Hierarchical Correlation Clustering and Tree Preserving Embedding

Morteza Haghir Chehreghani  
Chalmers University of Technology  
Gothenburg, Sweden  
morteza.chehreghani@chalmers.se

Mostafa Haghir Chehreghani  
Amirkabir University of Technology (Tehran Polytechnic)  
Tehran, Iran  
mostafa.chehreghani@gmail.com

Abstract

We propose a hierarchical correlation clustering method that extends the well-known correlation clustering to produce hierarchical clusters applicable to both positive and negative pairwise dissimilarities. Then, in the following, we study unsupervised representation learning with such hierarchical correlation clustering. For this purpose, we first investigate embedding the respective hierarchy to be used for tree preserving embedding and feature extraction. Thereafter, we study the extension of minimax distance measures to correlation clustering, as another representation learning paradigm. Finally, we demonstrate the performance of our methods on several datasets.

1. Introduction

Data clustering plays an essential role in unsupervised learning and exploratory data analysis. It is used in a variety of applications including web mining, network analysis, image segmentation, bioinformatics, user analytics and knowledge management. Its goal is to partition the data into groups in a way that the objects in the same cluster are more similar according to some criterion, compared to the objects in different clusters. Many clustering methods partition the data into $K$ flat clusters for example, $K$-means [55], spectral clustering [62,68] and correlation clustering [8]. In many applications, however, the clusters are preferred to be presented at different levels, encompassing both high-level and detailed information. Hierarchical clustering is useful to produce such structures, usually encoded by a dendrogram. A dendrogram is a tree data structure where each node corresponds to a cluster, with the leaf nodes (those at the bottom of the tree) containing only one object. Higher-level clusters are formed by aggregating lower-level clusters and the inter-cluster dissimilarity between them.

Hierarchical clustering can be performed either in an agglomerative (i.e., bottom-up) or in a divisive (i.e., top-down) manner [56]. Agglomerative methods are often computationally more efficient, making them more popular in practice [64]. In both approaches, the clusters are aggregated or split based on various criteria, such as single, average, centroid, complete and Ward. Several studies aim to improve these methods. The works in [49,52] focus on the statistical significance of hierarchical clustering. [24,25,65] formulate this problem as an optimization problem and propose approximate solutions. [82] considers multiple dissimilarities for a pair of clusters, and [11,17] suggest merging multiple clusters at each step instead of one. [6] employs global information to eliminate the influence of noisy similarities, and [19] proposes to apply agglomerative methods to small subsets of the data instead of individual data objects. [33,38] augment agglomerative methods with probabilistic models, and finally, [23,60] propose efficient but approximate methods for hierarchical clustering.

On the other hand, most clustering methods, either flat or hierarchical, assume non-negative pairwise (dis)similarities. However, in several practical applications, pairwise similarities can be any real number, positive or negative. For example, it could be preferable for a user or oracle to indicate whether two objects are similar (considered a positive relation) or dissimilar (considered a negative relation), rather than solely providing a positive (non-negative) pairwise similarity, even if the two objects are dissimilar. The former approach yields more precise information because, in the latter scenario, the dissimilarity between two objects (i.e., zero similarity) could be confused with a lack of available information. Some relevant applications for this setting include image segmentation with higher order correlation information [47,48], webpage segmentation [12], community detection over graphs [67], social media mining [73], analysis of connections over web [43], dealing with attraction/rejection data [26], automated label generation from clicks [3] and entity resolution [7,34].

Hence, a specialized clustering model known as correlation clustering has been developed to work with such data. This model was first introduced on the graphs with only $+1$ or $−1$ pairwise similarities [7,8], and then was generalized to the graphs with arbitrary positive or negative
edge weights [5, 13, 26]. The original model obtains the number of clusters automatically. The variant in [20] limits the number of clusters to fixed \( K \) clusters. Semidefinite programming (SDP) relaxation provides tight approximation bounds in particular for maximizing the agreements [13, 58], although it is computationally inefficient in practice [74]. Then, [15, 74] provide efficient greedy algorithms based on local search and Frank-Wolfe optimization with a fast convergence rate.

However, all of these methods produce flat correlation clusters. In this paper, we first propose a Hierarchical Correlation Clustering (HCC) method that handles both positive and negative pairwise (dis)similarities and produces clusters at different levels (Section 3). To the best of our knowledge, this work is one the first extensions of the well-known correlation clustering to hierarchical clustering.\(^1\) A hierarchical correlation clustering, also called HCC, is developed in [76]. This method offers a 0.4767 approximation, but lacking experimental evaluation. Furthermore, unlike our method, the method in [76] does not follow the generic agglomerative clustering procedure.

We note that unlike flat clustering, hierarchical clustering yields an ordering among the objects, i.e., objects that join earlier in the hierarchy are closer to each other than those that join at later steps. This implies that hierarchical clustering induces a new (dis)similarity measure between the objects, connected to the way the objects join each other to form clusters at different levels. Thereby, in the following, we consider two representation learning methods related to hierarchical clustering and study their adaptation to hierarchical correlation clustering. This enables us to not only use HCC for producing hierarchical clusters, but also to employ it for computing a suitable similarity/distance measure as an intermediate data processing step.

