COMPARISON BETWEEN RISS AND DCHARM FOR MINING GENE EXPRESSION DATA

Shaymaa Mousa

Department of management information system,
King abdalziz University, Jeddah, Saudi Arabia

ABSTRACT

Since the rapid advance of microarray technology, gene expression data are gaining recent interest to reveal biological information about genes functions and their relation to health. Data mining techniques are effective and efficient in extracting useful patterns. Most of the current data mining algorithms suffer from high processing time while generating frequent itemsets. The aim of this paper is to provide a comparative study of two Closed Frequent Itemsets algorithms (CFI), dCHARM and RISS. They are examined with high dimension data specifically gene expression data. Nine experiments are conducted with different number of genes to examine the performance of both algorithms. It is found that RISS outperforms dCHARM in terms of processing time..

KEYWORDS

Association Rules, Bioinformatics, Closed Frequent Itemsets, Data Mining, Gene Expression, Microarray.

1. INTRODUCTION

It is widely believed that thousands of genes and their products, RNA and proteins, in a given living organism open a new research area. The challenge is to extract useful information and discover knowledge from the biological data. Microarray technology promises to monitor the whole genome on a single chip to provide a better picture of the interactions among thousands of genes simultaneously [1, 5]. Microarray technology results in a new type of data format which implies the adaptation of computational techniques. Many bioinformatics and data mining researchers have been working on applying data mining techniques to analyze gene expression datasets such as classification, clustering, and association rules. Mining gene expression associations requires high computational capabilities in terms of memory space and processing time. These challenges inspired comparative researches among different algorithms.

The aim of this paper is to compare the performance of two Closed Frequent Itemsets Generation algorithms, RISS and dCHARM for microarray gene expression data in terms of processing time.

This paper is organized as follows: section two illustrates related work, section three introduces RISS algorithm, section four presents dCHARM algorithm, section five demonstrates the conducted experiments and their results, and section six concludes the research outcomes.

DOI : 10.5121/ijdkp.2013.3505

2. RELATED WORK

Most of the processing time in mining gene expression data is consumed in generating frequent itemsets. Therefore, researchers suggest the use of Closed Frequent Itemsets as an alternative to the usual Frequent Itemsets. The most common Closed Frequent Itemsets algorithms used for gene expression data are CARPENTER, COBBLER, FARMER, and MineTop-K [2-5]. CARPENTER addresses the problem of dealing with high dimensional datasets as gene expression by traversing row enumeration search space using Depth First Search (DFS) to discover closed frequent itemsets. COBBLER switches dynamically between column and row enumerations depending on the estimated cost of each during the mining process. FARMER addresses the problem of the huge number of resulted association rules by generating only interesting rule groups (IRG). IRG are a set of rules that are generated from the same group of rows and meet user interestingness constraints including minsupp, minconf, and minimum chi-square (minchi) threshold. MineTop-K focuses on the same problem of FARMER but generates most significant Topk covering rule groups rather than generating IRG. Topk covering rule groups are defined by first setting criterion for ranking and applying it to the resulted rules in the dataset. All of these works performed their experiments on the same data with different minsupp and different row lengths [3]. In most cases, COBBLER and FARMER outperform CARPENTER. All of these algorithms are reviewed in details in [6].

3. RISS ALGORITHM

Row Intersection Support Starting (RISS) algorithm is specially designed to handle datasets having a large number of items and relatively small number of rows such as gene expression dataset. RISS discovers closed frequent itemsets by traversing the Minimal Bottom-up row enumeration search space using BFS instead of the usual column enumeration traversal. It uses efficient search pruning techniques to yield a highly optimized algorithm. RISS traverse the row enumeration space using row enumeration-based mining algorithm which starts the search tree from the minsupp threshold deploying a vertical data format called RowSet (RS) [7].

