CWFM: Closed Contingency Weighted Frequent Itemsets Mining

Eunkyoung Park, Younghee Kim, Ieejoon Kim, Jaeyeol Yoon, Jiyeon Lim, and Ungmo Kim

Department of Computer Science and Engineering, Sungkyunkwan University,
300 Chunchun-Dong Jangan-ku, Suwon,
Gyeonggi-do 440-746 Republic of Korea
{wjbest527,younghee}@gmail.com,
{uk3080789,vntlffl,01039374479}@naver.com, umkim@ece.skku.ac.kr

Abstract. Weighted pattern mining have been studied the importance of items. So far, in weight constraint based pattern mining, the weight has been considered the item's price. The price considered as the weight has a limit. The weight characteristic of weighted pattern mining should be considered case-by-case situation. Thus, we motivate by considering the special and individual case-by-case situation to find the exact frequent patterns. We propose how to set weight into frequent patterns mining with a case-by-case condition, called CWFM (closed contingency weighted pattern miming). Moreover, we devise information tables by using statistical and empirical data as strategic decision. In addition, we calculate the contingency weight using outer variables and values which are from information tables. CWFM extracts more meaningful and appropriate patterns reflected case-by-case situation. The proposed new mining method finds closed contingency weighted frequent patterns having a significance which represents the case-by-case situation.

Keywords: closed weighted frequent patterns mining, information table, contingency weight, contingency weighted frequent patterns, outer-variables.

1 Introduction

Most algorithms use a support constraint to prune infrequent patterns and to reduce search space [4-7]. It is difficult for user to decide support value efficiently. One of the main limitations of the previous algorithms are treated all the items uniformly, but real items have different importance than other items [8-13]. For this reason, weighted frequent itemset mining algorithms [8-13] have been suggested that some items should be given priority, such as WFIM [8] is used to the price factor of item as priority. However, it dose not consider different factors for using weight. Here, it is necessary to explain factor in connection with weight. If special situation is considered, the substantial results can be to obtain frequent pattern considering the situation. In addition to the price factor, weights can be determined by other factors in a wide of dataset. That is, there are climate, area, age, wedding season, national holidays, gender and countries etc. For instance, sales

P. Perner (Ed.): ICDM 2012, LNAI 7377, pp. 166–177, 2012.

volume of a thermal underwear and bikini are influenced by season. Therefore, other factors except price may be various and can make more exact mining result.

In this paper, we consider to find the weighted frequent patterns with weighted factor. The proposed algorithm permits to specify the weight considering case-by-case situation, called CWFM. The weight value of item has differences among data, so that the process to determine the weight is needed before mining. It has two steps. First, Information tables: these collect information by using the substantial and empirical data from reliable institution. For example, National Statistical Office, Child Research Institute, Criminal & Investigation Laboratory and answer of survey etc. Secondly, Outer-variables: these are received data to reflect the special situation.

The chosen values from information tables compose a matrix. A determinant of matrix calculates by using the gauss elimination method. The calculated result value is called the contingency weight. Our main goal is to push the contingency weight constraints into the pattern growth algorithm. The main contributions of this paper for the real dataset are: 1) introduction of concept to provide the contingency weight based on the practical situation, 2) description of mining technique in closed frequent pattern mining with contingency weight, 3) implementation of our algorithm, CWFM.

The remainder of this paper is organized as follows: In section 2, we describe problem definition and related work. In section 3, we developed our proposed method and algorithm for weighted contingency frequent patterns. In the section 4, we present performance comparison. Finally, conclusions are given in Section 5.

2 Related Work and Problem Definition

2.1 Related Work

Weighted frequent patterns mining algorithms [8-13] based on the pattern growth method[4] improved the problems of the support constraints with weight constrains. The first, WFIM[8] suggested the method in which normalized weights with a weight range. The closed frequent pattern mining indicates the same knowledge of patterns. If a pattern X is closed frequent pattern mining, there are not proper superset X' in the transaction. The weight value of item has differences among data, so that the normalization process is needed. First, weights of items are given weight range table. According to items' importance, weight of items are given in $w(_i)_{min} \leq w(_i) \leq w(_i)_{max}$ for the item. The weight of a pattern is the sum of items divided by length of pattern. Maximum weight (MaxW) is defined the value of maximum weight among items in transaction. The weighted support of a pattern (WS(X)) is the value of multiplying the weight of pattern with the support of pattern. The weighted support is defined such as: WS(X) = Weight (X) * Support (X) and it should be grater than min_sup. Previous studies did not consider unique circumstance as the factor of weight. This paper can describe mining method considering contingency weight in special situation.

