T*[ACGT][AT]A* is taken. This experiment also finds total number of patterns which satisfy given regular expression. It also shows number of frequent sequential patterns which satisfy weights and lengths constraint and it display time (in Milliseconds) requiredto generate patterns.

This experiment also display discovered patterns, number of occurrence of pattern i.e. frequency, length of sequence and weight of sequence/pattern. Weight of patterns will be calculated using statically mean technique. This experiment shows that if frequency is less still patterns are frequent because of weight value of sequence.

Following table shows output summarized values of experimental implementation of algorithm.

Total Length	11585					
Element	A	C	G	T		
Weight	0.9	0.4	0.3	0.6		
Regular Expression	Average Weight	Min length	Max length	Total Pattern	Frequent Pattern	Time
A*[GC][TC]A*	0.4	3	10	91	86	18.0136
	0.6	3	10	91	82	20.0096
	0.8	3	10	91	7	69.0457
A*C*G*T*[ACGT][AT]A*	0.4	3	10	438	286	59.0369
	0.6	3	10	438	130	47.0311
	0.8	3	10	438	9	38.0253
A[G*T][CA*][AT]T*	0.4	3	10	31	30	6.004
	0.6	3	10	31	26	4.0039
	0.8	3	10	31	1	3.0034

Table 1:- Summarized result of implementation of algorithm with different RE

From table1 shows number of frequent pattern discovered for different regular expression and for different weights. From this it observed that as weight value increases, discovered patterns will be less. Also above table shows time required (in MS) to generates frequent pattern by different regular expression and for different weights. From this it conclude that as weights values increases, time required for discovering patterns will be less.

Following table summarizes the result of program for single regular expression.

Total Length	11585	Average Weight	0.6	
Element	A	C	G	T
Weight	0.9	0.4	0.3	0.6
Regular Expression	A*C*G*T*[ACGT][AT]A*			
Min length	Max length	Total Pattern	Frquent Pattern	Time
3	10	438	200	42.0287
4	10	438	188	42.027
5	10	438	162	40.0282
6	10	438	116	38.0261
7	10	438	70	37.0231
8	10	438	37	39.0261
9	10	438	16	35.0235
3	8	438	184	44.0284
3	9	438	196	56.0368
3	11	438	203	43.0284
3	12	438	204	43.0288
4	8	438	172	39.0261
4	9	438	184	43.028
5	9	438	158	40.0261

Table 2:- Summarized result of implementation of algorithm with Single RE

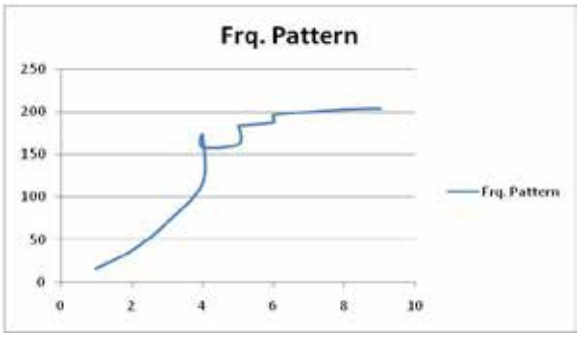
Following table 2 shows frequent sequential pattern generated for different length. Here present work vary min_length and max_length and find frequent sequential pattern and also calculate time require to generate that pattern.

Following table 3 generated from table 2. Here length of pattern will be calculated by taking difference between max_length and min_length.

i.e. Length of pattern= max_length - min_length

Length of pattern	**Frequent Pattern**
1	16
2	37
3	70
4	116
4	172
4	158
5	162
5	184
5	184
6	188
6	196
7	200
8	203
9	204

Table 3:- Shows length of patterns i.e. gap and Frequent patterns



Graph 1:- Shows length (Gap between max_length and min_length) vs generated pattern.

From above table 3 and graph 1, it is clearly shows that as gap between max_length and min_length increases, generated patterns also increase.

Following table 4 shows output of our experiment. Different parameters which are assigning to experiment are shown.

Average Weight	0.6	Length	min_length=3 and max_length=10
Regular Expression	A[G*T][CA*][AT]T*		
Time (in MS)	5.0027	Frequent Pattern:26	
Pattern/ Sequence	Frequency	Length	Weight of sequence
AGAA	66	4	0.75
AGAT	40	4	0.68
AGAAT	18	5	0.72
ATCA	32	4	0.7
ATCT	32	4	0.62
AGCA	60	4	0.62
ATAA	25	4	0.82
AGCAT	10	5	0.62
AGATTT	4	6	0.65
AGAATTT	1	7	0.69
AGATT	6	5	0.66
ATAT	20	4	0.75
AGCATTT	1	7	0.61
ATCTT	6	5	0.62
ATATT	7	5	0.72
ATAAT	2	5	0.78

Table 4 :- Output of experimental study shows for specific instance

Observation

In this algorithm three different constraints are used namely Regular expression, Weight and Length. Regular expression is useful to define user's interest for pattern requirements. From that requirement regular expression will be generated. Also present work use weight and length constraint to overcome drawback of support constraint.

