DISTANCE BASED CLUSTERING OF ASSOCIATION RULES

GUNJAN K. GUPTA , ALEXANDER STREHL AND JOYDEEP GHOSH
Department of Electrical and Computer Engineering
The University of Texas at Austin,
Austin, TX 78712-1084, USA

ABSTRACT:

Association rule mining is one of the most important procedures in data mining. In industry applications, often more than 10,000 rules are discovered. To allow manual insepection and support knowledge discovery the number of rules has to be reduced significantly by techniques such as pruning or grouping. In this paper, we present a new normalized distance metric to group association rules. Based on these distances, an agglomerative clustering algoritm is used to cluster the rules. Also the rules are embedded in a vector space by multi-dimensional scaling and clustered using a self organizing feature map. The results are combined for visualization. We compare various distance measures and illustrate subjective and objective cluster purity on results obtained from real data-sets.

INTRODUCTION

Massive amounts of data are being generated and stored every day in corporate computer database systems. Mining association rules [2] from transactional data is becoming a popular and important knowledge discovery technique [3]. For example, association rules (ARs) of retail data can provide valuable information on customer buying behavior. The number of rules discovered in a real data-set can easily exceed $10,000$. To manage this knowledge, rules have to be pruned and grouped, so that only a reasonable number of rules have to be inspected and analysed. In this paper we propose a new distance metric between two ARs. and propose a new grouping methodology using multi-dimensional scaling (MDS) and self organizing maps (SOMs). In this paper, we propose a new distance metric to cluster association rules (section 2) that improves upon the metric proposed in [8]. Based on the distance metric, we propose a new agglomerative clustering technique (section 3). Moreover, we embed the distances using multi-dimensional scaling and cluster the resulting points into a Euclidean space using a Self Organizing Feature Map (SOM)(section 4). We propose a visualization scheme to compare both techniques by color-coding the SOM results based on the agglomerative clustering results (section 4). Figure 1 depicts the overall process flow-diagram of our proposed system.

DISTANCE METRICS

A Euclidean distance could be defined on rule features such as support, confidence, lift or the bit-vector representation of BS. These direct features are very limited in capturing the interaction of rules on the data and characterize only a single rule. One way of defining distance between rules is in terms of the overlap of their market-baskets like the one proposed in [8]. One problem with this metric is that it grows as the number of market-baskets in the database increases. This can be corrected by normalizing (divide the measure by the size of the database $|r|$). However, the measure is still strongly correlated with support. High support rules will on average tend to have higher distances to everybody else. This is an undesired property. For example, two pairs of rules, both pairs consisting of non-overlapping rules, may have different distances. High support pairs have a higher distance than low support pairs. As an improvement to this metric, we propose a new distance measure based on a conditional probability estimate, as

$$\begin{aligned} d_{i,j} &= P(\overline{BS_i} \vee \overline{BS_j} | BS_i \vee BS_j) \qquad (1) \\ &= 1 - \frac{|m(BS_i, BS_j)|}{|m(BS_i)| + |m(BS_j)| - |m(BS_i, BS_j)|}, \end{aligned}$$

where the set BS_i is the union of items in the left and right hand sides of rule i, and $m(X)$ is the set of all transactions containing itemset X. We call $d_{i,j}$ the Conditional Market-Basket Probability (CMPB) Distance. Rules having no common MBs are at a distance of 1, and rules valid for an identical set of baskets are at a distance of 0. Let us call a distance interesting if it is neither 0 nor 1. Rule pairs with an interesting distance are called good neighbors. In most real databases, the majority of all rule pairs are not good neighbors. Manual exploration of a rule's good neighbors showed that intuitive relatedness was captured very well by this metric. For example, rules involving different items but serving equal purposes were found to be close good neighbors. Super-set relationships of the item-sets associated to the rules often lead to very small distances. The average time complexity for the computation of d is $O(N \cdot M^2 + K^2)$ where N is the number of transactions in the database, M is the average market-basket size in number of transactions and K is the number of discovered rules. The memory space complexity grows as $O(N + K^2)$. In most cases, a sparse matrix representation for $\mathbf{d}$ can cut down memory requirements significantly.

CLUSTERING

Combining SOM clustering results with Dimensionless Agglomerative Chain Clustering developed by us results in a good visualization interface. The distance measure described in Section 2 can directly be used for Agglomerative Clustering but a SOM needs a vector input.

