Clustering Based On Association Rule Hypergraphs [*]

Eui-Hong (Sam) Han
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
han@cs.umn.edu

George Karypis
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
karypis@cs.umn.edu

Vipin Kumar
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
kumar@cs.umn.edu

Bamshad Mobasher
Department of Computer Science
University of Minnesota
Minneapolis, MN 55455
mobasher@cs.umn.edu

Abstract

Clustering in data mining is a discovery process that groups a set of data such that the intracluster similarity is maximized and the intercluster similarity is minimized. These discovered clusters are used to explain the characteristics of the data distribution. In this paper we propose a new methodology for clustering related items using association rules, and clustering related transactions using clusters of items. Our approach is linearly scalable with respect to the number of transactions. The frequent item-sets used to derive association rules are also used to group items into a hypergraph edge, and a hypergraph partitioning algorithm is used to find the clusters. Our experiments indicate that clustering using association rule hypergraphs holds great promise in several application domains. Our experiments with stock-market data and congressional voting data show that this clustering scheme is able to successfully group items that belong to the same group. Clustering of items can also be used to cluster the transactions containing these items. Our experiments with congressional voting data show that this method is quite effective in finding clusters of transactions that correspond to either democrat or republican voting patterns. Compared to the existing clustering algorithm *Autoclass*, our algorithm produced comparable quality clusters in the congressional voting data.

1 Introduction

Clustering in data mining is a discovery process that groups a set of data such that the intracluster similarity is maximized and the intercluster similarity is minimized [CHY96]. These discovered clusters are used to explain the characteristics of the data distribution. For example, in many business applications, clustering can be used to characterize different customer groups and allow businesses to offer customized solutions, or to predict customer buying patterns based on the profiles of the cluster to which they belong.

Given a database of transactions, there are two different kinds of clustering that are potentially useful. First is the clustering of items present in the transactions. In many domains, the ability to cluster related items is of particular importance. Consider for instance the items that are sold in a grocery store. If we can cluster these items into item-groups that are often sold together, we can then use this knowledge to perform effective shelf-space organization as well as target sales promotions. The association rules [AMS$^+$96, HF95, HKK97] discovered in transaction databases can be used as clusters of items. However, this will lead to a very large number of clusters, each containing a small number of items. Hence, the knowledge that can be extracted out of these clusters is relatively fine grain. Going back to our grocery store example, we may be able to find rules that say *pasta-brand-A* and *sauce-brand-B* implies *parmesan-cheese-brand-C*, for different brands of pasta, sauce, and cheese; indicating in each case that the group of three items are related. However, by only looking at each rule, we cannot discover the higher-level knowledge that pasta, sauce, and parmesan-cheese are related. Generalized association rules [SA95] can potentially be used to discover some of this higher-level knowledge by providing to the association-rule discovery algorithm a taxonomy on the items. However, a given taxonomy cannot capture all interesting relationships among the items. For example consider the following two rules discovered in a typical census database:

Rule 1: {*self-employed* and *white* and *college-degree*}
$\Rightarrow$ {*capital-gains-over-10k*}

[*] This work was supported by NSF grant ASC-9634719, Army Research Office contract DA/DAAH04-95-1-0538, Cray Research Inc. Fellowship, and IBM partnership award, the content of which does not necessarily reflect the policy of the government, and no official endorsement should be inferred. Access to computing facilities was provided by AHPCRC, Minnesota Supercomputer Institute, Cray Research Inc., and NSF grant CDA-9414015. See **http://www.cs.umn.edu/~han** for an extended version of this paper and other related papers.

Rule 2: {*manager* and *white* and *college-degree*}
⇒ {*capital-gains-over-10k* }

From these two rules we can infer that the items *self-employed*, *manager*, *white*, *college-degree*, and *capital-gains-over-10k* are related; however, no taxonomy can naturally capture this relationship.

The second type of clustering is the clustering of transactions. For example, in direct mail order business, a transaction corresponds to products purchased by one client during a set time period. The major marketing/advertising costs of many direct mail order businesses is the cost of their direct mailings (catalogs). Effective clustering of the customers can be used to perform direct marketing by sending customized catalogs to groups of customers. Similarly, in Web-based organizations, clustering of client transactions (access patterns recorded in server logs) can be used to dynamically present users with customized information or with targeted advertising [MJHS96].

