International Journal of Hybrid Information Technology
Vol. 10, No.1 (2017), pp. 255-268
http://dx.doi.org/10.14257/ijhit.2017.10.1.22

Efficient Mining Maximal Trend Biclusters in Resource Effectiveness Matrix

Miao Wang[1, 2], Lihua Zhang[1, 2*] and Zhiyong Xiong[1, 2]

[1]*Science and Technology on Avionics Integration Laboratory, Shanghai, China*
[2]*China National Aeronautical Radio Electronics Research Institute, Shanghai, China*
wang_miao@careri.com, zhang_lihua@careri.com, xiong_zhiyong@careri.com

Abstract

The effectiveness of resources is the base for analyzing system safety or prognostics and health management system. This paper proposed two efficient bicluster mining algorithms: CoCluster algorithm and CeCluster algorithm, which mine trend bicluster respectively in discrete and real-valued resource effectiveness matrices. First, both algorithms construct a sample weighted graph; second, they mine maximal trend bicluster using sample-growth method in above constructed graph. In order to improve the mining efficiency, multiple pruning strategies are adopted for mining trend biclusters without candidate maintenance. Meanwhile, CoCluster algorithm and CeCluster algorithm can not only mine resource patterns with effectiveness in the downtrend, but also mine those with effectiveness in the uptrend. To improve the scalability, both algorithms can also mine resource patterns without change of effectiveness. The experimental results show our algorithms are more efficient than traditional algorithm. And the evaluated results of mean square error value show our algorithms can produce statistically significant biclusters.

Keywords*: bicluster; trend; candidate maintenance; resource*

1. Introduction

The level of efficiency of resources directly influences the effectiveness of the whole system. However, resource fault might cause some deficiencies of the system. Therefore, studying on the level of effectiveness of resources is the base for analyzing system safety or constructing prognostics and health management system [1]. It can be found which resources have a lower effective rate through -time recording of the effectiveness of all resources in the system. Frequent pattern mining method and association rules mining algorithm for resource effectiveness matrix in a period of time, can be efficient to discover incorrect resources in advance and thus start using spare resources earlier. However, some resources are normal in a period of time, *i.e.* satisfy the threshold value of support or confidence, but might present a downtrend of effectiveness in a certain period. For example, when the system implements a certain function in a short period, the implementation of this function will make some resources present an unhealthy state. Earlier discovery of such fault helps to conduct health management over potential faults and thus reduce the risk of poor health of the system.

The feature of resource pattern with the effectiveness presenting some trend described above, meets the feature of bicluster mining in data mining technology. Bicluster was first proposed by Cheng and Church [2] and used to find co-expression gene under specific experimental conditions in gene expression data. This algorithm uses a low square root residue to gradually delete redundant nodes. Many algorithms based on greedy strategy were proposed afterwards [3-9]. Various algorithms above use the following two mining

ISSN: 1738-9968 IJHIT

strategies: firstly, produce clustering globally according to the traditional clustering method and then optimize it gradually; secondly, mine biclusters respectively in two categories of data and then obtain the result through comparison and integration. However, neither strategy produces a high efficiency of algorithm. First, bicluster is a NP-hard problem [10]; second, while processing original data, bicluster needs to solve the problem of sensitivity of original data to noise. Meanwhile, bicluster algorithm should allow the overlap among clusterings, which increases the computation complexity of bicluster algorithm; finally, as bicluster algorithm directly processes original data, it should have a very strong flexibility for different types of bicluster.

In order to improve the mining efficiency of bicluster algorithm, Wang *et al*. [11, 12] uses sample-growth method to mine maximal bicluster in discrete data. However, the algorithm above can only mine biclusters for gene co-expression relation but cannot be used to mine trend bicluster. Based on the analysis above, this paper proposes two efficient bicluster mining algorithms: *CoCluster* algorithm and *CeCluster* algorithm, to mine trend bicluster respectively in discrete and real-valued resource effectiveness matrices. Firstly, both algorithms construct a sample weighted graph; secondly, they mine maximal trend bicluster using sample-growth method in above constructed graph. In order to improve the mining efficiency, multiple pruning strategies are adopted for mining trend biclusters without candidate maintenance. Meanwhile, *CoCluster* algorithm and *CeCluster* algorithm can not only mine resource patterns with effectiveness in the downtrend, but also mine those with effectiveness in the uptrend. To improve the scalability, both algorithms can also mine resource patterns without change of effectiveness. In a word, both algorithms can mine multiple biclusters: (1) basic trend bicluster with uptrend or downtrend considered; (2) traditional constant row bicluster, *i.e.* without variation trend; (3) bicluster with ratio relationship among columns; (4) bicluster with certain difference relationship among columns.

2. Problem Description of CoCluster Algorithm

Resource effectiveness matrix is defined as a two-dimensional real-valued matrix $D = R \times S$, where row collection *R* represents the resource name; column collection *S* refers to different sampling sites. Element D_{ij} of matrix *D* is a real-valued number which refers to the effective value (e.g. BIT value) of resource *i* under sampling site *j*. |*R*| is the number of resources in data set *D* and |*S*| is the number of sampling sites in data set *D*. For the convenience of mining, the original effective value in resource effectiveness matrix can be discretized into *1, 2, 3, n* values, where *1* represents the lowest health degree of resources and *n* represents very healthy resources. The number of discrete values is *4* in discrete resource effectiveness matrix shown in Table 1. Bicluster *B* means that resource in *R* satisfies the trend definition in the sampling site in *S*. Assuming that *M* is the set of all biclusters in *D*. Given a bicluster $N = K \times L (N \in M)$, if there does not exist another bicluster $P = S \times T (P \in M)$, in which $K \subseteq S$ and $L \subseteq T$, *N* is called as the maximal bicluster in *M*. A bicluster *B* can be defined as *Samples(Resources)*, where Resources refer to the set of resources in *B*. It can also be denoted as *B.Resources*; *Samples* refer to the set of sampling sites where these resources satisfy the trend definition, which can also be denoted as *B.Samples.*

