A Framework for Hierarchical Ensemble Clustering

LI ZHENG, School of Computer Science and Engineering, Nanjing University of Science and Technology; and School of Computer Science, Florida International University
TAO LI, School of Computer Science, Florida International University
CHRIS DING, Department of Computer Science, University of Texas at Arlington

Ensemble clustering, as an important extension of the clustering problem, refers to the problem of combining different (input) clusterings of a given dataset to generate a final (consensus) clustering that is a better fit in some sense than existing clusterings. Over the past few years, many ensemble clustering approaches have been developed. However, most of them are designed for partitional clustering methods, and few research efforts have been reported for ensemble hierarchical clustering methods. In this article, a hierarchical ensemble clustering framework that can naturally combine both partitional clustering and hierarchical clustering results is proposed. In addition, a novel method for learning the ultra-metric distance from the aggregated distance matrices and generating final hierarchical clustering with enhanced cluster separation is developed based on the ultra-metric distance for hierarchical clustering. We study three important problems: dendrogram description, dendrogram combination, and dendrogram selection. We develop two approaches for dendrogram selection based on tree distances, and we investigate various dendrogram distances for representing dendrograms. We provide a systematic empirical study of the ensemble hierarchical clustering problem. Experimental results demonstrate the effectiveness of our proposed approaches.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Clustering; I.2.6 [**Artificial Intelligence**]: Learning

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Hierarchical ensemble clustering, ultra-metric, ensemble selection

ACM Reference Format:
Li Zheng, Tao Li, and Chris Ding. 2014. A framework for hierarchical ensemble clustering. ACM Trans. Knowl. Discov. Data 9, 2, Article 9 (September 2014), 23 pages.
DOI: http://dx.doi.org/10.1145/2611380

1. INTRODUCTION

Data clustering arises in many disciplines and has a wide range of applications. The general goal of data clustering is to group a finite set of points in a multidimensional space into clusters so that points in the same cluster are similar to each other, whereas points in different clusters are dissimilar. The clustering problem has been extensively

This work is partially supported by the U.S. Department of Homeland Security under grant Award Number 2010-ST-062-000039; the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CIO001; the National Science Foundation under grants DBI-0850203, HRD-0833093, and DMS-0915110; and the Army Research Office under grants number W911NF-10-1-0366 and W911NF-12-1-0431.
Author's address: L. Zheng and T. Li, School of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China; School of Computer Science, Florida International University, 11200 SW 8th ST, Miami, FL 33199; email: {lzhen001, taoli}@cs.fiu.edu; C. Ding, Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, 76019; email: chqding@uta.edu.

DOI: http://dx.doi.org/10.1145/2611380

studied in the data mining, database, and machine learning communities, and many different approaches have been developed from various perspectives with various focuses. Based on the way the clusters are generated, these clustering methods can be roughly divided into two categories: partitional clustering and hierarchical clustering [Tan et al. 2005]. Generally, **partitional clustering** decomposes the dataset into a number of disjoint clusters that typically represent a local optimum of some predefined objective functions. Hierarchical clustering groups the data points into a hierarchical tree structure using bottom-up or top-down approaches. Also, equivalent dendrogram representation can be generated based on metric fitting.

Clustering is an inherently difficult problem. Different clustering algorithms and even multiple trials of the same algorithm may produce different results due to random initializations and stochastic learning methods. Recently, ensemble clustering has emerged as an important extension of the classical clustering problem because it can overcome the resulting instability and improve clustering performance. It refers to the following problem: Given a number of different (input) clusterings that have been generated for a dataset, find a single final (consensus) clustering that is a better fit in some sense than the existing clusterings [Strehl and Ghosh 2003]. Over the past few years, many ensemble clustering techniques have been proposed [Li et al. 2007, 2004 Azimi and Fern 2009; Fern and Brodley 2004; Gionis et al. 2005; Li and Ding 2008; Monti et al. 2003; Topchy et al. 2005; Luo et al. 2011].

However, existing ensemble techniques are primarily designed for partitional methods, and few research efforts have been reported for ensemble hierarchical clustering methods. In partitional clustering, the clustering results are "flat" and can be easily represented using vectors, clustering indicators, or connectivity matrices [Li and Ding 2008; Strehl and Ghosh 2003]. Different from partitional clustering, hierarchical clustering results are often more complex, and they are typically represented as dendrograms or trees.

In this work, we propose a novel **Hierarchical Ensemble Clustering** (HEC) framework in which the input can be both partitional clusterings and hierarchical clusterings. The output of the framework is a **consensus** hierarchical clustering. Three different cases are described here.

(1) In this case, the input clusterings are partitional clusterings. The **aggregate consensus distance** from these partitional clusterings is first constructed, and a consensus clustering using the consensus distance is then generated. These steps lead to the usual ensemble clustering. In HEC, a **structure hierarchy** can be further generated on top of the consensus clustering using the consensus distance.

Note that a structure hierarchy on top of a clustering solution is useful to organize and understand the discovered knowledge (topic or pattern). In addition, 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 (much research has been done on finding the most appropriate number of clusters in a dataset [Fraley and Raftery 1998; Sugar and James 2003; Tibshirani et al. 2001]). In ensemble clustering, we consider input partitional clusterings, including the **number of clusters** in each input partitional clustering, as meaningful results. Therefore, if the number of clusters of input partitional clusterings has a range of $[K_1, K_2]$, then the number of clusters in the final ensemble clustering should be $K \in [K_1, K_2]$. From this analysis, in the 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) In this case, the input clusterings are hierarchical clusterings (i.e., a set of dendrograms). A dendrogram is defined to be nested family of partitions, usually represented

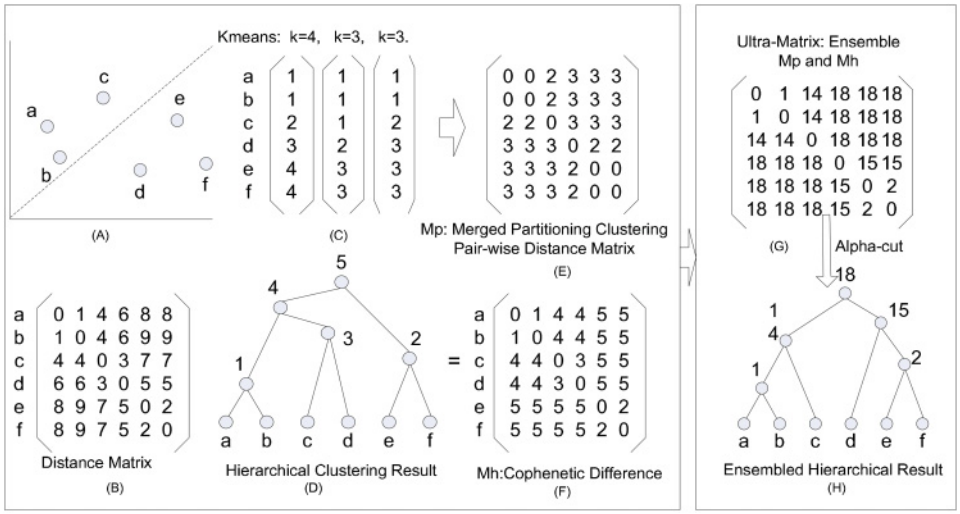


Fig. 1. An illustrative example of hierarchical ensemble clustering with both partitional and hierarchical clusterings as input. The dataset is shown in (A), and their distances are shown in (B). K-means clustering are performed in (C) and lead to a consensus distance matrix in (E). A hierarchical clustering 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).

graphically as a rooted tree [Podani 2000]. Dendrograms are often used to represent a hierarchical decomposition of the underlying data set.

The **aggregate dendrogram distance** is first constructed between objects and then a hierarchical clustering as the final solution is generated as the final solution.