One possibility for obtaining an embedding space for the rules is by defining a binary vector for each rule with one bit per item to describe its presence or absence. But such vectors are very sparse since the number of different items runs into thousands. The approach does not seem very attractive especially from the point of view of training a neural network. Multi-Dimensional Scaling[6] can be used to convert the distance information into an embedded space such that the distance information between rules is preserved.

Agglomerative Chain Clustering. We propose a Chaining algorithm that does not use any coordinate system and finds the clusters using the distance measures only. In this algorithm a point is joined to its closest neighbor found from the distance matrix. This process is applied to all the points in the space and results in a collection of graphs. All points joined together as a graph end up having the same label. The algorithm returns the labels of all the points. A nice property of the algorithm is that it scales the cluster sizes depending upon the density of the points in the neighborhood.

A more dense neighborhood results in a smaller more compact cluster and a more sparse region of the space returns a larger less dense cluster. It can be shown that the resultant clusters are unique and do not depend on the starting point.

Agglomerative Chain Clustering performs Chaining at multiple levels. At the end of the algorithm we get a tree structure that describes the multiple levels of clustering. It is similar to Single Link Agglomerative Clustering[4] but differs in its bias. The tree returned is shorter and the clusters more uniformly sized. The algorithm works as follows:

1. Perform chaining on all the points and retrieve all the clusters.
2. Find the centroid of each cluster.
3. Form a new space of N_c points representing the cluster centers.
4. If N_c is greater than 1 $then$ Go to Step 1 $else$ STOP.

An Agglomerative Chained Tree is shown in Figure 2. In this figure the height of a node is calculated as the average distance of the original points of the cluster from the centroid. This height represents the compactness of the clusters and is useful for extracting clusters of comparable compactness from the tree.

Dimensionless Agglomerative Chain Clustering. This is the method used for clustering Association Rules in this paper. It is a special case of the Agglomerative Chain Clustering and allows us to cluster rules together even in the absence of dimensional information. The first level chaining performs the clustering without any information of the location of the points. To repeat the chaining at the next level, we only need the distance between clusters (centers) and not the coordinates of the centers themselves. These distances can be estimated using many different methods. The method is used in this paper is a variation of the Lance and Williams Flexible Method[4]. According to their method, the distance between a group k and a group (ij) formed by the fusion of groups i and j satisfies the recurrence formula for the distance defined as follows:

$$d_{k(i,j)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta_{dij} + \gamma |d_{ki} - d_{kj}|, \tag{2}$$

where d_{ij} is the distance between the groups i and j and α, β and γ are parameters whose values depend on the definition of the center of clusters. By allowing β to vary, clustering schemes with various characteristics can be obtained; Lance and Williams suggest that probably the best value to assume for β is some small negative value, and in their example they use the value 0.25. For centroid the parameters α, β and γ take the following values:

$$\alpha_i = \frac{n_i}{n_i + n_j}; \alpha_j = \frac{n_j}{n_i + n_j}; \beta = -\alpha_i \alpha_j; \gamma = 0 \tag{3}$$

For three points, the formula becomes: $d_{k(ji)} = 0.5(d_{ki} + d_{kj}) - 0.25 d_{(ij)}$, where $d_{k(ji)}$ represents the distance of point k from the centroid of cluster ij. As we can see it is equal to the average distance minus one-fourth the distance between i and j. This distance measure is suitable for the single link clustering algorithm described by the author in [4].

In our clustering algorithm more than two points merge at one level. Hence the formula had to be modified to make it applicable. Using it, we can estimate distances between clusters at level $l+1$ using centroid distances between points at level l. The distance between clusters a and b is given by the formula:

$$d_{ab} = C_{ab} - 0.5(A_a + A_b); \tag{4}$$

$$C_{ab} = \frac{1}{n_a n_b} \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} d_{ij}; \;\; A_a = \frac{1}{n_a^2} \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} d_{ij}; \;\; A_b = \frac{1}{n_b^2} \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} d_{ij}; \tag{5}$$

where n_a represents the number of (original) points in cluster a and n_b represents the number of (original) points in cluster b. For the height of a node a in a tree the formula used is the mean distance of all the points in the cluster represented by the node as shown below.

$$H_a = \frac{2}{n_a(n_a-1)} \sum_{i=1}^{n_a} \sum_{j=1} d_{ij} \tag{6}$$

The points in the cluster a are represented by the leaf nodes of the agglomerative tree that have node a as an ancestor. n_a is the number of points in cluster a.

