Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

0-SM：A fast algorithm for mining Candidate Clusters in Pattern-based Clustering

Jingfeng Guo, Qian Ma,Hanfeng Liu

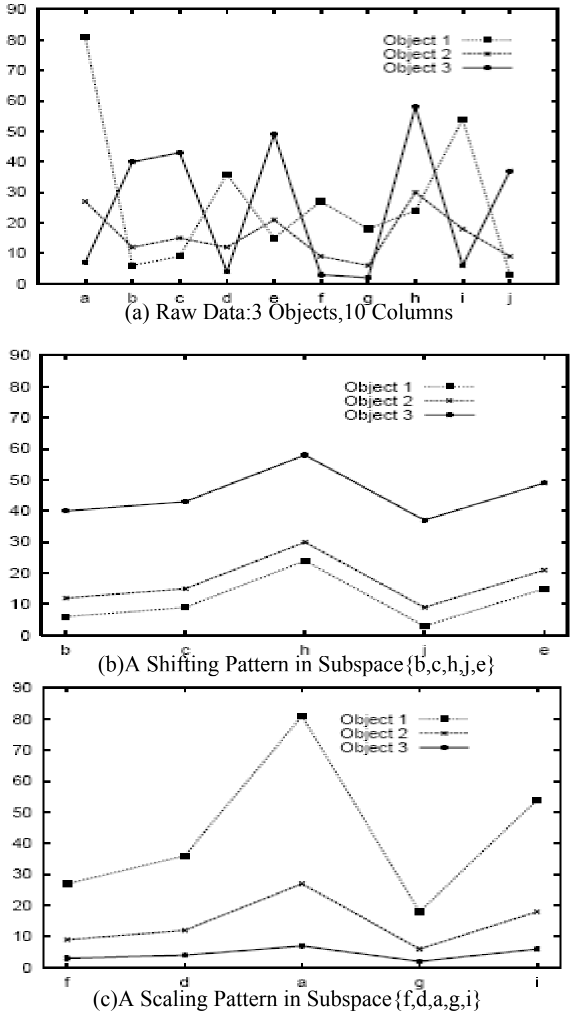
***Abstract*-Unlike traditional clustering methods that focus on grouping objects with similar values on a set of dimensions, clustering by pattern similarity finds objects that exhibit a coherent pattern of rise and fall in subspaces. Pattern-based clustering extends the concept of traditional clustering and benefits a wide range of applications, including large scale scientific data analysis, target marketing, web usage analysis, etc. However, state-of-the-art pattern-based clustering methods (e.g., the δ-pCluster algorithm), mining candidate clusters mostly by comparing each pair of attributes and objects, which have reduced the efficiency and makes them inappropriate for many real-life applications. This paper present a fast algorithm for mining candidate Clusters. We called it Zero-Sub-Matrix. It has a better efficiency than previous algorithms.**

1.INTRODUCTION

Clustering large datasets is a challenging data mining task with many real life applications. Much research has been devoted to the problem of finding subspace clusters [1, 2, 3, 4, 5]. Along this direction, we further extended the concept of clustering to focus on pattern-based similarity [6]. Several research work have since studied clustering based on pattern similarity [7, 8], as opposed to traditional value-based similarity. These efforts represent a step forward in bringing the techniques closer to the demands of real life applications, but at the same time, they also bring new challenges[9]. The standard of the pattern similarity based on pScore is inproved in this paper. A new standard of the pattern similarity based on spScore is proposed and a new algorithm of pattern clustering based on the spScore. The efficient of the new algorithm is better than the pClustering.

1.1 Pattern Similarity

We present the concept of subspace pattern similarity by an example in Figure 1. We have three objects. Here, the X axis represents a set of conditions, and the Y axis represents object values under those conditions. In Figure 1(a), the similarity among the three objects are not visibly clear, until we study them under two subsets of conditions. In Figure 1(b), we find the same three objects form a shifting pattern in subspace{ b, c, h, j}, and in Figure 1(c), a scaling pattern in subspace {f, d, a, g, i}.