(3) In this case, the input clusterings contain both partitional clusterings and hierarchical clusterings. The consensus distance from the partitional clusterings and the dendrogram distance from hierarchical clusterings are first constructed. These two distances into are then combined into a single distance, and a hierarchical clustering is generated as the final solution. An illustrative example is shown in Figure 1. Figure 1(A) shows the example dataset and Figure 1(B) shows the distance matrix. K-means clustering results with different numbers of clusters are presented in Figure 1(C) and lead to a consensus distance matrix shown in Figure 1(E). A hierarchical clustering is performed in Figure 1(D) and generates a dendrogram distance matrix shown in Figure 1(F). The consensus distance matrix of Figure 1(E) and the dendrogram distance matrix in Figure 1(F) are combined in Figure 1(G), and the final hierarchical clustering is generated in Figure 1(H).

Our preliminary work was presented at the International Conference on Data Mining (ICDM) 2010 [Zheng et al. 2010] in which we focused on the ensembles of hierarchical clustering and the related computational algorithms. In this journal article, we extend our previous work by systematically studying the following three important problems:

(1) ***Dendrogram Description***: How can we represent the dendrograms so that different hierarchical clustering solutions can be compared and combined?
(2) ***Dendrogram Combination***: How can we aggregate different dendrograms and generate final hierarchical solution?
(3) ***Dendrogram Selection***: Given a large collection of input hierarchical clusterings, how can we select a subset from the input collection to effectively build an ensemble solution that performs as well as or even better than using all available clusterings [Fern and Lin 2008]?

In particular, we investigate various descriptor matrices for representing dendrograms and propose a novel method for deriving a final hierarchical clustering by fitting an ultra-metric from the aggregated descriptor matrix. Here, we study the problem of combining both hierarchical and partitional clustering results, whereas our conference paper only focuses on the combination hierarchical clusterings. In this journal article, we present a method to first represent multiple partitional clustering results as a distance matrix and then effectively combine it with dendrogram descriptors. Thus, the final dendrogram naturally takes both types of clustering results into consideration. We formalize the ultra-metric transformation problem as an optimization problem and prove the correctness of our solution. This article also studies the problem of ensemble selection, which was ignored in our conference paper. The dendrogram selection mechanism, considering both the quality and the diversity of individual hierarchical clustering results, is presented and two approaches for dendrogram selection based on tree distances are developed. In addition, more experimental results, including using large datasets and different hierarchical clustering methods with different sets of base clusterings, are reported this article. 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.

The rest of the article is organized as follows: Section 2 discusses the related work; Section 3 discusses the ultra-metric and the general algorithm strategy for hierarchical ensemble clustering; Section 4 investigates various descriptor matrices for representing dendrograms; Section 5 describes the distance matrix used for representing partitional clustering results; Section 6 proposes a novel method for deriving final hierarchical clustering by fitting an ultra-metric from the aggregated distance matrix; Section 7 presents our approaches for dendrogram selection (i.e., selecting a subset of hierarchical clusterings from the input collection); Section 8 shows experimental evaluations and result analysis; and, finally, Section 9 concludes the paper and discusses future work.

2. RELATED WORK

2.1. Hierarchical Clustering

Hierarchical clustering algorithms are unsupervised methods to generate tree-like clustering solutions. They group the data points into a hierarchical tree structure using bottom-up (agglomerative) or top-down (divisive) approaches [Tan et al. 2005]. The typical bottom-up approach takes each data point as a single cluster to start with and then builds bigger clusters by grouping similar data points together until the entire dataset is encapsulated into one final cluster. The divisive approaches start with all data points in one cluster and then split the larger clusters recursively. Many research efforts have been reported on algorithm-level improvements to the hierarchical clustering process and on understanding hierarchical clustering [Wu et al. 2009; Zhao and Karypis 2002; Zheng and Li 2011].

2.2. Ensemble Clustering

Ensemble clustering refers to the problem of finding a combined clustering result based on multiple input clusterings of a given dataset. Many techniques can be used to obtain multiple clusterings, such as applying different clustering algorithms, using resampling to get subsamples of the dataset, utilizing feature selection methods to obtain different feature spaces, and exploiting the randomness of the clustering algorithm. Many approaches have been developed to solve ensemble clustering problems over the past few years [Azimi and Fern 2009; Fern and Brodley 2004; Gionis et al. 2005; Li and Ding 2008; Monti et al. 2003; Topchy et al. 2005]. However, existing ensemble clustering techniques are mainly designed for partitional clustering methods. The

problem of ensemble hierarchical clustering using dendrogram descriptors has been studied in Mirzaei et al. [2008]. The key difference here is that we present a coherent algorithm to learn the closest ultra-metric solution (matrix B in Equation (6)) whereas the approach in Mirzaei et al. [2008] requires many parameters that are selected in an ad hoc manner. In our approach, there are no parameters. In addition, we propose a hierarchical ensemble clustering framework that can naturally combine both partitional clustering and hierarchical clustering results, and we systematically study the problems related to dendrogram description, selection, and combination.

2.3. Consensus Tree

The problem of finding the consensus tree has been extensively studied in bioinformatics when comparing the evolution of species to reach a consensus or agreement [Adams 1986; Adams 1972]. Most techniques for solving the problem are based on agreement subtrees (e.g., the substructures that are common to all the trees) [Farach et al. 1995; Wilkinson 1994]. It is quite difficult for these consensus tree techniques to preserve structural information while including all the existing leaves from the input trees [Swofford 1991]. In our work, a framework based on descriptor matrices is proposed to preserve the common structures from the input clusterings and generate a full consensus tree.

2.4. Metric Fitting

Fitting a tree metric to the (dis-)similarity data has been studied quite extensively [Ailon and Charikar 2005]. Ultra-metric is a special kind of tree metric in which all elements of the input dataset are leaves in the underlying tree, and all leaves are at the same distance from the root. It naturally corresponds to a hierarchy of clusterings [Agarwala et al. 1999; Ailon and Charikar 2005]. 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 $p = 2$) [Agarwala et al. 1999]. In our work, a new method for fitting an ultra-metric to the aggregated descriptor matrix is developed.

2.5. Ensemble Decision Trees

In supervised classification, different decision trees can be combined using bagging [Breiman and Breiman 1996], boosting [Schapire and Singer 1999], stacking [Wolpert 1992], or random forests [Breiman and Breiman 2001]. Unlike our ensemble hierarchical clustering, these ensemble methods are designed for supervised classification. In addition, most of the decision tree ensembles do not generate a final tree and simply combine the output predictions of base trees.

2.6. Cluster Ensemble Selection

The problem of selecting a subset of input clusterings to form a smaller but better performing cluster ensemble than using all available solutions has been studied recently for partitional clustering [Azimi and Fern 2009; Fern and Lin 2008]. In this article, we develop cluster ensemble selection methods for hierarchical clustering based on tree distances.

There are also many related researches on combining multiple hierarchical clustering results from different perspectives [Hossain et al. 2012; Jalalat-evakilkandi and Mirzaei 2010; Koutroumbas et al. 2010; Lu and Wan 2012; Mirzaei and Rahmati 2008; Mirzaei and Rahmati 2010; Rashedi and Mirzaei 2011]. However, our proposed approach in this article is able to combine both multiple hierarchical clustering and partitional clustering results. In addition, we studied the problem of dendrogram

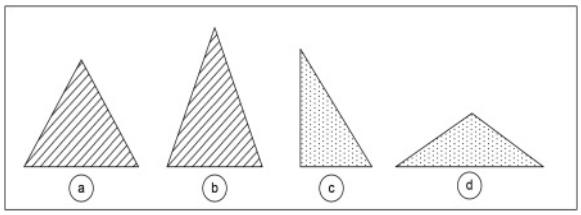


Fig. 2. A ultra-metric space example.

selection and also developed a method for learning the ultra-metric distance from the aggregated distance.

3. ULTRA-METRIC AND DENDROGRAM RECONSTRUCTION

A dendrogram is defined as a nested family of partitions, usually represented graphically as a rooted tree where leaves represent data objects and internal nodes represent clusters at various levels [Podani 2000]. The structural information is kept by pairwise cophenetic proximity that measures the level at which two data objects are first merged into a cluster [Jain and Dubes 1998].

