Open Problems in Relational Data Clustering

Adam Anthony AANTHON2@UMBC.EDU
Marie desJardins MARIEDJ@CS.UMBC.EDU
University of Maryland Baltimore County,
1000 Hilltop Circle, Baltimore, MD 21250

Abstract

Data clustering is the task of detecting patterns in a set of data. Most algorithms take non-relational data as input and are sometimes unable to find significant patterns. Many data sets can include relational information, as well as independent object attributes. We believe that clustering with relational data will help find significant patterns where non-relational algorithms fail. This paper discusses two open problems in relational data clustering: clustering heterogeneous data, and relation selection or extraction. Potential methods for addressing the problems are presented.

1. Introduction

Data clustering is the process of statistically grouping data objects with other similar objects. The motive for performing this task is to reveal hidden patterns in large data sets. With a great deal of prior research, computer scientists have the ability to find significant patterns in many different kinds of data such as numerical, text, or visual data. For many applications, current methods are sufficient. But in certain areas, clustering can fail to produce significant results.

In such situations, the incorporation of relational information may assist in producing a significant clustering for difficult data sets. *Relational data clustering* (which we will also refer to as relational clustering) is the design and use of clustering algorithms that use the relational structure of data instances to determine a set of clusters. This paper provides a brief overview of relational clustering research and proposes some open problems in relational clustering: namely, clustering heterogeneous data and relation selection/extraction in support of relational learning.

Presented at the ICML Workshop on Open Problems in Statistical Relational Learning, Pittsburgh, PA, 2006.

2. Related Work

There has been some recent research in relational clustering. Yin et al. (2005) describe a new technique for choosing descriptive, cross-relational features in a data set to produce a single object type that is a compound of features from other objects. When the feature selection algorithm halts, the CLARANS algorithm (Ng & Han, 2002) is used to cluster the set of compound objects. This algorithm is relational in the sense that it takes relational data as input. However, it does not use the significance of relations to assist the clustering, since it compounds the data into a link-less form before the clustering step.

Other relational clustering algorithms do use relations to assist the clustering process. Neville et al. (2003) adapted graph cutting algorithms to cluster only links, only attributes, or both, using a hybrid approach. Taskar et al. (2001) researched the use of probabilistic relational models and the EM algorithm to classify and cluster data. Han et al. (1997) developed hypergraph representations of data, and use the HMETIS (Karypis et al., 1999) system to partition the graph into a clustering. Finally, Bhattacharya and Getoor (2005) used relational clustering for a novel method of entity resolution in graphs. Their method involves a metric calculated as the linear combination of graph similarity and attribute similarity between two references.

3. Clustering Heterogeneous Data

Most non-relational machine learning methods require homogeneous data because it is difficult to compare heterogeneous data objects based on feature vectors. With added relational information, possibilities exist for defining the similarity between different object types.

A naive method is to consider two objects to be similar if they have some subset (based on a threshold) of linked objects in common. A way to think of this is to consider two objects similar if their *relational edges* are similar. This notion is a good starting point, but several complications exist. The most obvious complication is varying degree within a relational data set. An extreme case is where one object has a relation connecting it to every other object in the set. Every other object's relation set is a subset of this one object's edge set. With a naive similarity metric, this object could be considered similar to every other object in the set when it is actually very different.

The above naive similarity metric can be refined by adding constraints. For example, Adding the constraint that two objects must have a majority of their edge set in common would prevent the high degree object from being considered similar to every other object. However, care must now be taken in quantifying this similarity measure. More work can be done in this area of formalizing and refining this method, but perhaps other possibilities exist.

One possibility we have been thinking of is the idea of an *inter-cluster relation signature*. First, cluster one type of data objects based on their feature vectors using some traditional clustering method that is suitable to the data. Then, for each object of a different type, construct the inter-cluster relation signature as an n-dimensional vector i where n is the number of clusters produced in the initial clustering phase. Each dimension i_k is equal to the number of edges that an object has connecting itself to an object in cluster k. Note that this method may also be used for a novel homogeneous clustering method where the second phase is run for the same set of objects used for the first phase.

4. Relation Selection and Relation Extraction

Creating new methods for comparing heterogeneous objects could be important for pattern detection in many different kinds of sets. Another important open problem that is analogous to feature selection is relation selection. It is intuitive that, just as some features are not helpful for clustering a data set, some relations might provide little information for a relational clustering algorithm. To the authors' knowledge, no methods of relation selection exist.

Relation selection is simplified if the relation set is defined carefully. We can consider the *relation space* to be a set of k relational graphs $RG = \{G_1, G_2, ..., G_K\}$. Each relational graph can be viewed as $G_i = \{O_i, R_i\}$, where each element in O_i specifies a unique object in the feature space and R_i is a set of edges connecting objects, implying that connected objects are related in a particular way. For further simplification, assume that each graph only encodes a single type of relation, so to encode multiple relations, multiple graphs must be specified. The relations should be specified separately so that they can be analyzed separately. It is common in related research to specify all relations specified in a single graph. This alternative definition is the graph $G_{ALL} = \{O_{ALL}, R_{ALL}\}$ where O_{ALL} is the set of all unique objects found in the feature space, and $R_{ALL} = \{R_1 \cup R_2 \cup ... \cup R_K\}$

With this definition, relation selection can become an interesting problem. Applications as simple as using trivial isomorphism tests can help us reduce the relation space significantly. Additionally, subgraph isomorphism could be another application of graph theory that could help to remove redundant relation sets.

In addition to structural comparison, more work needs to be done to determine the amount of information that a relational set provides about a data set. For example, fully and sparsely connected sets could both be poor relational data sets for clustering because they do not suggest any natural grouping of the data. If such measures can be determined, they can be used to remove low-quality relations, to prevent clustering from being harmed by the poor information.

Finally, for relation extraction, there are two possible directions of research. One direction is the use of the feature space to extract a relation space. In the work described above, Han et al. (1997) generated their initial hypergraph using an association rule algorithm. Another possibility is to develop methods for combining two relation graphs into a stronger relation graph.

5. Conclusion

This paper listed two important problems in relational data clustering: heterogeneous data comparison and relation selection and extraction. Research in new methods for comparing two objects based on relational structure will help us to better detect patterns in heterogeneous data, but may also help us with homogeneous data sets as in the example of the inter-cluster relation signature. Relation selection and extraction will help us to reduce complex relational sets into a form that is more manageable. Selection and extraction will also help us to explore dimensions of the relational structure that are most likely to produce a strong natural grouping. Progress in either of these open areas will lead to progress in developing accurate relational clustering algorithms.

References

Bhattacharya, I., & Getoor, L. (2005). *Entity resolution in graph data* (Technical Report CS-TR-4758). University of Maryland.

Han, E.-H., Karypis, G., & Kumar, V. (1997). Hypergraph based clustering in high-dimensional data sets: A summary of results. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, *21*, 15–22.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, *32*, 68–75.

Neville, J., Adler, M., & Jensen, D. (2003). Clustering relational data using attribute and link information. *Proceedings of the Text Mining and Link Analysis Workshop*.

Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, *14*, 1003–1016.

Taskar, B., Segal, E., & Koller, D. (2001). Probabilistic classification and clustering in relational data. *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence* (pp. 870–878). Seattle, US.

Yin, X., Han, J., & Yu, P. S. (2005). Cross-relational clustering with user's guidance. *KDD '05: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 344–353). New York, NY, USA: ACM Press.