Machine Learning (2020) 109:1779–1802
https://doi.org/10.1007/s10994-020-05895-3

Learning representations from dendrograms

Morteza Haghir Chehreghani[1] · **Mostafa Haghir Chehreghani**[2]

Received: 11 November 2019 / Revised: 11 May 2020 / Accepted: 6 July 2020 / Published online: 16 August 2020
© The Author(s) 2020

Abstract
We propose unsupervised representation learning and feature extraction from dendrograms. The commonly used Minimax distance measures correspond to building a dendrogram with single linkage criterion, with defining specific forms of a level function and a distance function over that. Therefore, we extend this method to arbitrary dendrograms. We develop a generalized framework wherein different distance measures and representations can be inferred from different types of dendrograms, level functions and distance functions. Via an appropriate embedding, we compute a vector-based representation of the inferred distances, in order to enable many numerical machine learning algorithms to employ such distances. Then, to address the model selection problem, we study the aggregation of different dendrogram-based distances respectively in solution space and in representation space in the spirit of deep representations. In the first approach, for example for the clustering problem, we build a graph with positive and negative edge weights according to the consistency of the clustering labels of different objects among different solutions, in the context of ensemble methods. Then, we use an efficient variant of correlation clustering to produce the final clusters. In the second approach, we investigate the combination of different distances and features sequentially in the spirit of multi-layered architectures to obtain the final features. Finally, we demonstrate the effectiveness of our approach via several numerical studies.

Keywords Representation learning · Unsupervised learning · Ensemble method · Feature extraction · Dendrogram

Editors: Ira Assent, Carlotta Domeniconi, Aristides Gionis, Eyke Hüllermeier.

✉ Morteza Haghir Chehreghani
morteza.chehreghani@chalmers.se

Mostafa Haghir Chehreghani
mostafa.chehreghani@gmail.com

[1] Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden

[2] Department of Computer Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

1 Introduction

Real-world datasets often consist of complex and a priori unknown patterns and structures, requiring to improve the basic representation. Kernel methods are commonly used for this purpose (Hofmann et al. 2008; Shawe-Taylor and Cristianini 2004). However, their applicability is confined by several limitations (von Luxburg 2007; Nadler and Galun 2007; Chehreghani 2017b). (1) Finding the optimal parameter(s) of a kernel function is often nontrivial, in particular in an unsupervised learning task such as clustering where no labeled data is available for cross-validation. (2) The proper values of the parameters usually occur inside a very narrow range that makes cross-validation critical, even in presence of labeled data.

To overcome such challenges, some graph-based distance measures have been developed in the context of algorithmic graph-theory. In this setup, each object corresponds to a node in a graph, and the edge weights are the pairwise (e.g., squared Euclidean) distances between the respective objects (nodes). Then, different methods perform different types of inference on the graph to compute an effective distance measure between the pairs of objects. Link-based methods (Chebotarev 2011; Yen et al. 2008) first sum the edge weights on every path to compute the *path-specific* distances. The final distance is then obtained by summing up the *path-specific* distances of all paths between the two nodes. This distance measure can be obtained by inverting the Laplacian of the base distance matrix related to Markov diffusion kernel (Fouss et al. 2012; Yen et al. 2008). It requires an $\mathcal{O}(n^3)$ runtime, with n the number of objects.

Minimax distance measure is an alternative option that computes the minimum largest gap of all possible paths between the objects. Several previous works study the superior performance of Minimax distances, compared to metric learning or link-based choices (Farnia and Tse 2016; Fischer et al. 2003; Chehreghani 2016b; Kim and Choi 2007, 2013; Kolar et al. 2011; Li et al. 2017). Minimax distances have been first used with clustering problems in two ways, either as an input in the form of pairwise distance matrix (Chang and Yeung 2008; Pavan and Pelillo 2007), or integrated with some clustering algorithms (Fischer and Buhmann 2003). The straightforward approach to compute the pairwise Minimax distances is to use an adapted variant of the Floyd–Warshall algorithm, whose runtime is $\mathcal{O}(n^3)$ (Aho and Hopcroft 1974). However, the method in Fischer and Buhmann (2003) is computationally even more demanding, as its runtime is $\mathcal{O}(n^2|E| + n^3 \log n)$ ($|E|$ is the number of edges in the graph). Based on equivalence of Minimax distances over a graph and over any minimum spanning tree constructed on that, Chehreghani (2017b, 2020) propose to compute first a minimum spanning tree (e.g., using Prim's algorithm) and then obtain the Minimax distances over that via an efficient dynamic programming algorithm. Then, the runtime of computing pairwise Minimax distances reduces to $\mathcal{O}(n^2)$. Chehreghani (2017d) analyzes computing pairwise Minimax distances in different sparse and dense settings. Zhong et al. (2015) develops an approximate minimum spanning tree algorithm and investigates it for efficient computation of pairwise Minimax distances. Yu et al. (2014) and Liu and Zhang (2019) combine Minimax distances with specific clustering methods in closed-form ways.

Minimax distances have been also used for K-nearest neighbor search (Kim and Choi 2007, 2013; Chehreghani 2016b). The method in Kim and Choi (2007) presents a message passing method related to the sum–product algorithm (Kschischang et al. 2006) to perform K-nearest neighbor classification with Minimax distances. Even though its runtime is $\mathcal{O}(n)$, it needs computing a minimum spanning tree (MST) in advance that can require

$\mathcal{O}(n^2)$ runtime. Thereafter, the algorithm in Kim and Choi (2013) computes the Minimax K nearest neighbors via space partitioning whose runtime is $\mathcal{O}(\log n + K \log K)$. However, it is applicable only to sparse graphs built in Euclidean spaces. Finally, Chehreghani (2016b) has proposed an efficient Minimax K-nearest neighbor search method applicable to general graphs and dissimilarities. Its runtime, similar to the standard K nearest neighbor search is linear in general. Moreover, the method provides an outlier detection mechanism alongside performing K-nearest neighbor search, all with a linear runtime. The work in Chehreghani (2017a) investigate Minimax K nearest neighbor search for matrix (of user profiles) completion.

Besides Minimax distances, another related line of research has been developed in the context of *tree preserving embedding* (Shieh et al. 2011a, b), where the goal is to compute an embedding that preserves the *single* linkage dendrogram in the embedding.[1]

Both Minimax distances and tree preserving embedding correspond to computing a set of features representing *single* linkage dendrograms. Therefore, this limitation motivates us to extend the previous works on representation learning and feature extraction based on *single* linkage criterion and develop a generalized framework to compute different distance measures according to various dendrograms. In our framework the dendrogram, i.e., the way the inter-cluster distances called *linkage* are defined, can be constructed according to different criteria. The *single* linkage criterion (Sneath 1957) defines the linkages as the distance between the nearest members of the nodes. In contrast, the *complete* linkage criterion (Sorensen 1948; Lance and Williams 1967) defines the distance between two nodes as the distance between their farthest members, which corresponds to the maximum within-node distance of the new node. On the other hand, in *average* criterion (Sokal and Michener 1958) the average of inter-node distances is used as the linkage between two nodes. The *Ward* method (Ward 1963) uses the distances between the means of the nodes normalized by a function of the size of the nodes. Moseley and Wang (2017) analyzes in detail several of such criteria.

We study the embedding of the pairwise distances computed from a dendrogram into a new vector space such that the squared Euclidean distances in the new space equal to the dendrogram-based distances. This embedding provides us to employ dendrogram-based distances with a wide range of different machine learning methods, and yields a rich family of alternative dendrogram-based distances with Minimax distance measures and tree preserving embedding in Shieh et al. (2011a, b) being only special instantiations.

Then, we encounter a model selection problem which asks for the choice of the appropriate distance measure (and dendrogram). Therefore, in the context of model averaging and ensemble methods, we first study the aggregation of the distance measures from different dendrograms in the solution space. Assuming, for exaple the different dendrogram-based distance measures are used for an unsupervised clustering task, we build a graph with positive and negative edge weights based on the (dis)agreement of the respective nodes among different clustering solutions. Then, we employ an efficient variant of correlation clustering to obtain the final ensemble solution. Second, several recent studies demonstrate the superior performance of deep representation learning models that extract complex features via aggregating representations sequentially at different levels. Such models are highly over-parameterized and thus require huge amounts of training data to infer the parameters. However, unsupervised representation learning is expected to become far more

[1] Tree-based structures have been studied and analyzed in several other domain such as frequent pattern mining (Chehreghani et al. 2011, 2007) different from the setting in this paper.

important in longer term, as human and animal learning is mainly unsupervised (LeCun et al. 2015). Thereby, with the possibility of having access to a wide range of alternative feature extraction models, we investigate design of multi layer deep architectures in an unsupervised manner (in representation space, instead of solution space) which does not require inferring or fixing any critical parameter. Specifically, we study the sequential aggregation of the dendrogram-based features where for example the *single* linkage features are computed based on the features obtained from *average* linkage dendrogram, instead of using the original data features.

Our framework provides several options for choosing the dendrogram and the level function, where each option yields separate unsupervised representations and features. However, at the same time, we propose a principled way to aggregate and choose the best options (either in solution space or in representation space). Availability of such alternatives endows a rich family of unsupervised representation learning methods and is different from optimizing the free parameters of a kernel. We will discuss this model selection aspect with more detail in the experiments section.

Finally, we experimentally validate the effectiveness of our framework on UCI and real-world datasets.

2 Feature extraction from dendrograms

In this section, we first introduce the setup for computing distance measures from dendrograms, and then, based on the relation between Minimax distances and *single* linkage agglomerative clustering, we propose a generalized approach to extract features from dendrograms.