2.2 Problem Definition

Let I = $\{i, i, \ldots, i_n\}$ be a distinct set of items. A transaction database TDB is a set of transactions in each transaction. A A transaction is denoted as a tuple <tid, X>. The tid indicates a unique transaction identifier. The X= { $x_{1,}$ x_2, …, x_m }, $x_i \in I$, for $1 \leq i \leq m$, is a set of items, while m is called the length of itemset. An itemset is ordered list, X ∈ I ($2^n - 1$) where 2^n is the power set of I. An itemset = {x_1, x_2, …, x_n} is also represented as x_1 x_2, …x_n. An item is called a k-itemset if it contains k items. The support of a itemset X is the number of transactions containing X in the database. A weight of an item is importance or priority of item. The weighted frequent pattern mining is to find the complete set of patterns satisfying a support threshold and a weight constraint in the database. The closed frequent pattern is superset X′ if the support of pattern X equals to that of X’ and the length of X is less than that of X’ and every transaction containing X also contains X′. An example of the support constraint [1],[5],[12] is the support constraint. On the other hand, a pattern X is infrequent pattern, then, super patterns of the pattern must be infrequent patterns. Hence, infrequent patterns can be pruned to reduce search space.

Table 1. Transaction of database TDB

TID	Set of items	Frequent item list(min_sup =3)
100	a, e, h, g, u	a, e, h, g
200	a, h, g, n, u	h, g, a, n
300	e, g, n, t	g, n, e
400	a, e, h, g, t, w	h, g, a, e
500	h, g, e, w	h, g, e
600	e, g, n	g, n e

Example 1. Given a transaction database TDB in the Table 1. We have six transactions and 8 items: <a, e, h, g, n, t, u, w>. Suppose that we have minimum support = 3. A frequent list is:<a:3, e:5, h:4, g:6, n:3>. Items “t”, “u” and “w” are pruned because the support of these items is less than a min_sup. A super pattern “hga” is a closed frequent pattern because the support (3) of the pattern is equal to the support (3) of a pattern “ga” and the length(2) of “ga” is less than that(3) of “hga” and every transaction containing “ga” also contains “hga”.

3 CWFM

In this Section, we suggest the CWFM algorithm with the concept of contingency weight. The main approach of CWFM is to push contingency weight into the closed weighted frequent patterns mining algorithm based on the pattern growth method and prune uninteresting patterns. In CWFM, a new measure of weight is defined and related properties are described. Sequential patterns mining have been treated uniformly, but real items or datasets have different importance in special situation. We present our algorithm in detail and give statistical examples to explain the adaptation of contingency weight in the FP-tree construction, then show the projected FP-tree using by bottom up traversal of FP-tree.

3.1 Preprocessing

The preprocessing is to collect the necessary data in the actual situation. Before the mining, the preprocessing is needed. We will explain the health consultation in order to discover the association between diagnosed illness and other diseases. There are information table and outer-variable. The values of information tables can be found by using in the real world statistical data and will be used as the weight value.

Proposal 1. Information tables: These can be made usefully depending on the situation in each dataset and the number of information tables can vary. We collect the data that is associated with the characteristics of dataset. Here, the information tables are based on 2009 year data by Korea Statistical Information Services and are statistical data about deaths and illnesses. Our study consists of three information tables to explain the health consultation. One of them is total death rate table, other is death rate table for the generations and the other is death rate for smoking table. One of created tables should be set a standard table because value of item in standard table is used to header's weight of FP-tree. We will consider the death rate table(table 2) as the standard table. All of information tables, table1, table2, table3, table4, are shown below.

Table 2. Total death rate table

item	a	E	h	g	n	t	u	w
rate of death	**0.41**	**1.61**	**0.06**	**0.19**	**0.9**	**1.05**	**0.39**	**0.26**

Table 3. Death rate for the generations table

age item	a	e	h	g	n	t	u	w
0 ~ 9	9.76	8.51	14.92	8.48	30.55	34.83	11.23	11.42
10~19	6.52	5.68	9.97	5.68	20.41	23.27	7.5	7.64
20~29	6.54	5.70	10.0	5.69	20.46	23.34	7.53	7.65
30~39	6.56	5.72	10.06	5.73	20.54	23.46	7.57	7.70
40~49	6.53	5.69	10.14	5.78	20.65	23.63	7.62	7.77
50~59	6.46	5.41	10.20	5.91	20.82	23.92	7.67	7.94
60~69	6.19	4.63	9.93	6.15	21.12	24.44	7.67	8.28
70~79	5.52	3.44	8.48	6.69	21.73	24.81	7.31	9.04

