

# Hierarchical Ensemble Clustering

Li Zheng    Tao Li

*School of Computing and Information Sciences  
Florida International University  
Miami, FL, USA  
Email: {lzheng001, taoli}@cs.fiu.edu*

Chris Ding

*Department of Computer Science and Engineering  
University of Texas at Arlington  
Arlington, TX, USA  
Email: chqding@uta.edu*

**Abstract**—Ensemble clustering has emerged as an important elaboration of the classical clustering problems. Ensemble clustering refers to the situation in which a number of different (input) clusterings have been obtained for a particular dataset and it is desired to find a single (consensus) clustering which is a better fit in some sense than the existing clusterings. Many approaches have been developed to solve ensemble clustering problems over the last few years. However, most of these ensemble techniques are designed for partitional clustering methods. Few research efforts have been reported for ensemble hierarchical clustering methods. In this paper, we propose a hierarchical ensemble clustering framework which can naturally combine both partitional clustering and hierarchical clustering results. We notice the importance of ultra-metric distance for hierarchical clustering and propose a novel method for learning the ultra-metric distance from the aggregated distance matrices and generating final hierarchical clustering with enhanced cluster separation. Experimental results demonstrate the effectiveness of our proposed approaches.

**Keywords**-Hierarchical ensemble clustering; Ultra-metric.

## I. INTRODUCTION

Recently, as a way for overcoming the results instability and improving clustering performance, ensemble clustering has emerged as an important elaboration of the classical clustering problems. Ensemble clustering refers to the situation in which a number of different (input) clusterings have been obtained for a particular dataset and it is desired to find a single (consensus) clustering which is a better fit in some sense than the existing clusterings [20]. Many approaches have been developed to solve ensemble clustering problems over the last few years [5][9][11][12][15][23][13].

However, most of these existing ensemble techniques are designed for partitional clustering methods. Few research efforts have been reported for ensemble hierarchical clustering methods. Different from partitional clustering where the clustering results are “flat” and can thus be easily represented using vectors, clustering indicators, or connectivity matrices [20][12], the results of hierarchical clustering are more complex and typically represented as dendrograms or trees. In addition, hierarchical clustering methods have no global objective functions. These properties have made the ensemble hierarchical clustering problem more challenging.

In this paper, we propose a **hierarchical ensemble clustering** (HEC) framework. In this framework, the input could

be both partitional clusterings and hierarchical clusterings. The output is a **consensus** hierarchical clustering. We discuss three cases below.

(1) The input data consists of partitional clusterings. In this case, we first construct the **aggregate consensus distance** from these partitional clusterings. We then generate a consensus clustering using the consensus distance. If we stop here, that would be the usual ensemble clustering. In HEC, we can further build a **structure hierarchy** on top of the consensus clustering using the consensus distance.

Firstly, a structure hierarchy on top of a clustering solution is useful to organize and understand the discovered knowledge (topic or pattern). Secondly, the cluster structure hierarchy resolves a problem in the usual ensemble clustering when the input partitional clusterings have different number of clusters. In this case,  $K$ , the number of clusters in the final clustering solution is not uniquely determined (the issue of finding the most appropriate number of clusters in a dataset is an unsolved problem in data mining; in practice  $K$  is determined by trial and error.) On the other hand, in ensemble clustering, we consider input partitional clusterings are meaningful results, including the **number of clusters** in each input partitional clustering. Therefore, if the number of clusters of input partitional clusterings has a range  $[K_1, K_2]$ , then  $K$  of the final ensemble clustering should be  $K \in [K_1, K_2]$ . From this analysis, in HEC framework, we can set  $K = K_2$  for the bottom clusterings (leaves) of the structure hierarchy. In this way, the “true” number of clusters is guaranteed to be inside the cluster structure hierarchy.

(2) The input data consists of hierarchical clusterings, a set of dendrograms. In this case, we first construct the **aggregate dendrogram distance** between objects. From this distance, we then generate a hierarchical clustering as the final solution.