2.1 Pairwise distances over dendrograms

We are given a dataset of n objects with indices $\mathbf{O} = \{1, \dots, n\}$ and the corresponding measurements. The measurements can be for example the vectors in a feature space or the pairwise distances between the objects. In the former case, the measurements are shown by the $n \times d$ matrix $\mathbf{Y}$, wherein the ith row (i.e., $\mathbf{Y}_i$) specifies the d dimensional vector of the ith object. In the latter form, an $n \times n$ matrix $\mathbf{X}$ represents the pairwise distances between the objects. Then, we might show the data by graph $\mathcal{G}(\mathbf{O}, \mathbf{X})$, wherein $\mathbf{O}$ is the set of its vertices and $\mathbf{X}$ represents the edge weights. Note that the former is a specific form of the latter representation, where the pairwise distances are computed according to (squared) Euclidean distances.

A dendrogram D is defined as a rooted ordered tree such that,

1. each node v in D includes a non-empty subset of the objects, i.e., $v \subset \mathbf{O}, |v| > 0, \forall v \in D$, and
2. the overlapping nodes are ordered, i.e., $\forall u, v \in D$, if $u \cap v \neq 0$, 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 nodes at the lowest level (called the *final* nodes) are the singleton objects, i.e., node v is a final node if and only if $|v| = 1$. A node at a higher level contains the union of the objects of its children (direct descendants). The root of a dendrogram is defined as the node at the highest level (which has the maximum size), i.e., all other nodes are its descendants. $linkage(v)$ returns the distance between the children of v based on the criterion used to compute the dendrogram. For the simplicity of explanation, we assume each node has only two children. In the case that a parent node has multiple (more than two) child nodes, the different linkages among the children will have the same value, which will be assigned to the parent node. To encode a dendrogram, we use the data structure supported by SciPy in Python in particular the same way as the output of the *linkage* function.[2] This data structure is a $n-1$ by 4 matrix called $\mathbf{Z}$. Each individual object constitutes a separate singleton cluster where the cluster index is the object index. At each iteration i of the agglomerative algorithm, the indices of the two combined clusters are stored respectively in $\mathbf{Z}_{i,0}$ and $\mathbf{Z}_{i,1}$. The index of the new cluster is then $i+n$. We store the distance between the two clusters in $\mathbf{Z}_{i,2}$ and the size of the new cluster in $\mathbf{Z}_{i,3}$.

The level of node v, i.e., $level(v)$ is determined by $\max(level(c_l), level(c_r)) + 1$, where c_l and c_r indicate the two child nodes of v. For the final nodes, the $level()$ function returns 0. Every connected subtree of D whose final nodes contain only singleton objects from $\mathbf{O}$ constitutes a dendrogram on this set. We use $\mathcal{D}^D$ to refer to the set of all (sub)dendrograms derived in this form from D.

Thereby, the level of node v, i.e., $level(v)$ is determined by

$$level(v) = \begin{cases} \max(level(c_l), level(c_r)) + 1, \\ \qquad \text{if } linkage(v) > \max(linkage(c_l), linkage(c_r)). \\ \max(level(c_l), level(c_r)), \\ \qquad \text{if } linkage(v) = \max(linkage(c_l), linkage(c_r)). \end{cases} \tag{1}$$

where c_l and c_r indicate the two child nodes of v. Note that in an agglomerative method we always have $linkage(v) \geq \max(linkage(c_l, c_r))$. In particular, we usually expect $linkage(v) > \max(linkage(c_l, c_r))$, unless there are ties for example in the case of *single* linkage method, where then the new combination does not yield a higher level node. Rather, the new node has effectively three children instead of two, where two of them are combined to make an intermediate node. Without loss of generality and for the sake of simplicity of presentation, we assume that ties do not occur, i.e., we always have

$$level(v) = \max(level(c_l, c_r)) + 1. \tag{2}$$

We consider a generalized variant of the $level()$ function over a dendrogram D. Any function $f(v)$ that satisfies the following conditions is a *generalized level* function.

1. $f(v) = 0$ if and only if $v \subset \mathbf{O}, |v| = 1$.
2. $f(v) > f(u)$ if and only if v is an ancestor of u.

It is obvious that the basic function $level()$ satisfies these conditions. We use v^*_{ij} to denote the node at the lowest level which contains both i and j, i.e.,

[2] scipy.cluster.hierarchy.linkage: https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html.

$$v_{ij}^{*} = \arg\min_{v \in D} f(v) \quad \text{s.t. } i,j \in v. \tag{3}$$

Given dendrogram D, each node $v \in D$ represents the root of a dendrogram $D' \in \mathcal{D}^D$. Thereby, the dendrogram D' inherits the properties of its root node, i.e., $f(D') = \max_{v \in D'} f(v)$ and $linkage(D') = \max_{v \in D'} linkage(v)$, since the root node has the maximum linkage and level among the nodes of D'.

In this paper, we investigate inferring pairwise distances from a dendrogram computed according to an arbitrary criterion, i.e., beyond *single* linkage criterion. Moreover, our framework allows one to define the level function in a very flexible and diverse way. For this purpose, we consider the following generic distance measure over dendrogram D, where $\mathbf{D}_{ij}^D$ indicates the pairwise dendrogram-based distance between the pair of objects (final nodes) $i,j \in \mathbf{O}$.

$$\mathbf{D}_{ij}^D = \min f(D') \quad \text{s.t.} \quad i,j \in D', \text{ and } D' \in \mathcal{D}^D. \tag{4}$$

The level function $f(v)$ and the distance matrix $\mathbf{D}^D$ provide distinguishing outliers at different levels. The outlier objects do not occur in the nearest neighborhood of many other clusters or objects. Thus, they join the other nodes of the dendrogram only at higher levels. Hence, the probability of object i being an outlier is proportional to the level at which it joins to other objects/clusters. Therefore, such objects will have a large dendrogram-based distance from the other objects.

2.2 Minimax distances and single linkage agglomeration

We first study the relation between Minimax distances and *single* linkage agglomerative method. In particular, we elaborate that given the pairwise dissimilarity matrix $\mathbf{X}$, the pairwise Minimax distance between objects i and j is equivalent to $\mathbf{D}_{ij}^D$ where the dendrogram is produced with *single* linkage criterion and $\mathbf{D}_{ij}^D$ is defined by

$$\mathbf{D}_{ij}^D = \min linkage(D') \quad \text{s.t.} \quad i,j \in D' \text{ and } D' \in \mathcal{D}^D, \tag{5}$$

i.e., $f(D')$ in Eq. 4 is replaced by $linkage(D')$.

Theorem 1 *For each pair of objects $i,j \in \mathbf{O}$, their Minimax distance measure over graph $\mathcal{G}(\mathbf{O},\mathbf{X})$ is equivalent to their pairwise distance $\mathbf{D}_{ij}^D$ defined in Eq. 5 where the dendrogram D is obtained according to single linkage agglomerative method.*

Proof It can be shown that the pairwise Minimax distances over an arbitrary graph are equivalent to pairwise Minimax distances over 'any' minimum spanning tree computed from the graph. The proof is similar to the *maximum capacity* problem (Hu 1961) problem. Thereby, the Minimax distances are obtained by

$$\begin{aligned} \mathbf{D}_{i,j}^{MM} &= \min_{r \in \mathcal{R}_{ij}(\mathcal{G})} \left\{ \max_{1 \le l \le |r|-1} \mathbf{X}_{r(l)r(l+1)} \right\} \\ &= \max_{1 \le l \le |r_{ij}|-1} \mathbf{X}_{r(l)r(l+1)}, \end{aligned} \tag{6}$$

where r_{ij} indicates the (only) route between i and j, i.e., to obtain Minimax distances $\mathbf{D}^{MM}_{ij}$, we select the maximal edge weight on the only route between i and j over the minimum spanning tree.

On the other hand, single linkage method and the Kruskal's minimum spanning tree algorithm are equivalent (Gower and Ross 1969). Thus, dendrogram D represents the pairwise Minimax distances. Now, we only need to show that the Minimax distances in Eq. 6 equal the distances defined in Eq. 3 of the main text, i.e., $\mathbf{D}^{D}_{ij}$ is the largest edge weight on the route between i and j in the hierarchy.

Given i, j, let $D^* = \arg\min linkage(D') \quad \text{s.t.} \quad i, j \in D'$ and $D' \in \mathcal{D}^D$. Then, D^* represents a minimum spanning subtree, which includes a route between i, j (because the root node of D^* contains both i, j) and it is consistent with a complete minimum spanning on all the objects. On the other hand, we know that for each pair of nodes $u, v \in D^*$ which have direct or indirect parent–child relation, we have, $linkage(u) \geq linkage(v)$ iff $f(u) \geq f(v)$. This indicates that the linkage of the node root of D^* represents the maximal edge weight on the route between i and j induced by the dendrogram D. Thus, $\mathbf{D}^{D}_{ij}$ defined in Eq. 3 of the main text represents $\mathbf{D}^{MM}_{ij}$ and the proof is complete.□

Notice that the Minimax distances in Eq. 5 are obtained by replacing $f(D')$ with $linkage(D')$ in the generic form of Eq. 4.