Each item in the tables indicates name of a disease. Then, item "a" is stomach cancer, "e" is liver cancer, "h" is lung cancer, "g" is hypertensive heart disease, "n" is a cardiac disorder, "t" is a cerebrovascular disorder, "u" is diabetes mellitus and "w" is pneumonia in the tables. In 2009, the death toll is 246,942 persons (then, crude death rate of 497.3 per 100,000). The term "element" can be defined as value in the information table. The element of each item refers to the mortality of each disease in tables. For example, the element of item "e" represents 1.61 in the table 2. Every

1-row in tables indicates names of disease. The 1-columun in the table 3 means the generations. Finally, 1-columun in table 4 means the number of cigarette smoking per a day.

Table 4. Death rate for smoking table

num \ item	a	e	h	g	n	t	u	w
0~9	34	6.2	7.4	0.55	1.89	5.44	0.77	0.75
10~19	1.98	5.4	6.5	0.45	1.26	3.56	0.23	0.63
20~29	1.85	4.32	1.12	0.23	1.12	3.45	0.27	0.34
30~39	1.87	4.33	1.23	0.25	1.13	3.66	0.29	0.33
40~49	1.98	5.4	1.50	0.32	1.26	3.67	0.33	0.36
50~59	31.27	5.71	1.73	0.45	2.41	5.34	0.66	0.5
60~69	4.02	3.86	2.16	0.68	2.26	7.16	0.85	0.64
70~79	2.09	1.19	2.09	0.48	1.27	5.63	0.48	0.64

Proposal 2. Outer-variables: **I**t may be determined by reflecting the situation. In other word, there are items that represent the situation among items. The item values matching outer variables are selected in each table as an element of the matrix. For instance, assume that there is one man who is 43 years old and suffered from pneumonia in the past, and smokes 10 cigarettes per a day. Here, the outer variables set up 43 years old, pneumonia, 10 cigarettes. Here, outer variables are 'e' , 43 and 10.

3.2 Contingency Weight

As shown in the above method, the values extracted from each table become elements of square matrix. The determinant of matrix computes by using the Gauss elimination method and is defined det(Matrix). Then, we are set the value of '1' at empty elements. Because the value of 1 does not influence any other number even if multiply any number by 1. The det(Matrix) is multiplied to value of items in the standard table matching outer variables, respectively. At this time, calculated result value is called the contingency weight. Therefore, the weight can be increased by considering case-by-case situation as weight factor. The item value from other tables matching outer variable are extracted. Then, these constitute the matrix. The determinant of matrix computes using the Gauss elimination method. That is, it is way to increase the importance of item related special situation.

Definition 1. [Contingency Weight (CWeight)] There are given weight of item matching outer-value(E) in the standard table and the value of determinant of matrix det (matrix). Three tables made the matrix with row 3 and column 3.

The contingency weight is defined as follows.

$$\boxed{\text{CWeight (E) = weight (E) * det (matrix).}}$$

Example 2. Assume that outer variables are entered item “a” and item “e”, additionally, age "43" and the quantity of smoking per a day "10" are entered. The item “a” selects “0.41” and the item “e” picks “1.61” in the standard table (Table 2), respectively. Outer variables are treated ascending order in items. The each value becomes element of the matrix as 1-row 1-column and 1-row 2-column. Next, the 1-row 3-column is set "1" as the default value. Similarly, by using “43” entered as an outer variable, 43 and the item “a” intersect at the point value "6.53", "43" and the item "e" intersect at the point value "5.69 " in the death rate for the generations table. Finally, the item "a" and "10" intersect at the point value "1.98", the item "e" and "10" intersect at the point value "5.4" in the smoking table . As a result, the 3 x 3 matrix is constructed.

$$|\mathrm{Matrix}| = \begin{vmatrix} 0.41 & 1.61 & 1 \\ 6.53 & 5.69 & 1 \\ 1.98 & 5.4 & 1 \end{vmatrix}$$

Fig. 1. Matrix

According to the Gauss elimination method, the matrix changes as fig 2.

$$|\mathrm{Matrix}| = \begin{vmatrix} 0.41 & 1.61 & 1 \\ 0 & -19.95 & 14.93 \\ 0 & 0 & -6.61 \end{vmatrix}$$

Fig. 2. Result Matrix

Fig 2 is upper triangle matrix. The determinant of det (matrix) is about 54.07. So, item “a” and item “e” in the standard table(table2) are multiplied by 54.07 respectively. That is, a’s and e's contingency weight are 22.17 and 87.05 . The weight of item “a” is increased from “0.41” to “22.17” and the weight of item “e” is changed from “1.61” to “87.05” in the standard table.