Clustering of transactions has been studied in several areas including statistics [DJ80, Lee81, CS96], machine learning [SD90, Fis95], and data mining [NH94, CS96]. Most of the previous approaches in these areas are based on either probability or distance measure. If I is the number of different items, then each transaction is represented by a point in an I-dimensional space. Points (i.e., transactions) that are near-by in this I-dimensional space are clustered together. These clustering algorithms are able to effectively cluster transactions when the dimensionality of the space (i.e., the number of different items) is relatively small and most of the items are present in each transaction [DJ80, SD90, NH94]. However, these schemes fail to produce meaningful clusters, if the number of items is large and/or the fraction of the items present in each transaction is small. This type of data-sets are quite common in many data mining domains (e.g., market basket analysis), in which the number of different items is very large but each transaction has only few of these items. For example, a typical grocery store sells thousands of different items but each customer buys only a few of them (usually less than thirty).

In this paper we propose a new methodology for clustering related items using association rules, and clustering related transactions using clusters of items. Frequent item-sets, that are used to derive association rules, are sets of items of the database that meet a minimum support criterion [AMS$^+$96]. These frequent item-sets are used to group items into hypergraph edges, and a hypergraph partitioning algorithm [KAKS97] is used to find the clusters. The knowledge that is represented by clusters of related items can also be used to effectively cluster the actual transactions by looking at the clusters that these transactions belong to. For example, clusters of related items in a grocery store can be used to cluster together customers that are vegetarian, or that like certain ethnic foods. Similarly, clusters of items in census data can be used to cluster together individuals that belong to the same income levels.

Our algorithm is linearly scalable with respect to the number of transactions in the data base. The first step of our method, finding frequent item-sets, has been experimentally shown to be linearly scalable in [AMS$^+$96]. Once a hypergraph is constructed from the frequent item-sets, the remaining steps are independent of the number of transactions. The run time of the hypergraph partitioning depends only on the number of vertices and hyperedges. Hence, our algorithm is linearly scalable with respect to the number of transactions.

Our experiments with stock-market data shows that our item-clustering scheme is able to successfully generate clusters of companies that belong to the same industry group. Our experiments with congressional voting data show that this method is quite effective in clustering the votes into groups that are mostly supported by either republicans or democrats. When these clusters of votes is used to cluster transactions our scheme was quite effective in finding transaction clusters that correspond to either democrat or republican voting patterns. Compared to the existing clustering algorithm *Autoclass* [CS96], our algorithm produced comparable quality clusters in the congressional voting data.

The rest of this paper is organized as follows. Section 2 describes the proposed clustering techniques. Section 3 contains discussion and future works.

2 Clustering Based on Association Rules

The frequent item-sets [AMS$^+$96] found in transaction databases often reveal hidden relationships and correlations among data items. Association rule discovery in data mining has been used to discover such relationships in very large data repositories. Here we explore the feasibility and advantages of using the discovered association rules to cluster closely related data items into groups. Such clusters could be used in some domains to classify data items, to make predictions about similar data items, or to reduce the size of rule sets by eliminating those that are not interesting. We also, explore the use of discovered rules to cluster related transactions containing the data items. Clustering of transaction could be particularly useful in classifying users according to their behavior, such as the buying patterns of customers in supermarkets or client access patterns in a web-based environment [MJHS96].

2.1 Clustering of Data Items

Our method finds the clusters of items by constructing a weighted hypergraph from the frequent item-sets and partitioning this hypergraph using some similarity criteria related to the confidence of the association rules of these frequent item-sets. A hypergraph [Ber76] $H = (V, E)$ consists of a set of vertices (V) and a set of hyperedges (E). A hypergraph is an extension of a graph in the sense that each hyperedge can connect more than two vertices. In our model, the vertex set corresponds to the distinct items in the database and the hyperedges correspond to the frequent item-sets. For example, if {A B C} is a frequent item-set, then the hypergraph contains a hyperedge that connects A, B, and C. The weight of a hyperedge is determined by a function of the confidences of the all the association rules involving all the items of the hyperedge. For example, if $\{\text{A}\}\xrightarrow{0.8}\{\text{B C}\}$, $\{\text{A B}\}\xrightarrow{0.4}\{\text{C}\}$, $\{\text{A C}\}\xrightarrow{0.6}\{\text{B}\}$, $\{\text{B}\}\xrightarrow{0.4}\{\text{A C}\}$, $\{\text{B C}\}\xrightarrow{0.8}\{\text{A}\}$, and $\{\text{C}\}\xrightarrow{0.6}\{\text{A B}\}$ are all the possible association rules with confidence noted and the weighting function is the average

