Pattern Recognition 139 (2023) 109517

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

The impact of isolation kernel on agglomerative hierarchical clustering algorithms

Xin Han[a,b], Ye Zhu[c,*], Kai Ming Ting[d], Gang Li[c]

[a] School of Computer Science, Xi'an Shiyou University, Shaanxi 710065, China
[b] Asia-Pacific Academy of Economics and Management, University of Macau, Macau 999078, China
[c] Centre for Cyber Resilience and Trust, Deakin University, Geelong 3125, Australia
[d] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

ARTICLE INFO

Article history:
Received 1 June 2020
Revised 5 January 2023
Accepted 11 March 2023
Available online 13 March 2023

Keywords:
Agglomerative hierarchical clustering
Varied densities
Dendrogram purity
Isolation kernel
Gaussian kernel

ABSTRACT

Agglomerative hierarchical clustering (AHC) is one of the popular clustering approaches. AHC generates a dendrogram that provides richer information and insights from a dataset than partitioning clustering. However, a major problem with existing distance-based AHC methods is: it fails to effectively identify adjacent clusters with varied densities, regardless of the cluster extraction methods applied to the resultant dendrogram. This paper aims to reveal the root cause of this issue and provides a solution by using a data-dependent kernel. We analyse the condition under which existing AHC methods fail to effectively extract clusters, and give the reason why the data-dependent kernel is an effective remedy. This leads to a new approach to kernerlise existing hierarchical clustering algorithms including the traditional AHC algorithms, HDBSCAN, GDL, PHA and HC-OT. Our extensive empirical evaluation shows that the recently introduced Isolation Kernel produces a higher quality or purer dendrogram than distance, Gaussian Kernel and adaptive Gaussian Kernel in all the above mentioned AHC algorithms.

1. Introduction

Hierarchical clustering is one of the widely used clustering methods [1]. Given a set of data points, the goal of hierarchical clustering is not to find a single partitioning of the data, but a hierarchy of subclusters in a dendrogram. Because its clustering output of a dendrogram is easy to interpret, hierarchical clustering has been used in a wide range of applications, e.g., biomedical research [2], text classification [3] and financial market analysis [4].

The most widely used hierarchical clustering approach is agglomerative hierarchical clustering (AHC) [1]. AHC starts with merging the most similar individual points, and then iteratively merges the two most similar subclusters, based on a *linkage function* which measures the similarity of two subclusters, until all points belong to a single cluster. AHC has been studied in the theoretical community and used by practitioners [1].

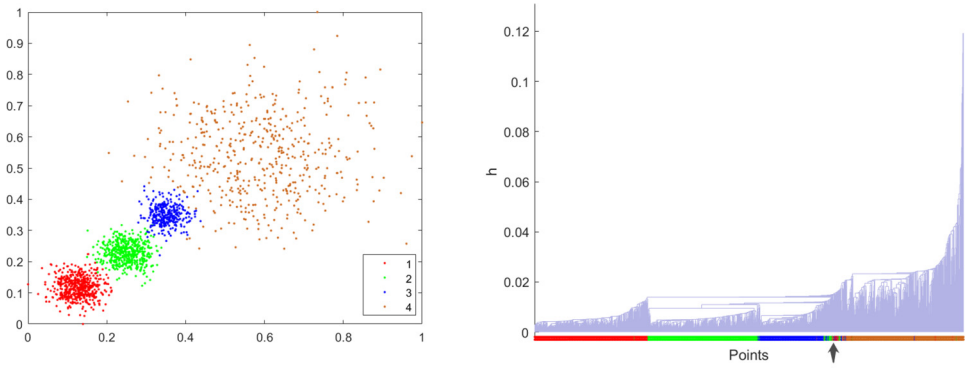
The linkage function used in an AHC relies on a distance (or similarity) measure. Many linkage functions have been proposed for hierarchical clustering, such as complete linkage [5] and average linkage [6]. In addition, in order to capture complex structures in the data, the graph-structural agglomerative clustering algorithms such as GDL [7] and PIC [8] have used a linkage function based on a *k*-nearest-neighbour graph.