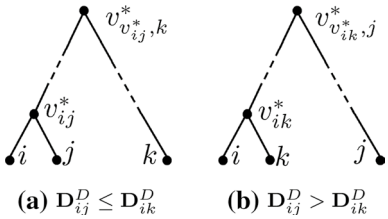
2.3 Vector-based representation of dendrogram-based distances

The generic distance measure defined in Eq. 4 yields an $n \times n$ matrix of pairwise dendrogram-based distances between objects. However, a lot of machine learning algorithms perform on a vector-based representation of the objects, instead of the pairwise distances. For instance, mixture density estimation methods such as Gaussian Mixture Models (GMMs) fall in this category. Vectors constitute the most basic form of data representation, since they provide a bijective map between the objects and the measurements, such that a wide range of numerical machine learning methods can be employed with them. Moreover, feature selection is more straightforward with this representation. Thereby, we compute an embedding of the objects into a new space, such that their pairwise squared Euclidean distances in the new space equal to their pairwise distances obtained from the dendrogram. For this purpose, we first investigate the feasibility of this kind of embedding. Theorem 2 verifies the existence of an $\mathcal{L}_2^2$ embedding for the general distance measure defined in Eq. 4.[3]

Theorem 2 *Given the dendrogram D computed on the input data $\mathbf{Y}$ or $\mathbf{X}$, the matrix of pairwise distances $\mathbf{D}^D$ obtained via Eq. 4 induces an $\mathcal{L}_2^2$ embedding, such that there exists a new vector space for the set of objects $\mathbf{O}$ wherein the pairwise squared Euclidean distances equal to $\mathbf{D}^{D}_{ij}$'s in the original data space.*

Proof First, we show that the matrix $\mathbf{D}^D$ yields an *ultrametric*. The conditions to be satisfied are:

[3] Note that $\mathbf{X}$ is not required to induce a *metric*, i.e., the triangle inequality might fail.



(a) $\mathbf{D}^D_{ij} \leq \mathbf{D}^D_{ik}$ **(b)** $\mathbf{D}^D_{ij} > \mathbf{D}^D_{ik}$

Fig. 1 The *ultrametric* property of $\mathbf{D}^D$

1. $\forall i,j : \mathbf{D}^D_{ij} = 0$ if and only if $i = j$. We investigate each of the conditions separately. (1) First, if $i = j$, then $\mathbf{D}^D_{ii} = \min f(i) = 0$. (2) If $\mathbf{D}^D_{ij} = 0$, then $v^*_{ij} = i = j$, because $f(v) = 0$ if and only if $v \in \mathbf{O}$. On the other hand, $\forall i \neq j, \mathbf{X}_{ij} > 0$, i.e., $f(v^*_{ij}) > 0$ if $i \neq j$.
2. $\forall i,j : \mathbf{D}^D_{ij} \geq 0$. We have, $\forall v, f(v) \geq 0$. Thus, $\forall D' \in \mathcal{D}^D, \min f(D) \geq 0$, i.e., $\mathbf{D}^D_{ij} \geq 0$.
3. $\forall i,j : \mathbf{D}^D_{ij} = \mathbf{D}^D_{ji}$. We have, $\mathbf{D}^D_{ij} = \{\min f(D) \text{ s.t. } i,j \in D', \text{ and } D' \in \mathcal{D}^D\} = \{\min f(D) \text{ s.t. } j,i \in D', \text{ and } D' \in \mathcal{D}^D\} = \mathbf{D}^D_{ji}$.
4. $\forall i,j,k : \mathbf{D}^D_{ij} \leq \max(\mathbf{D}^D_{ik}, \mathbf{D}^D_{kj})$. We first investigate $\mathbf{D}^D_{ik}$ where we consider the two following cases: (1) If $\mathbf{D}^D_{ij} \leq \mathbf{D}^D_{ik}$ (Fig. 1a), then $\mathbf{D}^D_{ik}$ does not yield a contradiction. (2) If $\mathbf{D}^D_{ij} > \mathbf{D}^D_{ik}$, then i and k join earlier than i and j, i.e., $f(v^*_{ij}) > f(v^*_{ik})$ (Fig. 1b). In this case, we have $f(v^*_{ij}) = f(v^*_{v^*_{ik}j})$ and $f(v^*_{kj}) = f(v^*_{v^*_{ik}j})$. Thus, we will have $f(v^*_{ij}) = f(v^*_{kj})$, i.e., $\mathbf{D}^D_{ij} = \mathbf{D}^D_{ik} \leq \max(\mathbf{D}^D_{ik}, \mathbf{D}^D_{kj})$. In a similar way, by investigating $\mathbf{D}^D_{jk}$ a similar result holds. Thereby, we conclude, a) if $\mathbf{D}^D_{ij} > \mathbf{D}^D_{ik}$, then $\mathbf{D}^D_{ij} = \mathbf{D}^D_{kj}$, and b) if $\mathbf{D}^D_{ij} > \mathbf{D}^D_{kj}$, then $\mathbf{D}^D_{ij} = \mathbf{D}^D_{ik}$. Thereby, we always have $\mathbf{D}^D_{ij} \leq \max(\mathbf{D}^D_{ik}, \mathbf{D}^D_{kj})$.

On the other hand, one can show that an *ultrametric* induces an $\mathcal{L}^2_2$ embedding (Deza and Laurent 1994). Therefore, $\mathbf{D}^D$ represents the pairwise squared Euclidean distances in a new vector space.□

After assuring the existence of such an embedding, we can use any method to compute it. In particular, we exploit the method introduced in Young and Householder (1938) and then further analyzed in Torgerson (1958). This method proposes first centering $\mathbf{D}^D$ to obtain a Mercer kernel and then performing an eigenvalue decomposition:[4]

1. Center $\mathbf{D}^D$ via

$$\mathbf{W}^D \leftarrow -\frac{1}{2}\mathbf{A}\mathbf{D}^D\mathbf{A}. \tag{7}$$

   $\mathbf{A}$ is obtained by $\mathbf{A} = \mathbf{I}_n - \frac{1}{n}\mathbf{e}_n\mathbf{e}_n^T$, where $\mathbf{e}_n$ is an n-dimensional constant vector of 1's and $\mathbf{I}_n$ is an identity matrix of size $n \times n$.
2. With this transformation, $\mathbf{W}^D$ becomes a positive semidefinite matrix. Thus, we decompose $\mathbf{W}^D$ into its eigenbasis, i.e., $\mathbf{W}^D = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{V} = (v_1, \ldots, v_n)$ contains the eigenvectors v_i and $\mathbf{\Lambda} = \texttt{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix of eigenvalues $\lambda_1 \geq \cdots \geq \lambda_l \geq \lambda_{l+1} = 0 = \cdots = \lambda_n$. Note that the eigenvalues are nonnegative, since $\mathbf{W}^D$ is positive semidefinite.

[4] In Roth et al. (2003), this method has been used to obtain an K-means variant for pairwise clustering, after adding a large enough constant to the off-diagonal elements of the input distance matrix.

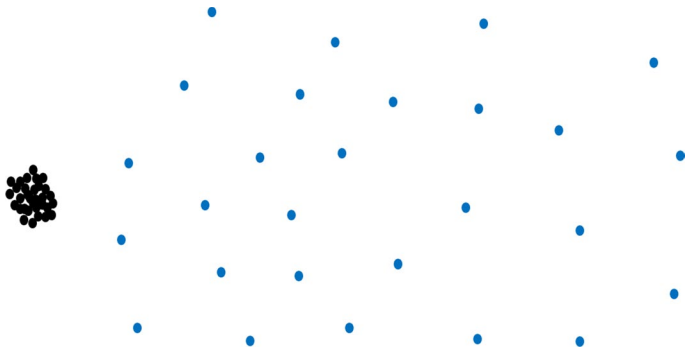


Fig. 2 Minimax distance measures might perform imperfectly on the data with diverse densities. An adaptive approach which takes into account the variance of different classes or clusters might be more appropriate

3. Calculate the $n \times l$ matrix $\mathbf{Y}_l^D = \mathbf{V}_l(\mathbf{\Lambda}_l)^{1/2}$, with $\mathbf{V}_l = (v_1, \dots, v_l)$ and $\mathbf{\Lambda}_l = \text{diag}(\lambda_1, \dots, \lambda_l)$, where l shows the dimensionality of the new vectors.

The new dendrogram-based dimensions are ordered according to the respective eigenvalues and one might choose only the first most representative ones, instead of taking all. Hence, an advantage of computing such an embedding is feature selection.

2.4 On the choice of level function

As mentioned before, Minimax distances as a particular instance of the dendrogram-based representations, are widely used in clustering and classification tasks. However, such distances (and equivalently the *single* linkage method) do not take into account the diverse densities of the structures or classes. For example, consider the dataset shown in Fig. 2 which consists of two clusters with different densities, marked respectively with black and blue colors. However, the intra-cluster Minimax distances for the members of the blue cluster are considerably large compared to the intra-cluster Minimax distances of the black cluster, or even the inter-cluster Minimax distances. Thereby, a clustering algorithm might split the blue cluster, instead of performing a cut on the boundary of the two clusters. According to Proposition 1, the Minimax distance between objects i and j seeks for a linkage with maximal weight on the path between them in the dendrogram. However, the absolute value of a linkage might be biased in a way that it does not precisely reflect the real coherence of the two nodes compared to the other nodes/objects. Thereby, in order to be more adaptive with respect to the diverse densities of the underlying structures, we will investigate the following choice in our experiments.

$$\mathbf{D}_{ij}^D = \min_{D'} level(D') \quad \text{s.t.} \quad i, j \in D', \text{ and } D' \in \mathcal{D}^D. \tag{8}$$

Note that our analysis is generic and can be applied to any definition of dendrogram-based distance measure and to any choice of f defined in Eq. 4. It only needs to satisfy the aforementioned conditions for generalized level functions.

3 Aggregation of multiple representations

3.1 Aggregation in solution space

As discussed earlier, a dendrogram can be constructed in several ways according to different criteria. Moreover, the choice of a level function and a distance function over a dendrogram renders another degree of freedom. Therefore, choosing the right method constitutes a model selection question. Let us assume such distances and features are used later in a clustering task, which is the most common unsupervised learning problem. Then, we address this problem via an ensemble method in the context of model averaging.

