International Journal of Image Processing and Vision Science

Volume 1 | Issue 4 Article 12

October 2013

A META CLUSTERING APPROACH FOR ENSEMBLE PROBLEM

DIVYA D J
Dept of Computer Science Engineering, B.T.L. Institute of Technology, Bangalore, India,
divyram@gmail.com

GAYATHRI DEVI B
Dept of Electrical Engineering, P E S Institute of Technology, Bangalore, India, gayinpes@gmail.com

A META CLUSTERING APPROACH FOR ENSEMBLE PROBLEM

DIVYA D J[1] & GAYATHRI DEVI B[2]

[1]Dept of Computer Science Engineering, B.T.L. Institute of Technology, Bangalore, India
[2]Dept of Electrical Engineering, P E S Institute of Technology, Bangalore, India
Email:divyram@gmail.com gayinpes@gmail.com

Abstract— A critical problem in cluster ensemble research is how to combine multiple clustering to yield a superior clustering result. Leveraging advanced graph partitioning techniques, we solve this problem by reducing it to a graph partitioning problem. We introduce a new reduction method that constructs a bipartite graph from a given cluster ensemble. The resulting graph models both instances and clusters of the ensemble simultaneously as vertices in the graph. Our approach retains all of the information provided by a given ensemble, allowing the similarity among instances and the similarity among clusters to be considered collectively in forming the clustering. Further, the resulting graph partitioning problem can be solved efficiently. We empirically evaluate the proposed approach against two commonly used graph formulations and show that it is more robust and achieves comparable or better performance in comparison to its competitors.

Keywords*-meta clustering approach; graph partitioning problem; ensemble problem; meta clustering*

I. INTRODUCTION

Clustering is to group analogous elements in a data set in accordance with its similarity such that elements in each cluster are similar while elements from different clusters are dissimilar. It doesn't require the class label information about the data set because it is inherently a data,driven approach. So, the most interesting and well developed method of manipulating and cleaning spatial data in order to prepare it for spatial data mining analysis is by clustering that has been recognized as a primary data mining method for knowledge discovery in spatial database Clustering fusion is the integration of results from various clustering algorithms using a consensus function to yield stable results. Clustering fusion approaches are receiving increasing attention for their capability of improving clustering performance. At present, the usual operation mechanism for clustering fusion is the "combining" of clusterer outputs. One tool for such combining or consolidation of results Clustering for unsupervised data exploration and analysis has been investigated for decades in the statistics, data mining, and machine learning communities. A recent advance of clustering techniques is the development of cluster ensemble or consensus clustering techniques which seek to improve clustering performance by generating multiple partitions of a given data set and then combining them to form a (presumably superior) clustering solution. Such techniques have been shown to provide a generic tool for improving the performance of basic clustering algorithms. Cluster ensembles can be generated in different ways. The resulting ensembles may differ and the same approach for solving the ensemble problems may perform differently accordingly. It is thus important for our experiments to consider different ways to generate cluster ensembles. Our experiments use two approaches, random subsampling and random projection [Fern & Brodley, 2003], to generate the ensembles. Note that for both approaches, K,means is used as the base clustering algorithm and the number K is pre,specified for each data set and remains the same for all clustering runs. Note that we also examined a third approach, randomly restarting K means, and it produced similar results to those of random subsampling. So we omit these results in the discussion of our experiments. Random projection should be diverse because it provides the base learner with different views of the data. On the other hand, we expect the quality of the clusterings produced by random subsampling to be higher because it provides the base learner with more complete information of the data.

A. Graph Partitioning Algorithms

Our goal is to evaluate different graph formulation approaches. To reduce the infuence of any chosen graph partitioning algorithm on our evaluation, we use two well, known graph partitioning algorithms that differ with respect to their search for the best partition.