(a) Raw Data:3 Objects,10 Columns

(b)A Shifting Pattern in Subspace{b,c,h,j,e}

(c)A Scaling Pattern in Subspace{f,d,a,g,i}

Fig. 1. Objects Form Patterns in Subspaces

This means, we should consider objects similar to each other as long as they manifest a coherent pattern in a certain subspace, regardless of whether their coordinate values in such subspaces are close or not. It also means many traditional distance functions, such as Euclidean, cannot effectively discover such similarity. A new clustering model is needed to capture not only the closeness of objects but also the similarity of the patterns exhibited by the objects. This

Manuscript received October 10, 2006. This work was supported by the National Nature Science Foundation of China under Grant 60673136.

J. F. Guo is with the College of Information and Science Technology, Yanshan University, Qinjhuangdao, Hebei, China.066004 (phone: 86-335-8050300; fax: 86-335-8074806; e-mail: jfguo@ysu.edu.cn).

Q. Ma, was with the College of Information and Science Technology, Yanshan University, Qinhuangdao, Hebei, China.066004 (e-mail: maqian@ysu.edu.cn)

H.F. Liu, was with the College of Information and Science Technology, Yanshan University, Qinhuangdao, Hebei, China.066004 (e-mail: liuhanfeng521@ysu.edu.cn)

1-4244-0705-2/07/$20.00 ©2007 IEEE

problem is known as Pattern Clustering. Clusters which is captured by Pattern Clustering approaches called pClusters.

1.2 Applications

Discovery of clusters in data sets based on pattern similarity is of great importance because of its potential for actionable insights.

- DNA micro-array analysis: Micro-array is one of the latest breakthroughs in experimental molecular biology. It provides a powerful tool by which the expression patterns of thousands of genes can be monitored simultaneously and is already producing huge amount of valuable data. Analysis of such data is becoming one of the major bottlenecks in the utilization of the technology. The gene expression data are organized as matrices tables where rows represent genes, columns represent various samples such as tissues or experimental conditions, and numbers in each cell characterize the expression level of the particular gene in the particular sample. Investigations show that more often than not, several genes contribute to a disease, which motivates researchers to identify a subset of genes whose expression levels rise and fall coherently under a subset of conditions, that is, they exhibit fluctuation of a similar shape when conditions change. Discovery of such clusters of genes is essential in revealing the significant connections in gene regulatory networks [10,11].
- E-commerce: Recommendation systems and target marketing are important applications in the E-commerce area. In these applications, sets of customers/clients with similar behavior need to be identified so that we can predict customers' interest and make proper recommendations. Let's consider the following example. Three viewers rate four movies of a particular type (action, romance, etc.) as (1, 2, 3, 6), (2, 3, 4, 7), and (4, 5, 6, 9), where 1 is the lowest and 10 is the highest score. Although the rates given by each individual are not close, these three viewers have coherent opinions on the four movies. In the future, if the 1st viewer and the 3rd viewer rate a new movie of that category as 7 and 9 respectively, then we have certain confidence that the 2nd viewer wi ll probably like the movie too, since they have similar tastes in that type of movies.

The above applications focus on finding cluster of objects that have coherent behaviors rather than objects that are physically close to each other. Algorithms such as the pClustering [6] have been proposed to find clusters of objects that manifest coherent patterns. Most process start with comparing each pair of attributes and objects to find MDS (Maximum Dimension Sets) which satisfy the user-specified δ threshold. As the Candidate Clusters, MDSs should be pruned and then merged to acquire pClusters.

Unfortunately, because of their requirement of comparing each pair of attributes and objects in data set, the process usually cost too much. So they can only handle datasets containing no more than thousands of records.

1.3 Our contributions:

We present a new clustering model with sp-Score as a new measure function of the subspace pattern similarity. It's a development of the function of pScore , which is more appropriate to capture coherent patterns after several calculations.

We propose a new algorithm called 0-SM (Zero-Sub-Matrix）to find candidate clusters immediately by computing of matrix and mining submatrix. Unlike many pattern-based clustering algorithms that find clusters in comparing each pair of attributes and objects.

The candidate clusters which captured by 0-SM algorithm are close to the final result rather than those captured by traditional MDS algorithms, a simple detection of overlapping clusters is needed only. We propose a algorithm of selecting candidate clusters, so that pClusters should finally be found.

The rest of the paper is organized as follows. We review related work in Section 2. Section 3 introduce a novel distance function for measuring subspace pattern similarity. An efficient clustering algorithm is proposed based on pattern-submatrix called 0-SM in Section 4. We conclude in Section 5.