We follow a two-step procedure to compute an aggregated clustering that represents a given set of clustering solutions (where, e.g., each solution is the result of a particular dendrogram and then a clustering algorithm). First, we construct a graph whose vertices represent the objects and its edge weights can be any integer number (i.e., positive, negative or zero), depending how often the respective vertices appear at the same cluster among the M different clustering solutions. More specifically, we initialize the edge weights by zero. Then, for each clustering solution $\mathbf{c}^m \in \{1, \ldots, K\}^n, 1 \leq m \leq M$ (each obtained from a different dendrogram-based representation), we compute a co-clustering matrix whose (i, j)th entry is $+1$ if $\mathbf{c}_i^m = \mathbf{c}_j^m$, and it is -1 otherwise (K indicates the number of clusters). Finally, we sum up the co-clustering matrices to obtain $\mathbf{S}^e$. Algorithm 1 describes the procedure in detail.[5]

Algorithm 1 Aggregation of M clustering solution by correlation clustering.

Require: A set of M clustering solutions $\mathbf{c}^m, 1 \leq m \leq M$ on the same set of objects $\mathbf{O}$.
Ensure: An ensemble clustering solution $\mathbf{c}^e$.

```
for i ∈ O do
    for j ∈ O do
        S^e_ij = 0
    end for
end for
for 1 ≤ m ≤ M do
    for i ∈ O do
        for j ∈ O do
            if c^m_i = c^m_j then
                S^e_ij = S^e_ij + 1
            else
                S^e_ij = S^e_ij − 1
            end if
        end for
    end for
end for
Apply Correlation Clustering on S^e to obtain final clustering solution c^e.
return c^e.
```

Given the graph with positive and negative edge weights, we use *correlation clustering* (Bansal et al. 2004) to partition it into K clusters. This model computes a partitioning that minimizes the disagreements, i.e., sum of the inter-cluster positive edge weights plus sum of the intra-cluster negative edge weights should be minimal. The cost function for a fixed number of clusters K is written by Bansal et al. (2004) and Chehreghani et al. (2012)

[5] The work in Chehreghani (2017c) suggests an adaptive shift approach to build the correlation matrix from a given dissimilarity matrix. However, in this work the correlation matrix is given by construction.

$$R(\mathbf{c}, \mathbf{S}^e) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j \in \mathbf{O}_k} (|\mathbf{S}^e_{ij}| - \mathbf{S}^e_{ij}) + \frac{1}{2} \sum_{k=1}^{K} \sum_{k'=k+1}^{K} \sum_{i \in \mathbf{O}_k} \sum_{j \in \mathbf{O}_{k'}} (|\mathbf{S}^e_{ij}| + \mathbf{S}^e_{ij}), \tag{9}$$

where $\mathbf{O}_k$ indicates the objects of the kth cluster, i.e., $\forall i : i \in \mathbf{O}_k$ iff $\mathbf{c}_i = k$. This model has been further analyzed in Thiel et al. (2019) in terms of convergence rate.

This ensemble clustering method yields a consistent aggregation of the clustering solutions obtained from different representations, i.e., in the case of $M = 1$ the optimal solution of Eq. 9 does not change the given clustering solution of this single representation.

Efficient optimization of correlation clustering cost function Finding the optimal solution of the cost function in Eq. 9 is NP-hard (Bansal et al. 2004; Demaine et al. 2006) and even APX-hard (Demaine et al. 2006). Therefore, we develop a *local search* method which computes a local minimum of the cost function. The good performance of such a greedy strategy is well studied for different clustering models, e.g., K-means (Macqueen 1967), kernel K-means (Schölkopf et al. 1998) and in particular several graph partitioning methods (Dhillon et al. 2004, 2005).[6] We begin with a random clustering solution and then we iteratively assign each object to the cluster that yields a maximal reduction in the cost function. We repeat this procedure until no further improvement is achieved, i.e., a local optimal solution is found.

At each step of the aforementioned procedure, one needs to investigate the costs of assigning every object to each of the clusters. The cost function is quadratic, thus, a single evaluation might take $\mathcal{O}(n^2)$. Thereby, if the local search converges after t steps, the total runtime will be $\mathcal{O}(tn^3)$. However, we do not need to recalculate the cost function for each individual evaluation. Let $R(\mathbf{c}, \mathbf{S}^e)$ denote the cost of clustering solution $\mathbf{c}$, wherein the cluster label of object i is k. To obtain a more efficient cost function evaluation, we first consider the contribution of object i in $R(\mathbf{c}, \mathbf{S}^e)$, i.e., $R_i(\mathbf{c}, \mathbf{S}^e)$, which is written by

$$R_i(\mathbf{c}, \mathbf{S}^e) = \frac{1}{2} \sum_{j \in \mathbf{O}_k} (|\mathbf{S}^e_{ij}| - \mathbf{S}^e_{ij}) + \frac{1}{2} \sum_{q=1, q \neq k}^{K} \sum_{j \in \mathbf{O}q} (|\mathbf{S}^e_{ij}| + \mathbf{S}^e_{ij}). \tag{10}$$

Then, the cost of the clustering solution $\mathbf{c}'$ being identical to $\mathbf{c}$ except for the object i which is assigned to cluster $k' \neq k$, i.e., $R(\mathbf{c}', \mathbf{S}^e)$ is computed by

$$R(\mathbf{c}', \mathbf{S}^e) = R(\mathbf{c}, \mathbf{S}^e) - R_i(\mathbf{c}, \mathbf{S}^e) + R_i(\mathbf{c}', \mathbf{S}^e), \tag{11}$$

where $R(\mathbf{c}, \mathbf{S}^e)$ is already known and $R_i(\mathbf{c}, \mathbf{S}^e)$ and $R_i(\mathbf{c}', \mathbf{S}^e)$ both require an $\mathcal{O}(n)$ runtime. Thus, we evaluate the cost function (9) only once for the initial random clustering. Then, iteratively and until the convergence, we compute the costs of assigning objects to different clusters via Eq. 11 and assign them to the clusters that yields a minimal cost. The total runtime is then $\mathcal{O}(tn^2)$.

[6] Consistently, for correlation clustering we observe a better performance with the local search method compared to the different approximation schemes such as those proposed in Bansal et al. (2004) and Demaine et al. (2006).

3.2 Aggregation in representation space

In this section, instead of an ensemble-based approach in the solution space, we describe the aggregation of different (dendrogram-based) distances in the representation space, independent of what the next task will be. The embedding phase of our general-purpose framework not only enables us to employ any numerical machine learning algorithm, but also provides an amenable way to successively combine different representations. In this approach, the features extracted from a dendrogram (e.g., single linkage) are used to build another dendrogram according to the same or a different criterion (e.g., average linkage), in order to yield more complex features. The degree of freedom (richness of the function class) can increase by the choice of a different level or distance function over dendrograms. Such a framework leads to a *nonparametric deep architecture* wherein a cascade of multiple layers of nonparametric information processing units are deployed for feature transformation and extraction. The output of each layer is a set of features, which can be fed into another layer as input. Note that in this architecture any other (nonparametric) unit can be employed at the layers, beyond the dendrogram-based feature extraction units. Each layer (dendrogram) extracts a particular type of features in the space of data representation.

4 Experiments

We empirically investigate the performance of dendrogram-based representations on different datasets and demonstrate the usefulness of this approach to extract suitable features. Our methods are unsupervised and do not assume availability of any labeled data. Thus, to fully benefit from this property, we consider an unsupervised representation learning strategy, such that no free parameter is involved in inferring the new features. Thereby, we apply our methods to clustering and density estimation problems, for which parametric feature extraction methods might be inappropriate, due to lack of labeled data for cross validation (to estimate the parameters). In particular, after extracting the new features, we apply the following algorithms to obtain a clustering solution: (1) Gaussian Mixture Model (GMM), (2) K-means, and (3) spectral clustering. In the case of GMM, after computing the assignment probabilities, we assign each object to the cluster (distribution) with a maximal probability. We run each method and as well as correlation clustering (to obtain the ensemble solution) 100 times and pick a solution with the smallest cost or negative log-likelihood.

UCI datasets We perform our experiments on the following datasets selected randomly from the UCI data repository.[7]

1. *Forest Type*: contains multi-temporal sensing information of 326 samples from a forested area in Japan each described with 27 features. The dataset consists of 5 clusters.
2. *Hayes-Roth*: contains 160 samples on human subjects study each described with 5 attributes.
3. *Lung Cancer*: each instance contains 56 attributes and is categorized as cancer or non-cancer.

[7] We observe similar results on several other datasets.

4. *Mammographic Mass*: consists of the BI-RADS attributes of the mammographic masses for 961 samples.
5. *One-Hundred Plant*: contains leaf samples for 100 plant species for each 16 samples with 64 features (1600 samples in total with 100 clusters).
6. *Perfume*: contains 560 instances (odors) of 20 different perfumes measured by a hand-held odor meter.
7. *Semeion Handwritten Digit*: features of 1593 handwritten digits from around 80 persons where each digit stretched in a rectangular box 16×16 in a gray scale of 256 values.
8. *Statlog (Australian Credit Approval)*: includes credit card data (described with 14 attributes) of 690 users.
9. *Urban Land Cover*: contains 168 high resolution aerial images of 9 types each represented by 148 features.
10. *Vertebral Column*: contains information of 6 biomechanical features of 310 patients categorized according to their status.