This research is motivated by the current state of two clustering research fronts. First, the impact of varied densities of clusters on density-based clustering algorithms has been well studied (e.g. [9]). But its impact on the traditional AHC algorithms (T-AHC) has not been investigated in the literature thus far. We think its impact has been overlooked because the dendrogram is said to have a 'complete' set of subclusters, assuming that some of these subclusters will be a good match for the true clusters. Yet, we show that this 'complete' set of subclusters often include ones that are not a good match to the ground truth clusters in a given dataset.

Our investigation uncovers a specific bias of T-AHC: using a distance-based linkage function, T-AHC tends to merge high-density subclusters first, before low-density subclusters, in the merging process. While there is a hint of this bias in the literature [10], no analysis of its cause has been conducted, as far as we know.

* Corresponding author.
E-mail addresses: xhan@tulip.academy (X. Han), ye.zhu@ieee.org (Y. Zhu), tingkm@nju.edu.cn (K.M. Ting), gang.li@deakin.edu.au (G. Li).

https://doi.org/10.1016/j.patcog.2023.109517



(a) Dataset: 4 clusters of different densities (b) Dendrogram produced by T-AHC

Fig. 1. A dendrogram produced by T-AHC with the distance-based single-linkage function on a dataset with four clusters of varied densities. The colours at the bottom of the dendrogram correspond to the true cluster labels of all points shown in Figure (a). The arrow in Figure (b) indicates the subtrees containing points from different clusters.

Another possible reason for its lack of attention in this matter is that this bias only becomes an issue if clusters are not well separated (see our definition in Section 4.) In many real-world datasets in which clusters are not well separated, we have observed that this bias often leads to a dendrogram of poor quality.

The impact of the bias on T-AHC is most revealing in a dataset, having clusters of varied densities and not well separated, as shown in Fig. 1. Clusters extracted from such dendrogram often lose many of their (true) members and/or have (false) members of other clusters.

Second, in the context of density-based clustering, the root cause of the bias has been established, i.e., the use of data-independent kernel/distance [11,12]. In addition, in the context of kernel/spectral (partitioning) clustering, the density bias has been recognised to be an issue [9]. A kernel which adapts to local density to replace a data-independent kernel has been proposed as a remedy, e.g., Adaptive Gaussian Kernel [13]. However, to the best of our knowledge, how to reduce the density bias in hierarchical clustering using data-dependent kernels is yet to be explored.

Here, we contend that the same root cause yields the bias in T-AHC. We then provide the formal analysis and explanation as to why a well-defined data-dependent kernel called Isolation Kernel [12,14] is an effective remedy.

The contributions of this paper are summarised as follows:

1. Defining the formal condition under which an AHC, which employs an existing kernel/distance, does not extract clusters of a dataset effectively.
2. Introducing a new concept called **entanglement** as a way to explain the merging process that leads to a poor quality dendrogram. Two indicators, i.e., the number of entanglements and the average entanglement level, are shown to be highly correlated to an objective measure of quality of dendrogram called dendrogram purity.
3. Identifying the root cause of a density bias of T-AHC. The bias merges points in the dense region first, before merging points in the sparse region; and its root cause is due to the use of a data-independent distance/kernel. Though this analysis is mainly based on T-AHC, the same root cause also applies to existing AHC algorithms.
4. Proposing a generic approach to improve existing distance-based AHC algorithms: simply replace the distance function with a data-dependent kernel, without modifying the algorithms. We also provide the reason why a data-dependent kernel can significantly reduce the above-mentioned bias.
5. Presenting the empirical evaluation results using five kinds of algorithms, i.e., T-AHC [15], HDBSCAN [16], GDL [7], PHA [17] and HC-OT [18] that:
   (i) kernels produce better clustering results than distance, and
   (ii) a recently introduced Isolation Kernel [12,14] performs better than Gaussian Kernel and Adaptive Gaussian Kernel [13].