(3) The input data consists of both partitional clusterings and hierarchical clusterings. In this case, we construct the consensus distance from the partitional clusterings and the dendrogram distance from hierarchical clusterings. We combine these two distances into a single distance, and then generate a hierarchical clustering as the final solution. Figure 1 illustrates this case. The dataset is shown in Fig.1(A) and their distances are shown in Fig.1(B). K-means clustering

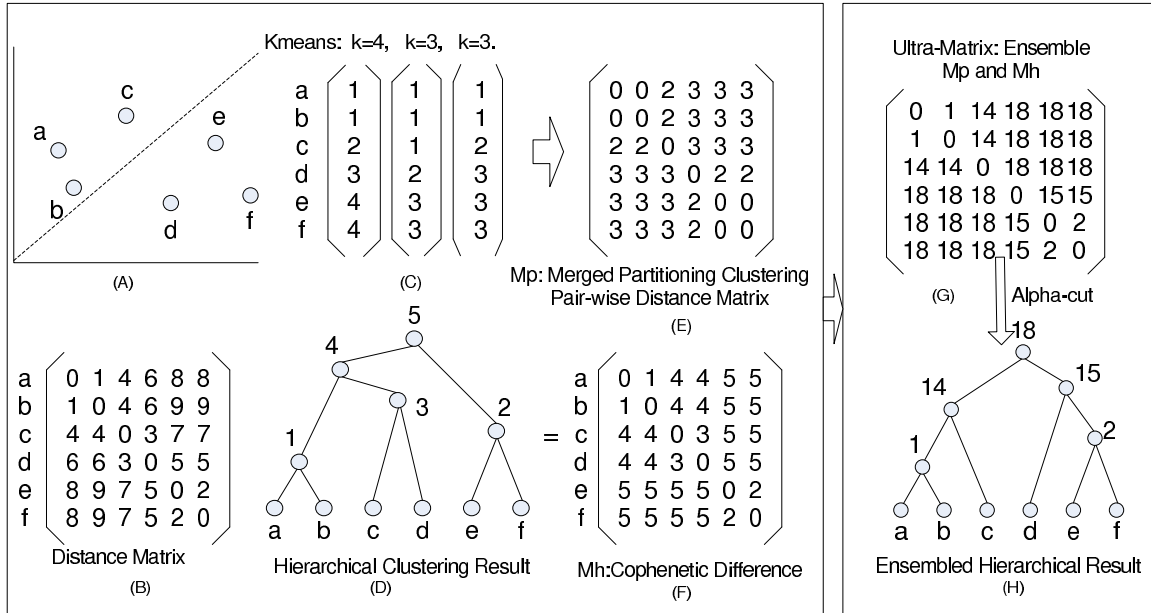


Figure 1. An illustrative example of hierarchical ensemble clustering with both partitional and hierarchical clusterings as input. The dataset is shown in (A) and their distances are shown in (B). K-means clustering are performed in (C) and lead to a consensus distance matrix in (E). A hierarchical clustering is done in (D) and leads to a dendrogram distance matrix in (F). The consensus distance matrix of (E) and the dendrogram distance matrix in (F) are combined in (G) and the final hierarchical clustering are generated in (H).

are performed with different numbers of clusters in Fig.1(C) and lead to a consensus distance matrix in Fig.1(E). A hierarchical clustering is done in Fig.1(D) and leads to a dendrogram distance matrix in Fig.1(F). The consensus distance matrix of Fig.1(E) and the dendrogram distance matrix in Fig.1(F) are combined in Fig.1(G) and the final hierarchical clustering are generated in Fig.1(H).

In this paper, we focus on the ensembles of hierarchical clustering and the related computational algorithms, because the ensemble of partitional clustering has been thoroughly studied. In particular we investigate various descriptor matrices for representing dendrograms and propose a novel method for deriving final hierarchical clustering by fitting an ultra-metric from the aggregated descriptor matrix. Our experimental evaluation also provides a systematic empirical study on the ensemble hierarchical clustering problem. Experimental results have demonstrated the effectiveness of our proposed approaches.