One way to perform representation learning from hierarchical clustering is to compute an embedding that corresponds to the respective hierarchy. Tree preserving embedding [69, 70] is a method that achieves this for the special case of single linkage method. Later, [21] develops tree preserving embedding for various standard agglomerative clustering methods. We then adapt these works (in particular the later work [21]) to develop a tree preserving embedding for HCC dendrograms (Section 4) where the embedded features can be used as a set of new features for an arbitrary downstream task. In this way, we can investigate HCC for the purpose of computing relevant features for a probabilistic method such as Gaussian Mixture Model (GMM), instead of solely using HCC for the purpose of hierarchical clustering. This enables us to apply a method like GMM for clustering the pairwise similarities that can be positive or negative numbers, a task that was not possible before.

Another representation learning paradigm that we study is called minimax dissimilarity, a graph-based method that is tightly connected to hierarchical clustering. It provides a sophisticated way to infer transitive relations and extract manifolds and elongated clusters in an unsupervised way [14, 32, 46, 54]. Thereby, for the first time, we study minimax dissimilarities on the graphs with positive and negative (dis)similarities, i.e., with correlation clustering (Section 5). We show that using minimax dissimilarities with correlation clustering not only helps for extracting elongated patterns, but also yields a significant reduction in the computational complexity, i.e., from NP-hardness to a polynomial runtime.

We finally perform several experiments on various datasets to demonstrate the effectiveness of our methods in different settings (Section 6).

### 2. Notations and Definitions

A dataset is characterized by a set of \( n \) objects with indices \( O = \{1, \ldots, n\} \) and a pairwise similarity or dissimilarity matrix. An \( n \times n \) matrix \( S \in \mathbb{R}^{n \times n} \) represents the pairwise similarities between the objects, whereas, the pairwise dissimilarities are shown by matrix \( D \in \mathbb{R}^{n \times n} \). Both of similarities and dissimilarities can be positive or negative. This property allows us to convert the pairwise similarities to dissimilarities by a simple transformation such as \( D = -S \), i.e., the pairwise dissimilarities are obtained by the negation of the similarities and vice versa.\(^2\) The objects and the pairwise (dis)similarities are represented by graph \( G = (O, S) \) or \( G = (O, D) \).

A cluster is represented by a set, e.g., \( v \), which includes the objects belong to that. The function \( dis(u, v) \) denotes the inter-cluster dissimilarity between clusters \( u \) and \( v \) that can be defined according to different criteria. A hierarchical clustering solution can be represented by a dendrogram \( T \) defined as a rooted ordered tree such that, i) each node \( v \) in \( T \) includes a non-empty subset of the objects corresponding to a cluster, i.e., \( v \subseteq O, |v| \geq 1, \forall v \in T \), with the leaf nodes including distinct single objects, and ii) the overlapping clusters are ordered, i.e., if \( u \cap v \neq \emptyset \), then either \( u \subseteq v \) or \( v \subseteq u \). The latter condition implies that between every two overlapping nodes an ancestor-descendant relation holds, i.e., \( u \subseteq v \) indicates \( v \) is an ancestor of \( u \), and \( u \) is a descendant of \( v \).

The clusters at the lowest level, called leaf clusters/node, are the individual distinct objects, i.e., \( v \) is a leaf cluster if

\(^1\)We note the so-called hierarchical correlation clustering methods proposed in [2, 36, 53] are irrelevant to the well-studied correlation clustering problem [7, 8]; they study for example the correlation coefficients for high-dimensional data.

\(^2\)Such a nonparametric transformation resolves the issues related to obtaining a proper similarity measure from pairwise dissimilarities. For example, with kernels, e.g., RBF kernels, finding the optimal parameter(s) is often crucial and nontrivial, and the optimal parameters occur inside a very narrow range [61, 80]. Moreover, the methods we develop in this paper are unaffected by the choice of the transformation \( D \); for example in Algorithm 1, we only use the pairwise similarities \( S \).
and only if \(|v| = 1\). A cluster at a higher level contains the union of the objects of its children. The root of a dendrogram is defined as the cluster at the highest level which has the maximum size, i.e., all other clusters are its descendants. \(\text{linkage}(v), v \in T\) returns the dissimilarity between the children of \(v\) based on the criterion used to compute the dendrogram (i.e., \(diss(c_1, c_r)\) where \(c_1\) and \(c_r\) indicate the two child clusters of \(v\)). For simplicity of explanation, w.l.g., we assume every non-leaf cluster has two children. The level of cluster \(v\), i.e., \(level(v)\), is determined by

\[
level(v) = \max(\text{level}(c_1), \text{level}(c_r)) + 1. \quad (1)
\]

For the leaf clusters, \(\text{level}()\) and \(diss()\) return 0. Every connected subtree of \(T\) whose leaf clusters contain only individual objects from \(O\) constitutes a dendrogram on this subset of objects. We require that every common node present in both \(T\) and the subtree must have the same child nodes or clusters. We use \(T_F\) to refer to the set of all (sub)dendrograms obtained in this way from \(T\).

3. Hierarchical Correlation Clustering

Agglomerative methods begin with each object in a separate cluster, and then at each round, combine the two clusters that have a minimal dissimilarity according to a criterion (defined by the \(diss(\ldots)\) function) until only one cluster remains. For example, the single linkage (SL) criterion \([71]\) defines the dissimilarity between two clusters as the dissimilarity between their nearest members (\(diss(u, v) = \min_{i \in u, j \in v} D_{i,j}\)), whereas, complete linkage (CL) \([50]\) uses the dissimilarity between their furthest members (\(diss(u, v) = \max_{i \in u, j \in v} D_{i,j}\)). On the other hand, the average linkage (AL) criterion \([72]\) considers the average of the inter-cluster dissimilarities as the dissimilarity between the two clusters (\(diss(u, v) = \sum_{i \in u, j \in v} \frac{D_{ij}}{|u||v|}\)). These methods can be shown to be shift-invariant, as mentioned in Proposition 1 \([18]\).