Given a dendrogram, our task is to assign distances between leaf nodes. This problem has been studied in the literature [Mirzaei et al. 2008; Podani 2000]. Several commonly used dendrogram distances (also called descriptors) are described in Section 4. 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.

3.1. Ultra-metric Distance

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

$$d_{ij} \geq 0,$$

if $d_{ij} = d(x_i, x_j) = 0$, then $x_i = x_j$; (2) symmetry

$$d_{ij} = d_{ji};$$

and (3) the **triangle inequality**

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

Although non-negativity and symmetry hold for many distance measures in data mining, the triangle inequality often does not always hold. A more restricted version of the triangle inequality is called the **ultra-metric inequality**:

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

for all triplets of points i, j, k. This is equivalent to saying that for any distinct triple i, j, k, the largest two distances among d_{ij}, d_{ik}, d_{jk} are equal and not less than the third one.

Definition 2. A distance measure is an ultra-metric if it satisfies the ultra-metric inequality, non-negativity, and symmetry.

To illustrate the ultra-metric, four triangles formed by three data points are shown in Figure 2. Those four triangles clearly satisfy the triangle inequality; however, only a and b satisfy the ultra-metric inequality. From Equation (1), it can be easily shown that, for those triangles shown in Figure 2, if the proximity measure is an ultra-metric,

then the triangle formed by all triples of points must be an isosceles triangle with the unequal leg no longer than the two legs of equal length. The example shows that ultra-metric properties impose more restrictions on sample relations.

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 converse is not true.

3.2. Dendrogram Reconstruction and Ultra-metric

In Single-Link (SL) and Complete-Link (CL) hierarchical clustering, a dendrogram is generated by repeatedly picking the closest pair of clusters from the distance matrix, merging these two clusters into one, and updating the distance matrix. Various schemes differ in how the distance between a newly formed cluster and the other clusters is defined. Let d be the final generated distance. It can be easily shown that d is an ultra-metric. To see why, consider three objects i, j, k. Without loss of generality, assume i and j merge first. Then we have $d(i, j) \leq d(i, k) = d(j, k)$. More details can be found in Jain and Dubes [1998].

In our HEC framework, ultra-metric distance plays a critical role due to its unique reconstruction property. We have the following proposition:

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.*

In fact, there are several ways to model the pairwise distance matrix between instances in a dendrogram (see Section 4). Using different dendrogram distance measures leads to different ultra-metric distances.

3.3. Hierarchical Ensemble Clustering Algorithm Strategy

With the aforementioned discussions on ultra-metric distances and dendrograms, the algorithmic strategy of our hierarchical ensemble clustering is outlined here:

(1) Use a dendrogram distance measure to generate an ultra-metric dendrogram distance for each input dendrogram (see Section 4). We also discuss the consensus distance matrix for partitional clustering results in Section 5.
(2) Aggregate the ultra-metric dendrogram distances, as well as the consensus distance for partitional clusterings (see Section 6).
(3) Find the closest ultra-metric distance from the aggregated distance (see Section 6).
(4) Construct the final hierarchical clustering (see Section 6).

4. DENDROGRAM DISTANCES

A dendrogram is usually used to represent the hierarchical clustering results for cluster analysis, and it is easy to interpret. The ultra-metric information contained in the pairwise distance matrix can be clearly mapped to dendrogram structural information. So, for each dendrogram, there is an ultra-metric matrix that uniquely characterizes it and can be used to recover this dendrogram [Mirzaei et al. 2008].

For instance, a dendrogram obtained from the SL 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 its covered leaves. The merge distance is usually called the *height*. If we replace the height of an internal node with its rank order (i.e., the *level*), which is maintained globally with respect to the whole dendrogram, then a weighted dendrogram becomes a fully ranked dendrogram [Podani 2000]. A dendrogram descriptor can be viewed as a distance function describing the relative position of a given pair of leaves in the dendrogram, and it is used to characterize a corresponding dendrogram.

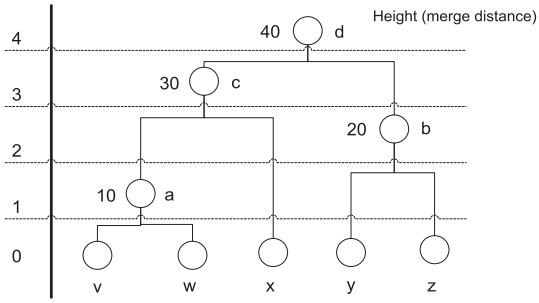


Fig. 3. A dendrogram example.

In the following paragraphs, we introduce several dendrogram descriptors used in our work. The first three dendrogram descriptors are based on a fully ranked dendrogram, and they all make use of the level information [Mirzaei et al. 2008; Podani 2000]. In other descriptors, the level information is not directly considered.

—***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 3 is 30.
—***Maximum Edge Distance*** **(MED)**: the depth of a node in a bottom-up view. All leaf nodes are assigned a depth of 0, and the depth of any internal node is generated in a bottom-up manner. Suppose C3 is the internal node at which C1 and C2 first merge; then, $\text{Depth}(C3) = \max(\text{Depth}(C1), \text{Depth}(C2)) + 1$. For example, MED of nodes v and x in Figure 3 is 2. Nodes v and x first merged at internal node c, so $\text{Depth}(c) = \max(\text{Depth}(a), \text{Depth}(x)) + 1 = \max(1, 0) + 1 = 2$, since $\text{Depth}(a) = \max(\text{Depth}(v), \text{Depth}(w)) + 1 = 1$.
—***Partition Membership Divergence*** **(PMD)**: PMD utilizes the property that a hierarchical clustering result implies a sequence of nested partitions and 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 that contains two specified leaves.
—***Subdendrogram Membership Divergence*** **(SMD)**: the number of subdendrograms in which two specified leaves are not included together.

For illustration purpose, an example dendrogram is given in Figure 3, and its various descriptor matrices are presented in Table I.

5. DISTANCE MATRICES FOR PARTITIONAL CLUSTERING RESULTS

As discussed in Section 1, 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 $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n data points. Given a partitional clustering C consisting of a set of clusters $C = \{C_1, C_2, \ldots, C_k\}$ where k is the number of clusters and $X = \bigcup_{\ell=1}^{k} C_\ell$, we can define the following associated distance matrix $D(C)$ whose ij-th entry is defined as

$$d_{ij} = \begin{cases} 0 & (i, j) \in C_\ell \\ 1 & \text{Otherwise,} \end{cases} \tag{2}$$

Table I. Dendrogram Descriptors for the Sample Dendrogram in Figure 3

1: CD

	v	w	x	y	z
v	0	10	30	40	40
w	10	0	30	40	40
x	30	30	0	40	40
y	40	40	40	0	20
z	40	40	40	20	0

2: CMD

	v	w	x	y	z
v	1	2	3	5	5
w	2	1	3	5	5
x	3	3	1	5	5
y	5	5	5	1	2
z	5	5	5	2	1

3:MED

	v	w	x	y	z
v	0	1	2	3	3
w	1	0	2	3	3
x	2	2	0	3	3
y	3	3	3	0	1
z	3	3	3	1	0

4: PMD

	v	w	x	y	z
v	0	1	3	4	4
w	1	0	3	4	4
x	3	3	0	4	4
y	4	4	4	0	2
z	4	4	4	2	0

5:SMD

	v	w	x	y	z
v	1	1	2	3	3
w	1	1	2	3	3
x	2	2	2	3	3
y	3	3	3	2	2
z	3	3	3	2	2

where $(i, j) \in C_\ell$ means that i-th data point and j-th data point are in the same cluster C_ℓ. In other words, if the i-th data point and the j-th data point are in the same cluster, then the distance between them is 0.

Given a set of s clusterings (or partitions) $\mathcal{P} = \{P^1, P^2, \ldots, P^s\}$ of the data points in X, the associated consensus distance matrix D can be represented as

$$D(\mathcal{P}) = \frac{1}{s} \sum_{i=1}^{s} D(P^i). \tag{3}$$

In other words, the ij-th entry of D indicates the average number of times that the i-th data point and the j-th data point are not in the same cluster.