3.1. Notations

Let T be a discretized gene expression table as shown in Table 1. It consists of a set of biological Samples B= {b1, b2, b3, …., bm} as its rows, and set of genes G= {g1, g2, g3, …., gn} as its columns. A table T is a triple (B, G, R), where $R \subseteq B \times G$ is a relation. For a $bi \in B$, and a $gj \in G$, $(bi, gj) \in R$ denotes that the gene gj is over expressed or under expressed in the biological sample bi.

Definition 1 Row Set (B')

Given a node in the enumeration tree, B' is the set of biological samples that are represented in this node. Each node in the tree represents a row set B'. For example, in Fig 1, the node 12 represents the row set {b1, b2}.

Definition 2 Supporting Feature Set G (B')

Given a row set B' in a gene expression table, G (B') is the maximal set of genes common to all the biological samples in B`. For example, in Fig 1, the supporting feature set of B'= {b4, b5} is G (B') = {g1, g2, g3, g5}

Definition 3 Row Set Frequency (Freq (G(B')))

Given a row set B' in a gene expression table, Freq (B') is the maximal set of biological samples numbers that contain G (B'). For example, in Fig 1, Freq (G (B')) = {1 2 3 4 5 6} is the row set frequency of B`= {b2, b3, b6}, and G (B`) = {g2}.

Table 1. Example Table

Biological Samples	**g1**	**g2**	**g3**	**g4**	**g5**	**g6**
b1	1	1	0	1	1	0
b2	0	1	1	0	1	0
b3	1	1	0	1	1	0
b4	1	1	1	0	1	0
b5	1	1	1	1	1	0
b6	0	1	1	1	0	1

3.2. Minimal Bottom-up Search Strategy

It checks row combinations from the smallest to the largest to traverse the search space, Fig 1. For example, 1- rowsets, then 2-rowsets, …, and finally n-rowsets. Minimal Bottom-up search strategy is a modification from the traditional one [1-3, 8] as it starts the search from the minsupp –rowsets rather than the 1- rowsets. This is valid as the maximum support for the K-rowsets equals K. For example if the minsupp =3 the enumeration tree starts from level 3 as shown in the rectangle shape in Fig 1. Each node in the enumeration tree is represented by RS format.

3.3. Pruning Techniques

RISS algorithm deploys two pruning techniques to speed up the mining process by decreasing the search space. The first pruning implies that there is no maximal genes common to the biological samples B, so no further enumeration will be required on the branch of this node. The second pruning action checks if the G (B`) exists in CFI, if it is true, current and further enumeration of this node is truncated. In other words, G (B`) does not discover any new closed frequent itemsets. This results in a condensed enumeration tree as shown in Fig2.

RISS does not need to perform the closure check among the discovered Itemsets since RISS extracts only the closed itemsets. The proof is that G (B`) cannot be a maximal gene set that is common to all biological samples B` unless it is a closed itemsets [9].

4. DCHARM ALGORITHM

dCHARM algorithm is based on diffset data structure [10]. dCHARM performs a search for closed frequent sets by exploring both the itemset space and transaction space over an IT-tree (itemset-tidset tree). It uses diffset vertical data representation for fast support computations. It is also reported that at a given level of support, the execution time is linearly increased with increasing number of transactions. Also, for many databases the intermediate diffsets can easily fit in memory even for low minsupp [11]. One of the reported limitations is that with databases

which have a large number of very short closed patterns and with low minsupp, dCHARM performance decreases.

4.1. Notations

Consider gene expression data in Table 1, Assume that X and Y are any two members in class [P] with X= g1 and Y= g3.

Definition 1 Diffset d(XY*)*

Let d(X) and d(Y) are the diffsets of X and Y correspondingly. The diffset of the itemset (XY) can be calculated as d(XY) = d(Y) _ d(X).