2. RELATION WORK

The study of clustering based on pattern similarity is related to previous work on subspace clustering. Many recent studies focus on mining subspace clusters embedded in high-dimensional spaces. Still, strong correlations may exist among a set of objects even if they are far apart from each other as measured by distance functions (such as Euclidean) used frequently in traditional clustering algorithms. It is essential to identify clusters of objects that manifest coherent patterns. A variety of applications, including DNA microarray analysis, E-commerce collaborative filtering, will benefit from fast algorithms that can capture such patterns.

Cheng et al [8] proposed the bicluster model, which captures the coherence of genes and conditions in a submatrix of a DNA micro-array. Based on mean squared residue score as a measure function, a random algorithm is designed to find such clusters.

Yang et al [13] proposed a move-based algorithm to find biclusters more efficiently. It starts from a random set of seeds (initial clusters) and iteratively improves the clustering quality.

Wang et al [6] proposed the δ-pCluster model which can be seen a classical model of pattern-based clustering. They introduced a concept of pScore as a measure of the coherence of the objects and attributes in a sub space of high dimension data set. They find MDSs by a pair wise algorithm, then prune object-pair MDSs and column-pair MDSs by turns to find all the candidate clusters. In a recent study [15], Pei et al. developed MaPle, an efficient algorithm to mine the complete set of maximal pClusters.

3. SPSCORE MODEL：

δ-pCluster model [6] introduced a concept of pScore as a measure of the coherence of the objects and attributes in a sub space of high dimension data set.

Let O be a subset of objects in the database ($O \subseteq D$), and let T be a subset of attributes ($T \subseteq A$). Pair(O, T) specifies a submatrix. Pair (O, T) forms a δ-pCluster if for any given x, y ∈ O, and a, b ∈ T, this 2 × 2 submatrix X in (O, T), we have :

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |\,(d_{xa}-d_{xb})-(d_{ya}-d_{yb})\,| \le \delta$$

for some δ ≥ 0.

From the conception of pScore, we can see obviously that pScore has the following property:

Property 1: reflexivity.

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{xa} & d_{xb} \end{bmatrix}\right) = |\,(d_{xa}-d_{xb})-(d_{xa}-d_{xb})\,| = 0 \le \delta$$

Property 2 : symmetry.

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right)$$

$$= |\,(d_{xa}-d_{xb})-(d_{ya}-d_{yb})\,|$$

$$= |\,(d_{xa}-d_{ya})-(d_{xb}-d_{yb})\,|$$

$$= pScore\left(\begin{bmatrix} d_{xa} & d_{ya} \\ d_{xb} & d_{yb} \end{bmatrix}\right) \le \delta$$

Property 3: reversibility:

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = pScore\left(\begin{bmatrix} d_{ya} & d_{yb} \\ d_{xa} & d_{xb} \end{bmatrix}\right) \le \delta$$

From those characters above, we can infer that:

$$pScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |\,(d_{xa}-d_{xb})-(d_{ya}-d_{yb})\,|$$

$$= pScore\left(\begin{bmatrix} d_{xa}-k & d_{xb}-k \\ d_{ya}-t & d_{yb}-t \end{bmatrix}\right)$$

$$= |\,((d_{xa}-k)-(d_{xb}-k))-((d_{ya}-t)-(d_{yb}-t))\,|$$

$$= pScore\left(\begin{bmatrix} d_{xa}-s & d_{xb}-w \\ d_{ya}-s & d_{yb}-w \end{bmatrix}\right)$$

$$= |\,((d_{xa}-s)-(d_{xb}-w))-((d_{ya}-s)-(d_{yb}-w))\,|$$

$$\le \delta$$

Where k,t,s,w can be any real numbers.

Definition 1: spScore

Let O be a subset of objects in the database ($O \subseteq D$), and let T be a subset of attributes ($T \subseteq A$). Pair(O, T) specifies a submatrix. Given x, y ∈ O, and a, b ∈ T , define the spScore of the 2 × 2 matrix as:

$$spScore\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = |\,((d_{xa}-k)-(d_{xb}-k))-((d_{ya}-t)-(d_{yb}-t))\,|$$

Where k, t are real numbers.