Equation (3) defines a way to aggregate multiple partitional clustering results into one consensus distance matrix. Also there are many different ways to define the consensus function, such as co-associations between data points or based on pairwise agreements between partitions. Some of the criteria are based on the similarity between data points, and some of them are based on the estimates of similarity between partitions. The relationship between consensus matrix and other measures is discussed and summarized in Li et al. [2010].

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 as the dendrogram descriptors.

6. DENDROGRAM COMBINATION

Given any similarity, we can do any kind of hierarchical clustering. However, there are many different choices here: SL, CL, average-link, and many other choices. Which one to choose? Our logic is that since the input individual descriptors are ultra-metric, and the consensus matrix is not ultra-metric, the most natural approach is to a find an ultra-metric that is as close to the consensus matrix as possible. Once this ultra-metric

is learned, the final hierarchical clustering is uniquely determined. There are other choices here. The entire approach is uniquely deterministic.

Let $D(\mathcal{P})$ be the computed consensus distance from the input partitional clusterings and let $D(H)$ be the aggregated dendrogram distance from the input hierarchical clusterings. The task of dendrogram combination includes the following steps:

(1) Finding an ultra-metric distance T which is the closest to $D = \frac{1}{2} \times (D(\mathcal{P}) + D(H))$
(2) Constructing the final hierarchical clustering based on T

Once the ultra-metric T is obtained, the final hierarchical clustering can be generated by performing the alpha-cut [Meyer et al. 2004]. In the remainder of this section, we concentrate on (1); that is, how to compute T.

It should be pointed out 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 that 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 3. The second reason is that we can use a transitive dissimilarity to construct T that could attract nearby data objects into a closer proximity.

6.1. Transitive Dissimilarity

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

The idea of transitive dissimilarity is to **preserve the transitivity** of a graph; more precisely, a social network with n people represented as $(V_1 \ldots, V_n)$. 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 $V_1 \to V_2 \to V_3$ can be enforced as

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

that is, the distance d_{13} should be no greater than either d_{12} or d_{23}.

Now consider four people. One can see that our enforcement satisfies associativity: 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

$$T(P_{ij}) = \max(d_{i,k_1}, d_{k_1,k_2}, d_{k_2,k_3}, \ldots, d_{k_{n-1},k_n}, d_{k_n,j}). \tag{4}$$

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:

$$m_{ij} = \min_{P_{ij}}(T(P_{ij})), \text{ for given vertices } V_i \text{ and } V_j. \tag{5}$$

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

Thus, the problem of obtaining the ultra-metric transformation of a consensus matrix can be formulated as the following optimization problem:

PROBLEM 1. *A is the consensus distance matrix; B is the desired ultra-metric to be computed:*

$$\min_B \sum_{ij} |A_{ij} - B_{ij}|, s.t. \;\; B_{ij} \leq A_{ij}. \tag{6}$$

The ultra-metric constraint on B is a hard constraint. The optimal solution is given by Algorithm 1. In other words, the desired ultra-metric distance is always smaller than input distance.

ALGORITHM 1: Modified Floyd-Warshall Algorithm to Compute the Minimum Transitive Dissimilarity of Weighted Graph G

Input: G: Pairwise distance matrix of dataset.
Output: M: Minimum transitive dissimilarity matrix closure of G.
Init: M = G.

```
for k ← 0 to N do
  for i ← 0 to N do
    for j ← 0 to N do
      m_ij = min(m_ij, max(m_ik, m_kj))
    end for
  end for
end for
return M
```

The modified Floyd-Warshall algorithm [Ding et al. 2006] is used to compute the updated transitive dissimilarity of all pairs of vertices in the weighted graph. Algorithm 1 describes the algorithm procedure where the adjacency matrix G of a weighted graph with N nodes is given as the input.

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

PROPOSITION 2. *Suppose the edge weights of a given graph satisfy the minimal transitive dissimilarities as defined in Equation (5). The transitive dissimilarities are equal to the edge weights.*

PROOF. We prove Proposition 2 using dynamic programming. Start from two-hop paths V_i-V_k-V_j between any given vertices V_i and V_j. As the edge weights d satisfy the minimal transitive dissimilarities, so d_{ij} must be less than or equal to two-hop transitive weight $T(P_{ikj})$ for any k. Since we have minimal transitive dissimilarity $m_{ij} \leq d_{ij}$ implied by Equation (5), so $m_{ij} \leq d_{ij} \leq T(P_{ikj})$ holds. For two-hop minimal transitive dissimilarity, we get $m_{ij} = d_{ij}$.

Given any three-hop path between V_i and V_j, denoted as V_i-V_k-V_l-V_j, we can change V_i-V_k-V_l to V_i-V_l, or change V_k-V_l-V_j to V_k-V_j based on the destination from two-hop paths. We apply transitive dissimilarity and the edge weight equivalence property again on path V_i-V_l-V_j or V_i-V_k-V_j again; then, we get $m_{ij} = d_{ij}$, for any path V_i-V_k-V_l-V_j.

For any n-hop path ($n \geq 2$), the same process can be applied. Thus, Proposition 2 is proved. □

PROPOSITION 3. *Given node pair V_i and V_j, let V_i-V_{k1}-$\cdots$-V_{km}-V_j) be the path with the eventual minimal transitive dissimilarity. After successive tightening of edges V_i-V_{k1}, V_{k1}-V_{k2}, $\ldots$, V_{km}-V_j in order, the transitive dissimilarity achieves the final optimal minimal transitive dissimilarity. This holds no matter what other edge relaxations occur.*

PROOF. Since the eventual path between V_i and V_j with minimal transitive dissimilarity is given, the length-2 minimal transitive dissimilarity (optimal solution) can be easily obtained. Also, the length-3 minimal transitive dissimilarity can be obtained based on the length-2 solution, and it is obviously the optimal solution. The conclusion holds when extending to the last edge of the path. Thus, Proposition 3 is proved. □

PROPOSITION 4. *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 3 ensures that the final correct solution will be reached no matter how internal vertices along the path are involved. Proposition 2 guarantees that any optimal solution obtained before traversing all the possible solutions will be maintained without change in the future. □

From these propositions, we know that the minimal transitive dissimilarity brings objects closer than the original distance matrix. Our experimental results in Section 8 show that the final hierarchical solutions arrived at by fitting an ultra-meric using transitive dissimilarity generally outperform the method that directly performs SL and CL hierarchical clusterings on the aggregated descriptor matrices. A formal analysis of cluster separation enhancement requires dedicated work and is one of our future projects.

7. DENDROGRAM SELECTION

Selecting a subset of input clusterings to form a smaller ensemble has been shown to achieve better performance than using all available solutions for partitional clustering methods [Azimi and Fern 2009; Fern and Lin 2008]. The selection is based on the quality and diversity of each individual clustering solution. For partitional clustering, since the clustering solutions are naturally represented using vectors or matrices [Li and Ding 2008; Strehl and Ghosh 2003], the diversity and quality of the clustering solutions can be easily computed. To perform dendrogram selection, the question is how to compute the diversity and quality of different hierarchical clustering solutions.

We propose two approaches to perform dendrogram selection based on tree distances. First, we introduce the tree distances to measure the similarities/differences between different hierarchies. Two distances are frequently used in the literature to compute the distance between two evolutionary dendrograms: Branch Score Distance (BSD) of Kuhner and Felsenstein [1994] and Symmetric Difference (SD) of Robinson and Foulds [1981]. Both distances are computed by considering all possible branches that could exist on the two trees. Note that each branch makes a partition of the given dataset into two groups—the ones connected to one end of the branch (the ones on a subtree) and the ones connected to the other (the others). BSD uses branch lengths, whereas SD does not use branch lengths and only uses the tree topologies. For BSD, each partition on a dendrogram has an associated branch length (i.e., the distance when merging two subclusters). BSD is then computed by taking the sum of squared differences between the branch lengths of two dendrograms. SD is calculated as the number of partitions that only exist in one of the dendrograms.

