Mining Spatial Gene Expression Data Using Association Rules

M.Anandhavalli anandhigautham@gmail.com
Reader, Department of Computer Science & Engineering
Sikkim Manipal Institute of Technology
Majitar-737136, India

M.K.Ghose mkghose2000@yahoo.com
Prof&Head, Department of Computer Science & Engineering
Sikkim Manipal Institute of Technology
Majitar-737136, India

K.Gauthaman gauthamank@gmail.com
Prof&Head, Department of Pharmacognosy
Himalayan Pharmacy Institute
Majitar, East Sikkim-737136, India

Abstract

One of the important problems in data mining is discovering association rules from spatial gene expression data where each transaction consists of a set of genes and probe patterns. The most time consuming operation in this association rule discovery process is the computation of the frequency of the occurrences of interesting subset of genes (called candidates) in the database of spatial gene expression data. In this paper, an efficient method for mining strong association rules from spatial gene expression data is proposed and studied. The proposed algorithm adopts Boolean vector with relational AND operation for discovering the frequent itemsets without generating candidate itemsets and generating strong association rules with fixed antecedents. Experimental results show that the proposed algorithm is fast and memory efficient for discovering of frequent itemsets and capable of discovering meaningful association rules in effective manner.

Keywords: Spatial Gene expression data, Association Rule, Frequent itemsets, Boolean vector, Similarity Matrix.

1. INTRODUCTION

The main contribution here has been a great explosion of genomic data in recent years. This is due to the advances in various high-throughput biotechnologies such as spatial gene expression database. These large genomic data sets are information-rich and often contain much more information than the researchers who generated the data might have anticipated. Such an enormous data volume enables new types of analyses, but also makes it difficult to answer research questions using traditional methods. Analysis of these massive genomic data has two important goals:

1) To determine how the expression of any particular gene might affect the expression of other genes
2) To determine what genes are expressed as a result of certain cellular conditions, e.g. what genes are expressed in diseased cells that are not expressed in healthy cells?

The most popular pattern discovery method in data mining is association rule mining. Association rule mining was introduced by [4]. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in transaction databases or other data repositories. The relationships are not based on inherent properties of the data themselves but rather based on the co-occurrence of the items within the database. The associations between items are commonly expressed in the form of association rules. In this setting, attributes which represents items are assumed to have only two attributes and thus referred as Boolean attributes. If an item is contained in a transaction, the corresponding attribute value will be 1; otherwise the value will be 0. Many interesting and efficient algorithms have been proposed for mining association rules for these Boolean attributes, for examples, Apriori [3], DHP [6], and partition algorithms [7]. Currently most association mining algorithms are dedicated to frequent itemsets mining. These algorithms are defined in such a

way that they only find rules with high support and high confidence. A characteristic of frequent itemsets mining is that it relies on there being a meaningful minimum support level that is sufficiently high to reduce the number of frequent itemsets to a manageable level. A huge calculation and complicated transaction process are required during the frequent itemsets generation procedure. Therefore, the mining efficiency of the Apriori-like algorithms is very unsatisfactory when transaction database is very large particularly spatial gene expression database.
In this paper, an attempt has been made to propose a novel, fast and memory efficient algorithm for discovering of frequent itemsets and for generating meaningful association rules in effective manner from spatial gene expression data.

2. MATERIALS AND METHODS

2.1 SPATIAL GENE EXPRESSION DATA

The Edinburgh Mouse Atlas gene expression database (EMAGE) is being developed as part of the Mouse Gene Expression Information Resource (MGEIR) [1] in collaboration with the Jackson Laboratory, USA. EMAGE (http://genex.hgu. mrc.ac.uk/Emage/database) is a freely available, curated database of gene expression patterns generated by in situ techniques in the developing mouse embryo. The spatial gene expression data are presented as N×N similarity matrix. Each element in the matrix is a measure of similarity between the corresponding probe pattern and gene-expression region. The similarity is calculated as a fraction of overlap between the two and the total of both areas of the images. This measurement is intuitive, and commonly referred to as the Jaccard index [2]. When a pattern is compared to itself, the Jaccard value is 1 because the two input spatial regions are identical. When it is compared to another pattern, the Jaccard Index will be less than one. If the Jaccard Index is 0, the two patterns do not intersect. If a Jaccard Index value is close to 1, then the two patterns are more similar.
However, biologists are more interested in how gene expression changes under different probe patterns. Thus, these similarity values are discretized such that similarity measure greater than some predetermined thresholds and converted into Boolean matrix.