## II. RELATED WORK

*Ensemble Clustering:* The problem of ensemble clustering is to find a combined clustering result based on multiple clusterings of a given dataset. However, most existing ensemble clustering techniques are designed for combining partitional clustering methods.

*Consensus Tree:* Consensus tree has been widely studied in bioinformatics when comparing the evolution of species to reach a consensus or agreement [1][2]. Most techniques for finding consensus tree are based on agreement subtrees (e.g., sub structures that are common to all

the trees) [8][24]. It is very difficult for these consensus tree techniques to preserve the structural information while including all the existing leaves from the input trees [21]. In this paper, we propose a framework for ensemble hierarchical clustering based on descriptor matrices to preserve the common structures from the input hierarchical clusterings and also generate a full consensus tree.

*Metric Fitting:* The problem of fitting a tree metric to the (dis-)similarity data on pairs of objects from a given set has been studied quite extensively [4]. Ultra-metric naturally corresponds to a hierarchy of clusterings of the data. Given a dissimilarity  $D$  on pairs of objects, the problem of finding the best ultra-metric  $d_u$  such that  $\|D - d_u\|_p$  is minimized is NP-hard for  $L_1$  and  $L_2$  norms (e.g., when  $p = 1$  and  $p = 2$ ) [3]. In this paper, we propose a new method for fitting an ultra-metric to the aggregated descriptor matrix.

## III. ULTRA-METRIC AND DENDROGRAM RECONSTRUCTION

A dendrogram is a rooted tree that represents the result of a hierarchical clustering. On the root, leaves represent data objects and internal nodes represent clusters at various levels. Given a dendrogram, our task is to assign distances between leaf nodes. This problem has been studied in literature. In Section IV, we will describe several commonly used dendrogram distances (also called descriptors). We note that each of these dendrogram distance is in fact an ultra-metric distance. This is important because given an ultra-metric distance matrix  $D = (d_{ij})$ , we can reconstruct the original tree.

### A. Ultra-metric Distance

*Definition 1:* A distance matrix  $D = (d_{ij})$  is a **metric**, if it has the following properties: the nonnegativity, the symmetry, and the **triangle inequality**

$$d_{ij} \geq 0, \quad d_{ij} \leq d_{ik} + d_{kj}, \quad i \neq k \neq j.$$

It should be noted that although the nonnegativity and symmetry hold for many distance measures in data mining, the triangle inequality does not always hold for many distances.

A more restricted version of the triangle inequality is called **ultra-metric inequality**:

$$d_{ij} \leq \max(d_{ik}, d_{jk}) \quad (1)$$

for all triplets of points  $i, j, k$ .

*Definition 2:* A distance measure is an ultra-metric if it satisfies the ultra-metric inequality and the nonnegativity and symmetry.

A distance measure automatically satisfies the triangle inequality if it satisfies the ultra-metric inequality. Thus an ultra-metric distance is also a metric distance; But the inverse is not true.

### B. Dendrogram reconstruction and Ultra-metric

Ultra-metric distance plays a critical role in our HEC frame because of the unique reconstruction property. Suppose, we are given a dendrogram  $G$  and we construct a dendrogram distance  $D$  using a particular method  $M$ .

The following proposition holds:

*Proposition 1:* From a given ultra-metric distance  $D$ , a unique dendrogram  $G$  can be constructed, in the sense that if we construct the distance from  $G$ , we recover  $D$  exactly.

Note that Proposition 1 does not exclude the possibility that two different ultra-metric distances  $D_1, D_2$  lead to the same reconstructed dendrogram  $G$ .