**Proposition 1** Single linkage, complete linkage and average linkage methods are invariant w.r.t. the shift of the pairwise dissimilarities by an arbitrary real number \(\alpha\).

Thus, we can still use these methods even with possibly negative pairwise dissimilarities as shifting the pairwise dissimilarities (by a large enough constant) to make them non-negative does not change the solution.

However, clustering the data consisting of positive and negative dissimilarities is usually conducted by correlation clustering. Thus, despite the applicability of single linkage, average linkage and complete linkage methods, we propose a novel hierarchical clustering consistent with the standard correlation clustering, called Hierarchical Correlation Clustering (HCC). This method is thus adapted to positive/negative pairwise (dis)similarities, and as our experiments confirm, it outperforms the other methods (i.e., SL, CL, and AL) when applied to such data.

The cost function for flat (standard) correlation clustering accounts for disagreements (i.e., negative similarities inside clusters and positive similarities between clusters) and is written by \([20]\)

\[
R_{CC}(v_1, \ldots, v_K; S) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in v_k} (|S_{ij}|-S_{ij}) + \frac{1}{2} \sum_{k=1}^{K} \sum_{k' \neq k} \sum_{i \in v_k, j \in v_{k'}} (|S_{ij}|+S_{ij}), \quad (2)
\]

where \(K\) is the number of clusters and \(v_k\)’s indicate the different clusters.

We may rewrite the cost function as

\[
R_{CC}(v_1, \ldots, v_K; S) = -\frac{1}{2} \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{i \in v_k, j \in v_{k'}} S_{ij} + \frac{1}{2} \sum_{k=1}^{K} \sum_{k' \neq k} \sum_{i \in v_k, j \in v_{k'}} |S_{ij}| + \frac{1}{2} \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{i \in v_k, j \in v_{k'}} S_{ij}, \quad (3)
\]

We then have

\[
R_{CC}(v_1, \ldots, v_K; S) \equiv constant + \sum_{k=1}^{K} \sum_{k'=1}^{K} \sum_{i \in v_k, j \in v_{k'}} D_{ij}. \quad (4)
\]

Therefore, correlation clustering aims to minimize the inter-cluster similarities, and in other words, it maximizes the inter-cluster dissimilarities. This formulation in Eq. 4 inspires us for a consistent way of defining a new inter-cluster dissimilarity function for hierarchical (agglomerative) correlation clustering of positive and negative (dis)similarities. At each step, we merge the two clusters that have a minimal dissimilarity (or a maximal similarity), where we define the dissimilarity between the two clusters

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
u and v as

\[
\text{dis}^{CC}(u, v) = \sum_{i \in u} \sum_{j \in v} D_{ij} = -\sum_{i \in u} \sum_{j \in v} S_{ij}. \tag{5}
\]

We emphasize that HCC is consistent with the generic agglomerative clustering framework applied with, for example, single linkage, average linkage, complete linkage and other criteria. The only difference is the definition of the inter-cluster dissimilarity function where with HCC we use \(\text{dis}^{CC}(\cdot, \cdot)\) defined in Eq. 5 (inspired from the cost function of flat correlation clustering). Other than this, the algorithmic procedure is consistent. Algorithm 1 in Appendix A describes the pseudocode of the HCC algorithm.

4. Feature Extraction from HCC

As mentioned, HCC represents the relations between objects according to the way they join to form the hierarchy. In this section, we use this intuition and develop a data representation consistent with HCC. For this purpose, we adapt the methods in [69, 70] and in particular [21] to our setting. Hereby, we first introduce distance functions over HCC, and then, investigate the embedding of such a distance function. This procedure leads to obtaining a set of features from HCC for each object which then can be used in the downstream task.

4.1. Distance functions over HCC

Given dendrogram \(T\), each cluster \(v \in T\) represents the root of a dendrogram \(T' \in T\). \(T'\) admits the properties of its root cluster, i.e., \(\text{level}(T') = \max_{u \in T'} \text{level}(u) = \text{level}(v)\) and \(\text{linkage}(T') = \max_{u \in T'} \text{linkage}(u) = \text{linkage}(v)\), since the root cluster has the maximum linkage and level among the clusters in \(T'\). Hence, in this way, we define functions such as \(\text{level}(\cdot)\) and \(\text{linkage}(\cdot)\) for the dendrograms as well.

The \(\text{linkage}(\cdot)\) function may seem to be a natural choice for defining a distance function over a HCC dendrogram. Specifically, one can define the dendrogram-based distance function \(X_{ij}\) over dendrogram \(T\) between \(i, j \in O\) as

\[
X_{ij} = \min \text{linkage}(T') \quad \text{s.t.} \quad i, j \in T', \text{ and } T' \in T. \tag{6}
\]

This choice corresponds to the linkage of the smallest cluster that includes both \(i\) and \(j\). This in particular makes sense for the single linkage dendrogram, and it would be consistent with the tree-preserving embedding in [69, 70]. If the original dissimilarity matrix \(D\) contains negative values, then using Proposition 1, one can sufficiently shift the pairwise dissimilarities to make all of them non-negative, without changing the structure of the dendrogram and the order of the clusters. Therefore, the conditions for a valid distance function including non-negativity still hold.