Our approach is distinguished from those used in existing AHC algorithms in two ways:

- Rather than creating new linkage functions that still employ distance measure [19], we propose to use a data-dependent kernel to replace the distance function in existing linkage functions.
- The methodology is generic which can be applied to different hierarchical clustering algorithms. Many existing linkage functions are tailor-made for a specific algorithm. We show that our approach can be applied to five existing algorithms, T-AHC, HDBSCAN [16], GDL [7], PHA [17] and HC-OT [18], even though each algorithm has its own specific linkage function(s).

The rest of the paper is organised as follows. We first describe the related work on AHC and kernels in Section 2. Then we define the Kernel-based AHC and the condition under which it fails to identify clusters in Section 3. Section 4 investigates the result of using a data-dependent kernel in addressing the density bias in Kernel-based AHC. An extensive empirical evaluation is reported in Section 5, followed by the conclusions in the last section.

2. Related work

2.1. Agglomerative hierarchical clustering

Hierarchical clustering can be categorised into agglomerative (bottom-up) and divisive (top-down) methods [1], depending on the direction in which the hierarchy in a dendrogram is created. Many works focus on improving the hierarchical clustering on the algorithm-level and understanding hierarchical clustering [1].

In this paper, we focus on AHC. An agglomerative method needs two functions: a distance function that measures the dissimilarity between two points; and a linkage function $h(C_i, C_j)$ that measures the dissimilarity of subclusters C_i and C_j [19]. Initially treating individual points in a given dataset as subclusters, AHC merges two most similar subclusters based on $h(\cdot,\cdot)$ iteratively to form a tree-based structure (dendrogram) until a single cluster is formed at the root of the tree.

When using a hierarchical clustering algorithm, it is important to choose a linkage function that is the most compatible with the dataset because different linkage functions have different properties [20]. For example, the complete-linkage function is sensitive to noise and outliers; and the average linkage function mainly finds globular clusters [20]. We found that all four commonly used link-

age functions in T-AHC have difficulty in handling clusters with varied densities (see Section 4).

To improve the performance of T-AHC, many variants of these common linkage functions have been introduced, attempting to address their limitations. HDBSCAN [16] uses a density-based linkage function to find clusters of varied densities.[1] PHA [17] is a hierarchical agglomerative clustering method that uses a potential field [21] to measure the similarity between subclusters to measure the similarity between clusters. The potential field is interpreted to represent both the global and local data distribution information. Recently HC-OT [18] is proposed using Optimal Transport-based distance to measure the similarity between clusters, which captures the distance between data distribution of the clusters. Both PHA and HC-OT are claimed to be robust to different types of data distribution in a dataset.

In order to capture the complex structure of a dataset, several graph-structural agglomerative clustering algorithms are proposed. Those algorithms first create a k-nearest-neighbour graph to obtain a set of small initial subclusters. Then they iteratively merge the two most similar subclusters until the target number of clusters is obtained. Chameleon [22] measures the similarity between two clusters based on relative interconnectivity and relative closeness, both of which are defined on the graph. Zell [23] describes the structure of a cluster and defines the similarity between two clusters based on the structural changes after merging. GDL [7] uses the product of average indegree and average outdegree in graphs to measure the similarity between two clusters. These algorithms typically use the pairwise distance to build the neighbourhood graph. Although they could handle clusters with varied densities to some degree, we show that simply replacing the distance with Isolation Kernel is able to significantly improve their clustering performance.

In addition, to reduce the runtime, many scalable hierarchical clustering algorithms have been developed recently [24–26]. However, these approaches trade off the clustering quality for fast linkage/similarity calculations based on sampling or approximation strategies.

All the above hierarchical clustering algorithms are based on a distance function to construct their linkage functions. As existing kernels like Gaussian kernel are data-independent, just as the distance function, the baseline of our investigation is AHC using kernel-based linkage functions, where the kernel is the commonly used Gaussian kernel. We will see later that both distance-based and such kernel-based linkage functions lead to the same bias we have mentioned in the introduction.