### C. Hierarchical ensemble clustering algorithm strategy

With above discussions on ultra-metric distances and dendrogram, we outline the algorithmic strategy of our Hierarchical ensemble clustering. Our central strategy is listed below:

- 1) Use a dendrogram distance measure to generate an ultra-metric dendrogram distance for each input dendrogram (see Section IV). The consensus distance matrix for partitional clustering results are discussed in Section IV.
- 2) Aggregate the ultra-metric dendrogram distances as well as the consensus distance for partitional clusterings. (see Section V)
- 3) Finding the closest ultra-metric distance from the aggregated distance. (see Section V)
- 4) Construct the final Hierarchical clustering. (see Section V)

## IV. DENDROGRAM DISTANCES

What is essential to a dendrogram is the ultra-metric information maintained by its pairwise distance matrix. For instance, a dendrogram generated by the single-link hierarchical clustering algorithm can be viewed as a weighted dendrogram, in which every internal node is associated with a continuous variable indicating the merge distance within all leaves covered by the internal node. The merge distance is called the *height*. If we replace the height of an internal node with its rank order (i.e., the *level*) which is maintained globally to the whole dendrogram, then a weighted dendrogram becomes a fully ranked dendrogram [16]. Dendrogram descriptor can be regarded as a distance function describing the relative position of a given pair of leaves in the dendrogram and is used to characterize a corresponding dendrogram.

We now introduce the dendrogram descriptors that will be used in our investigation. The first three dendrogram descriptors are based on a fully ranked dendrogram and use the level information [14][16]. The other descriptors do not directly consider level information.

- **Cophenetic Difference (CD)** : the lowest height (i.e., merge distance) of internal nodes in the dendrogram where two specified leaves are joined together. For example, CD between nodes  $v$  and  $x$  in Figure 2 is 30.
- **Maximum Edge Distance (MED)**: the depth of node in a bottom-up view. All leaf nodes are assigned as depth 0, the depth of any internal nodes is generated in a bottom-up manner where  $\text{Depth}(C1, C2) = \max(\text{Depth}(C1), \text{Depth}(C2)) + 1$ . For example, MED of nodes  $v$  and  $x$  in Figure 2 is 2 since  $\text{Depth}(v, x) = \max(\text{Depth}(v, w), \text{Depth}(x)) + 1 = \max(1, 0) + 1$ .
- **Partition Membership Divergence (PMD)**: By utilizing the property that a hierarchical clustering result implies a sequence of nested partitions obtained by cutting the dendrogram at every internal node, the PMD is defined as the number of partitions of the hierarchy in which two specified leaves are not in the same cluster.
- **Cluster Membership Divergence (CMD)**: the size of the smallest cluster in the hierarchy which contains two specified leaves.
- **Sub-dendrogram Membership Divergence (SMD)**: the number of sub-dendrograms in which two specified leaves are not included together.

To illustrate these descriptors, Table I presents the descriptor matrices for the example dendrogram given in Figure 2.

As discussed in Section I, our framework can be naturally extended to ensemble both partitional and hierarchical clustering results by representing the partitional clustering results with a distance matrix. Formally let  $X = \{x_1, x_2, \dots, x_n\}$  be a set of  $n$  data points. Given a set of  $T$  clusterings (or partitions)  $\mathcal{SP} = \{P^1, P^2, \dots, P^T\}$  of the data points in  $X$ , the  $ij$ -th entry of the associated consensus distance matrix

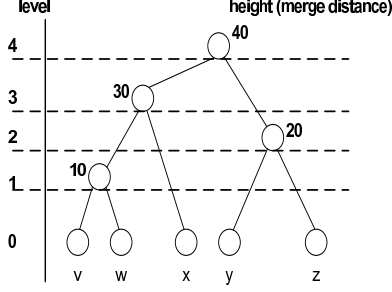


Figure 2. A Dendrogram Example.

Table I  
DENDROGRAM DESCRIPTORS FOR THE SAMPLE DENDROGRAM IN  
FIGURE 2.