To tell whether two objects exhibit a shifting pattern in a given subspace T, a simple way is to mormalize the two objects by subtracting d_k from each of their coordinate value d_i,($i \in T$), where $k \in T$ is a arbitrary dimension. As Figure 2 shows below:

Lemma1:

Pair (O, T) forms a pCluster if $\exists x \in O$, $\exists k \in T$ for $\forall y \in O$, $\forall k \in T$, have:

$$spScore\left(\begin{bmatrix} d_{xk} & d_{xt} \\ d_{yk} & d_{yt} \end{bmatrix}\right) \le \delta \text{ for some } \delta \ge 0.$$

Clearly, with a different choice of dimension k or object x, we may find the distance different. However, such difference is bounded by a factor of 2.

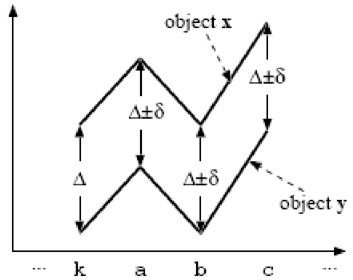


Fig. 2. a Shifting Pattern

Lemma 2:

For any two objects x, y, and a subspace T, if $\exists k \in T$, for $\forall a \in T$ such that spScore((x,y),(k,a)) ≤ δ, then $\forall b \in T$, spScore((x,y),(a,b)) ≤2δ.

Proof:

$$spCluster\left(\begin{bmatrix} d_{xa} & d_{xk} \\ d_{ya} & d_{yk} \end{bmatrix}\right) = \left|(d_{xa}-d_{xk})-(d_{ya}-d_{yk})\right| \le \delta$$

$$spCluster\left(\begin{bmatrix} d_{xb} & d_{xk} \\ d_{yb} & d_{yk} \end{bmatrix}\right) = \left|(d_{xb}-d_{xk})-(d_{yb}-d_{yk})\right| \le \delta$$

$$spCluster\left(\begin{bmatrix} d_{xa} & d_{xb} \\ d_{ya} & d_{yb} \end{bmatrix}\right) = \left|(d_{xa}-d_{xb})-(d_{ya}-d_{yb})\right| =$$

$$\left|(d_{xa}-d_{xk})-(d_{ya}-d_{yk})+(d_{xb}-d_{xk})-(d_{yb}-d_{yk})\right| \le$$

$$\left|(d_{xa}-d_{xk})-(d_{ya}-d_{yk})\right|+\left|(d_{xb}-d_{xk})-(d_{yb}-d_{yk})\right| \le 2\delta$$

Obviously, spScore also have the properties of reflexivity, symmetry and reversibility. Hence The spScore implement as a measure of pattern similarity as well as pScore. Furthermore, it's capable of capturing pattern similarities by spScore even after several operate of the original data.

4. 0-SM ALGORITHM

4.1 Pattern-submatrix

Most high dimension subspace data can form matrix. The data which a set of objects（O_1,O_2,…,O_m）expressed in a set of attributes (A1,A2,...,An) can form a m×n matrix D, the row of which represents objects and the column which represents attributes. This matrix is called expression matrix. As Figure 3 shows:

Hence, the task of clustering by pattern similarity can be converted into a problem that mining submatrixes which exhibit pattern similarity from a expression matrix.

Table 1：4 objects in 4 attributes

	A1	A2	A3	A4
O1	6	7	9	5
O2	8	9	11	7
O3	3	4	3	2
O4	5	5	4	4

Fig.3. expression matrix of table 1

$$D=\begin{pmatrix}6&7&9&5\\8&9&11&7\\3&4&3&2\\5&5&4&4\end{pmatrix}$$

Defination 2: Pattern-submatrix

Let D be a expression matrix, D' is a submatrix of D. D' is a pattern-submatrix if $\exists r \in O$,$\exists k \in T$, for any $\forall w \in O$, $\forall t \in T$, we have:

$$spScore\left(\begin{bmatrix} d_{rk} & d_{rt}\\ d_{wk} & d_{wt}\end{bmatrix}\right)\le\delta$$

Lemma 3:

A submatrix D'(O,T) is a pattern-submatrix if and only if $\exists r \in O, \exists k \in T$,subtract row r from every the rows of D' , and substract column k from every columns of D', every element of D' is approximate to 0 (in the range of (-δ, δ), δ is a user-specified clustering threshold).