2.2 DATA PREPROCESSING

Preprocessing is often required before applying any data mining algorithms to improve performance of the results. The preprocessing procedures are used to scale the data value either 0 or 1. The values contained in the spatial gene expression matrix had to be transformed into Boolean values by a so-called discretization phase. In our context, each quantitative value has given rise to the effect of four different discretization procedures [2]: Max minus x% method, Mid-range-based cutoff method, x% cut off and x% of highest value method.
Max minus x% procedure consists of identifying the highest expression value (HV) in the data matrix, and defining a value of 1 for the expression of the gene in the given data when the expression value was above HV – x% of HV where x is an integer value. Otherwise, the expression of the gene was assigned a value of 0 (Figure 1a).
Mid-range-based cutoff (Figure 1b) identifies the highest and lowest expression values in the data matrix and the mid-range value is defined as being equidistant from these two numbers (their arithmetic mean). Then, all expression values below or equal to the mid-range were set to 0, and all values strictly above the mid-range were set to 1.
x% of highest value approach (Figure 1c) identifies data in which its level of expression is in the 5% of highest values. These are assigned the value 1, and the rest were set to 0.
Value greater than x% approach (Figure 1d) identifies the level of expression and assigns the value 1 when it is greater than given percentage and the rest are set to 0.
From these four different procedures resulted in different matrix densities, the first and last procedure resulted in the same number of Boolean 1 results for all gene expressions, whereas the second and fourth procedure generated same densities of 1, depending on the gene expression pattern throughout the various data matrix.
From the similarity matrix, two different sets of transactions are constructed, which in turn lead to two different types of association rules.

1. The items I are genes from the data set, where a transaction $T \subseteq I$ consists of genes that all have an expression pattern intersecting with the same probe pattern.

2. The items I are the probe patterns, where a transaction $T \subseteq I$ consists of probe patterns all intersecting with the expression patterns in the same image.

To create the first type of transactions, we take for each probe pattern r, every gene g from which its associated gene expression pattern g satisfies the minimum similarity β, i.e., similarity(r, g) > β, to form the itemsets.
The second type of transactions is created in a similar way. For each gene expression pattern g in the database we create an itemsets that consists of a set of probe patterns that intersect with the gene expression pattern g. Each probe pattern r must satisfy the minimum similarity β, i.e.., similarity(r, g) > β, to get included in the itemsets.

	α (Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	0
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE 1a: Results of Max minus 25% method

	α (Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	0
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE 1b: Results of Mid-range-based cutoff

	α(Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	1
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE 1c: Results of x% of highest value approach

	α(Input)	α (after discretization)
a	0.096595	0
b	0.123447	0
c	0.291310	1
d	0.126024	0
e	0.155819	1
f	0.288394	1
g	0.000000	0
h	0.215049	1

FIGURE1d: Results of Value greater than x% approach

FIGURE 1: Schematic description of the discretization protocols used

2.3 ASSOCIATION RULE MINING

The Apriori-like algorithms adopt an iterative method to discover frequent itemsets. The process of discovering frequent itemsets need multiple passes over the data. .The algorithm starts from frequent 1-itemsets until all maximum frequent itemsets are discovered. The Apriori-like algorithms consist of two major procedures: the join procedure and the prune procedure. The join procedure combines two frequent k-itemsets, which have the same (k-1)-prefix, to generate a (k+1)-itemset as a new preliminary candidate. Following the join procedure, the prune procedure is used to remove from the preliminary candidate set all itemsets whose k-subset is not a frequent itemsets [3].
From every frequent itemset of k>=2, two subsets A and C, are constructed in such a way that one subset C, contains exactly one item in it and remaining k-1 items will go to the other subset A. By the downward closure properties of the frequent itemsets these two subsets are also frequent and their support is already calculated. Now these two subsets may generate a rule A →C, if the confidence of the rule is greater than or equal to the specified minimum confidence.