From experimental evaluation, it is clearly observed that algorithm generates user interested patterns within very less time. Also from table 1, it is observed that increase in weight constraints, discovered patterns will decrease. And average weight constraint is decreased, number of discovered patterns increases. From table 2 we observed that for small value of min_length, number of discovered frequent sequential patterns will increases and as we increase min_length, discovered frequent sequential patterns will decreases. Similarly in case of increase value of max_length, discovered sequential patterns will increase and as decrease max_length, discovered sequential patterns will decrease. From table 3 it observed that if gap between max_length and min_length is decreased, discovered frequent sequential patterns may be decrease. Length plays very important role because it can prune number of sequences. In some situation, user is only interested in specified length of pattern in such case this length constraint is important.

From table 4 we observed that although frequency of some patterns may be less, still its shows frequent. In above table4for example frequency of pattern ATCATT is 1 still it is frequent sequential pattern. This is because; pattern's weight is higher than minimum weight. It shows that items in these patterns are important according to the weights assigned.

Conclusion

Large numbers of algorithms were presented to mine frequent pattern but most of them used constant min support. One of problem for this technique is setting ideal support. If support value is low, huge number of patterns will be generated. Some of them are useful and some of them are not. If support value is high then it undiscovered some important patterns. Here proposer work uses Regular Expression constraint in pattern mining process to generate frequent pattern of user interest. Also in mining process present algorithm introduce weights of each item which are importance indicator of each item. This algorithm generates frequent pattern which satisfy min weight and regular expression constraint. Also in this algorithm uses length constraint which restricts length of pattern. Here presentwork introduces min_length and max_length property which takes length interval in integer form.

This algorithm works in five steps which takes input as regular expression, DNA sequence, minimum weight and length constraint and produce frequent sequential patterns. Here algorithm uses regular expression for discovering user interested patterns. Weights are used to discover pattern according to importance of items and length constraint is used to restrict length of pattern so as to reduce search space and find user interested pattern efficiently and effectively.

Gap between patterns plays important role in analysis of pattern mining. In future our plan will to add gap constraint in mining process. Also we may apply fuzzy weight in future.

REFERENCE

1. Yun, Unil. "WIS: Weighted Interesting Sequential Pattern Mining with a Similar Level of Support and/or Weight." ETRI Journal (Vol. 29, No. 3, June 2007). | 2. YUN, UNIL and KEUN HO RYU. "Discovering Important Sequential Patterns with Length-Decreasing Weighted Support Constraints." International Journal of Information Technology & Decision Making (Vol. 9, No. 4, 2010): 575-599. | 3. JianPei,Jiawei Han: Constrained frequent pattern mining: A pattern Growth view, SIGKDD Explorations Volume 4 Issue-1 page 31. | 4. Yun, Unil. "A new framework for detecting weighted sequential patterns in large sequence databases." Knowledge Based System (2008): 110-122 | 5.Ren, Jiadong, Shuo Li and Jun Dong. "Mining Weighted Closed Sequential Patterns by Memory Indexing ." International Journal of Digital Content Technology and its Applications (JDCTA) (Vol. 6, No. 19, October 2012): 432. | 6.Tao, Zaiping. "An Efficient Algorithm for Weighted Sequential Pattern Mining." International Journal of Advancements in Computer Technology (IJACT) (Vol. 2, No. 2, February 2012): 88. | 7. Dipak R. Kawade, Dr. Kavita S. Oza, "Exploration of DNA Sequences Using Pattern Mining", International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS) ISSN (Online): 2279-0055, ISSN (Print): 2279-0047, Issue 6 Volume 2, pp. 144-148, September-November, 2013 | 8. Minos N. Garofalakis, Rajeev Rastogi, Kyuseok Shim: SPIRIT: Sequential Pattern Mining with Regular Expression Constraints, Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, 1999. | 9. Sunil Joshi, Dr. R. S. Jadon and Dr. R. C. Jain, Sequential Pattern Mining Using Formal language Tools ,IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012, ISSN (Online): 1694-0814 | 10. Unil Yun and Keun Ho Ryu, Discovering Important Sequential Patterns With Length-Decreasing Weighted Support Constraints, International Journal of Information Technology & Decision Making, Vol. 9, No. 4 (2010) 575–599,World Scientific Publishing Company, DOI: 10.1142/S0219622010003968 | 11. Belkacem Chikhaoui, Shengrui Wang, Hélène Pigot, A New Algorithm Based On Sequential Pattern Mining For Person Identification In Ubiquitous Environments, SensorKDD'10, July 25, 2010, Washington, DC, USA. | 12. R. Agrawal, R. Srikant: "Fast Algorithms for Mining Association Rules", Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, Sept. 1994. Expanded version available as IBM Research Report RJ9839, June 1994. |