3.3 Contingency Weight Pattern

The frequent weight pattern found by using the Definition 1 means the contingency pattern. Our algorithm reflects to the weights by considering the characteristic of dataset and after collecting the empirical data. The different way of increase the importance of item proposes. There is strong point that user can get proper and accurate information.

3.4 FP (Frequent Pattern) Tree Structure

CWFP uses FP-trees as a compress structure and are used in pattern growth algorithms. CWFP computes local frequent items of a prefix by scanning its projected database once. The Contingency weight(CWeight) increases the weight of item matching outer-variables. So, CWeight is greater than weight in the standard table.

After this, sort the items in weight ascending order. A header table of FP-tree has four fields: “item-id”, “weight”, “support” and “node-link”. The first filed, Item-id is distinct item. The second field, weight indicates weight of items including contingency weight(CWeight). The third field, support is used to count item in the transaction. The last field, node-link is linked to child-node with same item in the FP-tree. Top of FP-tree has only a root node. The first node of root in the FP-tree has item-id and pointer. When an item is inserted in the FP-tree, sort the items in contingency weight ascending order. For instance, diseases are “a”, “e”, “n” and age is “43” and the quantity of smoking is 10 as outer variables. The item “a” selects “0.41” and the item “e” picks “1.61” and the item “n” select “0.9” in the standard table(table 2). The each value becomes element of the matrix as 1-row 1-column, 1-row 2-column and 1-row 3-column. In table 3, by using “43”, 43 and the item “a”, “e” and “n” intersect at the point value "6.53", "5.69" and "20.65", respectively. Finally, each item "a", "e", "n" and "10" intersect at the point value "1.98", "5.4" and "1.26" in table 4. As show in Fig 2, a result table, the matrix is constructed.

$$\begin{bmatrix} 0.41 & 1.61 & 0.9 \\ 6.53 & 5.69 & 20.65 \\ 1.98 & 5.4 & 1.26 \end{bmatrix}$$

Fig. 3. Matrix

The determinant of matrix is 25.29. As discussed above, a f_list is:<a:3, e:5, h:4, g:6, n:3 > from table 1. Items “t”, “u” and “w” are pruned since the support of them are less than a min_ sup (3). Before construct a header of FP-tree, contingency weight is computed. That is, weight list are < a:10.37, e:40.72, h:0.06, g:0.19, n:22.76 >. So, the weighted frequent items are sorted by the weight ascending order, {(h, g, a, e), (h, g, a, n), (g, n, e), (h, g, a, e), (h, g, e), (g, n, e)} in each transaction. The sorted weighted frequent items in the each transaction are sequentially inserted in a FP-tree along a path from the root to the corresponding node. If a new node is inserted in the path, only the support of the node is set to 1. If an existing node in the FP-tree is used, the support of the node is increased by one. Fig 4 presents the FP-tree and a corresponding header table.

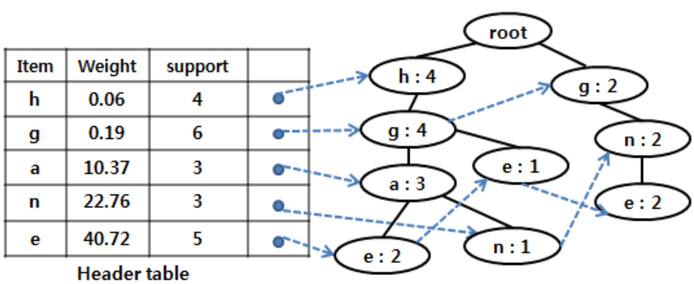

Fig. 4. Global FP-tree for CWFM

FP-tree with the contingency weight mine closed frequent itemsets by adapting bottom up traversal. FP-tree mines the patterns including item “e” which has the highest weight. First, e’s conditional database is generated by starting from e’s head and following e’s node-link. The conditional database for prefix “e” contains three transactions: {“hga:2”, “hg:1”, “gn:2”}. Item “a” and item “n” are pruned because item a’s support (2) and item n’s support (2) are less than a minimum support (3). A local conditional FP-tree with the prefix “e” are “hge:2”, “hg:1” and “gn:2”. All kinds of combinations of item related to the prefix “e” are “e:5”, “he:3”, “ge:5” and “hge:3”. The patterns “e:5” and “he:3” are pruned since the contingency weighted frequent pattern “ge:5” and “hge:3” are superset with same support according to closer property, respectively. As a result, closed contingency weighted frequent patterns for the prefix “e” are <ge:5>, and <hge:3>. It is shown fig 5 (a).