The goal of dendrogram selection is to select a diverse subset of dendrograms where each of them has good quality. We propose two approaches for dendrogram selection using tree distances. In both approaches, the size of the selected set of dendorgrams is given as an input. The first approach is to use a modified K-medoids algorithm (with the tree distances) to cluster those dendrograms and then select the medoids for each cluster. The medoid of a cluster is a representative object whose average similarity to all the other objects in the cluster is maximized; thus, the medoid dendrogram can be considered to best capture the information contained in the cluster and has good quality. On the other hand, selecting medoids from different clusters achieves diversity.

The second approach is based on the farthest-point heuristic [Gonzalez 1985]. The approach starts with the medoid of all the input clustering solutions. Then, pick a dendrogram that is as far from the selected dendrogram as possible. In general, the approach picks a dendrogram to maximize the distances to the nearest of all

Table II. Dataset Descriptions

Name	# Samples	# Attributes	# Classes
Wine	178	13	3
Parkinson Disease	195	22	2
Libras Movement	360	90	15
WebACE	2340	1000	12
Reuters	2787	1000	9

Table III. Experimental Results on Wine Dataset Using All Input Dendrograms

Descriptor	Ultra	Single-Link
CD	**0.392**	0.381
CMD	**0.443**	0.273
MED	**0.292**	0.288
PMD	**0.267**	0.232
SMD	**0.299**	0.290

The maximum CPCC value for any input dendrogram is 0.407, and the average value of all input dendrograms is 0.282.

dendrograms picked so far. Specifically, if $t_1, t_2, \ldots, t_{i-1}$ denote the selected dendrograms so far, then we pick t_i to maximize

$$min\{dist(s_i, s_1), dist(s_i, s_2), \ldots, dist(s_i, s_{i-1})\}. \tag{7}$$

The approach stops after the required number of dendrogram has been selected.

8. EXPERIMENTS

8.1. Experiment Setup

To evaluate our proposed ensemble framework, we focus on how well the ensemble hierarchical solution reflects the characteristics of the original dataset. **Co-Phenetic Correlation Co-efficiency (CPCC)** is used as the performance measure [Rohlf and Fisher 1968; Sokal and Rohlf 1962]. It aims to evaluate how faithfully a dendrogram preserves the pair-wise distances between the original data samples, and it can be calculated as

$$c = \frac{\sum_{i<j}(d(i,j) - d)(h(i,j) - h)}{\sqrt{[\sum_{i<j}(d(i,j) - d)^2][\sum_{i<j}(h(i,j) - h)^2]}}, \tag{8}$$

where $d(i, j)$ is the distance between the i-th and j-th data instances, $h(i,j)$ is the height of lowest common ancestor of the i-th and j-th data instances in ensemble dendrogram, d is the averages of $d(i, j)$ over all pairs, and h is the average of $h(i, j)$. Generally, the higher the CPCC value, the better the clustering performance.

We use five datasets from different domains to conduct the experiments: three datasets (Wine, Parkinson Disease, and Libras Movement) from UCI Machine Learning Repository,[1] and two benchmark text datasets for document clustering (WebACE and Reuters datasets) [Li and Ding 2008]. The datasets and their characteristics are summarized in Table II. The two text datasets are represented using the vector space model, and they are also preprocessed by removing the stop words and unnecessary tags and headers. All experiments are conducted under the environment of Windows XP operating system plus Intel P4 1.83GHz CPU and 4GB of RAM.

[1]The datasets can be downloaded from http://archive.ics.uci.edu/ml/.

Table IV. Experimental Results on Parkinson Disease Dataset Using All Input Dendrograms

Descriptor	Ultra	Single-Link
CD	**0.577**	0.554
CMD	**0.431**	0.419
MED	**0.485**	0.428
PMD	0.402	**0.417**
SMD	0.448	**0.491**

The maximum CPCC value for any input dendrogram is 0.381 and the average value of all input dendrograms is 0.201.

Table V. Experimental Results on Libra Movement Dataset Using All Input Dendrograms

Descriptor	Ultra	Single-Link
CD	**0.423**	0.419
CMD	**0.411**	0.389
MED	0.36	**0.363**
PMD	**0.279**	0.266
SMD	**0.45**	0.438

The maximum CPCC value for any input dendrogram is 0.334 and the average value of all input dendrograms is 0.25.

Table VI. Experimental Results on WebACE Dataset Using All Input Dendrograms

Descriptor	Ultra	Complete-Link
CD	**0.465**	0.4637
CMD	**0.4971**	0.4963
MED	**0.4787**	0.4699
PMD	0.4831	**0.4896**
SMD	**0.5056**	0.4781

The maximum CPCC value for any input dendrogram is 0.47 and the average value of all input dendrograms is 0.428.

8.2. Ensemble Hierarchical Clusterings

In this set of experiments, for each dataset, 10 input dendrograms are generated by using different hierarchical clustering methods on different attribute subsets. In particular, they are generated as follows: (1) five different attribute subsets are randomly constructed first, each of which contains 90% of all the attributes; and (2) SL and CL algorithms are applied to 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 SL and CL hierarchical clusterings on the aggregated descriptor matrices (denoted as *single-link*/*complete-link* or *SL*/*CL*.[2]

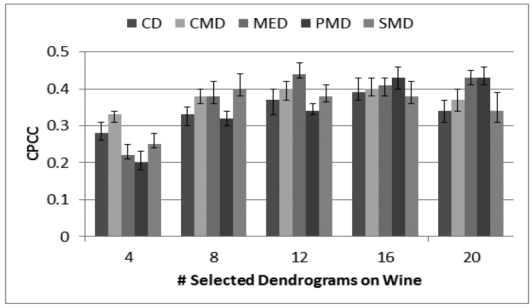
8.2.1. Results Using All Input Dendrograms. Tables III–VII present the experimental results on **six** datasets using all input dendrograms, respectively. Note that, unlike ensemble clustering for partitional clustering results, for hierarchical clustering ensembles, once the set of individual hierarchical clustering results is fixed, then the result of the ensemble is also determined. From the experimental results, we observe that:

[2]In our work, we apply *SL* on the aggregated descriptor matrices for four UCI datasets and apply *CL* on the aggregated descriptor matrices for two text datasets.

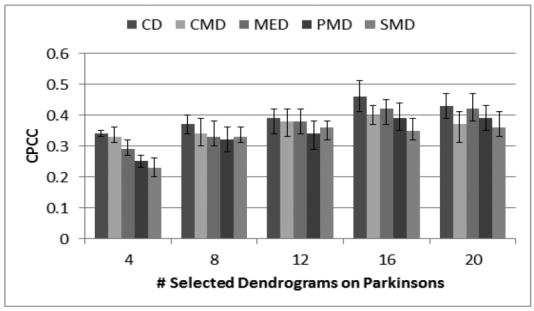
Table VII. Experimental Results on Reuters Dataset Using All Input Dendrograms

Descriptor	Ultra	Complete-Link
CD	**0.7349**	0.7312
CMD	**0.7822**	0.7435
MED	**0.7415**	0.7176
PMD	**0.7624**	0.6955
SMD	0.6475	**0.6479**

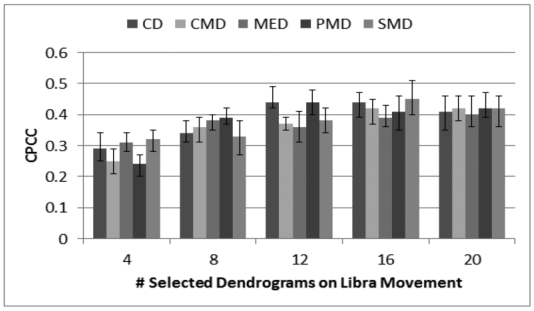
The maximum CPCC value for any input dendrogram is 0.7583 and the average value of all input dendrograms is 0.633.