2.4 ALGORITHM DETAILS

[1] Let $I=\{i_1, i_2, \ldots, i_n\}$ be a set of items, where each item ij corresponds to a value of an attribute and is a member of some attribute domain $Dh=\{d_1, d_2, \ldots, d_s\}$, i.e. $i_j \in D_h$. If I is a binary attribute, then the Dom (I)={0,1}. A transaction database is a database containing transactions in the form of (d, E), where d Є Dom(D) and E Є I.
[2] Let D be a transaction database, n be the number of transactions in D, and minsup be the minimum support of D. The new_support is defined as new_support = minsup × n.

[3] Proposition 1: By Boolean vector with AND operation, if the sum of '1' in a row vector Bi is smaller than k, it is not necessary for Bi to involve in the calculation of the k- supports.
[4] Proposition 2: According to [5], Suppose Itemsets X is a k-itemsets; |FK-1(j)| presents the number of items 'j' in the frequent set F_{K-1}. There is an item j in X. If | $F_{K-1}(j)$| is smaller than k-1, itemset X is not a frequent itemsets.
[5] Proposition 3: $|F_K|$ presents the number of k-itemsets in the frequent set F_K. If $|F_K|$ is smaller than k+1, the maximum length frequent itemsets is k.
[6] Lemma 1: If there exists two rules A→B and A→ {B U X}, where X ∉ AUB, then the confidence of the second cannot be larger than first one.

The proposed algorithm for finding the association rules in terms of spatial gene expression data in the form of similarity matrix consists of five phases as follows:
1. Transforming the similarity matrix into the Boolean matrix
2. Generating the set of frequent 1-itemsets F_1
3. Pruning the Boolean matrix
4. Generating the set of frequent k-itemsets F_k(k>1)
5. Generating association rules from the generated frequent itemsets with confidence value greater than a predefined threshold (minconfidence).

A detailed description of the proposed algorithm is described as follows:

Part 1: Algorithm for generating frequent itemsets

Input: Spatial Gene Expression data in similarity matrix (M), the minimum support.
Output: Set of frequent itemsets F.
1. Normalize the data matrix M and transformed into Boolean
Matrix B;
// Frequent 1-itemset generation
2. For each column Ci of B
3. If sum(C_i) >= new_support
4. F1 = { I_i};
5. Else delete C_i from B;
// By Proposition 1
6. For each row R_j of B
7. If sum(R_j) < 2
8. Delete R_j from B;
// By Proposition 2 and 3
9. For (k=2; | F_{k-1}| > k-1; k++)
10. {
// Join procedure
11. Produce k-vectors combination for all columns of B;
12. For each k-vectors combination { B_{i1}, B_{i2},…B_{ik}}
13. { E= B_{i1} ∩ B_{i2} ∩….∩B_{ik}
14. If sum(E) >= new_support
15. Fk = { I_{i1}, I_{i2},…I_{ik}}
16. }
// Prune procedure
17. For each item I_i in F_k
18. If |F_k(I_i)| < k
19. Delete the column Bi according to item Ii from B;
20. For each row R_j of B
21. If sum(B_j) < k+1
22. Delete Bj from B;
23. k=k+1
24. }
25. Return F = F_1UF_2….UF_k

This algorithm is capable of discovering all possible set of frequent itemsets subject to a user specified minimum confidence.

Part 2: Algorithm for generating association rules.

Input: Set of Frequent (F) with descending order of new_support count and minimum confidence.
Output: Set of Association rules

```
1.  For all f_k, f_k Є F, k=1 to max_size-1  do
2.  {
3.  req_support= new_support(f_k) X minconfidence
4.  total=0
5.  for all F_m , F_mЄ F, m=k+1 to max_size do
6.     {
7.     if new_support(F_m) >= req_support then
8.        {
9.  // By lemma 1
```

```
10.          If (Fk ⊆ Fm) then
11.             {
12.               total =totoal+1
13.               conf= new_support( Fm)/new_support(Fk)
14.                Generate the rule Fk ⟶ (Fm-Fk) &=conf and new_support=new_support(Fm)
15.             }
16.           else
17.             If ( total < 2) continue step1 with next k
18.            else
19.               total=0
20.           }
21.        }
22.  }
```