However, for the HCC dendrogram, the linkage function might not fulfill the conditions for a distance function. For example, consider a set of \(n\) objects where all the pairwise similarities are \(+1\), i.e., the dissimilarities are thus \(-1\). Then, the linkage function will always return negative values which would violate the non-negativity condition of a valid distance function. On the other hand, the HCC linkage \(\text{dis}^{CC}(\cdot, \cdot)\) is not shift-invariant (similar to the standard flat correlation clustering [18]) and we cannot use the shift trick in Proposition 1. Let \(D^{\alpha}\) shows the shifted pairwise dissimilarities, i.e., \(D^{\alpha}_{i,j} = D_{ij} + \alpha\). Then, \(\text{dis}^{CC}(u, v)\) between two clusters \(u\) and \(v\) based on \(D^{\alpha}\) is given by

\[
\text{dis}^{CC}(u, v) = \sum_{i \in u, j \in v} D^{\alpha}_{i,j} = \sum_{i \in u, j \in v} (D_{i,j} + \alpha) = \sum_{i \in u, j \in v} D_{i,j} + \alpha |u| |v|. \tag{7}
\]

With \(\alpha > 0\), this shift would induce a bias for the HCC linkage to choose imbalanced clusters. In other words, \(\text{dis}^{CC}(\cdot, \cdot)\) is not shift-invariant and we cannot shift the pairwise dissimilarities in \(D\) to make them nonnegative.

Therefore, we consider another choice, i.e., the \(\text{level}(\cdot)\) function used in [21]. It is nonnegative and satisfies the desired conditions. Then, \(X_{ij}\) is now computed by

\[
X_{ij} = \min \text{level}(T') \quad \text{s.t.} \quad i, j \in T' \text{ and } T' \in T. \tag{8}
\]

Intuitively, Eq. 8 selects the level of the smallest cluster/dendrogram that contains both \(i\) and \(j\). The lower the level at which the two objects join, the greater the similarity or proximity between them, indicating a closer relationship in the hierarchical clustering structure. In other words, a higher level in the dendrogram signifies a later fusion of the two objects, suggesting that they share fewer common characteristics compared to objects fused at lower levels.

4.2. Embedding the HCC-based distances

After applying the distance function in Eq. 8, we obtain an \(n \times n\) matrix representing pairwise HCC-based distances among objects. It is usually preferred to obtain vector-based representations for objects rather than pairwise distances. Models like Gaussian Mixture Models (GMMs) which involve mixture density estimation (see, e.g., [75]), can only be applied to vectors. Additionally, working with vector-based data simplifies feature selection. Hence, it is desired to compute an embedding of the objects into a new space, so that their pairwise squared Euclidean distances in the new space match their pairwise distances obtained from the dendrogram.

The matrix of pairwise distances \(X\) obtained via Eq. 8 induces an ultrametric [21, 51]. The primary distinction between a metric and an ultrametric is that the addition operation in the triangle inequality for a metric is replaced by a maximum operation, i.e., with ultrametric we have
2. With applying the transformation in step 1, the inequality in Eq. 9 is satisfied. Conversely, if \( X_{ij} > X_{ik} \), it implies that objects \( i \) and \( k \) are included in the same cluster (shown by \( c_{i,k} \)) before \( i \) and \( j \) join (to form cluster \( c_{i,j} \)). The bottom-up hierarchical clustering process then continues until \( c_{i,j,k} \) is formed, encompassing all three objects \( i, j, k \). Notice that \( i \) and \( k \) join \( j \) simultaneously via \( c_{i,k} \). In this case, according to Eqs. 8 and 9, and the relationships illustrated in Figure 1, we conclude

\[
X_{ij} = X_{kj} \leq \max(X_{ik}, X_{kj}).
\]  

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

The connection between ultrametric and trees is well-established in mathematics \([32, 57]\). Here we instantiate it to our setting via making the argument in \([21]\) more accurate. It is evident that when \( X_{ij} \leq X_{ik} \), the inequality in Eq. 9 is satisfied. Conversely, if \( X_{ij} > X_{ik} \), it implies that objects \( i \) and \( k \) are included in the same cluster (shown by \( c_{i,k} \)) before \( i \) and \( j \) join (to form cluster \( c_{i,j} \)). The bottom-up hierarchical clustering process then continues until \( c_{i,j,k} \) is formed, encompassing all three objects \( i, j, k \). Notice that \( i \) and \( k \) join \( j \) simultaneously via \( c_{i,k} \). In this case, according to Eqs. 8 and 9, and the relationships illustrated in Figure 1, we conclude

\[
X_{ij} = X_{kj} \leq \max(X_{ik}, X_{kj}).
\]  

3. We calculate the \( n \times l \) matrix \( A \) as the matrix of new data features:

\[
A = \frac{1}{l} (Y_l Z_l)^{1/2},
\]  

where \( Y_l = (y_1, \ldots, y_l) \) and \( Z_l = \text{diag}(z_1, \ldots, z_l) \), where \( l \) specifies the dimensionality of the new vectors.

In the embedded space, the new dimensions are arranged based on their corresponding eigenvalues. One can opt to select only the most significant ones, rather than utilizing all of them. Therefore, computing such an embedding also offers the benefit of feature selection.

We also note that many clustering methods can be written in matrix factorization form via for example spectral K-means \([28]\). This induces an embedding and hence a set of relevant features. However, for general positive/negative similarity matrices no exact embedding might be feasible due to violating positive semidefinite-ness. The method that we described here provides a solution to this challenge, i.e., enables us to extract features when the base pairwise similarities are positive and negative.