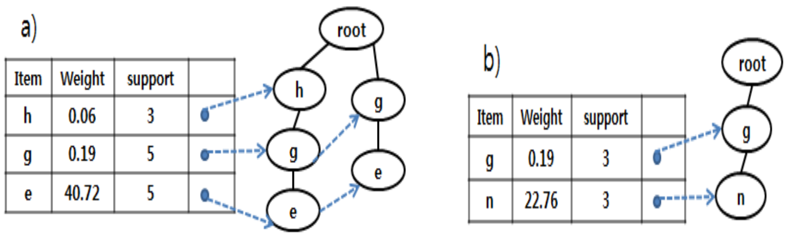

Fig. 5. Local conditional with the prefix “e” (a) and “n” (b)

From second the local conditional FP-tree with the prefix “n”, we can find out the contingency weighted frequent patterns are “hga:1” and “g:2”. We get closed contingency weighted frequent patterns related to “n:3” : <gn:3 >. So, it is closed contingency weighted pattern. Then, we can construct local conditional FP-tree from the FP-tree and mine closed contingency weighted frequent pattern from them recursively. Closed contingency weighted frequent pattern with the prefix “a” is <hga:3> and with the prefix “g” is <hg:4> and with the prefix “h” is empty.

3.5 CWFM Algorithm.

The algorithm for the CWFM is described as follows: Initially, the CWFM scans TDB once to find the f_list. Then, considering characteristic of dataset, analyze what data should be used to it and set outer-variables. This consist information tables and outer-variables. Secondly, a matrix is constructed weight of item in each table matching outer-value and det(matrix) is computed using the Gauss elimination method. Then, the contingency weight obtains, to adopt weight of item matching outer-value in standard table, value of multiplying item’s value with det(matrix). Thirdly, a global FP-tree is constructed from weight of f_list, in weight ascending order. Finally, contingency weighted frequent mining is performed.

CWFM Algorithm

Input :

D : A transaction database (TDB),

δ: minimum support threshold : min_sup,

õ: outer variables,

£ : information tables from the national statistical office

Output :

CWFP: The complete set of closed contingency weighted frequent pattern.

Method :

```
Generator in D1 ← {1-itemsets};        //D1 ← support-itemset (D1);
for each generator p ∈ D1 do begin
    if (support(p) < δ) then delete p from D1;
    else F_list ← p;                     //F_list: frequent item list of D1
end
õ[m] ← Input outer variables
   max = m;   // õ[m] → here, m is number of outer variable
    for ( i = 1; i <= max ; i++ )  £ [i];   //information tables create
     end
  for(y=1; y< number of õ; y++)    // matrix create
   for(z =1; z< number of õ; z++)
     for (i =1; i <= max; i++)    //for each information table
        for(n =0; õ[n] <= max; n++)
           for(row=1; row<=£[i].length; row++)
            if (£1[row] = = õ[n] ) v=row;
            end if
           end
        end
       for(n =0; õ[n] <= max; n++)
          for(col=1; col <= £1[i].length; col++)
            if (£1[col] = = õ[n] ) w=col;
                   end if
                end
             end
           matix[y][z] = £k[v][w];
      end
     end
   end
   det (matrix) ← matrix;
   for each o ∈  õ [i] do begin   //calculate contingency weight
      if (p = = o)   then ć= p (in standard table) * det (matrix) ;
      end if
```

Fig. 6. Pseudo code for the CWFM

```
    end
    Weight (p) ← ć // ćis contingency weight
  Header's weight ← Sort items in contingency weight ascending order.
  Call call FP-tree(f-list);
Call Conditional FP-tree (FP-tree, {}, CWFM);
```

Fig. 6. *(continued)*

4 Experimental Results.

4.1 Performance Comparison

In this section, we present the performance of our CWFM algorithm by using contingency weights. We also compared CWFM with WFIM[8]. CWFM was written in Java. Experiments were performed on a 366Mhz Pentium PC with 512MB main memory, running on Microsoft Windows/XP. WFIM adjusts by setting weight ranges to reduce the number of frequent itemsets when giving weights to each item.