(a) Wine

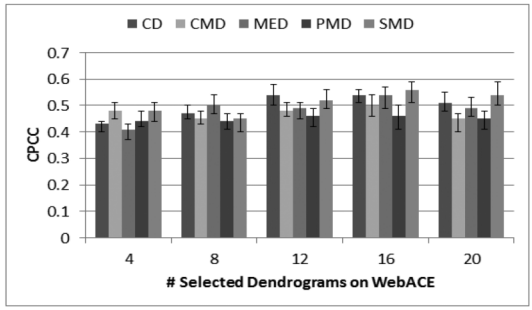


(b) Parkinsons

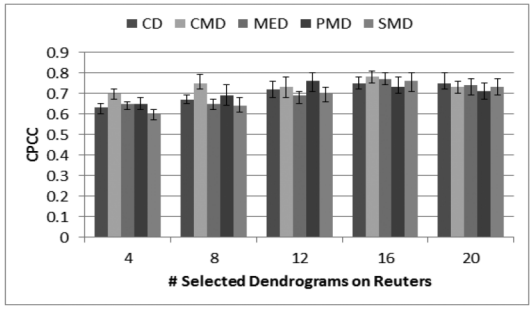


(c) Libra Movement

Fig. 4. The performance variation on different numbers of selected dendrograms over 20 trials.



(a) WebACE



(b) Reuters

Fig. 5. The performance variation on different numbers of selected dendrograms over 20 trials.

(1) Our proposed method *ultra* generally outperforms hierarchical clustering (*SL* or *CL*) across various descriptors on most counts, especially on large datasets (e.g., WebACE and Reuters), and (2) the ensemble solution using all input dendrograms may be worse than the best individual dendrogram, thus demonstrating the need for ensemble selection.

8.2.2. Results on Different Input Dendrograms. In order to provide more insights into our proposed method, we also conduct experiments with different sets of input dendrograms. Figures 4 and 5 show the experimental results on the three UCI datasets (Wine, Parkinsons, and Libra Movement) and the two text datasets (WebACE and Reuters), respectively, with different sets of input dendrograms. In particular, for a given size, we randomly select a set of input dendrograms, and then perform the experiments. The reported results are averaged over 20 different runs.

Based on our observation, the best performance is often obtained when the number of input dendrograms is around 4 or 5. Although this experiment is conducted by randomly selecting input dendrograms, it still demonstrates that using a subset of input dendrograms (rather than using all dendrograms) may improve the ensemble performance. The issue of using dendrogram selection strategies to form the candidate subset is discussed in Section 8.2.3 and Section 8.2.4, respectively.

8.2.3. Experiments on Ensemble Selection. We also conducted experiments to demonstrate the effects of ensemble selection. Note that dendrogram selection can be performed using two different approaches (K-medoid and Farthest neighbor, denoted as K and F) with two different distances (Branch Length Score Distance or Symmetric Distance, denoted as B and S). Tables VIII–Table XII present the experimental results

Table VIII. Experimental Results on Wine Dataset Using Six Selected Input Dendrograms

Desc.	Sel	Dis	max	ave	ultra	SL
CD	F	B	0.292	0.245	**0.352**	0.331
	K	B	0.306	0.251	**0.373**	0.357
	F	S	0.281	0.229	**0.329**	0.292
	K	S	0.299	0.238	0.336	**0.344**
CMD	F	B	0.292	0.245	**0.387**	0.378
	K	B	0.306	0.251	**0.373**	0.365
	F	S	0.281	0.229	**0.361**	0.329
	K	S	0.299	0.238	**0.35**	0.337
MED	F	B	0.292	0.245	**0.369**	0.348
	K	B	0.306	0.251	**0.355**	0.316
	F	S	0.281	0.229	**0.339**	0.318
	K	S	0.299	0.238	**0.357**	0.323
PMD	F	B	0.292	0.245	**0.296**	0.284
	K	B	0.306	0.251	0.315	**0.331**
	F	S	0.281	0.229	**0.316**	0.302
	K	S	0.299	0.238	0.305	**0.32**
SMD	F	B	0.292	0.245	**0.321**	0.307
	K	B	0.306	0.251	**0.338**	0.32
	F	S	0.281	0.229	**0.317**	0.293
	K	S	0.299	0.238	**0.309**	0.304

K and F denote K-medoid and Farthest Neighbor of ensemble selection methods, respectively, and B and S denote Branch Length Score Distance and Symmetric Distance of dendrogram distances, respectively.

Table IX. Experimental Results on Parkinson Disease Dataset Using Four Selected Input Dendrograms

Desc.	Sel	Dis	max	ave	ultra	SL
CD	F	B	0.438	0.256	**0.549**	0.521
	K	B	0.467	0.251	0.538	**0.544**
	F	S	0.493	0.273	**0.537**	0.505
	K	S	0.452	0.235	**0.526**	0.524
CMD	F	B	0.438	0.256	**0.56**	0.512
	K	B	0.467	0.251	**0.572**	0.542
	F	S	0.493	0.273	**0.553**	0.527
	K	S	0.452	0.235	0.524	**0.536**
MED	F	B	0.438	0.256	**0.574**	0.539
	K	B	0.467	0.251	**0.595**	0.532
	F	S	0.493	0.273	**0.54**	0.537
	K	S	0.452	0.235	**0.589**	0.527
PMD	F	B	0.438	0.256	**0.517**	0.492
	K	B	0.467	0.251	0.523	**0.531**
	F	S	0.493	0.273	**0.502**	0.499
	K	S	0.452	0.235	**0.544**	0.507
SMD	F	B	0.438	0.256	0.529	0.529
	K	B	0.467	0.251	**0.551**	0.504
	F	S	0.493	0.273	**0.547**	0.516
	K	S	0.452	0.235	0.498	**0.511**

K and F denote K-medoid and Farthest Neighbor of ensemble selection methods, respectively, and B and S denote Branch Length Score Distance and Symmetric Distance of dendrogram distances, respectively.

Table X. Experimental Results on Libra Movement Dataset Using Four Selected Input Dendrograms

Desc.	Sel	Dis	max	ave	ultra	SL
CD	F	B	0.287	0.199	0.392	**0.433**
	K	B	0.291	0.185	**0.441**	0.408
	F	S	0.274	0.167	**0.4**	0.396
	K	S	0.303	0.158	**0.398**	0.385
CMD	F	B	0.287	0.199	**0.432**	0.424
	K	B	0.291	0.185	**0.446**	0.418
	F	S	0.274	0.167	**0.410**	0.402
	K	S	0.303	0.158	**0.453**	0.391
MED	F	B	0.287	0.199	**0.49**	0.458
	K	B	0.291	0.185	0.442	**0.476**
	F	S	0.274	0.167	**0.483**	0.472
	K	S	0.303	0.158	0.453	**0.461**
PMD	F	B	0.287	0.199	**0.397**	0.346
	K	B	0.291	0.185	**0.383**	0.315
	F	S	0.274	0.167	**0.401**	0.359
	K	S	0.303	0.158	**0.394**	0.329
SMD	F	B	0.287	0.199	**0.437**	0.384
	K	B	0.291	0.185	**0.462**	0.391
	F	S	0.274	0.167	0.423	**0.439**
	K	S	0.303	0.158	**0.468**	0.379

K and F denote K-medoid and Farthest Neighbor of ensemble selection methods, respectively, and B and S denote Branch Length Score Distance and Symmetric Distance of dendrogram distances, respectively.

Table XI. Experimental Results on WebACE Dataset Using Four Selected Input Dendrograms

Desc.	Sel	Dis	max	ave	ultra	CL
CD	F	B	0.483	0.41	**0.491**	0.49
	K	B	0.474	0.409	**0.505**	0.499
	F	S	0.465	0.417	0.492	0.492
	K	S	0.487	0.405	**0.501**	0.494
CMD	F	B	0.483	0.41	**0.511**	0.501
	K	B	0.474	0.409	**0.509**	0.507
	F	S	0.465	0.417	0.498	**0.503**
	K	S	0.487	0.405	**0.505**	0.497
MED	F	B	0.483	0.41	**0.513**	0.502
	K	B	0.474	0.409	**0.504**	0.497
	F	S	0.465	0.417	**0.5**	0.497
	K	S	0.487	0.405	**0.507**	0.489
PMD	F	B	0.483	0.41	0.496	**0.498**
	K	B	0.474	0.409	0.492	**0.497**
	F	S	0.465	0.417	**0.501**	0.5
	K	S	0.487	0.405	**0.498**	0.49
SMD	F	B	0.483	0.41	**0.503**	0.491
	K	B	0.474	0.409	**0.5**	0.493
	F	S	0.465	0.417	**0.499**	0.484
	K	S	0.487	0.405	**0.507**	0.495

K and F denote K-medoid and Farthest Neighbor of ensemble selection methods, respectively, and B and S denote Branch Length Score Distance and Symmetric Distance of dendrogram distances, respectively.