5. Correlation Clustering and Minimax Dissimilarities

Finally, we study minimax dissimilarities for correlation clustering, a graph-based method that corresponds to constructing a hierarchical clustering.

Given graph \( G(O, D) \), the minimax (MM) dissimilarity between \( i \) and \( j \) is defined as

\[
D_{ij}^{MM} = \min_{p \in \mathcal{P}_{ij}(G)} \max_{1 \leq |p| \leq l-1} D_{p(p+1)},
\]  

where \( \mathcal{P}_{ij}(G) \) is the set of all paths between \( i \) and \( j \) over \( G(O, D) \). Each path \( p \) is specified by a sequence of object indices, i.e., \( p(l) \) indicates the \( l^{th} \) object on the path.

Minimax dissimilarities enable a clustering algorithm to capture the inherent patterns and manifolds in an unsupervised and nonparametric way by extracting the transitive connections \([16, 32]\). For example, if object \( i \) is similar to object \( j \), \( j \) is similar to \( k \), and \( k \) is similar to \( l \), then the minimax dissimilarity between \( i \) and \( l \) will be small, even though their direct dissimilarity might be large. The reason is that minimax dissimilarity finds the connectivity path \( i \to j \to \cdots \to k \to l \) and connects \( i \) and \( l \) via this path. This property is helpful in finding elongated clusters and manifolds of arbitrary shapes in an unsupervised way.

Minimax dissimilarities have been so far solely used with nonnegative pairwise dissimilarities. In the case of possible negative dissimilarities, we may use a trick similar to Proposition 1. As shown in Lemma 1, minimax paths are invariant w.r.t. the shift of the pairwise dissimilarities.

\textbf{Lemma 1} Consider graphs \( G(O, D) \) and \( G^{\alpha}(O, D^{\alpha}) \), where the pairwise dissimilarities (edge weights) in
$G^\alpha(O, D^\alpha)$ are shifted by constant $\alpha$, i.e., $D^\alpha_{ij} = D_{ij} + \alpha$. Then, the minimax paths between every pair of objects $i$ and $j$ are identical on graphs $G(O, D)$ and $G^\alpha(O, D^\alpha)$.

All the proofs are in Appendix B. Hence, given a dissimilarity matrix $D$, one can subtract $\alpha := \min(D)$ from all the elements to obtain $D^\alpha$. Then, the minimax dissimilarities can be computed from $G^\alpha(O, D^\alpha)$.

As mentioned, correlation clustering on an arbitrary similarity matrix $S$ is NP-hard [8, 26]. Therefore, using minimax (dis)similarities with correlation clustering not only helps for extracting elongated complex patterns, but also yields a significant reduction in the computational complexity, i.e., from NP-hardness to a polynomial runtime.

Among the approximate algorithms proposed for correlation clustering on complete graphs with discrete weights, the method in [5] provides a 3-factor approximation. with a randomly selected unclustered object in the graph, this method greedily finds the object’s positive neighbors (those with similarity +1) to form a new cluster. Then, it repeats this procedure for the remaining objects. One can conclude that in the optimal solution of correlation clustering on graph $G(O, S^{MM})$, only the positive neighbors of an object will be in the same cluster as the object itself, i.e., interestingly the 3-factor approximation algorithm in [5] becomes exact when applied to $G(O, S^{MM})$ (Theorem 2).

Theorem 1 The optimal clusters of the correlation clustering on graph $G(O, S^{MM})$ are equal to the connected components of graph $G(O, S')$.

Theorem 2 Assume the edge weights of graph $G(O, S)$ are either +1 or −1. Then, the approximate algorithm in [5] is exact when applied to the minimax similarities, i.e., to graph $G(O, S^{MM})$.

6. Experiments

In this section, we describe our experimental results on various datasets. We compare our methods with single linkage (SL), complete linkage (CL) and average linkage (AL).

As mentioned, there are several improvements over these basic methods. However, such contributions are orthogonal to our contribution. Moreover, it is unclear how such improvements can be extended to the dissimilarities that can be both positive and negative. Thus, we limit our baselines to these methods which as mentioned in Proposition 1, are applicable to such data.

In our studies, we have access to the true labels. Therefore, consistent with several previous studies on hierarchical clustering, e.g. [4, 9, 29], we evaluate the results according to the following criteria: i) Normalized Mutual Information (MI) [79] that measures the mutual information between the true and the estimated clustering solutions, and ii) Normalized Rand score (Rand) [41] that obtains the similarity between the two solutions. We do not use the labels to infer the clustering solution; they are only used for evaluation. Therefore, we are still in the unsupervised setting where the ground-truth labels play the role of an external evaluator.

In Appendix C, we present additional experimental results, in particular on datasets from other domains.

6.1. HCC on UCI data

We first investigate the hierarchical correlation clustering on the following six UCI datasets [45]. (i) Breast Tissue: includes electrical impedance measurements of freshly excised 106 tissue samples from the breast. The number of clusters is 6. (ii) Cardiotocography: contains 2126 measurements of fetal heart rate and uterine contraction features on cardiotocograms in 10 clusters. (iii) Image Segmentation: contains 2310 samples from images of 7 outdoor clusters. (iv) ISOLET: 7797 samples consisting of spoken attributes of different letters (26 clusters). (v) Leaf: 340 images of leaf specimens originating from 40 different plant species (clusters) each described by 16 attributes. (vi) One-Hundred Plant: 1600 samples of leaves each described by 64 features, from in total 100 types (clusters).