1: CD					2: CMD						
	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>		<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
<i>v</i>	0	10	30	40	40	<i>v</i>	1	2	3	5	5
<i>w</i>	10	0	30	40	40	<i>w</i>	2	1	3	5	5
<i>x</i>	30	30	0	40	40	<i>x</i>	3	3	1	5	5
<i>y</i>	40	40	40	0	20	<i>y</i>	5	5	5	1	2
<i>z</i>	40	40	40	20	0	<i>z</i>	5	5	5	2	1
3: MED					4: PMD						
	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>		<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
<i>v</i>	0	1	2	3	3	<i>v</i>	0	1	3	4	4
<i>w</i>	1	0	2	3	3	<i>w</i>	1	0	3	4	4
<i>x</i>	2	2	0	3	3	<i>x</i>	3	3	0	4	4
<i>y</i>	3	3	3	0	1	<i>y</i>	4	4	4	0	2
<i>z</i>	3	3	3	1	0	<i>z</i>	4	4	4	2	0
5: SMD											
	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>						
<i>v</i>	1	1	2	3	3						
<i>w</i>	1	1	2	3	3						
<i>x</i>	2	2	2	3	3						
<i>y</i>	3	3	3	2	2						
<i>z</i>	3	3	3	2	2						

$D$  is the number of times that the  $i$ -th data point and the  $j$ -th data point are not in the same cluster. Note that the distance matrix can be combined with the dendrogram descriptors to form the aggregated distance matrix for dendrogram combination. A weight can be assigned to the distance matrix to ensure that it is at the same scale of the dendrogram descriptors.

## V. DENDROGRAM COMBINATION

Suppose we have computed consensus distance  $D_p$  from the input partitional clusterings and aggregated dendrogram distances  $D_h$  from the input hierarchical clusterings. Our tasks now are

- 1) Find an ultra-metric distance  $T$  which is the closest to  $D = D_p + D_h$ .
- 2) Construct the final hierarchical clustering based on  $T$ .

For (2) after the ultra-metric  $T$  is obtained, we obtain the final hierarchical clustering by performing alpha-cut [6]. In the rest of this section we will concentrate on (1), i.e., how to compute  $T$ .

First we note that the aggregated distance  $D$  will not be ultra-metric, even if each individual dendrogram distance is an ultra-metric. We compute the ultra-metric distance

$T$  which is closest to  $D$ , instead of using  $D$  directly due to the following two reasons. The first reason is for the unique reconstruction of the eventual dendrogram, the final hierarchical clustering, as discussed in Section V.

The second reason is an interesting property of our way of constructing  $T$ , the close approximation of  $D$ . We use a transitive dissimilarity to construct  $T$ , which has the tendency that the solution for  $T$  attracts nearby data objects into a closer proximity and therefore enhances the cluster separation (thus improves the clustering quality). In the following, we first describe the algorithm to construct  $T$  and then demonstrate the clustering separation property.

### A. Transitive Dissimilarity

First, the nonnegative distance  $D$  can be viewed as the edge weight on a graph. Our task is to construct the transitive dissimilarity starting from  $D$ .

The idea of transitive dissimilarity is to **preserve transitivity** of a graph, more precisely a social network with  $n$  persons represented as  $(V_1 \cdots, V_n)$ . If person  $V_1$  knows person  $V_2$ , and person  $V_2$  knows person  $V_3$ , transitivity implies that person  $V_1$  knows person  $V_3$ . Turning this into distances, the transitivity of  $V_1 \rightarrow V_2 \rightarrow V_3$  can be enforced as

$$d_{13} \leq \max(d_{12}, d_{23}),$$

i.e., the distance  $d_{13}$  should be no worse than either  $d_{12}$  or  $d_{23}$ .

Now consider 4 persons. One can see the above enforcement satisfies the associativity: i.e., if both  $d_{13} \leq \max(d_{12}, d_{23})$  and  $d_{24} \leq \max(d_{23}, d_{34})$  hold, then

$$d_{14} \leq \max(d_{12}, d_{23}, d_{34}).$$

Generalizing to any path  $P_{ij}$  between  $i$  and  $j$ , on the graph, the **transitive dissimilarity** on a path  $P_{ij}$  (a set of edges connect  $V_i$  and  $V_j$ ) can be defined as

$$T(P_{ij}) = \max(d_{i,k_1}, d_{k_1,k_2}, d_{k_2,k_3}, \cdots, d_{k_{n-1},k_n}, d_{k_n,j}). \quad (2)$$

So for any given pair of vertices  $V_i$  and  $V_j$ , the transitive dissimilarity varies according to different paths chosen between  $V_i$  and  $V_j$ . The **minimal transitive dissimilarity** is defined as:

$$m_{ij} = \min_{P_{ij}}(T(P_{ij})), \text{ for given vertices } V_i \text{ and } V_j. \quad (3)$$

It is clear that  $m_{ij} \geq d_{ij}, \forall V_i$  and  $V_j$ , which implies that minimal transitive dissimilarity extends vertices further than the original distance matrix.