Example:

Since d(g1)= 2 6 and d(g3)= 1 3 then d(g1g3)= d(g3) - d(g1)= 1 3

Definition 2 Support *Supp (XY)*

Let d(X) and d(Y) are the diffsets of X and Y respectively. The support of the itemset (XY) can be computed as σ (XY) = σ (X) _ |d(XY)|

Example:

Since σ (g1)=4, σ (g1g3) = σ (g1) _ |d(g1g3)| = 4-2=2

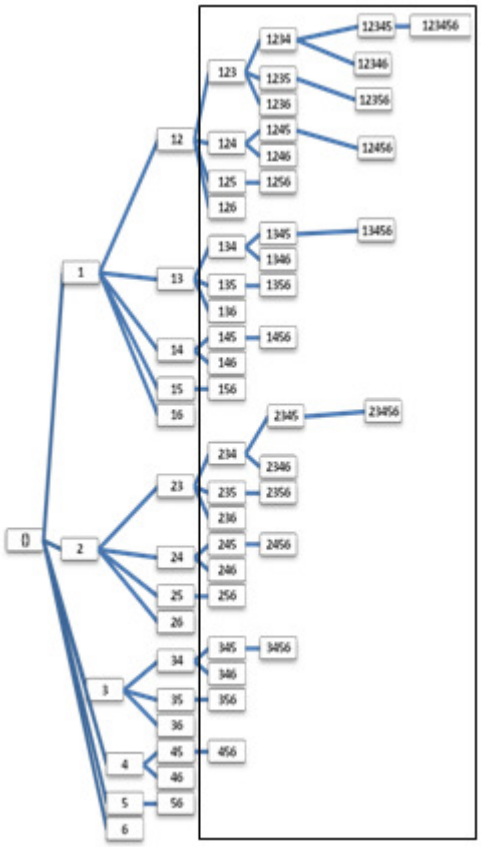

Figure 1. Row Enumeration Search Space

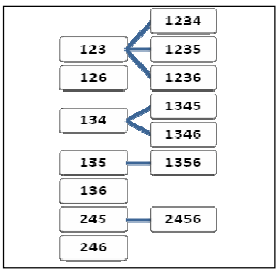

Figure 2. Pruning Enumeration Tree

Definition 3 Mismatch *m(X)*

m(X) and m(Y) denote the number of mismatches in the diffsets d(X) and d(Y).

Example:, m(g3)= 2 m(g1)=2

dCHARM performs a DFS for closed frequent sets over a novel Itemset-Diffset (ID-tree) search space which is in fact a prefix-based class. It uses the concept of a closure check to validate if a given itemset X is closed or not. The support of an itemset X is also equal to the support of its closure, i.e., $\sigma(X) = \sigma(c(X))$ [10].

The original CHARM algorithm uses TIDset for initial database, but uses diffsets thereafter which has been modified to use diffset data structure from the beginning. CHARM uses four basic properties of IT-pairs for pruning search space [10].

5. EXPERIMENTS AND RESULTS

5.1. Data

Array data is used of 24,483 gene measurements recorded for 19 breast cancer patients [12]. It contains: systematic name given to each gene or sequence and a description of what is known about gene's function. Also, it contains three values for each tumor sample profiled: Log10 (Intensity), Log10 (ratio), and P-value [13]. According to the biology specialists' opinion, the P-value is used in mining process.

5.2. Experiments

The aim is to compare the performance of the two algorithms in terms of time. Nine experiments have been conducted with different number of genes {2000, 4000, 6000, 8000} with minimum support threshold equals 42.1%. All experiments are performed on a PC with Core Duo 2 GHz CPU, 1GB RAM and a 120GB hard-disk and algorithm is coded in MATLAB.

5.3. Results

The processing time for generating closed frequent itemsets for each experiments is recorded, Table 2 and Fig 3. Both algorithms indicate that as the number of genes increases the processing time increases. Also, dCHARM processing time increases exponentially whereas RISS processing time increases more or less proportional.

Table 2. The Experiments Results.