The ground-truth labels are shown by $c^*$, i.e., $c^*_i$ shows the true label for object $i$. We assume an oracle reveals the pairwise similarities $S$ according to the (flip) noise parameter $\eta$. If $c^*_i = c^*_j$ then $S_{ij} = \mathcal{U}(0, 1)$ with probability $1 - \eta$ and $S_{ij} = \mathcal{U}(-1, 0)$ with probability $\eta$. If $c^*_i \neq c^*_j$ then $S_{ij} = \mathcal{U}(-1, 0)$ with probability $1 - \eta$ and $S_{ij} = \mathcal{U}(0, 1)$ with probability $\eta$. The function $\mathcal{U}(\ldots)$ returns a uniform random number within the specified range. For each $\eta$ we repeat the experiments 20 times and report the average results. This setup provides a systematic approach to study the robustness of various methods to noise.

Figure 2 shows the results for different datasets as a function of the noise level $\eta$ w.r.t. MI. Rand scores shown in Figure 5 in Appendix C exhibit a consistent behavior. We
6.2. Tree preserving embedding on UCI data

In the following, we investigate tree-preserving embedding and feature extraction. After computing the embeddings from different hierarchical clustering methods, we apply Gaussian Mixture Model (GMM) to the extracted features and evaluate the final clustering using the ground-truth solution. This kind of embedding enables us to apply methods like GMM to positive and negative pairwise similarities, a task that was not possible before. Since the extracted features appear in the form of vectors, thus, the final clustering method is not limited to GMM, and in principle, any numerical clustering can benefit from this embedding.

Table 1 shows the results for different UCI datasets. We observe that the embeddings obtained by HCC (i.e., ‘HCC+GMM’) yield significantly better results compared to the embeddings from other methods (i.e., ‘SL+GMM’, ‘CL+GMM’ and ‘AL+GMM’). It is worth noting that the results of embeddings (e.g., ‘HCC+GMM’) typically surpass the results obtained from the hierarchical clustering alone (e.g., ‘HCC’). This observation supports the idea that employing HCC to compute an embedding (for extracting new features) for a clustering method such as GMM may be advantageous, yielding superior results compared to using HCC exclusively for clustering purposes. This verifies why tree preserving embedding can be effective in general.

6.3. Experiments on Fashion-MNIST

Next, we investigate HCC and tree-preserving embedding on two randomly selected subsets of Fashion-MNIST dataset [81]. Fashion MNIST consists of $28 \times 28$ images of Zalando’s articles. Each subset consists of 5,000 samples/objects, where we compute the pairwise cosine similarities between them and then apply different methods. Table 2 shows the performance of different methods on these datasets. We observe that, consistent with the previous experiments, both ‘HCC’ and ‘HCC+GMM’ yield improving the results compared to the baselines. Furthermore, employing HCC to compute intermediate features for GMM (i.e., ‘HCC+GMM’) achieves higher scores compared to using ‘HCC’ alone for generating final clusters.

6.4. Correlation clustering with minimax dissimilarities

Finally, we investigate the use of minimax dissimilarities with correlation clustering. As mentioned, minimax dissimilarities are especially useful for extracting elongated and complex patterns in an unsupervised way. Thereby, we apply ‘minimax + correlation clustering’ to a number of datasets with visually elongated and arbitrarily shaped clusters, and compare the results with ‘correlation clustering’ alone. The datasets are shown in Figure 3. For each dataset, we simply construct the K-nearest neighbor graph (using

![](image-url)

**Figure 2.** MI score of different hierarchical clustering methods applied to UCI datasets, where x-axis shows the parameter $\eta$. **Table 1.** Performance of different tree preserving embedding methods on UCI datasets applied with GMM.

| method     | Breast Tissue | Cardiotocography | Image Segm. | ISOLET | Leaf | One-Hun. Plant |
|------------|---------------|------------------|-------------|--------|------|----------------|
| **SL**     | 0.006         | 0.007            | 0.008       | 0.009  | 0.008| 0.015          |
| **SL+GMM** | 0.093         | 0.077            | 0.120       | 0.239  | 0.192| 0.155          |
| **CL**     | 0.227         | 0.166            | 0.077       | 0.187  | 0.057| 0.081          |
| **CL+GMM** | 0.251         | 0.171            | 0.081       | 0.201  | 0.061| 0.140          |
| **AL**     | 0.542         | 0.519            | 0.391       | 0.518  | 0.257| 0.181          |
| **AL+GMM** | 0.550         | 0.513            | 0.422       | 0.522  | 0.240| 0.152          |
| **HCC**    | 0.903         | 0.900            | 0.987       | 0.945  | 0.938| 0.429          |
| **HCC+GMM**| **0.914**     | **0.911**        | **0.994**   | **0.960** | **0.941**| **0.462**     |

observe that among different methods, HCC performs significantly better and produces more robust clusters w.r.t. the noise parameter $\eta$. The results on Leaf and One-Hundred Plant are worse with all the methods. The reason is that these datasets are complex, having many clusters (respectively, 40 and 100 clusters) and fairly a small number of objects per cluster.
Table 2. Performance of different methods on MNIST and Fashion MNIST datasets. The embeddings by HCC yield better results.

| method   | Fashion MNIST 1 | Fashion MNIST 2 |
|---------|-----------------|-----------------|
|         | MI   | Rand  | MI   | Rand  |
| SL      | 0.322 | 0.206 | 0.241 | 0.196 |
| SL+GMM  | 0.411 | 0.335 | 0.384 | 0.340 |
| CL      | 0.403 | 0.293 | 0.546 | 0.379 |
| CL+GMM  | 0.478 | 0.426 | 0.574 | 0.362 |
| AL      | 0.464 | 0.468 | 0.602 | 0.534 |
| AL+GMM  | **0.608** | 0.551 | 0.647 | 0.553 |
| HCC     | 0.499 | 0.475 | 0.666 | **0.557** |
| HCC+GMM | 0.581 | **0.586** | **0.693** | 0.548 |

Figure 3. The datasets with arbitrarily shaped clusters, where ‘minimax + correlation clustering’ achieves perfect clustering.