*Proposition 2:* For any weighted dissimilarity graph, the minimal transitive dissimilarity between any pair of vertices holds the ultra-metric inequality:

$$m_{ij} \leq \max(m_{ik}, m_{kj}), \forall i, j, k.$$

*Proof:* Let  $P_{ij}$  is a set of all paths in which each element indicates an existing path connecting  $V_i$  and  $V_j$  as its end

points.  $(P_{ik}, P_{kj})$  is describing a path starting from  $V_i$  to  $V_j$  via  $V_k$  in a weighted graph. It is clear that  $(P_{ik}, P_{kj})$  is a subset of  $P_{ij}$ . We define  $W(P_{ij})$  as edge weights of any directly connected vertices in all possible paths  $P_{ij}$ .

$$\begin{aligned}
m_{ij} &= \min_{P_{ij}} \max[W(P_{ij})] \\
&\leq \min(P_{ik}, P_{kj}) \max(W(P_{ik}, P_{kj})) \\
&= \min(P_{ik}, P_{kj}) \max[\max[W(P_{ik})], \max[W(P_{kj})]] \\
&= \max[\min_{P_{ik}}(\max[W_{P_{ik}}]), \min_{P_{kj}}(\max[W_{P_{kj}}])] \\
&= \max(m_{ik}, m_{kj}).
\end{aligned}$$

Then, we use the modified Floyd-Warshall algorithm [7] to compute the updated transitive dissimilarity of all pair of vertices in the weighted graph. In particular, given input  $G$ , the adjacent matrix of a weighted graph with  $N$  nodes, the algorithm procedure is described in Algorithm 1.

**Input:**  $G$ : Pair-wise distance matrix of data set  
**Output:**  $M$ : Minimum Transitive dissimilarity matrix closure of  $G$

```

Init:  $M = G$ ;
for  $k \leftarrow 1$  to  $N$  do
  for  $i \leftarrow 1$  to  $N$  do
    for  $j \leftarrow 1$  to  $N$  do
       $m_{ij} = \min(m_{ij}, \max(m_{ik}, m_{kj}))$ ;
    end
  end
end
return  $H$  ;

```

**Algorithm 1:** Modified Floyd-Warshall algorithm to compute the minimum transitive dissimilarity of weighted graph  $G$

The following propositions are needed to show the correctness of the modified Floyd-Warshall algorithm.

*Proposition 3:* Suppose the edge weights of given graph satisfy the minimal transitive dissimilarity as defined in Eq.(3). The transitive dissimilarity are equal to the edge weights.

*Proposition 4:* Given a pair of nodes  $(i, j)$ . Let  $(i, k_1, \dots, k_m, j)$  be the path with the eventual maximal transitive dissimilarity. After successive relaxation (tightening) of edges  $(i, k_1), (k_1, k_2), \dots, (k_m, j)$  in order, the transitive dissimilarity achieves the final optimal maximal transitive dissimilarity. This holds no matter what other edges relax occur.

*Proposition 5:* Algorithm 1 correctly computes the minimum transitive dissimilarity.

*Proof:* The outer loop  $k = 1$  to  $N$  guarantees that all paths between any given vertices  $V_i$  and  $V_j$  will be considered to achieve the eventual optimal path. Proposition 4 ensures that final correct solution will be reached no

matter how internal vertices along the path are involved. Proposition 3 guarantees that any optimal solution obtained before traversing all the possible solutions will be maintained without change in the future. ■

## VI. EXPERIMENTS

### A. Experiment Setup

To evaluate our ensemble framework, we focus on how well the ensemble hierarchical solution reflects the characteristics of the original dataset. **Co-Phenetic Correlation Coefficiency (CPCC)**, which aims to evaluate how faithfully a dendrogram preserves the pair-wise distances between the original data samples [19] [17], is used as our performance measure. Generally, the higher the CPCC value, the better the clustering performance. Two datasets from UCI Machine Learning Repository<sup>1</sup> are used in our experiments. Both datasets have data labels which will be used in CPCC evaluation. The datasets and their characteristics are summarized in Table II. All experiments are conducted under the environment of Windows XP operating system plus Intel P4 1.83 GHz CPU and 4 GB of RAM.