The concept of mining *CoCluster* algorithm is the same variation trend of all resources within some consecutive sampling site, *i.e.* trend bicluster. The variation trend of resources that this paper pays attention to has three types: rising, decline and invariance. *CoCluster* algorithm does not distinguish the degree of rising or decline. That is to say, variations of discrete value from 1 to 2 and from 1 to 3 are not distinguished. It is only necessary that both resources present a rising or decline state.

Table 1. Discrete Resource Snapshot Matrix

	S_1	S_2	S_3	S_4	S_5	S_6	S_7
R_1	1	2	3	4	2	1	3
R_2	2	3	4	3	2	1	2
R_3	4	3	2	1	3	1	2
R_4	4	4	4	4	4	4	4
R_5	1	1	3	4	3	3	3
R_6	1	1	3	4	3	3	3
R_7	4	4	2	1	4	4	4

Definition 1. The relationship of two resources R_1 and R_2 in two consecutive samples S_1 and S_2 can be defined as follows:

1) If R_1 and R_2 both present uptrend or downtrend in S_1 and S_2, R_1 and R_2 are positively correlated and denoted as R_1R_2;

2) For R_1 and R_2, if one of them presents uptrend and the other presents downtrend in S_1 and S_2, R_1 and $R_2$are negatively correlated and denoted as R_1-R_2;

3) If R_1 and R_2 both present invariance trend in S_1 and S_2, R_1 and R_2 are uniformly correlated and denoted as R_1*R_2;

4) For R_1 and R_2, if one of them presents uptrend or downtrend and the other presents invariance trend in S_1 and S_2, R_1 and R_2 are not correlated;

According to four relations defined above, there are two types of resource pattern in bicluster mined with *CoCluster* algorithm: first, positive correlation or negative correlation among resources; second, uniform correlation among resource. In conclusion, the mining with *CoCluster* algorithm aims at mining all maximal biclusters meeting conditions in definition 1 from discrete resource effectiveness matrix. In order to improve the mining efficiency of the algorithm, *CoCluster* algorithm will use sample-growth and multiple pruning strategies for mining maximal biclusters without candidate maintenance. The detailed mining process will be introduced in the next section.

3. CoCluster Algorithm

As there are two types of resource pattern in bicluster mined by *CoCluster* algorithm: positive or negative correlation among resources and uniform correlation among resources. If the method of resource extension is used for mining, it is necessary to first calculate the sample set of each resource meeting the requirement of trend definition under all samples and then calculate the intersection of the same sample sets for resource extension, which will increase the complexity of the algorithm. Meanwhile, the larger number of resources will also influence the mining efficiency of the algorithm. Moreover, excessive resource trend sample information produced in step 1 will also cause low mining efficiency during the calculation of intersection for resource extension. Therefore, *CoCluster* algorithm mines trend bicluster from resource effectiveness matrix with the method of sample-growth. In addition, the method of sample-growth can also be specific to certain function or subsystem implementation period, *i.e.* mine resource collections with the same trend within a certain consecutive sample interval for the convenience of analysis on resource effectiveness by decision supporting system. The mining process of *CoCluster* algorithm can be divided into two steps: first, construct a sample weighted graph; second, mine all maximal trend biclusters with the method of sample-growth.

Different from the mining of bicluster with the traditional sample-growth method, the trend in bicluster is produced under consecutive time sample. Therefore, it is only necessary to build the weight on S_i and S_{i+1} edges when constructing the sample relational weighted graph. According to the analysis above, at most two groups of resource information exist simultaneously on the weight of each edge. In one of the group, resources have positive or negative correlation. In the other group, resources present invariance trend, *i.e.* uniform correlation. The sample weighted graph corresponding to

Table 1 is shown in Figure 1 where '0' refers to resource set with positive or negative correlation among resources and '1' refers to resource set with invariance trend among resources.

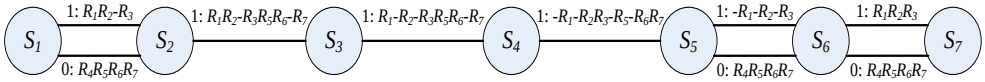

Figure 1. Sample Weighted Graph Corresponding To Table 1

Then, *CoCluster* algorithm uses sample-growth method to mine maximal trend bicluster in the constructed sample weighted graph. As resources in bicluster mined with this algorithm has trend consistency, all resources in trend bicluster have a consistent relationship in all adjacent samples. The variation trend of resources in bicluster is the same in all adjacent samples. Trend bicluster mined with *CoCluster* algorithm has the following three extension modes: (1) If it is positive correlation in initial two adjacent samples, it should be positive correlation or uniform correlation in all subsequent adjacent samples; (2) if it is negative correlation in initial two adjacent samples, it should be negative correlation or uniform correlation in all subsequent adjacent samples; (3) if it is uniform correlation in initial two adjacent samples, it should be uniform correlation or non-uniform correlation (positive or negative correlation) existed between two adjacent samples at first in all subsequent adjacent samples.