2.2. Kernels

Various kernel-based clustering algorithms have been developed to improve the performance of existing distance-based machine learning and data mining algorithms, such as kernel k-means [27], density-based clustering [12], and spectral clustering [28]. However, the study on the impact of a kernel on hierarchical clustering remains unexplored in the literature.

In this paper, we focus on a commonly used data-independent kernel, i.e., Gaussian Kernel, and two data-dependent kernels, i.e., Adaptive Gaussian Kernel [13] and Isolation Kernel [12,14]; and examine their impacts on AHC. The former has been applied to spectral clustering, and the latter has been applied to SVM classifiers and DBSCAN [29].

A brief description of each of these kernels is provided in the following. For any two points $x, y \in \mathbb{R}^d$,

Gaussian Kernel: The Gaussian Kernel defines the similarity between x and y as follows:

$$\mathcal{K}(x,y) = exp(\frac{-\parallel x-y \parallel^2}{2\sigma^2})$$

where σ is the bandwidth of the kernel.
Gaussian Kernel is a commonly used kernel, e.g., SVM for classification [30] and t-SNE [31] for visualisation.

Adaptive Gaussian Kernel: In order to make the similarity adaptive to local density, Adaptive Gaussian Kernel [13] is defined as:

$$K_{AG}(x,y) = exp(\frac{-||x-y||^2}{\sigma_x \sigma_y}) \tag{1}$$

where σ_x is the Euclidean distance between x and x's k-th nearest neighbour.
Adaptive Gaussian Kernel was introduced in spectral clustering to adjust the similarity locally for performing dimensionality reduction before clustering [13].

Isolation Kernel: Isolation Kernel is a recently introduced data-dependent kernel, which adapts to local distribution. The pertinent details of Isolation Kernel [12,14] are provided below.
Let $\mathbb{H}_\psi(D)$ denote the set of all partitions H that are admissible under the dataset D, where each H covers the entire space of $\mathbb{R}^d$; and each of the ψ isolating partitions, $\theta[z] \in H$, isolates one data point z from the rest of the points in a random subset $\mathcal{D} \subset D$, and $|\mathcal{D}| = \psi$. Here we use the Voronoi diagram [32] to partition the space, i.e., each $H \in \mathbb{H}_\psi(D)$ is a Voronoi diagram and each sample point $z \in \mathcal{D}$ is a cell centre.

Definition 1. Isolation Kernel of x and y with respect to D is defined to be the expectation taken over the probability distribution on all partitions $H \in \mathbb{H}_\psi(D)$ that both x and y fall into the same isolating partition $\theta[z] \in H, z \in \mathcal{D}$:

$$\begin{aligned} K_\psi(x,y \mid D) &= \mathbb{E}_{\mathbb{H}_\psi(D)}[\mathbb{1}(x,y \in \theta[z] \mid \theta[z] \in H)] \\ &= \mathbb{E}_{\mathcal{D} \subset D}[\mathbb{1}(x,y \in \theta[z] \mid z \in \mathcal{D})] \\ &= P(x,y \in \theta[z] \mid z \in \mathcal{D} \subset D) \end{aligned} \tag{2}$$

where $\mathbb{1}(\cdot)$ is an indicator function.

In practice, Isolation Kernel K_ψ is constructed using a finite number of partitions $H_i, i = 1, \ldots, t$, where each H_i is created using $\mathcal{D}_i \subset D$:

$$\begin{aligned} K_\psi(x,y \mid D) &= \frac{1}{t}\sum_{i=1}^{t} \mathbb{1}(x,y \in \theta \mid \theta \in H_i) \\ &= \frac{1}{t}\sum_{i=1}^{t}\sum_{\theta \in H_i} \mathbb{1}(x \in \theta)\mathbb{1}(y \in \theta) \end{aligned} \tag{3}$$

where θ is a shorthand for $\theta[z]$, and ψ is the sharpness parameter[2], i.e., the larger ψ, the sharper its kernel distribution.
As Equation (3) is quadratic, K_ψ is a valid kernel. The larger the ψ, the sharper the kernel distribution. ψ is a parameter having a similar function to σ in the Gaussian Kernel, i.e., the smaller σ, the narrower the kernel distribution.