In these datasets, the objects and as well as the features extracted from different dendrograms are represented by vectors. Thus, to obtain the pairwise distances, we compute the squared Euclidean distances between the respective vectors. Some clustering algorithms such as spectral clustering require pairwise similarities as input, instead of a vector-based representation. Therefore, as proposed in Chehreghani (2016a), we convert the pairwise distances $\mathbf{X}$ (or $\mathbf{D}^D$, if obtained from a dendrogram) to a similarity matrix $\mathbf{S}$ via $\mathbf{S}_{ij} = \max(\mathbf{X}) - \mathbf{X}_{ij} + \min(\mathbf{X})$, where the max(.) and min(.) operations return the maximal and minimal elements of the given matrix. Note that an alternative transformation is an exponential function in the form of $\mathbf{S}_{ij} = \exp(-\frac{\mathbf{X}_{ij}}{\sigma^2})$, which requires fixing the free parameter σ in advance. However, in particular in unsupervised learning, this task is nontrivial and the appropriates values of σ occur in a very narrow range (von Luxburg 2007).

Evaluation The ground truth solutions of these datasets are available. Therefore, we can quantitatively measure the performance of each method by comparing the estimated and the true cluster labels. For each estimated clustering solution, we compute three commonly used quality measures: (1) adjusted Mutual Information (Vinh et al. 2010), that gives the mutual information between the two estimated and true solutions, (2) adjusted Rand score (Hubert and Arabie 1985), that computes the similarity between them, and (3) V-measure (Rosenberg and Hirschberg 2007), that gives the harmonic mean of homogeneity and completeness. We compute the adjusted variants of these criteria, i.e., they yield zero for random solutions.

Results Tables 1 and 2 show the results on different UCI datasets. Each block row represents a separate dataset (in order, *Forest Type*, *Hayes-Roth*, *Lung Cancer*, *Mammographic Mass* and *One-Hundred Plant* in Table 1 and *Perfume*, *Semeion Handwritten Digit*, *Statlog*, *Urban Land Cover* and *Vertebral Column* in Table 2). For each dataset, we investigate the different feature extraction methods (*base*, PCA, LSA and those obtained by different dendrograms) with three different clustering algorithms. The goal of studying the three clustering algorithms is to demonstrate that our feature extraction methods can be used with various forms of clustering algorithms and are not limited to a specific algorithm. In this way, we investigate one probabilistic clustreing model (GMM), one which uses vector-based representation (K-means) and another that is applied to pairwise relations (spectral clustering). The three evaluation criteria that we use are the most common criteria for evaluating clustering methods. The results of the

Table 1 Permanence of different representations and clustering methods on different UCI datasets

Method	GMM			K-means			Spectral clustering		
	M.I.	Rand	V.M.	M.I.	Rand	V.M.	M.I.	Rand	V.M.
Base	0.3897	0.3306	0.3959	0.5197	0.4987	0.5279	0.4380	0.4303	0.4438
PCA	0.3755	0.3496	0.4139	0.4742	0.4181	0.3453	0.4331	0.4170	0.4274
LSA	0.3716	0.3472	0.3460	0.4633	0.4842	0.37 81	0.4484	0.4208	0.4513
Single	0.3544	0.3466	0.3592	0.3681	0.3321	0.3813	0.3705	0.3318	0.3838
Complete	0.3517	0.2792	0.3592	0.3517	0.2792	0.3592	0.3521	0.2785	0.3595
Average	**0.5294**	**0.5316**	**0.5370**	**0.5294**	**0.5316**	**0.5370**	**0.5325**	**0.5235**	**0.5417**
Ward	0.4718	0.3498	0.4812	0.4718	0.3498	0.4812	0.4718	0.3498	0.4812
Ensemble	*0.4771*	*0.3661*	*0.4855*	*0.4752*	*0.3641*	*0.4838*	*0.4752*	*0.3641*	*0.4838*
Base	0.1198	0.1175	0.1336	0.0138	0.0146	0.0005	0.0138	0.0146	0.0005
PCA	0.1113	0.1096	0.1159	0.0376	0.0244	0.0107	0.0260	0.0205	0.0187
LSA	0.1375	0.1440	0.1425	0.0756	0.0517	0.0346	0.0878	0.1060	0.1250
Single	**0.2379**	0.1624	**0.2589**	0.2562	0.2035	**0.3561**	0.2273	**0.1685**	**0.2909**
Complete	0.0446	0.0383	0.0588	0.0446	0.0383	0.0588	0.0446	0.0383	0.0588
Average	0.1945	**0.1787**	0.2118	**0.2610**	**0.2403**	0.2787	**0.2419**	0.1614	0.2752
Ward	0.0249	0.0496	0.0412	0.0249	0.0496	0.0412	0.0249	0.0496	0.0412
Ensemble	*0.1426*	*0.1193*	*0.1560*	*0.1249*	*0.1046*	*0. 1112*	*0.1042*	*0.1096*	*0.1152*
Base	0.1684	0.1698	0.2030	0.1997	0.2294	0.2356	0.1197	0.1294	0.1356
PCA	0.1170	0.1170	0.1743	0.1962	0.2362	0.2430	0.0609	0.0678	0.0890
LSA	0.1702	0.2162	0.2730	0.1962	0.2362	0.2430	0.0728	0.0419	0.0606
Single	0.1677	0.2316	0.1892	0.1525	0.2636	0.2425	0.1016	0.1316	0.1282
Complete	0.1537	0.2809	0.1810	0.1537	0.2809	0.1810	0.1537	0.2809	0.1810
Average	0.1475	0.2253	0.1795	0.2070	0.3533	0.2303	0.1239	0.1327	0.0742
Ward	0.1766	0.3388	0.2140	0.1766	0.3388	0.2140	**0.1766**	**0.3388**	**0.2140**
Ensemble	***0.2659***	***0.4345***	***0.2957***	***0.2659***	***0.4345***	***0.2957***	***0.1766***	***0.3388***	***0.2140***
Base	0.0036	0.0037	0.0059	0.0944	0.1133	0.0959	0.0944	0.1133	0.0959
PCA	0.0679	0.0454	0.0406	0.0944	0.1133	0.0959	0.0944	0.1133	0.0959
LSA	0.0550	0.0431	0.0603	0.0944	0.1133	0.0959	0.0944	0.1133	0.0959
Single	0.0407	0.0915	0.0639	**0.1523**	**0.2078**	**0.1542**	**0.1523**	**0.2078**	**0.1542**
Complete	0.0152	0.0113	0.0166	0.0152	0.0113	0.0166	0.0598	0.0191	0.0743
Average	**0.0834**	**0.0721**	**0.0895**	0.0834	0.0721	0.0895	0.0834	0.0721	0.0895
Ward	**0.0834**	**0.0721**	**0.0895**	0.0834	0.0721	0.0895	0.0834	0.0721	0.0895
Ensemble	***0.0834***	***0.0721***	***0.0895***	*0.0834*	*0.0721*	*0.0895*	*0.0834*	*0.0721*	*0.0895*
Base	0.4834	0.1956	0.6867	0.6765	0.4844	0.8138	0.4386	0.2427	0.6547
PCA	0.4510	0.2070	0.6791	0.6571	0.4580	0.8024	0.4704	0.2507	0.6881
LSA	0.4745	0.2942	0.7121	0.6593	0.4794	0.8034	0.4225	0.2185	0.6455
Single	0.4841	0.2426	0.6915	0.4809	0.2354	0.6884	0.4922	0.2625	0.6982
Complete	0.6381	0.4459	0.7893	0.6377	0.4427	0.7893	0.6377	0.4456	0.7891
Average	0.6975	0.5336	**0.8258**	0.6885	0.5176	0.8211	0.6788	0.5051	0.8159
Ward	0.6914	0.5249	0.8207	0.6852	0.5158	0.8174	**0.6876**	0.5151	**0.8184**
Ensemble	***0.6990***	***0.5408***	*0.8251*	***0.6925***	***0.5362***	***0.8218***	*0.6836*	***0.5197***	*0.8164*

The five block rows correspond to the first five datasets, respectively to *Forest Type*, *Hayes-Roth*, *Lung Cancer*, *Mammographic Mass* and *One-Hundred Plant*. The results of the ensemble method are shown in italics. For each clustering algorithm and each evaluation measure, the best result is bolded among the different feature extraction methods

ensemble method are shown in italics. For each clustering algorithm and each evaluation measure, the best result is bolded among the different feature extraction methods.

The *base* method indicates performing the GMM, K-means or spectral clustering on the original vectors without inferring any new features. We also investigate Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA) as two other baselines. As discussed in Theorem 2, the matrix of pairwise dendrogram-based distances satisfy the *ultrametric* conditions. Unltrametric is stronger than metric, i.e., any ultrametrci is a metric too. The only difference is the last condition in the proof of Theorem 2. For an ultrametric, we require $\forall i, j, k : \mathbf{D}^D_{ij} \leq \max(\mathbf{D}^D_{ik}, \mathbf{D}^D_{kj})$. It is obvious that this condition satisfies the triangle (metric) condition too, i.e., $\forall i, j, k : \mathbf{D}^D_{ij} \leq \mathbf{D}^D_{ik} + \mathbf{D}^D_{kj}$. Hence, $\mathbf{D}^D$ induces a metric. On the other hand, the different embedding methods usually rely on satisfying the metric conditions. Therefore, in principle any embedding and dimension reduction method can be applied to the dendrogram-based pairwise distances, the same way that it can be applied to the base pairwise distances too. Thus, further investigation of the results of different embedding methods is orthogonal to our contribution and we postpone it to future work.