Table 2. An Example Matrix 1

	S_1	S_2	S_3	S_4
R_1	1	2	3	4
R_2	1	2	3	4
R_3	4	3	2	1

Table 3. An Example Matrix 2

	S_1	S_2	S_3	S_4
R_1	1	2	3	3
R_2	1	2	3	3
R_3	4	3	2	2

Table 4. An Example Matrix 3

	S_1	S_2	S_3	S_4
R_1	1	1	3	4
R_2	1	1	3	4
R_3	4	4	2	1

Table 5. An Example Matrix 4

	S_1	S_2	S_3	S_4
R_1	3	3	3	3
R_2	3	3	3	3
R_3	4	4	4	4

Next, it will use an example for illustrating above. Given a bicluster $S_1S_2S_3S_4$(R_1R_2-R_3), for R_1R_2-R_3, if R_1 and R_2 are positively correlated, R_2 and R_3 are negatively correlated and R_1 and R_3 are negatively correlated in S_1 and S_2, R_1, R_2 and R_3 should also

satisfy the relevancy above in S_3, S_3 and S_4, as shown in Table 2, or R_1, R_2 and R_3 satisfy uniform correlation in a certain group of adjacent samples, as shown in Table 3. At this time, all resources in S_3 and S_4 present invariance trend, *i.e.* uniform correlation. Table 4 shows the situation in (3) above and resources have uniform correlation in S_1 and S_2. When there is R_1R_2-R_3 correlation in S_2 and S_3, there should be the same in S_3 and S_4. Bicluster shown in Table 5 is a form of mining result, *i.e.* it is invariance trend, *i.e.* uniform correlation in all adjacent samples.

It can be seen from the analysis above that multiple groups of resource collection satisfying the definition of resource trend will exist in a group of same sample set. Therefore, multiple groups of resource collection meeting the definition will be produced in real time during sample-growth. Thus, multiple groups of trend bicluster can be mined with the method of sample-growth for improving the mining efficiency of the algorithm. To improve the mining efficiency, *CoCluster* algorithm mines maximal bicluster without candidate maintenance. Pruning strategies used by this algorithm are designed based on the method of prior candidate sample detection, *i.e.* if the weight of the current candidate sample is the subset of a prior candidate sample weight, trend bicluster obtained by the extension of weight of the current candidate sample can be obtained by the extension of the weight of a prior candidate sample. Therefore, the weight of the current candidate sample can be pruned. Based on the analysis above, Lemma 1 can ensure that *CoCluster* algorithm can prune candidate sample without candidate maintenance.

Lemma 1. Assuming that P is the current bicluster to be extend, M is the candidate sample set of P and N is the prior candidate sample set of P, if a prior candidate sample $N_j(N_j \in N)$ making $PM_i.Resource$ a subset of $PN_j.Resource$ exists for candidate sample $M_i(M_i \in M)$, the bicluster obtained by extension of PM_i is a subset of that obtained by extension of PM_iN_j.

Proof. Proof by contradiction. Assuming that the resource collection of the current candidate sample M_i is not a subset of resource set of a prior candidate sample N_j before it, M_i can be pruned. It can be known from the assumption that a resource set not belonging to PN_j exists in PM_i. As the algorithm uses depth-first extension method for mining and N_j is extended earlier than M_i, there might be another sample making the resource collection of PM_iR_m not equal to that of $PM_iN_jR_m$. Therefore, M_i cannot be pruned, which is contradictory with the assumption. Thus, the original evidence is true.

However, candidate sample might have multiple weights. Only when all weights in the candidate sample are pruned, this candidate sample can be pruned. Based on the lemma above, *CoCluster* algorithm uses the following two pruning strategies for pruning of candidate sample, thus improving the mining efficiency of the algorithm.

Pruning 1. Assuming that P is the current bicluster to be extended, M is candidate sample set of P and N is prior candidate sample set of P, if a prior candidate sample $N_j(N_j \in N)$ making $PM_i.Resource$ a subset of $PN_j.Resource$ exists for candidate sample $M_i(M_i \in M)$, $PM_i.Resource$ should be pruned.

Pruning 2. Assuming that P is the current bicluster to be extended, M is candidate sample set of P and N is prior candidate sample set of P, if a prior candidate sample $N_j(N_j \in N)$ making $PM_i.Resourcem$ a subset of $PN_j.Resource$ exists in all $PM_i.Resourcem(m=1,2,\cdots,n)$ for candidate sample $M_i(M_i \in M)$, PM_i should be pruned.

Those satisfied pruning conditions should be directly pruned and those not satisfied pruning conditions should continue extension. However, whether the current extension mode is output should be judged according to the following output strategy (which actually meets the definition of maximal bicluster):

Output strategy. Assuming that P is the current bicluster to be extended and M is candidate sample set of P, if P has n weights and $P.Resource_m(m=1,2,\cdots,n)$ does not have a candidate sample $M_i(M_i \in M)$ making $P.Resource_m$ a subset of PM_i.Resource(does not meet maximal definition), $P.Resource_m$ can be output.

Based on the analysis above, this algorithm can directly mine maximal trend bicluster with the method of sample-growth without storing the trend bicluster of candidate in internal memory. Figure 2 illustrates the mining process of *CoCluster* algorithm. Example data are shown in table 1 and the threshold of minimum number of samples and resources is 2.

Algorithm: CoCluster algorithm

Input: threshold of number of samples or resources in bicluster: $r_{\min}$, resource effectiveness data: *D*

Output: all maximal trend biclusters meeting the threshold

Initialization: sample weighted graph: *G* =Null, current bicluster to be extended *Q* =Null, S_i=Null and S_j=Null.