Genes	Processing Time (m second)	
	RISS	dCHARM
2000	1611.70	8058.50
4000	5150.600	36054.200
6000	13038.00	130380
8000	24464.30	293571.6

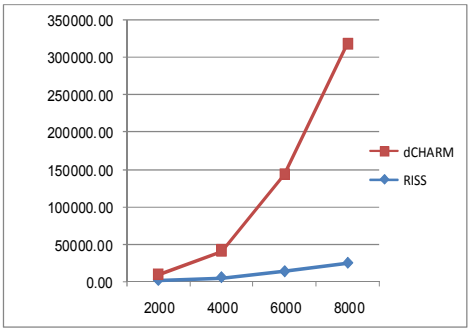

Figure 3. CFI processing Time of RISS and dCHARM.

6. CONCLUSIONS

This paper compares between RISS or dCHARM for microarray gene expression data rather than specifying their characteristics. RISS has four advantages over dCHARM: First, RISS does not require to check itemsets for minsupp condition. Second, RISS guarantees that the generated itemsets are closed ones without the need to check closure property. Third, RISS reduces search space by 57.14% while dCHARM pruned only 35.7% of its search space. Fourth, RISS is extremely faster than dCHARM especially in high dimension data.

REFERENCES

[1] J. Hancock, and M. Zvelebil, Dictionary of Bioinformatics and Computational Biology, A John Wiley & Sons, Inc., Publication, 2004.

[2] J. Cohen, "Bioinformatics-An Introduction for Computer Scientists", ACM Computing Surveys, vol. 36, no. 3, pp. 122-168, June 2004.

[3] V. Bajic, V. Brusic, J. Li, S-K. Ng, L. Wong, "From Informatics to Bioinformatics", Conferences in Research and Practice in Information Technology, Vol. 19, 2003.

[4] N. M. Luscombe, D. Greenbaum, M. Gerstein, "What is Bioinformatics?A Proposed Definition and Overview of the Field", Department of Molecular Biophysics and Biochemistry Yale University, New Haven, USA.2001.

[5] A. Krause, "Large Scale Clustering of Protein Sequences", Ph.D. Dissertation, Berlin, 2002.

[6] A.Sharaf_Eldin, M.Hana, T.Soliman, S.Rashad, ”A Comparative Study of Association Rules for Mining Gene Expression Databases”, International Journal of Intelligent Computing and Information Science, Volume 7, January 2007.

[7] A.Sharaf_Eldin, M.Hana, S.Kassim, S.Rashad,“Associations System For Breast Cancer Microarray Data”, International Journal of Intelligent Computing and Information Science, 2009.

[8] Hegde P, Qi R, Abernathy K, Gay C, Dharap S, Gaspard R, Hughes JE, Snesrud E, Lee N, Quackenbush J, “A concise guide to cDNA microarray analysis”, Biotechniques 3, 548-554, 2000.

[9] C. Tang, and A. Zhang. "Mining Multiple Phenotype Structures Underlying Gene Expression Profiles", CIKM'03, New Orleans, Louisiana, USA, November 3-8, 2003.

[10] M. Zaki, and K. Gouda, "Fast Vertical Mining Diffsets", RPI Technical Report 01-1. Rensselaer Polytechnic Institute, Troy, NY 12180 USA. New York, 2001.

[11] Wang DG, Fan JB, Siao CJ, Berno A, Young P, Sapolsky R, et al. “Large-scale identification, mapping, and genotyping of single nucleotide polymorphisms in the human genome”. Science, 280:1077–82, 1998.

[12] Ladina Joos, Emel Eryüksel, Martin H. Brutsche, “Functional genomics and gene microarrays the use in research and clinical medicine”, SWISS MED WKLY, 133:31–38, 2003.

[13] M. Ritchie, "Bioinformatics Approaches for Detecting Gene–Gene and Gene–Environment Interactions in Studies of Human Disease", Neurosurg. Focus , Volume 19 , October, 2005.