Different dendrogram-based feature extraction methods are specified by the name of the criterion used to build the deprogram. The ensemble method refers to the aggregation of the different solutions and then preforming correlation clustering. According to the equivalence of *single* linkage method, Minimax distances and the tree preserving embedding method in Shieh et al. (2011a), this method can be seen as another baseline which also constitutes a special instantiation of the dendrogram-based feature extraction methodology. Note that the superior performance of Minimax distances (*single* linkage features) over methods such as metric learning or link-based methods has been demonstrated in previous works (Kim and Choi 2007, 2013; Chehreghani 2016b, 2017b) (see for example Figure 1 in Kim and Choi (2013)).[8]

We interpret the results of Tables 1 and 2 as follows. For each dataset (block row) and each clustering algorithm, we investigate whether "some" of the dendrogram-based features (i.e., *single*, *complete*, *average* or *Ward*) perform better (according to the three evaluation criteria) than the baseline methods (*base*, PCA and LSA). If so, then we conclude our framework provides a rich and diverse family of non-parametric feature extraction methods wherein some instances yield more suitable features for the data at hand. Thus, a user has more freedom and options to choose the correct features. However, the user might not have sufficient information to choose the correct features (dendrograms), thus, we propose to use the ensemble variant, in the context of averaging (aggregating) multiple learners.

According to the results reported in Tables 1 and 2, we observe: (1) extracting features from dendrograms yields better representations that improve the evaluation scores of the final clusters. The dendrogram might be built in different ways which correspond to computing different types of features. In particular, we observe the features extracted via *average* linkage and *Ward* linkage often lead to very good results. *Single* linkage (Minimax) features are more suitable for low-dimensional data wherein connectivity paths still exists. However, in higher dimensions, the other methods might perform better due to robustness and flexibility. (2) The ensemble method works well in particular compared to the baselines and most of the dendrogram-based approaches. Note that the ensemble method

[8] Moreover, methods such as metric learning often require fixing free parameter(s) which is non-trivial in unsupervised settings such as clustering.

Table 2 Permanence of different representations and clustering methods on different UCI datasets

Method	GMM			K-means			Spectral clustering		
	M.I.	Rand	V.M.	M.I.	Rand	V.M.	M.I.	Rand	V.M.
Base	0.8350	0.6783	0.8944	0.8555	0.7243	0.8974	0.2353	0.3981	0.4070
PCA	0.8916	0.7051	0.9159	0.8174	0.7430	0.8731	0.7933	0.6890	0.8942
LSA	0.7853	0.5912	0.8485	0.7982	0.6237	0.8625	0.8038	0.6049	0.8521
Single	0.8975	0.7924	0.9178	0.8967	0.7939	0.9245	0.8943	0.7960	0.9213
Complete	0.8941	0.7842	0.9169	0.8752	0.7474	0.9025	0.8632	0.7197	0.8981
Average	0.9054	0.8193	0.9229	0.9116	0.8288	0.9298	0.9041	0.8088	0.9263
Ward	**0.9390**	**0.8831**	**0.9516**	**0.9348**	**0.8729**	**0.9491**	**0.9348**	**0.8729**	**0.9491**
Ensemble	*0.9183*	*0.8393*	*0.9357*	*0.9133*	*0.8411*	*0.9379*	*0.9087*	*0.8244*	*0.9342*
Base	0.5253	0.4064	0.5312	0.5313	0.4037	0.5382	0.4884	0.3596	0.4970
PCA	0.5095	0.3685	0.5095	0.5291	0.4130	0.5179	0.4909	0.2928	0.5434
LSA	0.5130	0.3619	0.5406	0.5217	0.4097	0.5226	0.4982	0.2641	0.4849
Single	0.4961	0.3806	0.5132	0.4943	0.3214	0.5258	0.5065	0.2740	0.5612
Complete	0.3911	0.2508	0.4010	0.4110	0.2780	0.4212	0.4206	0.2769	0.4349
Average	**0.5879**	**0.4648**	0.5740	**0.6004**	**0.4712**	**0.6132**	0.5524	0.3682	**0.5896**
Ward	0.5362	0.3842	0.5495	0.5353	0.3770	0.5502	0.5587	0.3915	0.5736
Ensemble	*0.5661*	*0.4214*	***0.5858***	*0.5588*	*0.4211*	*0.5705*	***0.5759***	***0.4181***	*0.5863*
Base	0.0074	0.0038	0.0162	0.0038	0.0022	0.0099	0.0232	0.0116	0.0425
PCA	0.0074	0.0038	0.0162	0.0038	0.0022	0.0099	0.0525	0.0278	0.0261
LSA	0.0074	0.0038	0.0162	0.0074	0.0038	0.0162	0.0305	0.0374	0.0316
Single	0.0580	0.0859	0.0593	0.0580	0.0859	0.0593	0.0219	0.0203	0.0357
Complete	0.0399	0.0510	0.0411	0.0298	0.0445	0.0309	0.0570	0.0715	0.0709
Average	**0.0864**	0.1271	**0.0898**	0.0367	0.0484	0.0476	**0.0719**	**0.0972**	**0.0830**
Ward	0.0848	0.1251	0.0881	**0.0848**	**0.1251**	**0.0881**	0.0074	0.0038	0.0162
Ensemble	***0.0864***	***0.1272***	*0.0896*	***0.0848***	***0.1251***	***0.0881***	*0.0291*	*0.0259*	*0.0458*
Base	0.1465	0.0844	0.1747	0.0909	0.0339	0.1277	0.1392	**0.0963**	0.1645
PCA	0.1465	0.0844	0.1747	0.0909	0.0339	0.1277	0.0705	0.0817	0.0763
LSA	0.1465	0.0844	0.1747	0.0909	0.0339	0.1277	0.1208	0.0953	0.1281
Single	0.0973	0.0409	0.1236	0.0939	0.0458	0.1258	0.0898	0.0364	0.1272
Complete	**0.1688**	**0.0902**	**0.1910**	**0.1640**	**0.0798**	**0.1858**	**0.1563**	0.0689	**0.1769**
Average	0.1489	0.0732	0.1708	0.1436	0.0659	0.1650	0.1493	0.0721	0.1711
Ward	0.1515	0.0796	0.1746	0.1406	0.0594	0.1632	0.1406	0.0594	0.1632
Ensemble	*0.1534*	*0.0857*	*0.1771*	*0.1491*	*0.0756*	*0.1717*	*0.1501*	*0.0735*	*0.1724*
Base	0.1159	0.0825	0.1257	0.2072	0.1051	0.1953	**0.1722**	0.1042	**0.1779**
PCA	0.1398	0.06472	0.1534	0.1948	0.1601	0.1692	0.1209	0.1075	0.1383
LSA	0.1308	0.1179	0.1445	0.1609	0.1388	0.1846	0.1630	**0.1252**	0.1715
Single	0.1528	0.1002	0.1643	0.1906	0.2687	0.2001	0.1092	0.1188	0.1211
Complete	0.1696	0.1053	0.1773	0.1696	0.1053	0.1773	0.0705	0.0645	0.0941
Average	**0.3080**	**0.3278**	**0.3247**	**0.3080**	**0.3278**	**0.3247**	0.1242	0.0560	0.1444
Ward	0.1443	0.2216	0.1512	0.1443	0.2216	0.1512	0.1191	0.0583	0.1399
Ensemble	*0.2475*	*0.2846*	*0.2613*	*0.2322*	*0.2776*	*0.2452*	*0.1165*	*0.0953*	*0.1376*

The five block rows correspond to the second five datasets, respectively to *Perfume*, *Semeion Handwritten Digit*, *Statlog*, *Urban Land Cover* and *Vertebral Column*. The results of the ensemble method are shown in italics. For each clustering algorithm and each evaluation measure, the best result is bolded among the different feature extraction methods

Table 3 Aggregation of two representations on the *Perfume* dataset

M.I.				
	S	C	A	W
S	0.9509	0.9120	0.8998	0.9182
C	0.8738	0.8787	0.9116	0.9246
A	0.9305	0.9197	0.9305	0.9125
W	**0.9612**	**0.9612**	**0.9612**	**0.9612**
Rand				
	S	C	A	W
S	0.9071	0.8289	0.8114	0.8385
C	0.7517	0.7622	0.8195	0.8443
A	0.8678	0.8480	0.8678	0.8399
W	**0.9360**	**0.9360**	**0.9360**	**0.9360**
V.M.				
	S	C	A	W
S	0.9595	0.9255	0.9161	0.9325
C	0.8991	0.9020	0.9302	0.9411
A	0.9441	0.9350	0.9441	0.9263
W	**0.9667**	**0.9667**	**0.9667**	**0.9667**

Higher scores are highlighted in bold

The first and the second dendrograms are indicated by the rows and the columns, respectively. GMM is used to perform the clustering on the final features. The best combination is using first *Ward* and then any of the four options

is more than just averaging the results. It can be interpreted as obtaining a good (strong) learner from a set of weaker learners. Thereby, in several cases, the ensemble method performs even better than all the other alternatives.

Aggregation of representations As a side study, we investigate the sequential aggregation of different dendrogram-based features in representation space, i.e., we consider the combination of every two such feature extractors. For this purpose, we first compute a dendrogram and extract the respective features. Then, we use these features to compute a second dendrogram from which we obtain a new set of features. Finally, we apply a clustering method (GMM, *K*-means and spectral clustering) and evaluate the results w.r.t. Mutual Information, Rand score and V-measure.