Proof :

Let D' (O,T) be a pattern-matrix. Given r∈O, k∈T, for each2×2 submatrix{(r,w),(k,t)} of D', (w∈O，t∈T), have:

$$R\begin{pmatrix} d_{rk} & d_{rt}\\ d_{wk} & d_{wt}\end{pmatrix}\Rightarrow\begin{pmatrix} d_{rk}-d_{rk} & d_{rt}-d_{rt}\\ d_{wk}-d_{rk} & d_{wt}-d_{rt}\end{pmatrix}$$

$$\Rightarrow\begin{pmatrix}(d_{rk}-d_{rk})-(d_{rk}-d_{rk}) & (d_{rt}-d_{rt})-(d_{rk}-d_{rk})\\ (d_{wk}-d_{rk})-(d_{wk}-d_{rk}) & (d_{wt}-d_{rt})-(d_{wk}-d_{rk})\end{pmatrix}$$

$$\Rightarrow\begin{pmatrix}0 & 0\\ 0 & \pm spScore(R)\end{pmatrix}\Rightarrow R'\begin{pmatrix}0 & 0\\ 0 & (-\delta,\delta)\end{pmatrix}$$

Defination 3: Pattern-similar matrix

Let D be a expression matrix. Matrix S is a transform of subtracting values of some rows (or columns) from all the rows (or columns) of D. S can be seen as a Pattern-similar matrix of D.

Lemma 4:

Let D and S be both expression matrix. Tell S is a pattern-similar matrix of D, if and only if D and S have the same pattern-submatrix.

Proof:

According to the concept of spScore, if D' is a pattern-submatrix of expression matrix D, it can still follow the rule in Lemma 1 after several subtracting of rows and columns. So we can see obviously that pattern-similar matrixes S and D have the same pattern-submatrix.

4.2 0-SM

Definition 4: Zero-submatrix

Let D be a expression matrix, D' be the submatrix of D. We define D' be a **zero-submatrix** of D if all elements of D' are approximate to 0 (in the range of (-δ, δ)). Usually, we call it "0-SM" for short.

According to the concept of 0-SM, if D' is a 0-SM of D, and there must exist elements which are not approximate to 0 in any D''⊃D', then D' is a maximal 0-SM.

0-SM is actually a special pattern-submatrix. The problem of how to mining all the pattern-submatrix of expression matrix D can be covert to a process of finding all the maximal 0-SM of all the pattern-similar matrix of D.

Hence, in order to find pattern-submatrix of expression matrix D, we should do operations as below:

1) First, select a row i of D, subtract each coordinate value of i from every row of D;
2) Select a column j of D, subtract each coordinate value of j from every column of D;
3) Mining all the maximal 0-SM of D.
4) Select other rows and columns to continue subtracting until all the 0-SM have been found.

Example:

D is the expression matrix of Table 1.

$$D=\begin{pmatrix}6&7&9&5\\8&9&11&7\\5&4&3&2\\5&5&7&4\end{pmatrix}\text{ subtract the first row}$$

$$\Rightarrow\begin{pmatrix}0&0&0&0\\2&2&2&2\\-3&-3&-6&-3\\-1&-2&-2&-1\end{pmatrix}\text{ subtract the first column}$$

$$\Rightarrow\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&3&0\\0&1&1&0\end{pmatrix}$$

We can see that submatrix $\{(o_1,o_2),(a_1,a_2)\}$ is a 0-SM, $\{(o_1,o_2),$ (a1,a2,a3,a4)} and $\{(o_1,o_2,o_3),(a_1,a_2)\}$ are two maximal 0-SMs. They are both pattern-submatrix of D.

This process can be summarized as algorithm 1:

Input : ***D:*** expression matrix with m rows and n columns; ***δ:***cluster threshold; ***X:***a null 0-1 matrix with m rows and n columns; ***BR:***the record list; ***nc:***User-specified minimum of columns; ***nr:***User-specified minimum of rows;

Output: all zero-submatrix of D

i←0;

```
while i<m do
 {for each row of D do
   { Subtract the $i_{th}$ row from each row of D;
   j←0;
   while j<n do
        {for each column of D do
             {Subtract the $j_{th}$ column from each column of D;
              if -δ≤D[k][t]≤δ then
               X[k][t]←0;
              else
               X[k][t]←1;
              end}
            SM(X, δ,nr,nc,BR) ; //mining 0-SM
            Output BR;
         j++;}}
   i++;}
```

4.3 Search for 0-SM

How to find 0-SM efficiently is a key problem. Operations on bit strings can be used to detect 0-SM,for it is known that the speed of doing bit operation is very fast.