2.3. Dendrogram evaluation

An AHC algorithm produces a dendrogram or cluster tree. To evaluate the quality of a dendrogram, *Dendrogram Purity* has been

[1] HDBSCAN can be interpreted as a kind of AHC algorithm that relies on a single linkage function and a particular dissimilarity measure, see the details in Appendix A.

[2] This parameter corresponds to the σ parameter in the Gaussian kernel, i.e., the smaller σ is, the more concentrated its kernel distribution.

created to measure the hierarchical clustering result [24,33]. Given a dendrogram, the procedure finds the smallest subtree containing two leave nodes belonging to the same ground-truth cluster; and measures the fraction of leave nodes in that subtree that belongs to the same cluster. The dendrogram purity is the expected value of this fraction. The dendrogram purity will be 1 if and only if all leave nodes belonging to the same cluster are rooted in the same subtree. The dendrogram purity [33] is calculated as follows.

Given a dendrogram $\mathcal{T}$ produced from a dataset $D = \{x_1, \ldots, x_n\}$. Let $C^\star = \left\{C_t^\star\right\}_{t=1}^{\kappa}$ be the true labels in κ clusters, and $\mathcal{P}^\star = \left\{\left(x, x'\right) | x, x' \in \mathcal{X}, C^\star(x) = C^\star\left(x'\right)\right\}$ be the set of pairs of points that are in the same ground-truth cluster. Then the dendrogram purity of $\mathcal{T}$ is

$$\text{Purity}(\mathcal{T}) = \frac{1}{|\mathcal{P}^\star|} \sum_{t=1}^{\kappa} \sum_{x_i, x_j \in C_t^\star} \text{pur}\left(\text{lvs}\left(\text{LCA}\left(x_i, x_j\right)\right), C_t^\star\right) \tag{4}$$

where $\text{LCA}(x, y)$ is the least common ancestor of x and y in $\mathcal{T}$; $\text{lvs}(z) \subset D$ is the set of points in all the descendant leaves of z in $\mathcal{T}$; and $\text{pur}(S, C) = \frac{|S \cap C|}{S}$ computes the fraction of S that matches the ground-truth label of C.

We use dendrogram purity as an objective measure to assess the quality of the dendrograms produced by different clustering algorithms.

3. Kernel-based hierarchical clustering

An agglomerative method creates a dendrogram and extracts clusters from it. To build a dendrogram, it iteratively merges two most similar subclusters (as measured by a linkage function based on a similarity measure) until all points in a dataset are grouped into one cluster.

A dendrogram contains a complete set of all possible subclusters grouped by the linkage function. It is richer than a flat clustering result produced by a partitioning clustering algorithm; and it shows the hierarchical relationship between subclusters.

To extract the most meaningful clusters from a dendrogram, a cluster extraction algorithm applies cuts in the dendrogram to produce subtrees, where each subtree represents a cluster.

Many kernel methods have been proposed to improve the performance of existing distance-based machine learning and data mining algorithms. For clustering, the use of kernel enables a method to capture the nonlinear relationship inherent in the data distribution and separate non-convex clusters that would otherwise be impossible for distance-based methods. For example, kernel k-means [27] and spectral clustering [13] have been shown to enrich the types of clusters that can be detected by distance-based k-means.

An existing distance-based agglomerative clustering algorithm can be easily kernelised by replacing the distance matrix with the kernel similarity matrix, leaving the rest of the procedure unchanged. The procedure is shown in Algorithm 1 which employs $\hbar$ as a kernel linkage function.[3] By setting κ to 1, it produces a dendrogram. Table 1 shows the kernel versions of the four commonly used linkage functions in T-AHC.