Figure 4. Embeddings of vehicle motion trajectories computed via dynamic time wrapping and t-SNE [27, 39].

The ordinary Euclidean distance with a typical K such as 3). The edges in the nearest neighborhood receive a positive weight (e.g., +1), and a negative weight (e.g., -1) otherwise. We then apply either ‘minimax + correlation clustering’ or ‘correlation clustering’ to the resultant graph. We observe that for all the datasets, ‘minimax + correlation clustering’ yields the perfect clustering, i.e., MI and Rand are equal to 1. Whereas with ‘correlation clustering’ alone these scores are very low. We acknowledge the existence of other clustering methods, such as DBSCAN [30], capable of achieving perfect clustering on these datasets. However, these methods often involve crucial hyperparameters, the tuning of which can be challenging in unsupervised learning. To our knowledge, in addition to computational and theoretical benefits, ‘minimax + correlation clustering’ stands out as the sole method capable of achieving perfect results on datasets with elongated clusters of arbitrary shapes, and fulfills the following two promises: i) it eliminates the need to fix critical hyperparameters, a task often intricate in unsupervised learning, and ii) it automatically determines the correct number of clusters without requiring prior fixing.

In the following, we consider the interesting application of clustering vehicle motion trajectories for ensuring safety in self-driving [39]. In this application, 1,024 trajectories consisting of drive-by left, drive-by right and cut-in types are prepared, where some of them are collected in real-world and some others are generated using Recurrent Auto-Encoder GAN [27]. Next, dynamic time wrapping [10] and t-SNE [77] are employed to map the temporal data onto a two-dimensional space, as illustrated in Figure 4 [27, 39]. We then apply ‘minimax + correlation clustering’ to this data and compare it with ‘correlation clustering’. Similar to the datasets in Figure 3, ‘minimax + correlation clustering’ achieves perfect clustering with MI = Rand = 1 and accurately determines the correct number of clusters.

7. Conclusion

We proposed a new hierarchical clustering method, called HCC, suitable when the (dis)similarities can be positive or negative. This method is consistent with the general algorithmic procedure for agglomerative clustering and only differs in the way the inter-cluster dissimilarity function is defined. We then considered embedding the HCC dendrograms, which provides extracting useful features to apply for example GMM for clustering positive and negative similarities. In the following, we studied the use of minimax dissimilarities with correlation clustering and showed that it yields reduction in the computational complexity, in addition to a possibility for extracting elongated manifolds. Finally, we demonstrated the effectiveness of the methods on several datasets in different settings.

Acknowledgement

The work of Morteza Haghir Chehreghani was partially supported by the Swedish Research Council VR (grant number 2023-04809) and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
References