The main purpose of experiments is to certain how efficiently the weighted frequent pattern can be found by using contingency weight. In this performance test, number of information tables and number of pattern are checked. Fig 6 and Fig 7 show the results of performance evaluation based on the connect dataset which contain 100k to 700k transactions. This dataset is real dataset and have 43items in the each transaction.

Fig 6 illustrates that performance when using and CWFM is better than using WFIM.

Although the number of information tables increase, Fig 7 also show that the slope of CWFM is lower than that of WFIM. CWFM sees somewhat better scalability than WFIM.

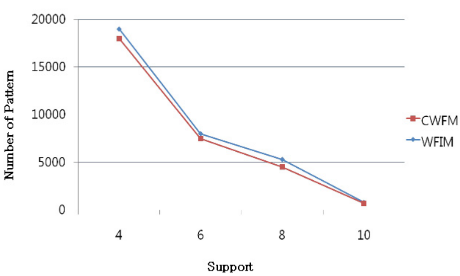

Fig. 7. number of patterns

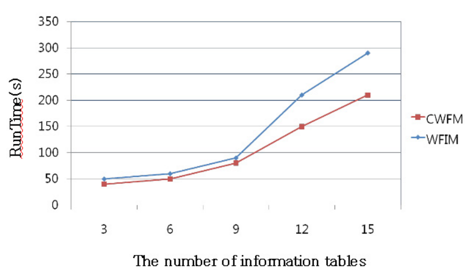

Fig. 8. Runtime for the CWFM

Table 5. The number of frequent patterns by increasing outer_variable

min_support	Num of the outer_variables	Num of contingency weight pattern
54046(80%)	10	3,565
	20	935
	30	203

Table 5 lists the difference of number of frequent patterns by increasing the number of the outer-variables. It shows that the number of frequent patterns decrease considerably by increasing the number of outer_vatriables or information table. Of course, the values in the information table and the value of the contingency weight (CWeight) that reflects the real world. Therefore, the number of patterns may be different.

5 Conclusion

In this paper, we studied the problem of weight factor in weighted frequent pattern mining. In previous studies, the items are used prices as weight factor. However, the factors of items are various in dataset or application. We introduced information tables and the concept of contingency weighted pattern by using statistical and empirical data. CWFM focuses on closed weighted frequent pattern by considering special situation. To assign the weights of items in special dataset, outer-variables are emphasis. Hence, the contingency patterns are considerably valuable for considering special situations. In the forward, weighted pattern mining should be considered opinion of experts and the number of information table. This is to reflect substantial part in terms of information. The weights based information tables can be support to extract more accurate prediction information in the case-by-case situation and valuable patterns by adopting the contingency weight.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD 1993 (1993)
2. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: ICDE 1995 (1995)
3. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 1–17. Springer, Heidelberg (1996)
4. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000 (May 2000)
5. Pei, J., Han, J.: CLOSET: An Efficiently Algorithm for Mining Frequent Closed Itemsets. In: DMKD 2000 (May 2000)
6. Pei, J., Han, J., Mortazavi-Asi, B., Pino, H.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth. In: ICDE 2001 (2001)

7. Wang, J., Han, J., Pei, J.: CLOSET+: searching for the best strategies for mining frequent closed itemsets. In: ACM SIGKDD 2003 (August 2003)
8. Yun, U., Leggett, J.J.: WFIM: Weighted Frequent Itemset Mining with a weight range and a weight range and minimum weight. In: SDM 2005 (April 2005)
9. Yun, U., Leggett, J.J.: WLPMiner: Weighted Frequent Pattern Mining with Length-Decreasing Support Constraints. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 555–567. Springer, Heidelberg (2005)
10. Yun, U., Leggett, J.J.: WIP: Weighted Interesting Patterns with a strong weight and/or support affinity. In: SDM 2006 (April 2006)
11. Yun, U., Leggett, J.J.: WSpan: Weighted Sequential Pattern mining in large sequence databases. In: Proc. of the Third Int'l Conf. on IEEE Intelligent Systems (September 2006)
12. Yun, U.: Mining Lossless Closed Frequent Patterns with Weight Constraints. Knowledge Based Systems 20, 86–97 (2007)
13. Yun, U.: WIS: Weighted Interesting Sequential Pattern Mining with a Similar Level of Support and/or Weight. ENRI Journal 29(3) (June 2007)