This algorithm is capable of finding all association rules with a fixed antecedent and with different consequents from the frequent itemsets subject to a user specified minimum confidence very quickly. The proposed algorithm is avoiding the unnecessary checking for the rules based on the above lemma 1.The algorithm generate the rules with a fixed antecedent part. When all the rules with that antecedent are generated it will go to the next antecedent. For a given antecedent if all rules in the level, where k is the number of items in the consequent, have confidence less than the threshold, i.e. no rules are generated, and then the confidence of any rule in k+1 level also cannot be more than threshold. So checking for rules from this level onward can be avoided without missing any rules. Now the maximum possible confidence of the rule in the k+1 level will be minimum confidence of the two itemsets from which this is constructed. Since the confidence of only one of them is larger than the threshold, others must be less than the threshold. So the confidence of the rule in k+1 will be less than threshold. So, it is not necessary to check for the rules in the next level without missing any valid rule. So it can be concluded that the proposed algorithm is complete.

3. RESULTS AND DISCUSSION

The proposed algorithm was implemented in Java and tested on Linux platform. Comprehensive experiments on spatial gene expression data has been conducted to study the impact of normalization and to compare the effect of proposed algorithm with Apriori algorithm. Figure 2 and 3 gives the experimental results for execution time (generating frequent itemsets and finding rules) vs. user specified minimum supports and shows that response time of the proposed algorithm is much better than that of the Apriori algorithm. In this case, confidence value is set 100% for the rule generation, which means that all the rules generated are true in 100% of the cases.

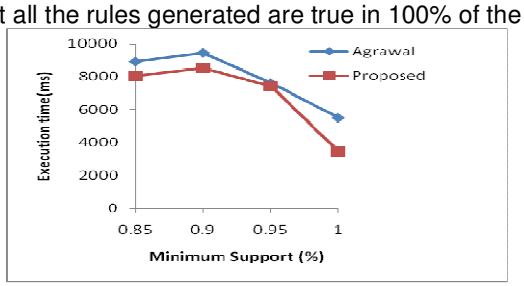

FIGURE 2: Performance on Stage 14 of EMAGE Spatial Gene expression data (Minsupport vs. Execution time)

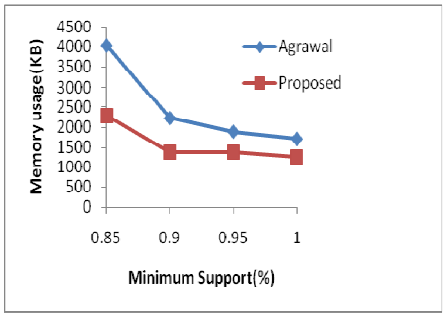

FIGURE 3: Performance on Stage 17 of EMAGE Spatial Gene expression data (Minsupport vs. Execution time)

Figure 4 and 5 gives the experimental results for memory usage vs. user specified minimum supports and results show that proposed algorithm uses less memory than that of Apriori algorithm because of the Boolean and relational AND bit operations.

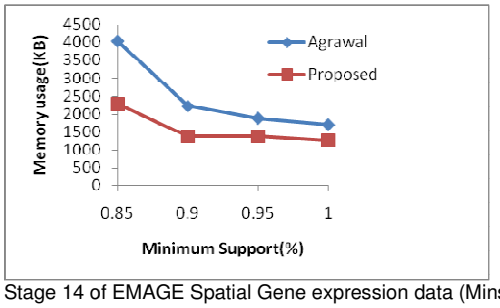

FIGURE 4: Performance on Stage 14 of EMAGE Spatial Gene expression data (Minsupport vs. Memory usage)

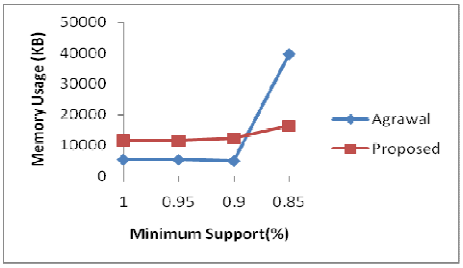

FIGURE 5: Performance on Stage 17 of EMAGE Spatial Gene expression data (Minsupport vs. Memory usage)