Name	# samples	# attributes	# classes
Libras Movement	360	90	15
Madelon	2000	500	2

Table II  
DATASET DESCRIPTIONS

### B. Ensemble Hierarchical Clusterings

For experiments on ensemble hierarchical clusterings, 10 input dendrograms are generated for each dataset by using different hierarchical clustering methods on different attribute subsets. We evaluate our proposed method for generating the final hierarchical solution by fitting an ultra-meric using all five dendrogram descriptors (i.e., CD, CMD, MED, PMD, SMD). We also compare our proposed method (denoted as *ultra* in the experimental results) with the method that directly performs single-link hierarchical clustering on the aggregated descriptor matrices (denoted as *single-link* or *SL*). Table III and Table IV present the experimental results on two datasets using all input dendrograms, respectively. From the experimental results, we observe that: 1) Our proposed method *ultra* generally outperforms *single-link* across various descriptors; and 2) the ensemble solution using all input dendrograms may be worse than the best individual dendrogram, thus demonstrating the need for ensemble selection.

### C. Ensemble Partitional and Hierarchical Clusterings

We also conduct experiments to evaluate our proposed method for combining both partitional and hierarchical clusterings on the datasets. For each dataset, 10 partitional clustering results are obtained by running K-means 10 times and they are combined with 5 input dendrograms. The

<sup>1</sup>The datasets can be downloaded from <http://archive.ics.uci.edu/ml/>.

Descriptor	ultra	single-link
CD	0.414	0.416
CMD	0.406	0.387
MED	0.35	0.361
PMD	0.258	0.261
SMD	0.438	0.44

Table III

EXPERIMENTAL RESULTS ON LIBRA MOVEMENT DATASET USING ALL INPUT DENDROGRAMS. THE MAXIMUM CPCC VALUE FOR ANY INPUT DENDROGRAM IS 0.329 AND THE AVERAGE VALUE OF ALL INPUT DENDROGRAMS IS 0.18.

Descriptor	ultra	single-link
CD	0.08	0.052
CMD	0.076	0.077
MED	0.042	0.027
PMD	0.075	0.075
SMD	0.049	0.05

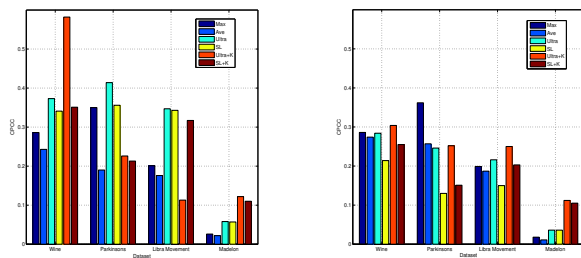
Table IV

EXPERIMENTAL RESULTS ON MADELON DATASET USING ALL INPUT DENDROGRAMS. THE MAXIMUM CPCC VALUE FOR ANY INPUT DENDROGRAM IS 0.064 AND THE AVERAGE VALUE OF ALL INPUT DENDROGRAMS IS 0.018.

experimental results are shown in Figure 3. The results demonstrate that our ensemble framework is able to combine both partitional and hierarchical clusterings and improve the performance. The results also show that our proposed method *ultra* clearly outperforms *SL* on all datasets.

## VII. CONCLUSION

In this paper, we propose a framework for ensemble hierarchical clusterings. We introduce and study three important components of the framework: Hierarchical Ensemble Selection, Dendrogram Description and Dendrogram Combination. We investigate five different dendrogram descriptor matrices, and develop a novel method for fitting an ultrametric from the aggregated descriptor matrix. Our descriptor matrices based framework can be naturally generalized to ensemble both partitional clustering and hierarchical clustering results as partitional clustering results can be easily



(a) The 5 dendrograms are represented by Cluster Membership Divergence(CMD). (b) The 5 dendrograms are represented by Cluster Membership Divergence(CMD).