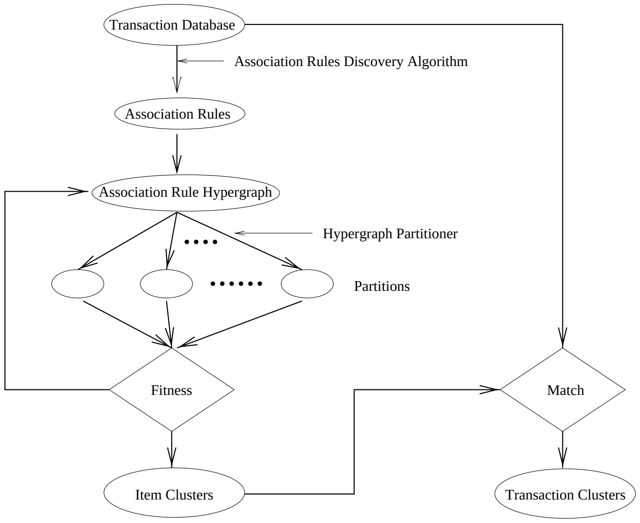

Figure 1: Clustering Based on Association Rule Hypergraphs

of the confidences, then the weight of the hyperedge connecting A,B, and C is 0.6.

In our preliminary experiments, we used HMETIS [KAKS97] to partition this hypergraph. HMETIS is a multi-level partitioning algorithm that has been shown to quickly produce high quality partitions (i.e., the sum of the weights that straddle partitions is minimized). HMETIS produces balanced k-way partitions, where k (i.e., the number of partitions) is specified by the user.

In order to evaluate the validity of our clustering algorithm, we conducted an experiment using stock activity data for S&P 500 companies to see if the proposed technique can correctly cluster various publicly traded companies according to their industry groups. One transaction in this experiment is one day's S&P500 stock price movements compared to the previous day's closing price. The items in a transaction are company names with either up or down indicator. We gathered 717 transactions of S&P500 stock data from Jan. 1994 to Oct. 1996. We found 20 clusters of which 16 were clean clusters (i.e., we could label the cluster with an industry based on the items in the cluster) and 4 were mixed clusters. Some of the clean clusters found from this data are shown in Table 1.

The clusters in Table 1 show stocks that move together. With the clusters found, we may focus on the association rules within the clusters to find out strongly connected items. We may also focus on the association rules that cross the clusters and that have high confidence. Such intercluster rules may be useful in defining cluster level relationships or characterizing higher level association rules [HF95, SA95].

2.2 Clustering of Transactions

Our method performs clustering of the transactions based on the clusters of items discovered using the techniques described in Section 2.1. The items in a cluster serve as the description of the cluster of which transactions belong to. So given the clusters of items and a transaction, one can determine the cluster that the transaction belongs to by calculating the score of each cluster based on the items in the transaction and the items in the clusters. A simple score function might be the ratio $|T \bigcap C_i|/|C_i|$, where T is the transaction and C_i is a cluster of items. A transaction belongs to the cluster which has the highest score with respect to that transaction.

We performed clustering of transactions on 1984 United States Congressional Voting Records Database provided by [MM96]. The data set includes 435 transactions each corresponding to one Congressman's votes on 16 key issues. We removed class values from each transaction, and we followed the steps specified in Section 2.1 to generate the clusters of items. These clusters are shown in Table 2.

Given the item clusters of Table 2, we used the simple score function described earlier to cluster the transactions. We then evaluated these transaction clusters to see to what extent they represent voting records of congressmen belonging to the same party. In particular, for each transaction cluster, we counted the number of democrats and republicans that belong to it. Table 3(a) shows that cluster 1 represents mostly republican congressmen, cluster 2 represents mostly democrat congressmen and cluster 3 also represents democrat congressmen. The characteristics of the clusters are more evident when we just include transactions to a cluster only if the matching score is 1.0 for the item cluster, i.e. the transaction contains all the items of the item cluster. Table 3(b) shows the result on this clustering.