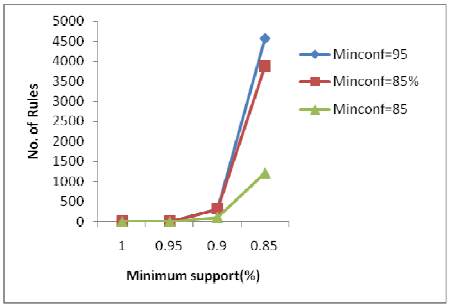

FIGURE 6: Association rules and Minsup in Apriori algorithm Stage 14 of EMAGE Spatial Gene expression

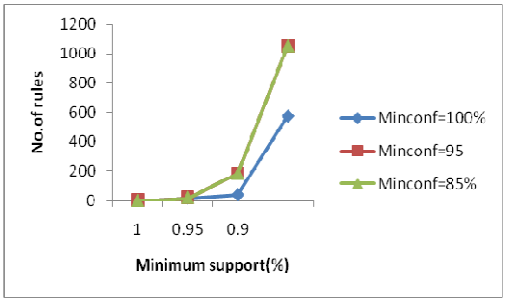

FIGURE 7: Association rules and Minsup in Proposed algorithm Stage 14 of EMAGE Spatial Gene expression

The number of association rules decreases along with an increase in minimum support under a given specific minimum confidence, which shows an appropriate Minsupport (or Minconfidence) can constraint the number of association rules and avoid the occurrence of some association rules so that it cannot yield a decision. These results have shown in Figures 6-7 for the Stage 14 of EMAGE spatial gene expression data. The results are as expected and quite consistent with our intuition.

4. CONCLUSION

In this paper, a novel method of mining frequent itemsets and strong association rules from the spatial gene expression data has been proposed to generate frequently occur genes very quickly. The proposed algorithm does not produce candidate itemsets, it spends less time for calculating k-

supports of the itemsets with the Boolean matrix pruned, and it scans the database only once and needs less memory space when compared with Apriori algorithm. The proposed algorithm is good enough for generating association rules from spatial gene expression data and it is very fast and memory efficient. Finally, the large and rapidly increasing compendium of data demands data mining approaches, particularly association rule mining ensures that genomic data mining will continue to be a necessary and highly productive field for the foreseeable future.

5. ACKNOWLEDGMENT

This study has been carried out as part of Research Promotion Scheme (RPS) Project under AICTE, Govt. of India.

6. REFERENCES

1. Baldock,R.A., Bard,J.B., Burger,A., Burton,N., Christiansen,J., Feng,G., Hill,B., Houghton,D., Kaufman,M., Rao,J. et al., "EMAP and EMAGE: a framework for understanding spatially organized data", Neuroinformatics, 1, 309–325, 2003.
2. Pang-Ning Tan, Micahel Steinbach, Vipin Kumare, "Intoduction to Data Mining Pearson Education", second edition, pp.74, 2008.
3. Agrawal, R. & Srikant, R., "Fast Algorithms for Mining Association Rules in large databases". In Proceedings of the 20th International Conference on Very Large Databases pp. 487-499. Santiago, Chile, 1994.
4. Agrawal, R., Imielinski, T., & Swami, A., "Mining association rules between sets of items in large databases". Proceedings of the ACM SICMOD conference on management of data", Washington, D.C, 1993.
5. Xu, Z. & Zhang, S., "An Optimization Algorithm Base on Apriori for Association Rules". Computer Engineering 29(19), 83-84, 2003.
6. J S. Park and M -S. Chen and PS. Yu, "An effective hash-based algorithm for mining association rules", Proceedings of the ACM SIGMOD International Conference on Management of Data", San Jose, CA, May 1995.
7. A Savasere, E. Ommcinskl and S Navathe, "An efficient algorithm for mining association rules in large databases", In Proceedings Of the 21st International Conference on Very Large Databases, Zurich, Switzerland, September 1995.