Figure 3. The performance comparison of combining 10 partitional clustering results with 5 selected dendrograms. *max* represents the maximum CPCC value for any input dendrogram, and *ave* represents the average CPCC value for the input dendrograms. *ultra* and *SL* represents the recovery approaches for ensemble dendrograms by using ultra-matrix transformation and single-link respectively. *ultra+K* and *SL+K* represents the combination of K-means clustering results and previous two methods.

represented using distance matrices. Experiments are performed to evaluate our proposed approaches and the results demonstrate their effectiveness.

## ACKNOWLEDGEMENTS

The work is supported in part by NSF grants DBI-0850203 and HRD-0833093.

## REFERENCES

- [1] E. Adams. Consensus techniques and the comparison of taxonomic trees. *Syst. Zool.*, 21:390–397, 1972.
- [2] E. Adams. N-trees as nestings: complexity, similarity, and consensus. *Journal of Classification*, 3:299–317, 1986.
- [3] R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing*, 28(3):1073–1085, 1999.
- [4] N. Ailon and M. Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *FOCS '05*, pages 73–82, 2005.
- [5] J. Azimi and X. Fern. Adaptive cluster ensemble selection. In *IJCAI'09*, pages 992–997, 2009.
- [6] H. De Meyer, H. Naessens, and B. De Baets. Algorithms for computing the min-transitive closure and associated partition dendrogram of a symmetric fuzzy relation. *European Journal of Operational Research*, 155(1):226–238, May 2004.
- [7] C. Ding, H. X. X. He, and H. Peng. Transitive closure and metric inequality of weighted graphs: detecting protein interaction modules using cliques. *International Journal of Data Mining and Bioinformatics*, 1(2):123–142, Jan 2006.
- [8] M. Farach, T. Przytycka, and M. Thorup. On the agreement of many trees. *Information Processing Letters*, 55:297–301, 1995.
- [9] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML '04*, page 36, 2004.
- [10] X. Z. Fern and W. Lin. Cluster ensemble selection. *Stat. Anal. Data Min.*, 1(3):128–141, 2008.
- [11] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. In *ICDE*, pages 341–352, 2005.
- [12] T. Li and C. Ding. Weighted consensus clustering. In *Proceedings of 2008 SIAM International Conference on Data Mining*, 2008.
- [13] T. Li, C. Ding, and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM*, 2007.
- [14] A. Mirzaei, M. Rahmati, and M. Ahmadi. A new method for hierarchical clustering combination. *Intelligent Data Analysis*, 12(6):549–571, November 2008.
- [15] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning Journal*, 52(1-2):91–118, 2003.
- [16] J. Podani. Simulation of random dendrograms and comparison tests: some comments. *Journal of Classification*, 17(1):123–142, Jan 2000.
- [17] S. R. R. and F. J. Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11(1):33–40, May 1962.
- [18] D. Robinson and L. Foulds. Comparison of phylogenetic dendrograms. *Math. Biosci.*, 53(2):131–147, June 1981.
- [19] F. J. Rohlf and D. R. Fisher. Test for hierarchical structure in random data sets. *Systematic Zoology*, 17(4):407–412, Dec 1968.
- [20] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *JMLR*, 3:583–617, March 2003.
- [21] D. Swofford. When are phylogeny estimates from molecular and morphological data incongruent? In *M. M. Miyamoto and J. Cracraft, editors, Phylogenetic analysis of DNA sequences*, pages 295–333, 1991.
- [22] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2006.
- [23] A. P. Topchy, A. K. Jain, and W. F. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1866–1881, 2005.
- [24] M. Wilkinson. Common cladistic information and its consensus representation: reduced adams and reduced cladistic consensus trees and profiles. *Systematic Biology*, 43(3):343–368, 1994.