B. Spectral Graph Partitioning

Spectral graph partitioning is a well studied area with many successful applications. We choose a popular multi, way spectral graph partitioning algorithm proposed by Ng et al. [Alexander Strehl and J. Ghosh,2002], which seeks to optimize the normalized cut criterion [Shi & Malik, 2000]. We refer to this algorithm as SPEC. SPEC can be simply described as follows. Given a graph G = (V;W), it computes the degree matrix D, which is a diagonal matrix such that D(i; i) =Pj W(i; j). Based on D, it

then computes a normalized weight matrix K largest eigenvectors u1; u2; _ _
_ ; uK to form matrix U = [u1; _ _ _ ; uK]. The rows of U are then normalized to have unit length. Treating the rows of U as K dimensional embeddings of the vertices of the graph, SPEC produces the clustering solution by clustering the embedded points using K,means. Intuitively, SPEC embeds the vertices of a graph onto a K,dimensional space and then performs clustering in the K,dimensional space. For graphs generated by IBGF and CBGF, the clusters and instances are embedded and clustered separately. Interestingly, for HBGF, the clusters and instances are simultaneously embedded onto the same space and clustered together. Here we argue that this potential advantages over IBGF and CBGF. Compared to IBGF, the inclusion of the cluster vertices may help define the structure of the data and make it easier for K,means to and the structure in the K, dimensional space. In comparison to CBGF, it is expected to be more robust because even when the cluster vertices are not well structured, possibly due to the lack of a correspondence structure in the clusters, K,means can still perform reasonably well using the instance vertices.

II. EXISTING GRAPH FORMULATIONS FOR CLUSTER ENSEMBLES

This section introduces two existing techniques proposed for formulating graphs from cluster ensembles. We rename these two techniques as instance,based and cluster,based approaches to characterize the differences between them.

A. Instance-Based Graph Formulation

Instance,Based Graph Formulation (IBGF) constructs a graph to model the pair wise relationships among instances of the data set X. Recall that the commonly used agglomerative approach generates a similarity matrix from the cluster ensemble and then performs agglomerative clustering using the similarity matrix. IBGF uses this matrix in conjunction with graph partitioning. Below we formally
describe IBGF. Given a cluster ensemble IBGF constructs a fully connected graph G = (V;W), where V is a set of n vertices, each representing an instance of X.W is a similarity matrix and W(i; j) =1R PRr=1 I(gr(Xi) = gr(Xj)), where I(_) is an indicator n function that returns 1 if the argument is true and 0 otherwise; gr(_) takes an instance and returns the cluster that it belongs to in Cr. W(i; j) measures how frequently the
instances i and j are clustered together in the given ensemble. In recent work this similarity measure has been shown to give satisfactory performance in domains where a good similarity 1. Note that in some cases this bias maybe unwarranted (or distance) metric is otherwise hard to . Once a graph is constructed, one can solve the graph partitioning problem using any graph partitioning technique and the resulting partition can be directly output as the clustering solution. Note that IBGF constructs a fully connected graph, resulting in a graph partitioning problem of size n2, where n is the number of instances. Depending on the algorithm used to partition the graph, the computational complexity of IBGF may vary. But generally it is computationally more expensive than the cluster based approach and our proposed approach, which is a key disadvantage of IBGF.

B. *Cluster-Based Graph Formulation*

Note that clusters formed in different clusterings may contain the same set of instances or largely overlap with each other. Such clusters are considered to be corresponding (similar) to one another. Cluster,Based Graph Formulation (CBGF) constructs a graph to model the correspondence (similarity) relationship among different clusters in a given ensemble and partitions the graph into groups so that the clusters of the same group correspond to one another. Once a partition of the clusters is obtained, we can produce a clustering of instances as follows. First we consider each group of clusters as a metacluster. For each clustering, an instance is considered to be associated with a metacluster if it contains the cluster to which the instance belongs. Note that an instance may be associated with different meta clusters in different runs, we assign an instance to the metacluster with which it is most frequently associated. Ties are broken randomly. The basic assumption of CBGF is the existence of a correspondence structure among different clusters formed in the ensemble. This poses a potential problem in cases where no such correspondence structure exists, this approach may fail to provide satisfactory performance. The advantage of CBGF is that is computationally efficient. The size of the resulting graph partitioning problem is t2, where t is the total number of clusters in the ensemble. This is significantly smaller than the n2 of IBGF, assuming .hypergraph based approach, which models clusters as hyperedges and instances as vertices in a hypergraph and uses a hypergraph artitioning algorithm to produce a partition. onceptually, this approach forms a different type of graph and has the limitation that it can not model soft clustering. Practically, we observed that it performed worse than IBGF and CBGF on our datasets.

