Pattern Preserving Clustering

Hui Xiong
Rutgers University, USA

Michael Steinbach
University of Minnesota, USA

Pang-Ning Tan
Michigan State University, USA

Vipin Kumar
University of Minnesota, USA

Wenjun Zhou
Rutgers University, USA

INTRODUCTION

Clustering and association analysis are important techniques for analyzing data. Cluster analysis (Jain & Dubes, 1988) provides insight into the data by dividing objects into groups (clusters), such that objects in a cluster are more similar to each other than to objects in other clusters. Association analysis (Agrawal, Imielinski & Swami, 1993), on the other hand, provides insight into the data by finding a large number of strong patterns -- frequent itemsets and other patterns derived from them -- in the data set. Indeed, both clustering and association analysis are concerned with finding groups of *strongly related* objects, although at different levels. Association analysis finds strongly related objects on a local level, i.e., with respect to a subset of attributes, while cluster analysis finds strongly related objects on a global level, i.e., by using all of the attributes to compute similarity values.

Recently, Xiong, Tan & Kumar (2003) have defined a new pattern for association analysis -- the hyperclique pattern -- which demonstrates a particularly strong connection between the overall similarity of all objects and the itemsets (local pattern) in which they are involved. The hyperclique pattern possesses a *high affinity property*: the objects in a hyperclique pattern have a guaranteed level of global pairwise similarity to one another as measured by the cosine similarity (uncentered Pearson correlation coefficient). Since clustering depends on similarity, it seems reasonable that the hyperclique pattern should have some connection to clustering.

Ironically, we found that hyperclique patterns are mostly destroyed by standard clustering techniques, i.e., standard clustering schemes do not preserve the hyperclique patterns, but rather, the objects comprising them are typically split among different clusters. To understand why this is not desirable, consider a set of hyperclique patterns for documents. The high affinity property of hyperclique patterns requires that these documents must be similar to one another; the stronger the hyperclique, the more similar the documents. Thus, for strong patterns, it would seem desirable (from a clustering viewpoint) that documents in the same pattern end up in the same cluster in many or most cases. As mentioned, however, this is not what happens for traditional clustering algorithms. This is not surprising since traditional clustering algorithms have no built in knowledge of these patterns and may often have goals that are in conflict with preserving patterns, e.g., minimize the distances of points from their closest cluster centroids.

More generally, the breaking of these patterns is also undesirable from an application point of view. Specifically, in many application domains, there are fundamental patterns that dominate the description and analysis of data within that area, e.g., in text mining, collections of words that form a topic, and in biological sciences, a set of proteins that form a functional module (Xiong et al. 2005). If these patterns are not

respected, then the value of a data analysis is greatly diminished for end users. If our interest is in patterns, such as hyperclique patterns, then we need a clustering approach that preserves these patterns, i.e., puts the objects of these patterns in the same cluster. Otherwise, the resulting clusters will be harder to understand since they must be interpreted solely in terms of objects instead of well-understood patterns.

There are two important considerations for developing a pattern persevering clustering approach. First, in any clustering scheme, we must take as our starting point the sets of objects that comprise the patterns of interest. If the objects of a pattern of interest are not together when we start the clustering process, they will often not be put together during clustering, since this is not the goal of the clustering algorithm. Second, if we start with the sets of objects that comprise the patterns of interest, we must not do anything in the clustering process to breakup these sets. Therefore, for pattern preserving clustering, the pattern must become the basic *object* of the clustering process. In theory, we could then use any clustering technique, although modifications would be needed to use sets of objects instead of objects as the basic starting point. Here, we use hyperclique patterns as our patterns of interest, since they have some advantages with respect to frequent itemsets for use in pattern preserving clustering: hypercliques have the high affinity property, which guarantees that keeping objects together makes sense, and finding hypercliques is computationally much easier than finding frequent itemsets. Finally, hyperclique patterns, if preserved, can help cluster interpretation.

BACKGROUND

Cluster analysis has been the focus of considerable work, both within data mining and in other fields such as statistics, psychology, and pattern recognition. Several recent surveys may be found in Berkhin (2002), Han, Kamber & Tung (2001), and Jain, Murty & Flynn (1999), while more extensive discussions of clustering are provided by the following books (Anderberg, 1973; Jain & Dubes, 1988; Kaufman & Rousseeuw, 1990).

While there are innumerable clustering algorithms, almost all of them can be classified as being either *partitional*, i.e., producing an un-nested set of clusters that partitions the objects in a data set into disjoint groups, or *hierarchical*, i.e., producing a nested sequence of partitions, with a single, all-inclusive cluster at the top and singleton clusters of individual points at the bottom. While this standard description of hierarchical versus partitional clustering assumes that each object belongs to a single cluster (a single cluster within one level, for hierarchical clustering), this requirement can be relaxed to allow clusters to overlap.

Perhaps the best known and widely used partitional clustering technique is K-means (MacQueen, 1999), which aims to cluster a dataset into K clusters---K specified by the user---so as to minimize the sum of the squared distances of points from their closest cluster centroid. (A cluster centroid is the mean of the points in the cluster.) K-means is simple and computationally efficient, and a modification of it, bisecting K-means (Steinbach, Karypis & Kumar, 2000), can also be used for hierarchical clustering.

Traditional hierarchical clustering approaches (Jain & Dubes, 1988) build a hierarchical clustering in an *agglomerative* manner by starting with individual points or objects as clusters, and then successively combining the two most similar clusters, where the similarity of two clusters can be defined in different ways and is what distinguishes one agglomerative hierarchical technique from another. These techniques have been used with good success for clustering documents and other types of data. In particular, the agglomerative clustering technique known as Group Average or UPGMA, which defines cluster similarity in terms of the average pairwise similarity between the objects in the two clusters, is widely used because it is more robust than many other agglomerative clustering approaches.

As far as we know, there are no other clustering methods based on the idea of preserving patterns. However, we mention two other types of clustering approaches that share some similarity with what we are doing here: constrained clustering and frequent item set based clustering. Constrained clustering (Wagstaff and Cardie, 2000, Tung, Ng, Lakshmanan & Han, 2001, Davidson and Ravi, 2005) is based on the idea of using standard clustering approaches, but restricting the clustering process. Our approach can be viewed as constraining certain objects to stay together during the clustering process. However, our constraints are automatically enforced by putting objects in hypercliques together, before the clustering process begins, and thus, the general framework for constrained clustering is not necessary for pattern preserving clustering.