We examined the quality of dimensionless clustering by considering only the distance information between data points in different dimensional Euclidean spaces. For a very high dimensional space the error was small and since the Association Rule space is inherently high dimensional, the formulas for cluster width and distance estimation work well. The clusters obtained during simulations using the dimensionless approximation are identical to dimension based clustering for up to 60 points for a 2-D space and even more for higher dimension space. At 1000 points and 40 dimensions the number of points grouped differently in the clustering tree is less than 1% on an average.

SOM CLUSTERING & VISUALIZATION

The scalar distance between the rules cannot be used as an input to a SOM directly. Hence Multi-Dimensional Scaling is performed using Singular Value Decomposition. The data is normalized to zero mean distribution and the suitable value for the embedding dimension is obtained by monitoring the Stress Factor[6]. Given the Matrix M as the original distance matrix and M' as the corresponding matrix in the projected space, the Stress between M and M' is given by:

$$Stress = \frac{\sum_{k=1}^{N} \sum_{i=1}^{k-1} (d_{ik} - d'_{ik})^2}{\sum_{k=1}^{N} \sum_{i=1}^{k-1} (d_{ik})^2} \tag{7}$$

A Stress threshold with a shortened binary search gives a very close estimate of the correct number of dimensions in 2-4 trials. For a search range of 1 to N, the Shortened Binary Search examines stress value at $L = \frac{N}{2}, \frac{3N}{4}, \frac{7N}{8}, ..$ until the Stress is less than threshold. For 1000 rules the cutoff was reached with Shortened Binary Search at L=750 with 2.3 % Stress.

The embedded space obtained from Multi-Dimensional Scaling can now be used with any clustering algorithm that needs a vector input. In particular, mapping the input space to a 2-D SOM output space seemed to be very suitable since it provides an easy visualization of the clustering. But how do we verify results from SOM given the abstract nature of ’distance’ between rules ? We developed a novel technique based on defining a hierarchical color spectrum over the Agglomerative Clustering Tree. Thus the more similar two clusters are in color, the more closely they appear together in the Agglomerative Clustering tree. This allows us to evaluate our SOM results by coloring the SOM with the colors from the Agglomerative Clustering Tree hierarchy.

RESULTS

The test data-set consists of 172,000 cash register transactions of a home improvement store. From this data 2831 frequent item sets, 4782 association rules and 1311 hashed association rules are extracted. The new CMBP meets the intuitive expectations of a distance metric much better. Agglomerative Clustering was performed on a rules distance matrix of size 1,311x1,311. The number of different levels available for splitting the tree obtained is 289. The split that is used for defining the colors of the SOM is such that it results in 19 clusters at 208th split level and 715 clusters at 210th split level. The web-site *http://www.ece.utexas.edu/ gunjan/aclu/* has SOM results for various number of epochs. Clusters of relatively pure color clearly show the correlation between the clusters discovered by Dimensionless Agglomerative Chain Clustering algorithm and the SOM. Given that SOM is mapping 715 clusters onto a 10x10 grid, we expect on an average of 7 clusters to fall onto one point. Most of the overlaps should be with other clusters that are close to the given clusters. This should result in a color localization on the SOM. Since there is no ground truth, the visualization using SOM allows us to only get a good overall idea of the clusters. A good overlap between SOM and Agglomerative Clustering might imply that the clusters are more reliable. It also allows the user to inspect such clusters first. But the only way to see if the clusters are good is by printing out the text form of the Association Rules. The rules do appear to be correlated as opposed to randomly picked rules. Some examples are listed at our web-site at *http://www.ece.utexas.edu/ gunjan/aclu/clustertext* .