Table XII. Experimental Results on Reuters Dataset Using Four Selected Input Dendrograms

Desc.	Sel	Dis	max	ave	ultra	CL
CD	F	B	0.73	0.682	**0.747**	0.739
	K	B	0.741	0.635	0.785	**0.794**
	F	S	0.737	0.696	**0.792**	0.786
	K	S	0.729	0.64	**0.769**	0.75
CMD	F	B	0.73	0.682	**0.793**	0.767
	K	B	0.741	0.635	**0.798**	0.752
	F	S	0.737	0.696	**0.794**	0.755
	K	S	0.729	0.64	**0.782**	0.751
MED	F	B	0.73	0.682	**0.779**	0.754
	K	B	0.741	0.635	**0.783**	0.781
	F	S	0.737	0.696	0.765	**0.77**
	K	S	0.729	0.64	**0.752**	0.75
PMD	F	B	0.73	0.682	**0.782**	0.763
	K	B	0.741	0.635	**0.775**	0.755
	F	S	0.737	0.696	**0.787**	0.761
	K	S	0.729	0.64	0.74	**0.745**
SMD	F	B	0.742	0.726	**0.797**	0.784
	K	B	0.744	0.727	**0.782**	0.753
	F	S	0.736	0.730	**0.771**	0.767
	K	S	0.731	0.722	0.75	0.75

K and F denote K-medoid and Farthest Neighbor of ensemble selection methods, respectively, and B and S denote Branch Length Score Distance and Symmetric Distance of dendrogram distances, respectively.

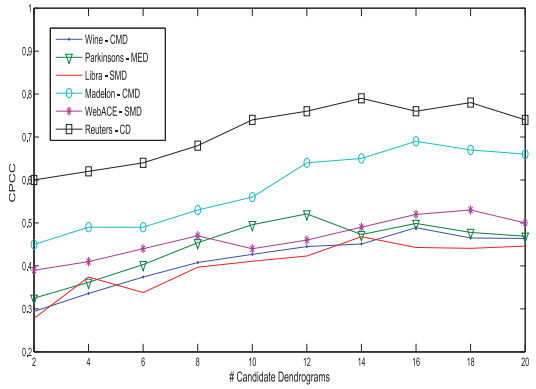
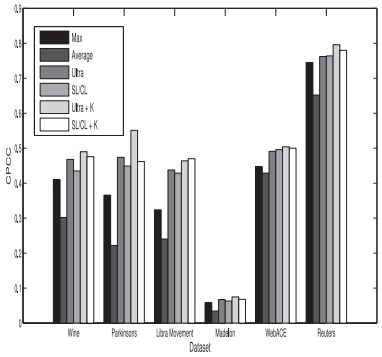


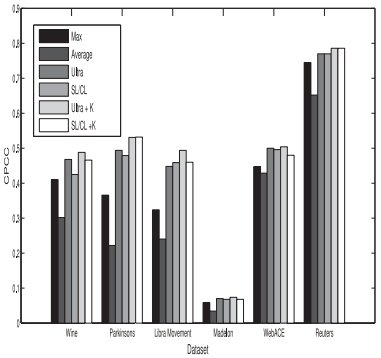
Fig. 6. The performance variation on all datasets with different numbers of candidate dendrograms.

on the **six** datasets using four selected input dendrograms, respectively.[3] In these tables, *Sel* denotes the ensemble selection approaches, *Dis* represents the tree distances, *max* represents the maximum CPCC value for any input dendrogram, and ave represents the average CPCC value for the input dendrograms. The experiments show that: (1) with ensemble selection, the results of both *ultra* and hierarchical clustering (*SL* or *CL*) have improved; (2) *ultra* still outperforms hierarchical clustering (*SL* or *CL*) in most cases; and (3) in many cases, the experiment results of *ultra* and hierarchical

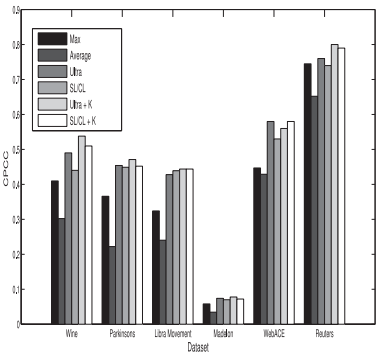
[3]The value of 4 is chosen based on our experiments on ensemble size selection, and it seems to provide good results in our experiments. How to come up with a principled way to determine ensemble size selection is one of our future projects.



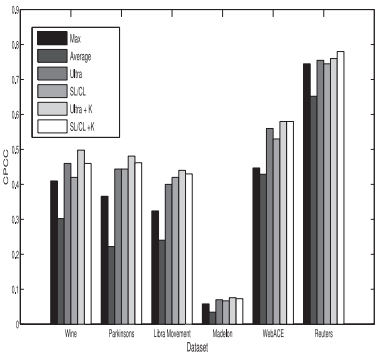
(a) The five dendrograms are represented by Cophenetic Distance Matrix (CD) and are selected using Farthest Neighbor ensemble selection and Branch Score Distance.



(b) The five dendrograms are represented by Cophenetic Distance Matrix (CD) and are selected using K-Medoid ensemble selection and Symmetric Distance.



(c) The five dendrograms are represented by Cluster Membership Divergence (CMD) and are selected using K-Medoid ensemble selection and Branch Score Distance.



(d) The five dendrograms are represented by Cluster Membership Divergence (CMD) and are selected using Farthest Neighbor ensemble selection and Symmetric Distance.

Fig. 7. The performance comparison of combining 10 partitional clustering results with five 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*/*CL* represent the recovery approaches for ensemble dendrograms by using ultra-matrix transformation and hierarchical clustering, respectively. *ultra+K* and *SL*/*CL+K* represent the combination of K-means clustering results and the previous two methods.

clustering (*SL* or *CL*) outperform the best dendrogram in the candidate set, which means those ensemble dendrograms could be more representative of the original set.

8.2.4. Experiments on Ensemble Size. To demonstrate the effect of the size of the ensemble, Figures 4 and 5 show the performance variation on different numbers of selected dendrograms on all datasets. We apply K-Medoid selection methods on SD to choose candidate dendrograms. For each dataset, we vary the group size of candidate dendrograms and use CMD as the descriptor to conduct the dendrogram selection.

Figure 6 shows the CPCC value for each dendrogram group, averaging over 20 runs. Note that for better readability, the plotted value of the Madelon dataset is 10 times its actual value. The performance slightly decreases once the number of ensemble dendrograms reaches a certain size. So selecting a relatively smaller subset is likely to

produce better ensemble results. It also shows that ensemble selection can influence the ensemble results and can be used to produce better hierarchical solutions.

8.3. Ensemble Partitional and Hierarchical Clusterings

In this set of experiments, we evaluate our proposed method for combining both partitional and hierarchical clusterings on all datasets. For each dataset, 10 partitional clustering results are obtained by running K-means 10 times, and they are combined with five input dendrograms. Figure 7 presents the experimental results. From the experimental results, we conclude that our ensemble framework is able to combine both partitional and hierarchical clusterings and improve the performance on most datasets. The results also show that our proposed method *ultra* clearly outperforms *SL*/*CL* on all datasets, and *ultra+K* generally outperforms *SL*/*CL+K* in most cases.