[1] Clustering datasets. https://github.com/millaan9/Clustering-Datasets/. 8
[2] Elke Achtert, Christian Böhm, Peer Kröger, and Arthur Zimek. Mining hierarchies of correlation clusters. In 18th International Conference on Scientific and Statistical Database Management, pages 119–128, 2006. 2
[3] Rakesh Agrawal, Alan Halverson, Krishnaram Kenthapadi, Nina Mishra, and Panayiotis Tsaparas. Generating labels from clicks. In International Conference on Web Search and Web Data Mining, WSDM, pages 172–181, 2009. 1
[4] Julien Ah-Pine. An efficient and effective generic agglomerative hierarchical clustering approach. J. Mach. Learn. Res., 19:42:1–42:43, 2018. 6
[5] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. J. ACM, 55(5):23:1–23:27, 2008. 2, 6, 13
[6] Maria-Florina Balcan, Yingyu Liang, and Pramod Gupta. Robust hierarchical clustering. J. Mach. Learn. Res., 15(1):3831–3871, 2014. 1
[7] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In 43rd Symposium on Foundations of Computer Science FOCS, 2002. 1, 2
[8] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. Machine Learning, 56(1-3):89–113, 2004. 1, 2, 6
[9] MohammadHossein Bateni, Soheil Behnezhad, Mahsa Derakhsh, MohammadTaghi Hajiaghayi, Raimondas Kiveris, Silvio Lattanzi, and Vahab S. Mirrokni. Affinity clustering: Hierarchical clustering at scale. In Advances in Neural Information Processing Systems (NIPS), pages 6864–6874, 2017. 6
[10] R. Bellman and R. Kalaba. On adaptive control processes. Automatic Control, IRE Transactions on, 4(2):1–9, 1959. 8
[11] Michel Bruynooghe. Méthodes nouvelles en classification automatique de données taxinomiques nombreuses. Statistique et analyse des données, 2(3):24–42, 1977. 1
[12] Deepayan Chakrabarti, Ravi Kumar, and Kunal Punera. A graph-theoretic approach to webpage segmentation. In International Conference on World Wide Web, WWW, pages 377–386, 2008. 1
[13] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In 44th Symposium on Foundations of Computer Science (FOCS), pages 524–533, 2003. 2
[14] Morteza Haghir Chehreghani. Classification with minimax distance measures. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pages 1784–1790, 2017. 2
[15] Morteza Haghir Chehreghani. Clustering by shift. In IEEE International Conference on Data Mining (ICDM), pages 793–798, 2017. 2
[16] Morteza Haghir Chehreghani. Unsupervised representation learning with minimax distance measures. Mach. Learn., 109(11):2063–2097, 2020. 5
[17] Morteza Haghir Chehreghani. Reliable agglomerative clustering. In International Joint Conference on Neural Networks, IJCNN, pages 1–8. IEEE, 2021. 1
[18] Morteza Haghir Chehreghani. Shift of pairwise similarities for data clustering. Mach. Learn., 112(6):2025–2051, 2023. 3, 4, 14
[19] Morteza Haghir Chehreghani, Hassan Abolhassani, and Mostafa Haghir Chehreghani. Improving density-based methods for hierarchical clustering of web pages. Data Knowl. Eng., 67(1):30–50, 2008. 1
[20] Morteza Haghir Chehreghani, Alberto Giovanni Busetto, and Joachim M. Buhmann. Information theoretic model validation for spectral clustering. In International Conference on Artificial Intelligence and Statistics (AISTATS), 2012. 2, 3
[21] Morteza Haghir Chehreghani and Mostafa Haghir Chehreghani. Learning representations from dendrograms. Mach. Learn., 109(10):1779–1802, 2020. 2, 4, 5
[22] Dongdong Cheng, Qingsheng Zhu, Jinlong Huang, Quan-wang Wu, and Lijun Yang. A novel cluster validity index based on local cores. IEEE Trans. Neural Networks Learn. Syst., 30(4):985–999, 2019. 8
[23] Michael Cochez and Hao Mou. Twister tries: Approximate hierarchical agglomerative clustering for average distance in linear time. In International Conference on Management of Data (ACM SIGMOD), pages 505–517, 2015. 1
[24] Vincent Cohen-Addad, Varun Kanade, Frederik Mallmann-Trenn, and Claire Mathieu. Hierarchical clustering: Objective functions and algorithms. In SODA, pages 378–397, 2018. 1
[25] Vincent Cohen-Addad, Varun Kanade, and Frederik Mallmann-Trenn. Hierarchical clustering beyond the worst-case. In Advances in Neural Information Processing Systems (NIPS), pages 6201–6209, 2017. 1
[26] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. Theor. Comput. Sci., 361(2-3):172–187, 2006. 1, 2, 6
[27] Andreas Demetriou, Henrik Alfsveg, Sadegh Rahrovar, and Morteza Haghir Chehreghani. A deep learning framework for generation and analysis of driving scenario trajectories. SN Comput. Sci., 4(3):251, 2023. 8
[28] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In Tenth ACM International Conference on Knowledge Discovery and Data Mining, SIGKDD, pages 551–556, 2004. 5
[29] Laxman Dhulipala, David Eisenstat, Jakub Lacki, Vahab S. Mirrokni, and Jessica Shi. Hierarchical agglomerative graph clustering in nearly-linear time. In International Conference on Machine Learning (ICML), 2021. 6
[30] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), pages 226–231, 1996. 8
[31] Miroslav Fiedler. Ultrametric sets in euclidean point spaces. ELA. The Electronic Journal of Linear Algebra, 3:23–30, 1998. 5
[68] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 1

[69] A. Shieh, T. B. Hashimoto, and E. M. Airoldi. Tree-preserving embedding. In *International Conference on Machine Learning (ICML)*, pages 753–760, 2011. 2, 4

[70] A. D. Shieh, T. B. Hashimoto, and E. M. Airoldi. Tree-preserving embedding. *Proceedings of the National Academy of Sciences*, 108(41):16916–16921, 2011. 2, 4

[71] Peter Henry Andrews Sneath. The application of computers to taxonomy. *Journal of General Microbiology*, 17:201–226, 1957. 3

[72] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *Uni. of Kansas Science Bulletin*, 38:1409–1438, 1958. 3

[73] Jiliang Tang, Yi Chang, Charu C. Aggarwal, and Huan Liu. A survey of signed network mining in social media. *ACM Comput. Surv.*, 49(3):42:1–42:37, 2016. 1

[74] Erik Thiel, Morteza Haghir Chehreghani, and Devdatt P. Dubhashi. A non-convex optimization approach to correlation clustering. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pages 5159–5166, 2019. 2

[75] D. M. Titterington, etc., A. F. M. Smith, and U. E. Makov. *Statistical analysis of finite mixture distribution*. Probability & Mathematical Statistics S. John Wiley & Sons, 1985. 4

[76] D. Vainstein, V. Chatziafratis, G. Citovsky, A. Rajagopalan, M. Mahdian, and Y. Azar. Hierarchical clustering via sketches and hierarchical correlation clustering. In *AISTATS*, pages 559–567, 2021. 2

[77] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 8

[78] Richard S. Varga and Reinhard Nabben. On symmetric ultrametric matrices. In *Numerical Linear Algebra*, 1993. 5

[79] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, 11:2837–2854, 2010. 6

[80] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 2

[81] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. 7

[82] Pelin Yildirim and Derya Birant. K-linkage: A new agglomerative approach for hierarchical clustering. 17:77–88, 2017. 1

[83] Gale Young and A. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938. 5