C. Cluster-based Similarity Partitioning Algorithm (CSPA)

Based on a coarse resolution viewpoint that two objects have a similarity of 1 if they are in the same cluster and a similarity of 0 otherwise, a binary

similarity matrix can be readily created for each clustering. The entry,wise average of r such matrices representing the r sets of groupings yields an overall similarity matrix S with a resolution. The entries of S denote the fraction of clusterings in which two objects are in the same cluster, and can be computed in one sparse matrix multiplication S =rHHy. the generation of the cluster,based similarity matrix Now, we can use the similarity matrix to recluster the objects using any
reasonable similarity,based clustering algorithm. We hose to partition the induced similarity graph (vertex = object, edge weight = similarity) using METIS [Karypis and Kumar, 1998] because of its robust and scalable properties. CSPA is the simplest and most obvious heuristic, but its computational and storage complexity are both quadratic in n, as opposed to the next two approaches that are near linear in n.

D. HyperGraph-Partitioning Algorithm (HGPA)

The second algorithm is a direct approach to cluster ensembles that re,partitions the data using the given clusters as indications of strong bonds. The cluster ensemble problem is formulated [Kunal Punera, Joydeep Ghosh] as partitioning the hypergraph by cutting a minimal number of hyperedges. We call this approach the hypergraph, partitioning algorithm (HGPA). All hyperedges are considered to have the same weight. Also, all vertices re equally weighted. Note that this includes n`,way relationship information, while CSPA only considers pairwise relationships.

E. Representing Sets of Clusterings as a Hypergraph

The first step for both of our proposed consensus functions is to transform the given cluster label vectors into a suitable hypergraph representation. In this subsection, we describe how any set of clusterings can be mapped to a hypergraph. A hypergraph consists of vertices and
hyperedges. An edge in a regular graph connects xactly
two vertices. A hyperedge is a generalization of an edge in that it can connect any set of vertices. For each label vector _(q) 2 Nn, we construct the binary membership indicator matrix H(q), with a column for each cluster (now represented as a hyperedge) All entries of a row in the binary membership indicator matrix H(q) add to 1, if the row corresponds to an object with known label. Rows for objects with unknown label are all zero.The concatenated block matrix H = H(1;:::;r) = (H(1) : : : H(r)) defines the adjacency matrix of a hypergraph with n vertices and Pr q=1 k(q) hyperedges. Each column vector ha specifies a hyperedge ha, where 1 indicates that the vertex corresponding to the row is part of that hyperedge and 0 indicates that it is not. Thus, we have mapped each cluster to a hyperedge
and the set of clusterings to a hypergraph

III. META CLUSTERING ALGORITHM

We introduce the algorithm to solve the cluster ensemble problem. The Meta,Clustering Algorithm (MCLA) is based on clustering clusters. It also yields object,wi se confidence estimates of cluster membership. We represented each cluster by a hyperedge. The idea in MCLA is to group and collapse related hyperedges and assign each object to the collapsed hyperedge in which it participates most strongly. The hyperedges that are considered related for the purpose of collapsing are determined by a graph,based clustering of hyperedges. We refer to each cluster of hyperedges as a meta,cluster C(M).
Collapsing reduces the number of hyperedges from:

$$\sum_{q=1}^{r} k^{(q)} \qquad (1)$$

to k. The detailed steps are :

A. Construct Meta Graph

Let us view all the indicator vectors h (the hyperedges of H) as vertices of another regular undirected graph, the meta,graph. The edge weights are proportional to the similarity between vertices. A suitable similarity measure here is the binary Jaccard measure, since it is the ratio of the intersection to the union of the sets of objects corresponding to the two hyperedges. Formally, the edge weight wa,b between two vertices ha and hb as defined by the binary Jaccard measure of the corresponding indicator vectors ha and hb is: Since the clusters are non,overlapping there are no edges amongst vertices of the same clustering H(q) and, thus,
the meta,graph is r,partite

B. Cluster Hyperedges

Find matching labels by partitioning the meta,graph into k balanced meta,clusters. Each vertex is weighted proportional to the size of the corresponding cluster. Balancing ensures that the sum of vertex,weights is approximately the same in each meta,cluster. We use the graph partitioning package METIS in this step. This results
in a clustering of h vectors. Since each vertex in the meta, graph represents a distinct cluster label, a meta,cluster represents a group of corresponding labels.

C. *Collapse Meta Clusters*

For each of the k meta,clusters, we collapse the hyperedges into a single meta,hyperedge. Each meta, hyperedge has an association vector which contains an entry for each object describing its level of

association with the corresponding meta,cluster. The level is computed by averaging all indicator vectors h of a particular meta,cluster. An entry of 0 or 1

indicates the weakest or strongest association, respectively.