We also compared our transaction clusters to the clusters found by *Autoclass* [CS96]. Table 4 shows the number of democrats and republicans that belong to the clusters found by *Autoclass*. *Autoclass* found 5 clusters where as our method found 3 clusters. Both results found 2 big clusters (clusters 1 and 2) that have one dominant party line. In addition to these two big clusters, our method found cluster 3 which is relatively small and has mostly democrat congressmen. On the other hand, *Autoclass* found cluster 3 which is evenly divided by democrat and republican congressmen, cluster 4 which has mostly democrat congressmen and cluster 5 which is relatively small and has more republican congressmen than democrat congressmen. These comparisons show that our method produced clusters that are comparable (if not better) to those of *Autoclass* in quality.

3 Discussion and Future Works

In this paper, we have presented preliminary results on new methods for clustering items and transactions based on the frequent item-sets discovered in the association mining process. Our algorithm is linearly scalable with respect to the number of transactions in the data base. The first step of our method, finding frequent item-sets, has been experimentally shown to be linearly scalable in [AMS$^+$96]. Once a hypergraph is constructed from the frequent item-sets, the remaining steps are independent of the number of transactions. The run time of HMETIS is $O(\log k(V+E))$, where k is the depth of bisection partition tree, V is the number of vertices and E is the number of hyperedges. The number of vertices in an association-rules hypergraph is the same as the number of distinct items in the data base which is fixed for

Discovered Clusters	Industry Group
APPLIED MATL↓, BAY NETWORK↓, 3 COM↓, CABLETRON SYS↓, CISCO↓, DSC COMM↓, HP↓, INTEL↓, LSI LOGIC↓, MICRON TECH↓, NATL SEMICONDUCT↓, ORACLE↓, SGI↓, SUN↓, TELLABS INC↓, TEXAS INST↓	Technology 1
APPLE COMP↓, AUTODESK↓, ADV MICRO DEVICE↓, ANDREW CORP↓, COMPUTER ASSOC↓, CIRC CITY STORES↓, COMPAQ↓, DEC↓, EMC CORP↓, GEN INSTRUMENT↓, MOTOROLA↓, MICROSOFT↓, SCIENTIFIC ATL↓	Technology 2
FANNIE MAE↓, FED HOME LOAN↓, MBNA CORP↓, MORGAN STANLEY↓	Financial
BAKER HUGHES↑, DRESSER INDS↑, HALLIBURTON HLD↑, LOUISIANA LAND↑, PHILLIPS PETRO↑, SCHLUMBERGER↑, UNOCAL↑	Oil
BARRICK GOLD↑, ECHO BAY MINES↑, HOMESTAKE MINING↑, NEWMONT MINING↑, PLACER DOME INC↑	Gold
ALCAN ALUMINUM↓, ASARCO INC↓, CYPRUS AMAX MIN↓, INLAND STEEL INC↓, INCO LTD↓, NUCOR CORP↓, PRAXAIR INC↓, REYNOLDS METALS↓, STONE CONTAINER↓, USX US STEEL↓	Metal

Table 1: Clustering of S&P 500 Stock Data

Cluster	Voting Items
1	adoption-of-the-budget-resolution-NO, physician-fee-freeze-YES, el-salvador-aid-YES religious-groups-in-schools-YES, anti-satellite-test-ban-NO, aid-to-nicaraguan-contras-NO, mx-missile-NO, education-spending-YES, crime-YES, duty-free-exports-NO
2	adoption-of-the-budget-resolution-YES, physician-fee-freeze-NO, el-salvador-aid-NO, anti-satellite-test-ban-YES, aid-to-nicaraguan-contras-YES, mx-missile-YES, education-spending-NO, crime-NO
3	handicapped-infants-NO, water-project-cost-sharing-YES, immigration-YES, synfuels-corporation-cutback-YES, superfund-right-to-sue-YES, export-administration-act-south-africa-NO

Table 2: Clustering of Congressional Voting Items

(a) Clustering of the full transaction

Class	Cluster 1	Cluster 2	Cluster 3
Republican	159	7	2
Democrat	38	216	13

(b) Clustering of transactions with matching score 1.0

Class	Cluster 1	Cluster 2	Cluster 3
Republican	86	0	1
Democrat	2	133	2

Table 3: Clustering of Congressional Voting Data Set

Class	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Republican	0	130	28	2	8
Democrat	163	22	30	48	4

