Techniques for Weighted Clustering Ensembles

Carlotta Domeniconi
George Mason University, USA

Muna Al-Razgan
George Mason University, USA

INTRODUCTION

In an effort to achieve improved classifier accuracy, extensive research has been conducted in classifier ensembles. Very recently, cluster ensembles have emerged. It is well known that off-the-shelf clustering methods may discover different structures in a given set of data. This is because each clustering algorithm has its own bias resulting from the optimization of different criteria. Furthermore, there is no ground truth against which the clustering result can be validated. Thus, no cross-validation technique can be carried out to tune input parameters involved in the clustering process. As a consequence, the user is not equipped with any guidelines for choosing the proper clustering method for a given dataset.

Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature. Cluster ensembles can provide more robust and stable solutions by leveraging the consensus across multiple clustering results, while averaging out emergent spurious structures that arise due to the various biases to which each participating algorithm is tuned.

In this chapter, we discuss the problem of combining multiple *weighted clusters*, discovered by a locally adaptive algorithm (Domeniconi, Papadopoulos, Gunopulos, & Ma, 2004) which detects clusters in different subspaces of the input space. We believe that our approach is the first attempt to design a cluster ensemble for subspace clustering (Al-Razgan & Domeniconi, 2006).

Recently, several subspace clustering methods have been proposed (Parsons, Haque, & Liu, 2004). They all attempt to dodge the curse of dimensionality which affects any algorithm in high dimensional spaces. In high dimensional spaces, it is highly likely that, for any given pair of points within the same cluster, there exist at least a few dimensions on which the points are far apart from each other. As a consequence, distance functions that equally use all input features may not be effective.

Furthermore, several clusters may exist in different subspaces comprised of different combinations of features. In many real-world problems, some points are correlated with respect to a given set of dimensions, while others are correlated with respect to different dimensions. Each dimension could be relevant to at least one of the clusters.

Global dimensionality reduction techniques are unable to capture local correlations of data. Thus, a proper feature selection procedure should operate locally in input space. Local feature selection allows one to embed different distance measures in different regions of the input space; such distance metrics reflect local correlations of data. In (Domeniconi, Papadopoulos, Gunopulos, & Ma, 2004) we proposed a *soft* feature selection procedure (called LAC) that assigns weights to features according to the local correlations of data along each dimension. Dimensions along which data are loosely correlated receive a small weight, which has the effect of elongating distances along that dimension. Features along which data are strongly correlated receive a large weight, which has the effect of constricting distances along that dimension. Thus the learned weights perform a directional local reshaping of distances which allows a better separation of clusters, and therefore the discovery of different patterns in different subspaces of the original input space.

The clustering result of LAC depends on two input parameters. The first one is common to all clustering algorithms: the number of clusters k to be discovered in the data. The second one (called h) controls the strength of the incentive to cluster on more features. The setting of h is particularly difficult, since no domain knowledge

for its tuning is likely to be available. Thus, it would be convenient if the clustering process automatically determined the relevant subspaces.

In this chapter we discuss two cluster ensemble techniques for the LAC algorithm. We focus on setting the parameter h and assume that the number of clusters k is fixed. We leverage the diversity of the clusterings produced by LAC when different values of h are used, in order to generate a consensus clustering that is superior to the participating ones.

BACKGROUND

In many domains it has been shown that a classifier ensemble is often more accurate than any of the single components. This result has recently initiated further investigation in ensemble methods for clustering. In (Fred & Jain, 2002) the authors combine different clusterings obtained via the *k*-means algorithm. The clusterings produced by *k*-means are mapped into a co-association matrix, which measures the similarity between the samples. Kuncheva et al. (Kuncheva & Hadjitodorov, 2004) extend the work in (Fred & Jain, 2002) by choosing at random the number of clusters for each ensemble member. The authors in (Zeng, Tang, Garcia-Frias, & Gao, 2002) introduce a meta-clustering procedure: first, each clustering is mapped into a distance matrix; second, the multiple distance matrices are combined, and a hierarchical clustering method is introduced to compute a consensus clustering. In (Hu, 2004) the authors propose a similar approach, where a graph-based partitioning algorithm is used to generate the combined clustering. Ayad et al. (Ayad & Kamel, 2003) propose a graph approach where data points correspond to vertices, and an edge exists between two vertices when the associated points share a specific number of nearest neighbors. In (Fern & Brodley, 2003) the authors combine random projection with a cluster ensemble. EM is used as clustering algorithm, and an agglomerative approach is utilized to produce the final clustering. Greene et al. (Greene, Tsymbal, Bolshakova, & Cunningham, 2004) apply an ensemble technique to medical diagnostic datasets. The authors focus on different generation and integration techniques for input clusterings to the ensemble. *K-means*, *K*-medoids and fast *weak clustering* are used as generation strategies. The diverse clusterings are aggregated into a co-occurrence matrix. Hierarchical schemes are then applied to compute the consensus clustering. Greene's approach follows closely Fred and Jain's approach (Fred & Jain, 2002). However, they differ in the generation strategies. Similarly, in (Boulis & Ostendorf, 2004) the association between different clusterings produced by various algorithms is investigated. Techniques based on constrained and unconstrained clustering and on SVD are considered. (Gionis, Mannila, & Tsaparas, 2005) approach finds an ensemble clustering that agrees as much as possible with the given clusterings. The proposed technique does not require the number of clusters as an input parameter, and handles missing data.

In (Strehl & Ghosh, 2003) the authors propose a consensus function aimed at maximizing the normalized mutual information of the combined clustering with the input ones. Three heuristics are introduced: Cluster-based Similarity Partitioning Algorithm (CSPA), HyperGraph Partitioning Algorithm (HGPA), and Meta-Clustering Algorithm (MCLA). All three algorithms transform the set of clusterings into a hypergraph representation.

In CSPA, a binary similarity matrix is constructed for each input clustering. An entry-wise average of all the matrices gives an overall similarity matrix S. S is utilized to recluster the data using a graph-partitioning based approach. HGPA constructs a hypergraph in which each hyperedge represents a cluster of an input clustering. The algorithm seeks a partitioning of the hypergraph by cutting a minimal number of hyperedges. The partition gives k unconnected components of approximately the same size. MCLA is based on the clustering of clusters. It provides object-wise confidence estimates of cluster membership. Hyperedges are grouped, and each data point is assigned to the collapsed hyperedge in which is participates most strongly.

MAIN FOCUS

In the following we introduce two consensus functions to identify an emergent clustering that arises from multiple clustering results. We reduce the problem of defining a consensus function to a graph partitioning problem (Dhillon, 2001; Fern & Brodley, 2004; Strehl & Ghosh, 2003). In fact, the *weighted clusters* computed by the LAC algorithm offer a natural way to define a similarity measure to be integrated as weights associated to