Algorithm description: CoCluster($r_{\min}$, *D, Q,* S_i, S_j)

(1) **If** *G* is null, scan data set *D* and make its weight graph. S_i is the first sample in the weighted graph;

(2) **For** each sample S_j linked to sample S_i,

(3) **If** all resource linked lists in S_j satisfies pruning conditions,

(4) **Continue**;

(5) **Else**

(6)**For** resource linked lists not satisfying pruning conditions, *Q.Sample=Q.Sample* $\cup S_j$ *; Q. Resource=Q.Resource*$\cap S_iS_j$*.Resource;*

(7) CoCluster ($r_{\min}$, *D, Q,* S_i, S_j->next);

(8) **Endfor**

(9) **Endif**

(10) **If** *Q* satisfies output conditions and threshold,

(11) **Output** *Q*;

(12) **Endif**;

(13) $S_i = S_i$->next;

(14) **Return**

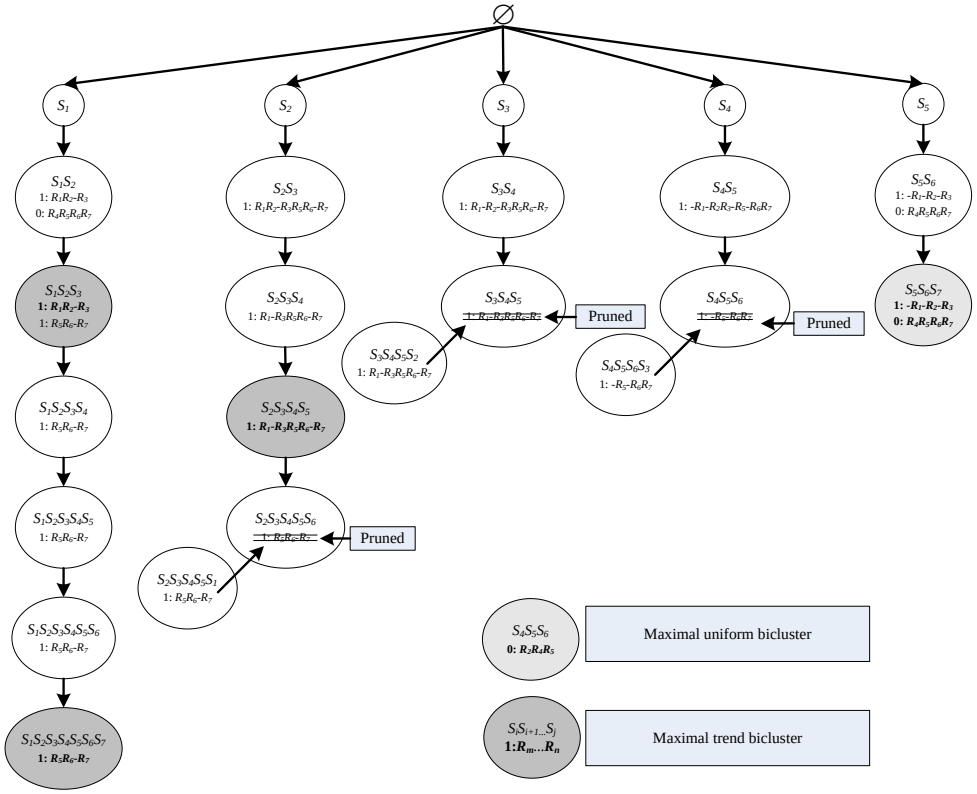

Figure 2. Example Mining Process of *Cocluster* Algorithm

4. CeCluster Algorithm

The mining process of *CeCluster* algorithm is similar to *CoCluster* algorithm, *i.e.* first construct a sample weighted graph that satisfies resource trend definition and then mine all maximal trend biclusters with the method of sample-growth. The same to *CoCluster* algorithm, the variation trend of resources in *CeCluster* algorithm has three types: rising, decline and invariance. As long as the variation of resource effectiveness exceeds certain threshold, it is considered as uptrend or downtrend as long as both resources present uptrend or downtrend. Similarly, if the variation of resource effectiveness is within the constraining range of certain threshold, it is considered as invariance trend. In conclusion, definition 2 provides how to define the trend of resources in real-valued resource effectiveness matrix.

Definition 2. Assuming that values of resource R_1 in two consecutive samples S_1 and S_2 are denoted as V_1 and V_2, the trend of resource R_1 in S_1 and S_2 can be defined as:

1) If $\frac{V_2 - V_1}{\min\{V_2, V_1\}} \geq \alpha$, resource R_1 present uptrend between S_1 and S_2;

2) If $\frac{V_2 - V_1}{\min\{V_2, V_1\}} \leq -\alpha$, resource R_1 present downtrend between S_1 and S_2;

3) If $-\beta \leq \frac{V_2 - V_1}{\min\{V_2, V_1\}} \leq \beta$, resource R_1 present invariance trend between S_1 and S_2;

where α and β are the used defined threshold.

The trend of all resources between some pair of adjacent samples can be obtained from definition 2. Then, the trend relationship between two or among more resources can be obtained according to definition 2. Thus, the relationship between resources in real-valued

resource effectiveness matrix can be obtained. After making clear the relationship of resources between samples, *CeCluster* algorithm constructs a sample weighted graph. Different from the weighted graph constructed by *CoCluster* algorithm, *CeCluster* algorithm constructs the graph in real-valued resource effectiveness matrix according to definitions 2 and with the same method as *CoCluster* algorithm. The sample weighted graph shown in Figure 3 is made according to the real-valued resource effectiveness matrix shown in Table 6.