We observe for most of the datasets, aggregation of different features either improves the results or preserves the accuracy of the results as same as the first representation. However, aggregation of the clustering solutions usually yields more significant changes (improvements) compared to the aggregating the representations. One of the significant changes happens on the *Perfume* dataset. See the results in Tables 3, 4 and 5, where respectively GMM, *K*-means and spectral clustering have been applied to the final features to produce the clusters. The first and the second dendrograms are indicated by the rows and the columns, respectively (where S refers to *single*, C to *complete*, A to *average*, and W to *Ward*, the different ways of obtaining the features). These results should

Table 4 Aggregation of two representations on the *Perfume* dataset, where *K*-means is used for the clustering of the final features

M.I.				
	S	C	A	W
S	0.9164	0.8945	0.9192	0.9104
C	0.8666	0.8653	0.8842	0.9057
A	0.9091	0.9091	0.9091	0.8911
W	**0.9481**	0.9383	0.9383	0.9379
Rand				
	S	C	A	W
S	0.8341	0.7942	0.8416	0.8175
C	0.7333	0.7332	0.7685	0.8061
A	0.8225	0.8225	0.8225	0.7946
W	**0.8963**	0.8824	0.8824	0.8805
V.M.				
	S	C	A	W
S	0.9313	0.9236	0.9342	0.9300
C	0.9019	0.8999	0.9145	0.9267
A	0.9295	0.9295	0.9295	0.9116
W	**0.9603**	0.9516	0.9516	0.9515

Higher scores are highlighted in bold

W-S (*Ward* and then *single*) is the best combination

be compared with the block row in Table 2 that corresponds to the *Perfume* dataset (the first block row). We observe that over this dataset, feature aggregation often improves the results for different clustering methods. However, as mentioned before, such an aggregation is usually less significant (on other datasets).

We observe that on this dataset, the W-S combination (extracting the features first via *Ward* and then via *single* linkages) consistently yields the best results, among all different combinations. In Table 6, we compare these results with the best feature extractor for the *perfume* dataset, which is based on the *Ward* linkage. *Single* linkage even though does not yield very good results itself, but improves the *Ward* features the most. According to Table 6, except spectral clustering, using the *single* linkage features helps the clustering algorithm to produce better results. However, the best result is obtained with GMM for which combining *Ward* with any option is helpful.

Model selection Our framework provides several options for choosing the dendrogram and the level function, and at the same time a principled way to aggregate and choose the best options (either in solution space or in representation space).

Availability of such alternatives endows a rich family of unsupervised models for representation learning and feature extraction. We note that this availability is different than optimizing the free parameters of a kernel.

Table 5 Aggregation of two representations on the *Perfume* dataset, where spectral clustering is applied to the final features to cluster them

M.I.				
	S	C	A	W
S	0.9147	0.8768	0.8832	0.9104
C	0.8373	0.8339	0.8538	0.8717
A	0.8578	0.8460	0.8458	0.8756
W	**0.9217**	0.9161	0.9124	0.9139
Rand				
	S	C	A	W
S	0.8366	0.7643	0.7788	0.8175
C	0.6875	0.6736	0.7165	0.7412
A	0.7070	0.6958	0.6901	0.7448
W	**0.8368**	0.8327	0.8339	0.8350
V.M.				
	S	C	A	W
S	0.9366	0.9093	0.9133	0.9300
C	0.8836	0.8819	0.8972	0.9075
A	0.9005	0.8922	0.8905	0.9107
W	**0.9426**	0.9364	0.9367	0.9375

Higher scores are highlighted in bold

W-S (*Ward-single*) is the best combination

Table 6 Comparison of *Ward*(W) and *Ward-single*(W-S) features on the *perfume* dataset

Method	GMM			K-means			Spectral clustering		
	M.I.	Rand	V.M.	M.I.	Rand	V.M.	M.I.	Rand	V.M.
W	0.9390	0.8831	0.9516	0.9348	0.8729	0.9491	**0.9348**	**0.8729**	**0.9491**
W-S	**0.9612**	**0.9360**	**0.9667**	**0.9481**	**0.8963**	**0.9603**	0.9217	0.8368	0.9426

Higher scores are highlighted in bold

Performing *single* linkage on the *Ward* features improves the final clustering

1. In our framework, the number of the choices is very limited, whereas for a kernel function the free parameter(s) can usually take a wide (continuous) range of different values. Moreover, the optimal values of the kernel parameters usually occur inside very narrow ranges that makes it difficult to find them via search or cross-validation, even using the labeled data (Nadler and Galun 2007; von Luxburg 2007).
2. In our framework, every choice has an explicit interpretation that makes model selection more straightforward. For example, single linkage is more suitable for elongated structures and patterns, whereas average linkage suits better for high-dimensional data.

Table 7 Comparison of the runtimes of different methods for optimization the correlation clustering objective used to obtain the ensemble solutions

Dataset	Local search (s)	LP (s)	SDP
Forest Type	2.8	149.3	More than 10 h
Hayes-Roth	2.1	93.0	8.45 h
Lung Cancer	1.1	26.7	2.16 h
Mammographic Mass	16.5	729.5	More than 10 h
One-Hundred Plant	27.2	1015.9	More than 10 h
Perfume	4.7	312.9	More than 10 h
Semeion Handwritten Digit	4.9	335.7	More than 10 h
Statlog	6.2	369.8	More than 10 h
Urban Land Cover	2.2	95.2	9.51 h
Vertebral Column	2.6	127.8	More than 10 h

On the other hand, the proposed level function in Eq. 8 is better adapted to the density-diverse structures.

3. Finally, as we demonstrated on all the datasets, our framework also provides a consistent way for computing an ensemble of the different choices and options. According to the experimental results, the ensemble solution performs very well compared to the individual choices. Computing such an ensemble solution is nontrivial for many kernels.

Here, as a side study, we compare the two choices for level function on the ensemble solution, i.e., the option defined in Eq. 5 and the one defined in Eq. 8. As explained before, Eq. 8 suggests a context-sensitive level function that takes into account the data diversity. According to the results in Tables 1 and 2, with the level function in Eq. 8, the ensemble solution of GMM on different UCI datasets yields the following MI scores: 0.4771, 0.1426, 0.2659, 0.0834, 0.6990, 0.9183, 0.5661, 0.0864, 0.1534, 0.2475. However, with the level function in Eq. 5, the ensemble solution of GMM on different UCI datasets gives the following MI scores: 0.4148, 0.1301, 0.2738, 0.0834, 0.6364, 0.9075, 0.5747, 0.0864, 0.1451, 0.2262. We observe that on two datasets (*Mammographic Mass* and *Statlog*) the two variants yield the same results. Among the remaining eight datasets, on six of them the level function in Eq. 8 performs better, whereas on only two datasets (*Lung Cancer* and *Semeion Handwritten Digit*) the level function in Eq. 5 yields higher scores. It is notable that however the results from both of the choices are acceptable.

Efficiency of correlation clustering optimization In our framework, we employ an efficient optimization of correlation clustering to compute the ensemble solution. We have studied its effectiveness in terms of the quality of the ensemble solution. Here, we investigate the efficiency of its optimization procedure in terms of runtimes. In particular, we compare our local search optimization with the Linear Programming (LP) method (Demaine et al. 2006) and the Semidefinite Programming relaxation (SDP) (Charikar et al. 2003; Mathieu and Schudy 2010). Table 7 shows the different runtime results. We observe that the local search method performs significantly faster compared to the alternatives. It is notable that the SDP method encounters memory issues for the datasets of larger than 200 objects. We stop it when its runtime exceeds 10 h.

Experiments on scientific datasets At the end, we investigate the proposed methods on two real-world datasets collected within a scientific data analytics project. The goal is to

Table 8 Permanence of different representations and clustering methods on two scientific datasets

Method	GMM			K-means			Spectral clustering		
	M.I.	Rand	V.M.	M.I.	Rand	V.M.	M.I.	Rand	V.M.
Base	0.3291	0.3136	0.4065	0.3530	0.3264	0.3168	0.3705	0.3530	0.3822
PCA	0.3205	0.3021	0.3437	0.3474	0.3628	0.3340	0.3606	0.3502	0.3797
LSA	0.3066	0.3139	0.3839	0.3522	0.3328	0.3874	0.3430	0.3588	0.3678
Single	0.3628	0.3750	0.4196	0.3965	0.3921	0.4391	0.3729	0.3780	0.4125
Complete	0.5043	0.3951	0.5784	0.4966	0.3902	0.5557	0.4831	0.3877	0.5484
Average	**0.5067**	**0.4193**	**0.5728**	**0.7007**	**0.7423**	**0.7196**	**0.6241**	**0.6825**	**0.6654**
Ward	0.4638	0.3404	0.5381	0.4972	0.3753	0.5688	0.4535	0.3260	0.5212
Ensemble	*0.4726*	*0.3687*	*0.5447*	*0.4956*	*0.4032*	*0.5669*	*0.4949*	*0.3728*	*0.5456*
Base	0.2718	0.1528	0.3204	0.2652	0.1502	0.3077	0.2883	0.2040	0.3184
PCA	0.2744	0.1475	0.3254	0.2683	0.1434	0.3114	0.2818	0.2159	0.3401
LSA	0.2803	0.1614	**0.3390**	0.2413	0.1332	0.2932	0.2743	0.2115	0.3217
Single	**0.2915**	0.1683	0.3372	**0.2949**	0.1655	**0.3367**	0.3132	0.2006	0.3485
Complete	0.2907	0.1819	0.3362	0.2576	0.1588	0.3022	0.2913	0.2265	0.3625
Average	0.2354	0.1419	0.2804	0.2848	**0.1725**	0.3303	0.2987	0.1940	0.3321
Ward	0.2810	**0.1878**	0.3354	0.2451	0.1518	0.2827	0.2852	0.1960	0.3229
Ensemble	*0.2779*	*0.1653*	*0.3277*	*0.2234*	*0.1495*	*0.2670*	***0.3348***	***0.2273***	***0.3666***

The first block row corresponds to the *computer science* dataset and the second block row corresponds to *electrical engineering* dataset. The results of the ensemble method are shown in italics. For each clustering algorithm and each evaluation measure, the best result is bolded among the different feature extraction methods

extract clusters of different subjects and topics. The extracted clusters help to analyze (1) how an automated approach can distinguish the scientific outcomes in different subjects and accordingly categorize the respective authors, (2) how separable or related the different subjects and topics are. The first dataset contains 10,000 published scientific articles in 10 different topics of computer science including algorithms, database, machine learning, networks, hardware, software engineering, formal methods, security, logic and information systems. The second dataset contains 10,000 published scientific articles in different topics of electrical engineering. Each ground truth cluster consists of 1000 articles. For each dataset, we obtain the TF-IDF vectors of the articles where we remove the step words. The number of features is 5823 and 5495 respectively for computer science and for electrical engineering datasets. We compute the base pairwise distances based on the squared Euclidean distances between the TF-IDF vectors. We then use them to compute the dendrograms.