Table 4: Clustering of Congressional Voting Data Set by *Autoclass*

a given data base. The number of hyperedges is the same as the number of frequent item-sets with support greater than the minimum support. We can control the number of hyperedges by changing the minimum support and/or lower limit on the weights of the hyperedges. In particular, we can raise the minimum support to decrease the number of frequent item-sets and we can include hyperedges in the hypergraph only if they have at least minimum weight. Hence, our algorithm is linearly scalable with respect to the number of transactions.

Our preliminary experiments indicate that clustering using association rule hypergraphs holds great promise in several application domains. Our experiments with the S&P 500 stock data show that the clustering of items based on frequent item-sets work quite well. The preliminary experiments with congressional voting data also show that the clustering of transactions based on clusters of items is quite effective. Compared to the existing clustering algorithm *Autoclass*, our algorithm produced comparable quality clusters in the congressional voting data.

We are investigating ways to improve the modeling such that the relationship among items are more accurately captured in hypergraph. More specifically, we are evaluating different weight functions for the hypergraph edges. We are also working on to improve HMETIS such that the number of partitions can be determined automatically by evaluating fitness of the partitions. In the clustering of transactions, we are evaluating different matching schemes that can be used in matching transactions against item clusters.

Our algorithm for constructing clusters of items and transactions can be used in many different ways. For example, discovered clusters can be used for improving accuracy, efficiency and robustness of classification algorithm, and for detecting deviations. It also appears possible to find interesting high confidence association rules that have very low support using discovered clusters.

References

[AMS$^+$96] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[Ber76] C. Berge. *Graphs and Hypergraphs.* American Elsevier, 1976.

[CHY96] M.S. Chen, J. Han, and P.S. Yu. Data mining: An overview from database perspective. *IEEE Transactions on Knowledge and Data Eng.*, 8(6):866–883, December 1996.

[CS96] P. Cheeseman and J. Stutz. Baysian classification (autoclass): Theory and results. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press, 1996.

[DJ80] R. Dubes and A.K. Jain. Clustering methodologies in exploratory data analysis. In M.C. Yovits, editor, *Advances in Computers*. Academic Press Inc., New York, 1980.

[Fis95] D. Fisher. Optimization and simplification of hierarchical clusterings. In *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pages 118–123, Montreal, Quebec, 1995.

[HF95] J. Han and Y. Fu. Discovery of multiple–level association rules from large databases. In *Proc. of the 21st VLDB Conference*, Zurich, Switzerland, 1995.

[HKK97] E.H. Han, G. Karypis, and V. Kumar. Scalable parallel data mining for association rules. In *Proc. of 1997 ACM-SIGMOD Int. Conf. on Management of Data*, page to appear, Tucson, Arizona, 1997.

[KAKS96] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Hypergraph partitioning: Applications in VLSI domain. Technical Report TR-96-060, Department of Computer Science, University of Minnesota, Minneapolis, 1996.

[KAKS97] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in VLSI domain. In *Proceedings ACM/IEEE Design Automation Conference*, 1997.

[Lee81] R.C.T. Lee. Clustering analysis and its applications. In J.T. Toum, editor, *Advances in Information Systems Science*. Plenum Press, New York, 1981.

[MJHS96] B. Mobasher, N. Jain, E.H. Han, and J. Srivastava. Web mining: Pattern discovery from world wide web transactions. Technical Report TR-96-050, Department of Computer Science, University of Minnesota, M inneapolis, 1996.

[MM96] C.J. Merz and P.M. Murphy. UCI repository of machine learning databases. In *http://www.ics.uci.edu/ mlearn/MLRepository.html*, 1996.

[NH94] R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. of the 20th VLDB Conference*, pages 144–155, Santiago, Chile, 1994.

[SA95] R. Srikant and R. Agrawal. Mining generalized association rules. In *Proc. of the 21st VLDB Conference*, pages 407–419, Zurich, Switzerland, 1995.

[SD90] J.W. Shavlik and T.G. Dietterich. *Readings in Machine Learning.* Morgan-Kaufman, 1990.