Table 6. Real-Valued Resource Snapshot Matrix

	S_1	S_2	S_3	S_4	S_5	S_6
R_1	0.73	0.81	0.9	1	0.89	0.77
R_2	0.62	0.71	0.83	0.69	0.68	0.71
R_3	0.2	0.3	0.38	0.45	0.38	0.2
R_4	0.62	0.54	0.3	0.39	0.4	0.41
R_5	0.8	0.9	0.6	0.68	0.69	0.72
R_6	0.98	0.87	0.79	0.68	0.75	0.84

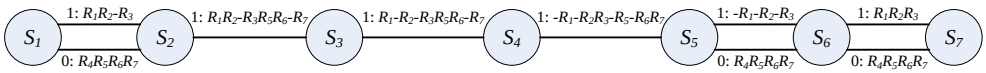

Figure 3. Sample Weighted Graph Corresponding To Table 6

After constructing the weighted graph, *CeCluster* algorithm mines maximal trend *bicluster* using sample-growth method and prior sample detection without candidate maintenance. The extension method and pruning strategies used in *CeCluster* algorithm are the same as described in *CoCluster* algorithm and will not be described here again. Figure 4 illustrates the mining process of *CeCluster* algorithm. Example data are shown in table 6 and the threshold of minimum number of samples and resources is 2.

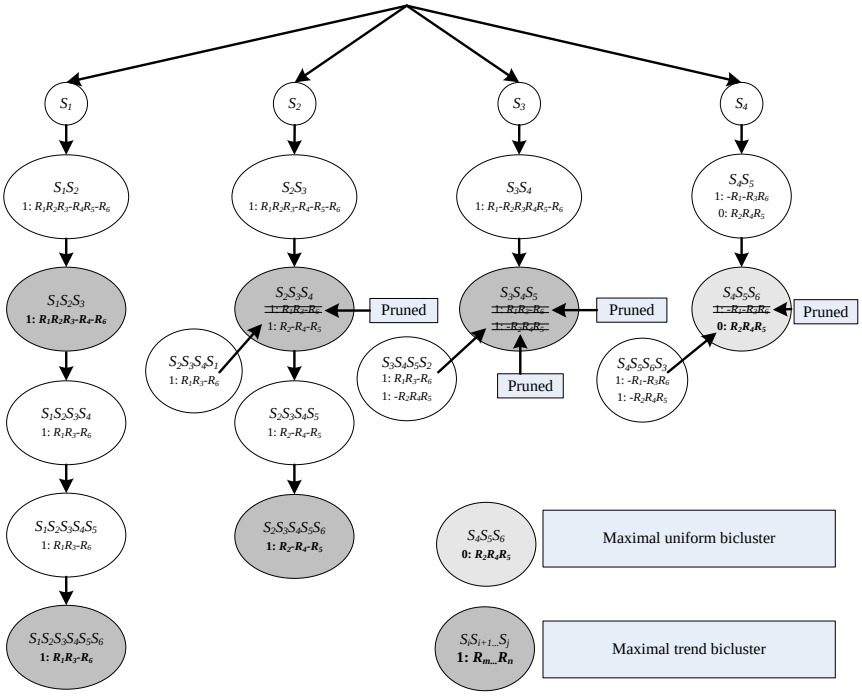

Figure 4. Example Mining Process of *Cecluster* Algorithm

Table 7. The Distribution of Each Sub-Block

1-200	uptrend	random	
201-400	random	downtrend	
401-600	random	uniform trend	random
601-800	random		

5. Experimental Result and Analysis

In this section, we will make an experimental comparison on the mining efficiency and result of the algorithm above and existing algorithms. The hardware environment of the experiment is desktop computer: Intel(R) Core(TM)2 Duo 2.53GHz CPU and 4G internal memory; the software environment is Microsoft Windows 7 SP1 operating system; the algorithm programming and operating environment is Microsoft Visual C++ 6.0 SP6. Experimental data used in this paper are simulation data. The method of data generation in block is used: some region is set as uptrend (or downtrend or uniform trend) and all resources R in this region present uptrend (or downtrend or uniform trend). The data set contains 20 sampling sites and 800 resources. Table 7 describes the distribution of each sub-block in the data set and the proportion of each sub-block is random.

5.1. Experimental Result of Cocluster and Analysis

CoCluster algorithm proposed in this paper is used to mine maximal trend bicluster in discrete resource effectiveness matrix. Therefore, we will discretize data with the method of k-means clustering proposed in D_iB_iCLUS algorithm and classify the real-valued value of each resource in all sampling sites into K. The initial central point in each classification is random. Discretization with k-means has certain disadvantage, *i.e.* the result obtained by each discretization might be different. The reason is that the selection of initial central point might be difference each time. Therefore, to avoid the influence of the selection of central point, we discretize each resource for 10 times and the result with minimum mean square error will be used as the final discretization result.