Table 8 shows the permanence of different representations and clustering methods on these datasets, where the first block row corresponds to the *computer science* dataset and the second block row corresponds to the *electrical engineering* dataset. The results of the ensemble method are shown in italics. For each clustering algorithm and each evaluation measure, the best result is bolded among the different feature extraction methods. We observe consistent results to the UCI datasets. (1) Using different dendrogram-based features often improves the results for different clustering methods w.r.t. the evaluation criteria. (2) The ensemble solution yields either the best results or yields very close results to the best choice, i.e., it can effectively address the model selection problem.

5 Conclusion

We extended the previous Minimax and tree preserving representation learning methods that correspond to building a *single* linkage dendrogram, and proposed a generic framework to compute representations from different dendrograms, beyond *single* linkage. Then, we studied the embedding to extract vector-based features for such distances. This property extends the applicability to a wide range of machine learning algorithms. Then, we considered the aggregation of different dendrogram-based features in solution space and representation space. First, based on the consistency of the cluster labels of different objects, we build a graph with positive and negative edge weights and then apply correlation clustering to obtain the final clusters. In the second approach, in the spirit of deep learning models, we apply different dendrogram-based features sequentially, such that the input of the next layer is the output of the current one, and then we apply the particular (clustering) algorithm to the final features. Our experiments on several datasets revealed the effectiveness of the proposed framework.

Acknowledgements Open access funding provided by Chalmers University of Technology. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Parts of this work have been done at Xerox Research.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Aho, A. V., & Hopcroft, J. E. (1974). *The design and analysis of computer algorithms* (1st ed.). Boston: Addison-Wesley Longman Publishing Co.

Bansal, N., Blum, A., & Chawla, S. (2004). Correlation clustering. *Machine Learning*, *56*(1–3), 89–113.

Chang, H., & Yeung, D.-Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, *41*(1), 191–203.

Charikar, M., Guruswami, V., & Wirth, A. (2003). Clustering with qualitative information. In *44th Symposium on foundations of computer science FOCS* (pp. 524–533).

Chebotarev, P. (2011). A class of graph-geodetic distances generalizing the shortest-path and the resistance distances. *Discrete Applied Mathematics*, *159*(5), 295–302.

Chehreghani, M. H. (2016a). Adaptive trajectory analysis of replicator dynamics for data clustering. *Machine Learning*, *104*(2–3), 271–289.

Chehreghani, M. H. (2016b). K-nearest neighbor search and outlier detection via minimax distances. In *SDM '16* (pp. 405–413).

Chehreghani, M. H. (2017a). Feature-oriented analysis of user profile completion problem. In *39th European conference on information retrieval (ECIR)* (pp. 304–316).

Chehreghani, M. H. (2017b). Classification with minimax distances. In *Thirty-first AAAI conference on artificial intelligence (AAAI)*.

Chehreghani, M. H. (2017c). Clustering by shift. In *IEEE international conference on data mining, ICDM* (pp. 793–798).

Chehreghani, M. H. (2017d). Efficient computation of pairwise minimax distance measures. In *IEEE international conference on data mining, ICDM* (pp. 799–804).

Chehreghani, M. H. (2020). Unsupervised representation learning with minimax distance measures. *Machine Learning*. https://doi.org/10.1007/s10994-020-05886-4.

Chehreghani, M. H., Rahgozar, M., Lucas, C., & Chehreghani, M. H. (2007) Mining maximal embedded unordered tree patterns. In *Proceedings of the IEEE symposium on computational intelligence and data mining, CIDM* (pp. 437–443).

Chehreghani, M. H., Chehreghani, M. H., Lucas, C., & Rahgozar, M. (2011). Oinduced: An efficient algorithm for mining induced patterns from rooted ordered trees. *IEEE Transactions on Systems, Man, and Cybernetics Part A*, *41*(5), 1013–1025.

Chehreghani, M. H., Busetto, A. G., & Buhmann, J. M. (2012). Information theoretic model validation for spectral clustering. In *Fifteenth international conference on artificial intelligence and statistics (AISTATS)* (pp. 495–503).

Demaine, E. D., Emanuel, D., Fiat, A., & Immorlica, N. (2006). Correlation clustering in general weighted graphs. *Theoretical Computer Science*, *361*(2–3), 172–187.

Deza, M., & Laurent, M. (1994). Applications of cut polyhedra–i. *Journal of Computational and Applied Mathematics*, *55*(2), 191–216.

Dhillon, I. S., Guan, Y., & Kulis, B. (2004). Kernel k-means: Spectral clustering and normalized cuts. In *ACM KDD '04* (pp. 551–556). ACM.

Dhillon, I. S., Guan, Y., & Kulis, B. (2005). A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25.

Farnia, F., & Tse, D. (2016). A minimax approach to supervised learning. In *NIPS '16* (pp. 4233–4241).

Fischer, B., & Buhmann, J. M. (2003). Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(4), 513–518.

Fischer, B., Roth, V., & Buhmann, J. M. (2003). Clustering with the connectivity kernel. In *NIPS '03* (pp. 89–96).

Fouss, F., Francoisse, K., Yen, L., Pirotte, A., & Saerens, M. (2012). An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. *Neural Networks*, *31*, 5372.

Gower, J. C., & Ross, G. J. S. (1969). Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society*, *18*, 54–64.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *Annals of Statistics, 36*(3), 1171–1220.

Hu, T. C. (1961). The maximum capacity route problem. *Operations Research*, *9*, 898–900.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Kim, K.-H., & Choi, S. (2007). Neighbor search with global geometry: A minimax message passing algorithm. In *ICML* (pp. 401–408).

Kim, K.-H., & Choi, S. (2013). Walking on minimax paths for k-nn search. In *AAAI*.

Kolar, M., Balakrishnan, S., Rinaldo, A., & Singh, A. (2011). Minimax localization of structural information in large noisy matrices. In *NIPS '11* (pp. 909–917).

Kschischang, F. R., Frey, B. J., & Loeliger, H. A. (2006). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, *47*(2), 498–519.

Lance, G. N., & Williams, W. T. (1967). A general theory of classificatory sorting strategies 1. Hierarchical systems. *The Computer Journal*, *9*(4), 373–380.

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, T., Yi, X., Carmanis, C., & Ravikumar, P. (2017). Minimax Gaussian classification and clustering. In A. Singh & J. Zhu (Eds.), *AISTATS '17* (Vol. 54, pp. 1–9).

Liu, Q., & Zhang, R. (2019) Global optimal path-based clustering algorithm. *CoRR*, arXiv:1909.07774.

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *In 5th Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Mathieu, C., & Schudy, W. (2010) Correlation clustering with noisy input. In M. Charikar (Ed.), *Proceedings of the twenty-first annual ACM-SIAM symposium on discrete algorithms, SODA* (pp. 712–728).

Moseley, B., & Wang, J. (2017) Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. In *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017* (pp. 3094–3103).

Nadler, B., & Galun, M. (2007). Fundamental limitations of spectral clustering. *Advanced in Neural Information Processing Systems*, *19*, 1017–1024.

Pavan, M., & Pelillo, M. (2007). Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 167–172.

Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL* (pp. 410–420). ACL.

Roth, V., Laub, J., Kawanabe, M., & Buhmann, J. M. (2003). Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(12), 1540–1551.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computing*, *10*(5), 1299–1319.

Shawe-Taylor, J. & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Shieh, A., Hashimoto, T. B., & Airoldi, E. M. (2011a). Tree preserving embedding. In *Proceedings of the 28th international conference on machine learning, ICML* (pp. 753–760).

Shieh, A. D., Hashimoto, T. B., & Airoldi, E. M. (2011b). Tree preserving embedding. *Proceedings of the National Academy of Sciences*, *108*(41), 16916–16921.

Sneath, P. H. A. (1957). The application of computers to taxonomy. *Journal of General Microbiology*, *17*, 201–226.

Sokal, R. R., & Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, *38*, 1409–1438.

Sorensen, T. (1948). *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons*. Biologiske Skrifter: Det Kongelige Danske Videnskabernes Selskab. I kommission hos E. Munksgaard.

Thiel, E., Chehreghani, M. H., & Dubhashi, D. P. (2019). A non-convex optimization approach to correlation clustering. In *Thirty-third AAAI conference on artificial intelligence (AAAI)* (pp. 5159–5166).

Torgerson, W. S. (1958). *Theory and methods of scaling*. Hoboken: Wiley.

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, *11*, 2837–2854.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244.

Yen, L., et al. (2008). A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances. In *KDD* (pp. 785–793).

Young, G., & Householder, A. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, *3*(1), 19–22.

Yu, Z., Xu, C., Meng, D., Hui, Z., Xiao, F., Liu, W., & Liu, J. (2014). Transitive distance clustering with k-means duality. In *2014 IEEE conference on computer vision and pattern recognition, CVPR* (pp. 987–994).

Zhong, C., Malinen, M. I., Miao, D., & Fränti, P. (2015). A fast minimum spanning tree algorithm based on k-means. *Information Science*, *295*, 1–17.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.