Conclusions and Future Work. A key reason for clustering rules is to obtain more concise and abstract descriptions of the data. We plan on merging rules of the same cluster into joined meta-rules. However, this is not a trivial problem. We are currently investigating the use of meta-data (such as product hierarchies) to support merging decisions. More powerful agglomerative clustering techniques that have other height definitions and splitting properties may yield better clusters. Trying out other distance measures for cluster centers is required for comparing results. Another idea is to modify the SOM to allow training on dimensionless distance data. Another alternative worth exploring is to represent data at multiple levels of product hierarchy before extracting, clustering and merging of association rules. This may provide more abstract descriptions of the data’s association rules that better capture customer buying behavior.

Acknowledgements. We thank Knowledge Discovery One (*http://www.kd1.com*) for providing us with real world data-sets and especially Mark Davis for his support and suggestions.

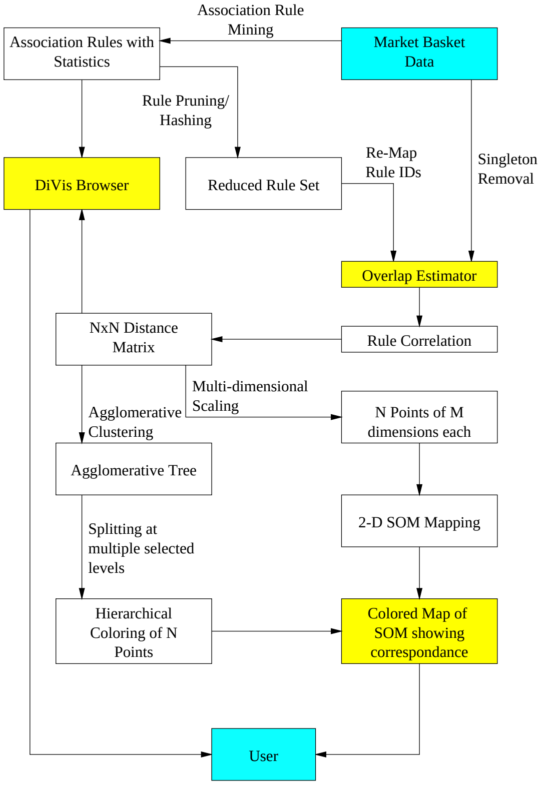

Figure 1: Flow diagram of distance computation and clustering process.

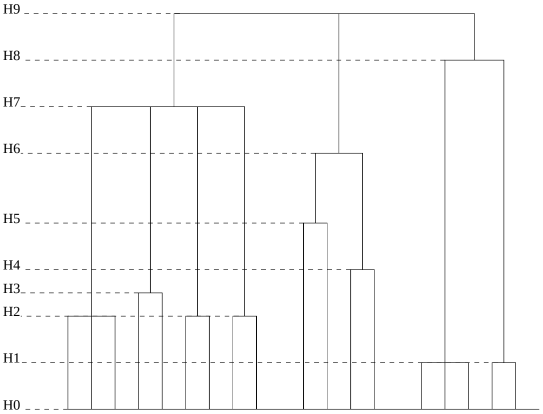

Figure 2: An example of an Agglomerative Chain Clustering Tree with nine unique splitting points each resulting in a different cluster set. For example splitting at H4 would give the following cluster sets (1,2,3), (4,5), (6,7), (8,9), (10), (11), (12,13), (14,15,16), (17,18). H4 is also the height of nodes 12 and 13. Nodes are numbered from left to right.

References

[1] R. Agrawal and R. Srikant. Fast algorithm for mining association rules in large databases. In *Proceedings 20th International Conference on Very Large Data Bases*, pages 478–499, September 1994.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD-93*, pages 207–216, May 1993.

[3] Ming-Syan Chen, Jiawei Han, and Philip S. Yu. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):866–883, December 1996.

[4] Brian Everitt. *Cluster Analysis, 2nd Edition*, chapter 3. Halsted Press, 1980.

[5] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proceedings 21th International Conference on Very Large Data Bases*, pages 420–431, September 1995.

[6] Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*, chapter 12. Prentice Hall, 1982.

[7] Jong Soo Park, Ming-Syan Chen, and Philip S. Yu. Using a hash-based method with transaction trimming for mining association rules. *IEEE Transactions on Knowledge and Data Engineering*, 9(5):813–825, September/October 1997.

[8] H. Toivonen, M. Klemettinen, P. Ronkainen, and H. Mannila. Pruning and grouping discovered association rules. In *MLnet Workshop on Statistics, Machine Learning and Discovery in Databases*, pages 47–52, April 1995.