In this section, a comparison will be made on the efficiency of *CoCluster* algorithm and TCB_i*clustera*lgorithm [13] (TCB for short) and *CoCluster* algorithm without using pruning strategies(denoted as *CoCluster_nonpruning*). Figures 5(a)-5(c) provide the comparison of running time of three algorithms above with values in discretization respectively 5, 10 and 20 and the number of resources respectively 100, 200, 300, 400, 500 and 600. It can be seen from these figures that the mining time of these three algorithms increases progressively with the increase of the number of resources in data set. Meanwhile, the mining efficiency of *CoCluster* algorithm is higher than that of the other two algorithms under each data size. Especially when the number of resources in data source is higher, the mining efficiency of *CoCluster* algorithm is almost 1000 times higher than that of TCB algorithm. The reason is that the pruning strategy used by TCB

algorithm has a lower efficiency. With the increase of the number of resources in data set, this algorithm needs more pruning judgments to mine all biclusters meeting the threshold constraint. However, due to low success rate of pruning, the cost of pruning judgment is too high, thus influencing the mining efficiency of the algorithm. *CoCluster* algorithm uses high-efficiency pruning strategies for mining and will produce more maximal biclusters especially when the number of resources in data set is high and data are dense. Thus, the pruning efficiency of *CoCluster* algorithm will be higher.

Then, we use mean square error (MSE) [2] to measure the difference degree of the model. Mean square error can be used to measure the relevancy of a group of resources in a group of sampling sites. Lower score of mean square error indicates lower difference degree and high relevancy of resources in a group of consecutive sampling sites. Assuming that I and J are respectively the collection of all sampling sites in a group of resources and data set and D_{ij} is the real-valued value of resource i in sampling site j. The score of mean square error of this group of resources in all sampling sites can be calculated by the following formula.

$$M(I,J)=\frac{1}{|I||J|}\sum_{i\in I,j\in J}(D_{ij}-D_{iJ}-D_{Ij}+D_{IJ})^2 \quad , \text{ where } \quad D_{iJ}=\frac{1}{|J|}\sum_{j\in J}D_{ij} \quad \text{and} \quad D_{Ij}=\frac{1}{|I|}\sum_{i\in I}D_{ij}$$

are the mean value in row i and column j; $D_{IJ}=\frac{1}{|I||J|}\sum_{i\in I,j\in J}D_{ij}$ is the mean value of expression value of this group of genes under all experimental conditions.

Figures 6(a)-6(f) provide the distribution of MSE value of mining result in data sets with different number of resources and degree. It can be seen from these figures that MSE value of almost all results is lower than 0.1, indicating that the mining result of *CoCluster* algorithm has certain correlation though it is mined from discrete data, thus showing that the effective value of resources in maximal trend bicluster mined with this algorithm has small variation. Thus, biclusters with weak trend can be mined from a lot of data so as to discover resources with fault trend in time in the earlier stage.

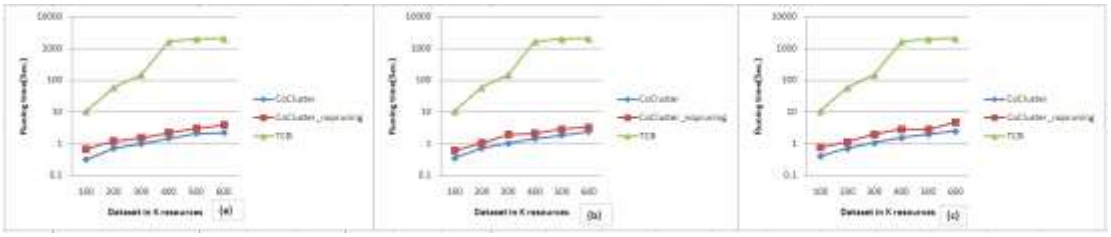

Figure 5. Comparison of Operating Time under Different Number of Resources in Data Sets with Different Discretization Degree: (a) K=5; (b) K=10; (c) K=20

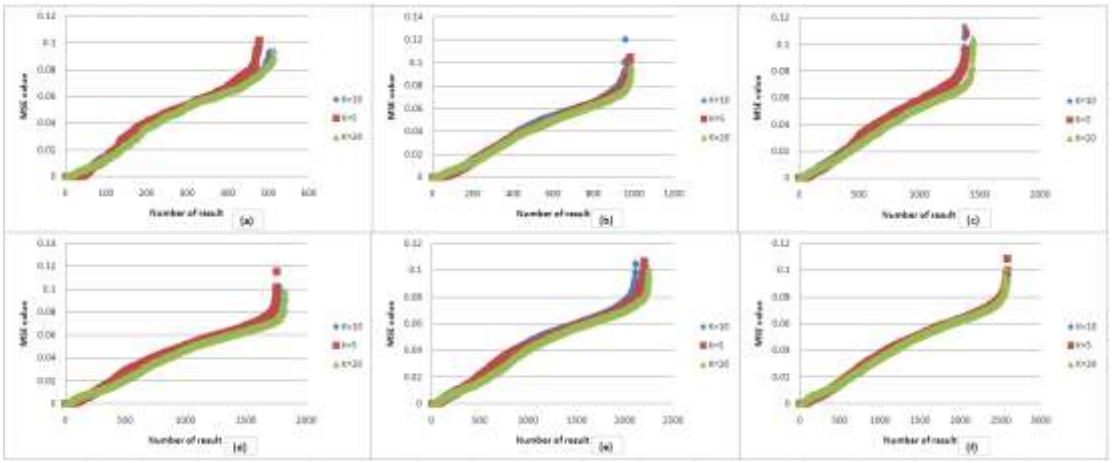

Figure 6. Distribution of MSE Value Of Mining Result in Data Sets with Different Number of Resources and Degree: (a)100 Resources; (b) 200 Resources; (c) 300 Resources; (d)400 Resources; (e) 500 Resources; (f) 600 Resources

