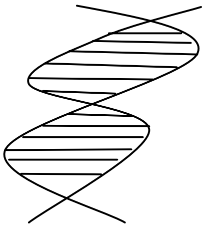

A report on:

Mining Gene Expression Databases for Association Rules

Chad Creighton and Samir Hanash

University of Michigan, Ann Arbor USA

by Arpit Gattani
gattani@cc.usu.edu

Problem Statement:

Association rules served business world for many years by predicting the nature of sells. Now computer scientists are looking for more areas where associations can be applied. Genetics is one of those fields. Association rules can uncover many of the hidden rules in gene and protein databases. Authors of the papers applied Apriori algorithm to a database consisting of 300 expression profiles for yeast to mine association rules. They tried to reveal the expressions in the dataset which were associated with other expressions. Authors were also concerned by the minimum Support and maximum Confidence they needed to set for the experiment. Another problem authors required to deal with was How to stop the generation of the redundant item sets and How to limit the candidate generation.

Contribution of the paper:

Problem of mining association rules in gene expression was tackled by binning the each measured value as highly expressed, highly repressed or neither. It helps lessen the effect of the noise on the data and hence results in more authenticate data.

After this the frequent itemsets were found. Apriori algorithm was used to find the frequent itemsets. Apriori algorithm depends on a very basic property, i.e. for a itemset to be frequent, each of its subset must also be a frequent itemset. The algorithm starts with a single item in the set and then runs iteratively with each frequent itemset detected in previous level increases by one. By this approach it is very hard to leave any frequent dataset.

Next approach after finding the frequent itemsets is to mine association rules for these frequent itemsets. An association rule defined in the paper is of kind *LHS* => *RHS*, where LHS and RHS are sets of items in a frequent itemset, *X*. The support of this rule with respect to a transaction T, can be defined as *LHS* U *RHS* with respect to *T*. The confidence can be defined as (*LHS* U *RHS*) / support(LHS). Apriori algorithm applied to find frequent itemsets, finds all the frequent itemsets in the given dataset. But many of the itemsets found to be frequent are redundant which are actually the subsets of the larger frequent itemsets. Approach taken by the paper is to limit down the search space of the candidates generated.

Implementation: the database for the application was taken from Hughes et al. A expression profile of about 6316 transcripts corresponding to 300 different mutations and chemical analysis of yeast were considered for application. A database application was developed to implement the Apriori algorithm. Application was built in Microsoft Access Database with the connection to SQL Server. Input was given by one or more spreadsheets with items in rows and experiments in column. The application was run to find the frequent itemsets in the data. Application was also provided with the flexibility to select any other constraints than the values of support and confidence. After finding the frequent itemsets the association rules were mined. Association rules of the kind LHS => RHS, where LHS is a single item, were found. Minimum length of the frequent itemset was kept 7 so that no itemset less than 7 items should appear in the frequent itemsets as that would increase the candidate generation.

Results: Binning of the expression was done at a value greater than 0.2 for being up and -0.2 for being down. The expressions between these values were considered as neither up nor down. It was found that among the 6316 data set, 197 were found up in at least 10% of the experiments, 47 were found down in same percentage of experiments. Rests were found neutral. Also a randomized data set was found and the same application was run on that randomized data set. Its purpose was to confirm that the association rules found in yeast data wasn't by chance.

Algorithm was implemented on these two separate data sets, randomized and yeast datasets. The value of minimum support and confidence was kept 10% and 80% respectively in both datasets. It was found that there was at least one association rule for each frequently found itemsets with LHS = one item. The application was run on a Pentium 4 processor and it took about a day to find all frequent itemsets. On the other side, in the randomized dataset only one frequent itemset was found. So it was proved that the association rules generated in yeast data were not by chance.

With frequent itemsize greater than seven or more, application found about 40 practically important rules. These rules were interpreted with there biological importance. The generated rules were listed in two different tables, one the rules found were listed. It was found that each gene in this list were up and none were found down. Second table presents the detailed description of each of the rules in table 1.

While interpreting the rules generated, it was found that under defined minimum support and confidence, where ever the gene YHM1 is up, all of the RHS to it were also up. Other rules found were also defined in the same fashion. It was also discovered that for association rules found, where number of itemsets were more, support and confidence were very near to the defined minimum.

Limitations of the Approach:

Paper approaches the problem of finding the association rules in gene expression through a very straightforward method. Apriori algorithm is used by the authors to discover association rules in the dataset. Though version of Apriori algorithm used in the paper have many advantages like capability to find maximum frequent patterns, accuracy and controlled candidate generation, it has some limitations. First limitation that paper have is authors implemented the algorithm on a very small dataset. For a larger dataset it would have been quite difficult to handle the candidate generation and redundant frequent itemset. It is possible that even after tuning the various parameters the program will run out of memory. Also the data is provided to program via spreadsheet, which is very inconvenient when data is large. Algorithm in itself doesn't define any technique to take care of this problem. Secondly, value of the minimum "support" and "confidence" are also chosen independently to find rules in frequent itemsets. Interpretation of the results of the paper suggest that most of the association rules were found at the edge of the minimum value of support and confidence. To find useful results in mining such genetic data, its very important to have some criteria for selecting the minimum values for support and confidence. Third important limitation is that association rules found were found with pattern *LHS* => *RHS* where independent item (LHS) is a single item. In most genetic data there are more complex rules to be defined.

Space for Further Enhancements:

Though the paper is aimed to find simple association rules in yeast dataset and compare its authenticity, the approach can be enhanced to mine frequent patterns and association rules in larger datasets. One of such enhancements could be adopting constraints based mining of association rules. This approach can solve problems like repeated candidate generation and generation of redundant frequent itemsets. Also this will allow us to set our minimum support and confidence values dynamically, thus giving us more precise results. Alternate technique which can be applied to the problem is to apply clustering on the predefined association rules. This technique will allow us to mine for new association rule in a more manageable environment.

Another enhancement that can be made in the present algorithms is that it can be made to process at different levels. A iterative algorithm will allow us to keep track of redundant candidate generation at each level and we can leave the repeated ones at that level.

One of the lately invented techniques to find frequent substructures in large dataset is Graphical representation of data. Such techniques guarantees the accurate results and easily manageable implementation. Graphical representation provide us with the facility to mine association rules with complete control as we can define the time and level of pruning.