D. Compete for Objects

In this step, each object is assigned to its most associated meta,cluster: Specifically, an object is assigned to the meta,cluster with the highest entry in the association vector. Ties are broken randomly. The confidence of an assignment is reflected by the winner's share of association (ratio of the winner's association to the sum of all other associations). Note that not every meta,cluster can be guaranteed to win at least one object. Thus, there are at most k labels in the final combined clustering λ.

E. Multilevel Graph Partition: METIS

Metis a multilevel graph partitioning system, approaches the graph partitioning problem from a different angle. It partitions a graph using three basic steps: (1) coarsen the graph by collapsing b vertices and edges; (2) partition the coarsened graph and (3) refine the partitions. In comparison to other graph partitioning algorithms, Metis is highly efficient and achieves competitive performance.

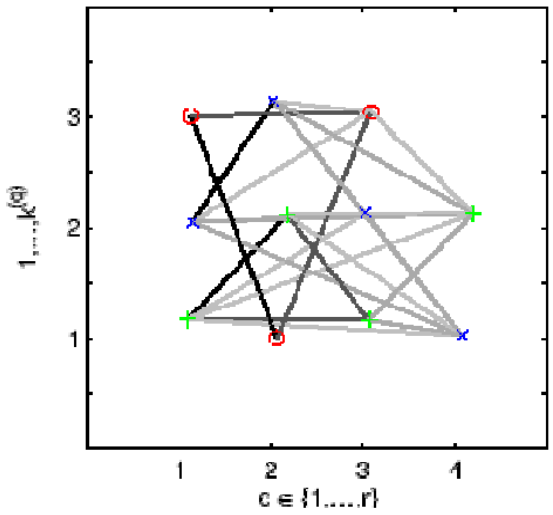

Figure 1. Meta clustering

IV. ADVANTAGES OF META CLUSTERING

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

1) Provides for a method to represent the consensus across multiple runs of a clustering algorithm, to determine the number of clusters in the data, and to assess the stability of the discovered clusters.

2) The method can also be used to represent the consensus over multiple runs of a clustering algorithm with random restart so as to account for its sensitivity to the initial conditions.

3) It also provides for a visualization tool to inspect cluster number, membership, and boundaries.

4) We will be able to extract lot of features / attributes from multiples runs of different clustering algorithms on the data. These features can give us valuable information in doing a final consensus clustering.

V. CONCLUSION

sMCLA extends MCLA by accepting soft clusterings as input. sMCLA's working can be divided into the following steps:

1) Construct Soft Meta,Graph of Clusters
2) Group the Clusters into Meta,Clusters
3) Collapse Meta,Clusters using Weighting
4) Compete for Object

Other worthwhile future work includes a thorough theoretical analysis of the average normalized mutual information (ANMI) objective, including how it can be applied to soft clusterings. We also plan to explore possible sMCLA schemes in more detail. The CSPA scheme introduced is not very practical by itself. However, it can be used as a post,processing step to refine good solutions when n is not too large. For example, one can use the supra,consensus labeling as the initialization instead of the best single input clustering. Preliminary experiments indicate that this post,processing a. Another direction of future work is to better understand the biases of the three proposed consensus functions. We would also like to extend our application scenarios. Cluster ensembles could enable federated data mining systems to work on top of distributed and heterogeneous databases.

REFERENCES

[1] Alexander Strehl and J. Ghosh, "Cluster ensembles – a knowledge reuse framework for combining multiple partitions", Journal of Machine Learning Research, vol. 3, 2002, pp. 583,617

[2] Kunal Punera, Joydeep Ghosh, "Consensus Based Ensembles of Soft Clusterings", Applied Artificial Intelligence: An International Journal, Aristides vol. 22, Issue 7,8, 2008, pp. 780,810 [3] Fern, X. Z., & Brodley, C. E.. Random projection for high dimensional data clustering: A cluster ensemble approach Hongjun Wang,

Hanhuai Shan, Arindam Banerjee. Bayesian Cluster Ensembles, SIAM International Conference on Data Mining, SDM 09, 2003

[4] Topchy, A., Jain, A. K., & Punch, W, "Combining multiple weak clusterings", ICDM. 2003

[5] Strehl, A., & Ghosh, J., "Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Machine Learning Research",2002

[6] Karypis, G., & Kumar, V., "A fast and high quality multilevel scheme for partitioning irregular graphs", 1998.