5.2. Experimental Result of CeCluster and Analysis

In this section, a comparison will be made on the operating efficiency of *CeCluster* algorithm and TCB algorithm and *CeCluster* algorithm without using pruning strategies (denoted as *CeCluster_nonpruning*). Figures 7(a)-7(d) provide the comparison of running time of three algorithms above with different parameters and the number of resources respectively 200, 400, 600, 800 and 1000. It can be seen from these figures that the mining time of these three algorithms increases progressively with the increase of the number of resources in data set. Meanwhile, the mining efficiency of *CeCluster* algorithm is higher than that of the other two algorithms under each data size. Especially when the number of resources in data source is high, the mining efficiency of *CeCluster* algorithm is almost 1000 times higher than that of TCB algorithm. The reason is that the pruning strategy used by TCB algorithm has a lower efficiency. With the increase of the number of resources in data set, this algorithm needs more pruning judgments to mine all biclusters satisfying the threshold constraint. However, due to low success rate of pruning, the cost of pruning judgment is too high, thus influencing the mining efficiency of the algorithm. *CeCluster* algorithm uses high-efficiency pruning strategies for mining and will produce more maximal biclusters especially when the number of resources in data set is high and data are dense. Thus, the pruning efficiency of *CeCluster* algorithm will be higher. Figures 6(a)-6(f) provide the distribution of MSE value of mining result in data sets of *CeCluster* algorithm with different mining parameters when the number of resources is 1000. It can be seen from these figures that MSE value of almost all results is lower than 0.1, indicating that the mining result of *CeCluster* algorithm has certain correlation, thus showing that the effective value of resources in maximal trend bicluster mined with this algorithm has small variation. Thus, biclusters with weak trend can be mined from a lot of data so as to discover resources with fault trend in time in the earlier stage.

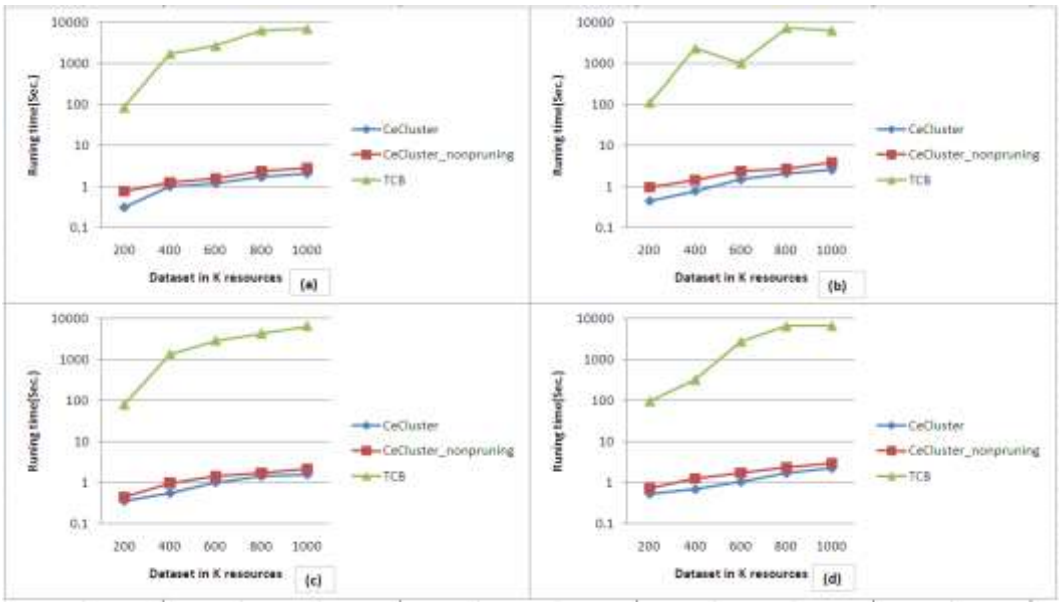

Figure 7. Comparison of Operating Time of Three Algorithms under Different Parameters: (a) α=0.2, β=0.03; (b) α=0.2, β=0.06; (c) α=0.3, β=0.03; (d) α=0.3, β=0.06