We should transform expression matrix D to 0-1 matrix by letting 0 be the description of elements whose value is in the range of （-δ, δ）,and 1 be the description of other elements.

$$D\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \to X \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Thus this 0-1 matrix can be seen as a set of bit strings. Every rows of X forms a bit string, they are 1110 0000, 1101 0000, 1011 0010 and 0111 0110. The higher 4 bit describes the rows included in the pattern, and the lower 4 bit describes the columns included.. We can find relative 0-SM from searching for the bits expresses 0 simultaneously, which can be found by doing "bitwise AND"(∧) to the higher bits and "bitwise OR" (∨) to the lower bits.

Example:

For the 1st row and the 3rd row in matrix X, we do "bitwise AND" to the higher 4 bits and "bitwise OR" to the lower 4 bits. We have:

$$\begin{array}{cccc} & 1110 & & 0000 \\ \wedge & 1011 & \vee & 0010 \\ \hline = & 1010 & & 0010 \end{array}$$

The bits express 0 means the relative rows and columns is included in a pattern. Thus submatrix {(0,2),(0,1,3)} is a 0-SM of D.

Our task is to find all the maximal 0-SM. So we proposed a bit string record list structure to do bit string operations. Those bit strings, which has more than nr (user-specialized minimal number of rows) bits express 0 in the higher m bits and nc(user-specialized minimal number of columns) bits express 0 in the lower n bits, can be inserted to the record list. Do "OR" operations for each pair of bit strings in record list, if the result bit string's rows count ≥ nr and columns count ≥ nc, insert it to record list as a new record. If the result is the same to one of its original operators, delete the original record from the record list. Thus replace old patterns by larger patterns continually until the all the records have been processed.

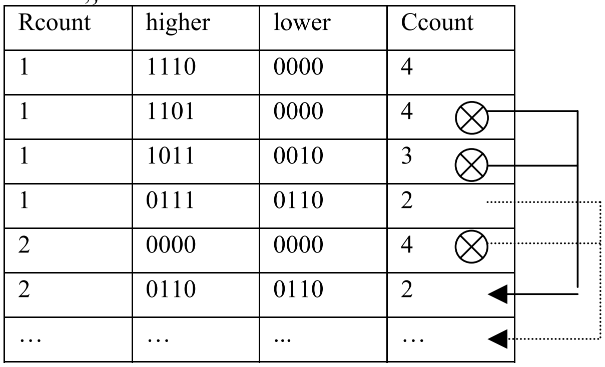
Example:

Partial operations of a record list as Figure 4 shows.

The process of finding 0-SM can be summarized as algorithm 2.

SM(X, δ, nr, nc,BR)

```
{ BR←null;
  for each byte string $S_i$ do
       {if $S_i$.Rcount≥nr&&$S_i$.Ccount≥nc then
         insert $S_i$ to BR;}
  i←0;
  while i<|BR| do
    { j←i+1;
      while j<|BR| do
         { P.lower =BR[i].lower∨BR[j].lower;
          if P.Ccount ≥ nc then
            { P.higher =BR[i].higher∧BR[j].higher;
              insert P to BR;
              if P.higher==BR[j].higher &&
                       P.lower==BR[j].lower then
                  delete BR[j] form BR;
              if P.higher==BR[i].higher &&
                  P.lower==BR[i].lower  then
                  {delete BR[i] form BR;
                   break;}}
         j++;}
     i++;}
  i←0;
  while i<|BR| do
   {if BR[i].Rcount<nr then
         delete BR[i] form BR;
     i++;}
  return BR;}
```



Rcount	higher	lower	Ccount
1	1110	0000	4
1	1101	0000	4 ⊗
1	1011	0010	3 ⊗
1	0111	0110	2
2	0000	0000	4 ⊗
2	0110	0110	2
…	…	...	…

Fig. 4. example of bit string record list

4.4 Merge of candidate clusters

The clusters which are mined by 0-SM algorithm may have problems of overlapping. So it's important to merge the overlapping records before output. The merge process performs as below:

Propose a pCluster record list structure, insert each record which is found in the 0-SM mining process. The insert principle is:

Insert pattern p to the record list if there isn't the same patterns and supersets of p in the record list,.

If there exist p' which is the subsets of pattern p in the record list, we should delete p' from the list.