9. CONCLUSION AND FUTURE WORK

A framework for ensemble hierarchical clusterings based on descriptor matrices is proposed in this article. Three important components of the framework (dendrogram selection, dendrogram description, and dendrogram combination) are studied. In particular, two ensemble selection schemes based on tree distances are proposed, five different dendrogram descriptor matrices are investigated, and a novel method for fitting an ultra-metric from the aggregated descriptor matrix is developed. Since partitional clustering results can be easily represented using distance matrices, our descriptor matrices-based framework can be naturally generalized to ensemble both partitional clustering and hierarchical clustering results as partitional clustering results. Experiments are conducted to demonstrate the effectiveness of our proposed approaches.

There are several avenues for future work. First, we plan to investigate the techniques for scaling up the ensemble process to large-scale datasets. Second, our studies show that selecting a relatively smaller subset is likely to produce better ensemble results. One interesting question is how to determine the ensemble size. Another interesting yet related direction is that, rather than picking representative dendrograms, we can associate every generated dendrogram with a weight. So, when considering the ensemble, dendrograms with larger weights can contribute more than can dendrograms with smaller weights. Third, another aspect of interest is to provide a formal analysis on cluster separation enhancement using transitive dissimilarity.

REFERENCES

E. N. Adams. 1986. N-trees as nestings: Complexity, similarity, and consensus. *Journal of Classification* 3, 299–317. 10.1007/BF01894192.

E. N. Adams III. 1972. Consensus techniques and the comparison of taxonomic trees. *Systematic Zoology* 21, 4, 390–397.

R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. 1999. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing* 1073–1085.

N. Ailon and M. Charikar. 2005. Fitting tree metrics: Hierarchical clustering and phylogeny. In *Proceedings of the Symposium on Foundations of Computer Science*. 73–82.

J. Azimi and X. Fern. 2009. Adaptive cluster ensemble selection. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'09)*. 992–997.

L. Breiman and L. Breiman. 1996. Bagging predictors. *Machine Learning* 24, 2, 123–140, Aug. 1996.

L. Breiman and L. Breiman. 2001. Random forests. *Machine Learning* 5–32.

C. Ding, X. He, H. Xiong, H. Peng, and S. R. Holbrook. 2006. Transitive closure and metric inequality of weighted graphs: Vdetecting protein interaction modules using cliques. *International Journal of Data Mining and Bioinformatics* 1, 162–177.

M. Farach, T. M. Przytycka, and M. Thorup. 1995. On the agreement of many trees. *Information Processing Letter* 55, 297–301.

X. Z. Fern and C. E. Brodley. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*. ACM, New York, NY, 36.

X. Z. Fern and W. Lin. 2008. Cluster ensemble selection. *Statistical Analysis and Data Mining* 1, 128–141.

C. Fraley and A. E. Raftery. 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal* 41, 578–588, 1998.

A. Gionis, H. Mannila, and P. Tsaparas. 2005. Clustering aggregation. In *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*. 341–352.

T. Gonzalez. 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38, 293–306, 1985.

M. Hossain, S. M. Bridges, Y. Wang, and J. E. Hodges. 2012. An effective ensemble method for hierarchical clustering. In *Proceedings of the 5th International C* Conference on Computer Science and Software Engineering*. ACM, 18–26.

A. Jain and R. Dubes. 1998. *Algorithms for Clustering Data*. Prentice Hall advanced reference series. Prentice Hall, 1988.

M. Jalalat-evakilkandi and A. Mirzaei. 2010. A new hierarchical-clustering combination scheme based on scatter matrices and nearest neighbor criterion. In *Proceedings of the 2010 5th International Symposium on Telecommunications (IST'10)*. IEEE, 904–908.

K. Koutroumbas, I. Tsagouri, and A. Belehaki. 2010. On the clustering of foF2 time series corresponding to disturbed ionospheric periods. *Advances in Space Research* 45, 9, 1129–1144.

M. K. Kuhner and J. Felsenstein. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Molecular Biology and Evolution* 11, 3, 459–68.

T. Li and C. Ding. 2008. Weighted consensus clustering. In *Proceedings of the SIAM International Conference on Data Mining*. 798–809.

T. Li, C. Ding, and M. I. Jordan. 2007. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of the 2007 7th IEEE International Conference on Data Mining (ICDM'07)*. IEEE Computer Society, Washington, DC, 577–582.

T. Li, M. Ogihara, and S. Ma. 2004. On combining multiple clusterings. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management (CIKM'04)*. ACM, New York, NY, 294–303.

T. Li, M. Ogihara, and S. Ma. 2010. On combining multiple clusterings: an overview and a new perspective. *Applied Intelligence* 33, 2, 207–219.

Y. Lu and Y. Wan. 2012. PHA: A fast potential-based hierarchical agglomerative clustering method. *Pattern Recognition* 46, 5, 1227–1239, May 2013.

D. Luo, C. Ding, H. Huang, and F. Nie. 2011. Consensus spectral clustering in near-linear time. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering (ICDE'11)*. IEEE Computer Society, Washington, DC, 1079–1090.

H. D. Meyer, H. Naessens, and B. D. Baets. 2004. 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.

A. Mirzaei and M. Rahmati. 2008. Combining hierarchical clusterings using min-transitive closure. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR'08)*. IEEE, 1–4.

A. Mirzaei and M. Rahmati. 2010. A novel hierarchical-clustering-combination scheme based on fuzzy-similarity relations. *IEEE Transactions on Fuzzy Systems* 18, 1, 27–39.

A. Mirzaei, M. Rahmati, and M. Ahmadi. 2008. A new method for hierarchical clustering combination. *Intelligent Data Analysis* 12, 549–571.

S. Monti, P. Tamayo, J. Mesirov, and T. Golub. 2003. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning* 52, 91–118.

J. Podani. 2000. Simulation of random dendrograms and comparison tests: Some comments. *Journal of Classification* 17, 123–142.

E. Rashedi and A. Mirzaei. 2011. A novel multi-clustering method for hierarchical clusterings based on boosting. In *Proceedings of the 2011 19th Iranian Conference on Electrical Engineering (ICEE'11)*. IEEE, 1–4.

D. F. Robinson and L. R. Foulds. 1981. Comparison of phylogenetic trees. *Mathematical Bioscience*, 53, 131–147.

F. J. Rohlf and D. R. Fisher. 1968. Tests for hierarchical structure in random data sets. *Systematic Zoology* 17, 4, 407–412.

R. E. Schapire and Y. Singer. 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37, 297–336, 1999.

R. R. Sokal and F. J. Rohlf. 1962. The comparison of dendrograms by objective methods. *Taxon*, 11, 2, 1962.

A. Strehl and J. Ghosh. 2003. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3, 583–617, March 2003.

C. A. Sugar and G. M. James. 2003. Finding the number of clusters in a data set: An information theoretic approach. *Journal of the American Statistical Association* 98, 750–763, 2003.

D. Swofford. 1991. When are phylogeny estimates from molecular and morphological data incongruent? In M. M. Miyamoto and J. Cracraft, editors, *Phylogenetic Analysis of DNA Sequences*. Oxford University Press, 295–333.

P.-N. Tan, M. Steinbach, and V. Kumar. 2005. *Introduction to Data Mining* (1st ed.). Addison-Wesley Longman, Boston, MA.

R. Tibshirani, G. Walther, and T. Hastie. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B* 63, 2, 411–423.

A. Topchy, A. Jain, and W. Punch. 2005. Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 12, 1866–1881.

M. Wilkinson. 1994. Common cladistic information and its consensus representation: Reduced adams and reduced cladistic consensus trees and profiles. *Systematic Biology*, 43, 3, 343–368, 1994.

D. H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5, 241–259, 1992.

J. Wu, H. Xiong, and J. Chen. 2009. Towards understanding hierarchical clustering: A data distribution perspective. *Neurocomputing*, 72, 10–12, 2319–2330, 2009.

Y. Zhao and G. Karypis. 2002. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM'02)*. ACM, New York, NY, 515–524.

L. Zheng and T. Li. 2011. Semi-supervised hierarchical clustering. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining (ICDM'11)*, 982–991, 2011.

L. Zheng, T. Li, and C. H. Q. Ding. 2010. Hierarchical ensemble clustering. In *ICDM'10*, 1199–1204, 2010.

Received July 2013; revised January 2014; accepted March 2014