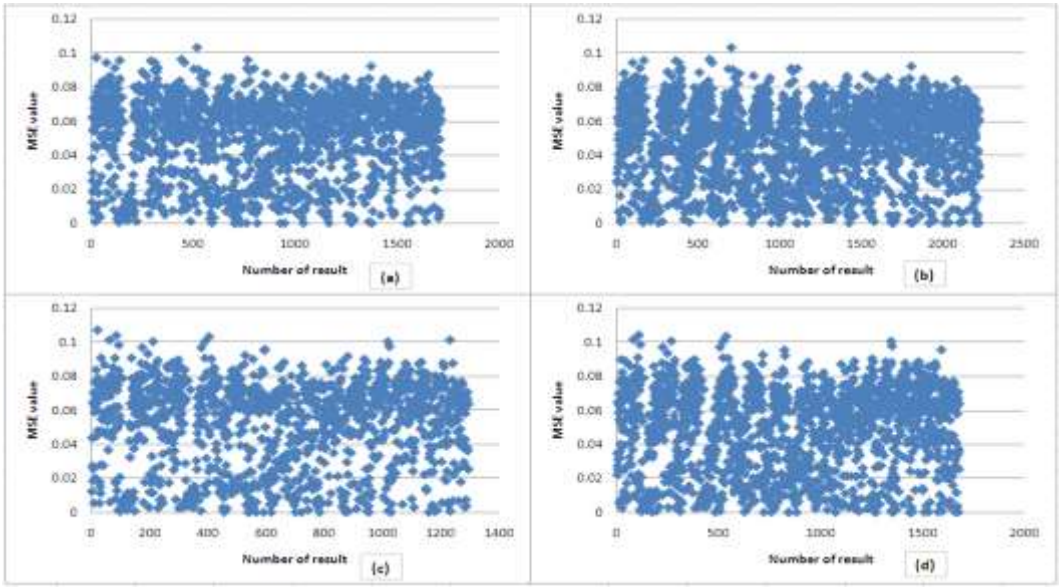

Figure 8. Distribution of MSE Value of Mining Result of *CeCluster* Algorithm under Different Parameters when the Number of Resources is 600: (a) α=0.2, β=0.03; (b) α=0.2, β=0.06; (c) α=0.3, β=0.03; (d) α=0.3, β=0.06

6. Conclusion

This paper proposed two efficient bicluster mining algorithms: *CoCluster* algorithm and *CeCluster* algorithm, to mine trend bicluster respectively in discrete and real-valued resource effectiveness matrices. To improve the mining efficiency, both algorithms mine maximal trend bicluster using the method of sample-growth and multiple pruning strategies without candidate maintenance. Meanwhile, *CoCluster* algorithm and *CeCluster* algorithm can not only mine resource patterns with effectiveness in the

downtrend, but also mine those with effectiveness in the uptrend. However, due to the lack of real test data, all experimental results of the algorithm in this paper are mined based on artificially generated data. Our next research direction is to mine trend biclusters from resource effectiveness matrix measured in real environment.

Acknowledgements

This paper is supported by Aviation Foundation under Grant No. 201510281326321 and National Key Basic Research Program of China under Grant No. 2014CB744900.

References

[1] M. Pecht, "A prognostics and health management roadmap for information and electronics-rich systems", Microelectronics Reliability, **(2010)**, pp. 317–323.
[2] Y. Cheng and G.M. Church, "Biclustering of Expression Data", Proc. 8th Int'l Conf. Intelligent Systems for Molecular Biology (ISMB00), ACM Press, **(2000)**, pp. 93–103.
[3] Z. Xu, Y. Yin, J. Wang and J.-U. Kim, "A Game-theoretic Approach for Efficient Clustering in Wireless Sensor Networks", International Journal of Hybrid Information Technology, vol.7, no.1, **(2014)**, pp. 67-80.
[4] C.-N. Zhang and Y.-R. Li, "A Kind of Chaotic Particle Swarm and Fuzzy C- mean Clustering Based on Genetic Algorithm", International Journal of Hybrid Information Technology, vol.7, no.4, **(2014)**, pp. 287-298.
[5] L. Zhao and M. J. Zaki, "MicroCluster: An Efficient Deterministic Biclustering Algorithm for Microarray Data", in IEEE Intelligent Systems, special issue on Data Mining for Bioinformatics, vol. 20, no. 6, **(2005)**, pp. 40-49.
[6] J. Kaur and N. Madan, "Association Rule Mining: A Survey", International Journal of Hybrid Information Technology, vol.8, no.7, **(2015)**, pp. 239-242.
[7] G. Pandey, G. Atluri, M. Steinbach, C. L. Myers and V. Kumar, "An association analysis approach to biclusting", In Proc. ACM Conf. on Kownlege Discovery and Data Mining, **(2009)**, pp. 677-686.
[8] A. Serin and M. Vingron, "Debi: Discovering differentially expressed biclusters using a frequent itemset approach", Algorithms for Molecular Biology, vol. 6, no. 1, **(2011)**, pp. 18-29.
[9] H. Tao and T. Yanna, "Clustering Outlier Detection Algorithm", International Journal of Hybrid Information Technology, vol.8, no.5, **(2015)**, pp. 129-134.
[10] R.Gupta, N. Rao and V. Kumar, "Discovery of Error-tolerant Biclusters from Noisy Gene Expression Data", Proceedings of BIOKDD'10, Washington DC, USA, **(2010)**.
[11] M. Wang, X. Shang, S. Zhang and Z. Li, "FDCluster：Mining frequent closed discriminative bicluster without candidate maintenance in multiple microarray datasets", ICDM 2010 workshop on Biological Data Mining and its Applications in Healthcare, **(2010)**, pp. 779-786.
[12] M. Wang, X. Shang, M. Miao, Z. Li and W. Liu, "FTCluster: Efficient Mining Fault-Tolerant Biclusters in Microarray Dataset", Proceedings of ICDM 2011 workshop on Biological Data Mining and its Applications in Healthcare, **(2011)**, pp. 1075-1082.
[13] M. Yang and X. Shang, "Bicluster algorithm facing the time-series gene expression data", Application Research of Computers, vol. 30, no. 8, **(2013)**, pp. 2308-2314.

Authors

Miao Wang, he is an engineer at science and technology on avionics integration laboratory and China aeronautical radio electronics research institute. He completed his doctor and master degree from northwestern polytechnical university in 2013 and 2018, respectively. He is the director of system integrated testing and verification technology research unit. He is a member of China computer federation. His research interests mainly include data mining, PHM, avionics and safety.

Lihua Zhang, she is an engineer at science and technology on avionics integration laboratory and China aeronautical radio electronics research institute. She completed his doctor and master degree from northwestern polytechnical university in 2014 and 2008, respectively. Her current research interests are PHM, avionics, data mining and safety.

Zhiyong Xiong, he is research fellow and the director of science and technology on avionics integration laboratory office. His research interest is IMA.