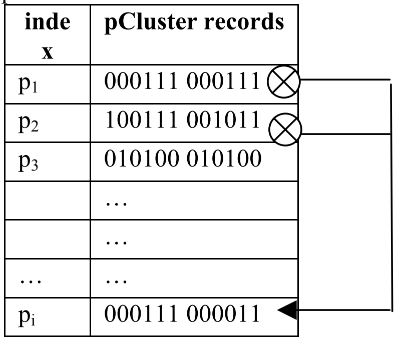
Then output all the patterns in the pCluster record list, which is all the pClusters been found.

Example:

Figure 5 shows a example of pCluster record list.

When p_i(000111 000011) comes, since there are two subsets $p_1, p_2 \subseteq p_i$, and none supersets of it, we should delete p_1 and p_2 from the record list and insert pi as a new record to the tail of the list.

When p_j(010101 000111) comes, since there is one superset $p_3 \supseteq p_j$, pi can not be inserted to.



inde x	pCluster records
p_1	000111 000111 ⊗
p_2	100111 001011 ⊗
p_3	010100 010100
	…
	…
…	…
p_i	000111 000011 ◀

Fig.5. a pCluster record list

5.Conclusion

Clustering by pattern similarity is an interesting and challenging problem. The computational complexity problem of subspace clustering is further aggravated by the fact that we are concerned with patterns of rise and fall instead of value similarity. Unfortunately, most of previous pattern-based Clustering approaches have to find pattern candidates such as MDS by comparing each pair of objects and attributes, which makes them limited in efficiency advancement of algorithm. In this paper we introduced a model of spScore, mining pattern candidates by finding pattern submatrix from repetitious computing in expression matrix. We proposed 0-SM algorithm to mining patterns, since byte string computing which is used to detect patterns is easy to process, the efficiency can be advanced largely.

The algorithm based on the pScore has time complexity $O(M^2N\log M+N^2M\log N)$ where M is the number of columns and N is the number of objects. The worst case for pruning is $O(kM^2N^2)$ .The algorithm proposed in this paper has time complexity $O(MN+M(M+N)+M^2N+N^2M)$.

How to prune insignificant patterns in clustering process to reduce workload and advance efficiency more can be a further study.

1、 Reference

[1] C. C. Aggarwal, C. Procopiuc, J.Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In SIGMOD, 1999.

[2] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In SIGMOD, pages 70-81, 2000.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Authomatic subspace clustering of high dimensional data for data mining applications. In SIGMOD, 1998.

[4] C. H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In SIGKDD, pages 84-93, 1999.

[5] H. V. Jagadish, Jason Madar, and Raymond Ng. Semantic compression and pattern extraction with fascicles. In VLDB, pages 186-196, 1999.

[6] HaixunWang,WeiWang, Jiong Yang, and Philip S. Yu. Clustering by pattern similarity in large data sets. In SIGMOD, 2002.

[7] Jiong Yang, Wei Wang, Haixun Wang, and Philip S. Yu. δ-clusters: Capturing subspace correlation in a large data set. In ICDE, pages 517-528, 2002.

[8] Jian Pei, Xiaoling Zhang, Moonjung Cho, Haixun Wang, and Philip S. Yu. Maple: A fast algorithm for maximal pattern-based clustering. In ICDM, 2003.

[9] Haixun Wang, Fang Chu, Wei Fan, Philip S. Yu, Jian Pei, A Fast Algorithm for Subspace Clustering by Pattern Similarity In *SSDBM,* 2004.

[10] P. O. Brown and D. Botstein. Exploring the new world of the genome with DNA microarrays. Nature Genetics, 21:33-37, 1999.

[11] P. D'haeseleer, S. Liang, and R. Somogyi. Gene expression analysis and genetic network modeling. In Pacific Symposium on Biocomputing, 1999.

[12] Y. Cheng and G. Church. Biclustering of expression data. In Proc. of 8th International Conference on Intelligent System for Molecular Biology, 2000.

[13] J. Yang, W. Wang, H. Wang, and P. S. Yu. δ-clusters: Capturing subspace correlation in a large data set. In ICDE, pages 517–528, 2002.

[14] Pei, J., Zhang, X., Cho, M., et al. MaPle: A Fast Algorithm for Maximal Pattern-based Clustering. ICDM'03.

[15] Daxin Jiang, Jian Pei, Aidong Zhang, A General Approach to Mining Quality Pattern-Based Clusters from Microarray Data. In DASFAA, pages 188-200, 2005.