Here we provide the definitions associated with Algorithm 1 and its resultant dendrogram; and a theorem stating the condition under which two ground-truth clusters can be successfully extracted from the dendrogram.

Definition 2. Given a set of points $D = \{x_1, \ldots, x_n\}$, a set of subclusters are initialised as $\mathbb{C}_1 = \{C_1, C_2, \ldots, C_n\}$, where $C_i = \{x_i\}$. The

Algorithm 1: AHC - Agglomerative hierarchical clustering.

Input : M - pairwise similarity matrix ($n \times n$ matrix); κ - target number of clusters
Output: $\mathbb{C} = \{C_1, C_2, \ldots, C_\kappa\}$

1 **for** $j = 1, 2, \ldots, n$ **to do**
2 $\quad C_j = \{x_j\}$;
3 **end**
4 $\mathbb{C} = \{C_1, C_2, \ldots, C_n\}$;
5 **while** $|\mathbb{C}| > \kappa$ **to do**
6 $\quad$ Find the most similar pair C_p and C_q based on linkage function $\hbar$:$\{C_p, C_q\} = \arg\max_{C_i, C_j \in \mathbb{C}, i \neq j} \hbar(C_i, C_j)$;
7 $\quad$ Merge C_p and C_q in $\mathbb{C}$;
8 **end**

Table 1
Kernel-based linkage functions $\hbar$ used in T-AHC. C is a cluster which consists of data points; and K is a kernel function.

Single-linkage	$\hbar(C_i, C_j) = \max_{x \in C_i, y \in C_j} K(x, y)$
Complete-linkage	$\hbar(C_i, C_j) = \min_{x \in C_i, y \in C_j} K(x, y)$
Average-linkage	$\hbar(C_i, C_j) = \frac{1}{\lvert C_i \rvert \lvert C_j \rvert} \sum_{x \in C_i, y \in C_j} K(x, y)$
Weighted-linkage	$\hbar(C_i, C_j) = \begin{cases} \frac{\hbar(C_p, C_j) + \hbar(C_q, C_j)}{2}, & \text{if } C_i = C_q \cup C_p \\ K(C_i, C_j), & \text{if } \lvert C_i \rvert = \lvert C_j \rvert = 1 \end{cases}$

kernel agglomerative clustering algorithm recursively merges the two most similar subclusters C_p and C_q at each step s and updates the set of subclusters to $\mathbb{C}_s = \{\mathbb{C}'_{s-1} \cup C_{pq}\}$, where $\mathbb{C}'_{s-1} = \mathbb{C}_{s-1} \setminus \{C_p, C_q\}$ and $C_{pq} = C_p \cup C_q$, as measured by a kernel-based linkage function $\hbar$, i.e., $\{C_p, C_q\} = \arg\max_{C_i, C_j \in \mathbb{C}_{S-1}, i \neq j} \hbar(C_i, C_j)$, until all subclusters are merged into one final cluster.

Definition 3. A dendrogram is a tree structure representing the order of the subcluster merging process. The height at the root of a subtree indicates the value of the kernel-based linkage function which is used to merge the two subclusters to form the subtree.

Definition 4. A cluster extracted by a cut η on the dendrogram is a subtree from which its next merged height is more than η.

Theorem 1. *Given two non-overlapping ground-truth clusters ζ_i and ζ_j in a dataset, to correctly identify them from the dendrogram produced by the agglomerative clustering algorithm with a kernel linkage function $\hbar$, both clusters must satisfy the following condition:*

$$\forall_{\imath \leq I, \jmath \leq J} \ \min(\hat{\hbar}(\mathbb{C}_\imath^i), \hat{\hbar}(\mathbb{C}_\jmath^j)) \geq \hat{\hbar}(\mathbb{C}_\imath^i, \mathbb{C}_\jmath^j) \tag{5}$$

where $\mathbb{C}_\imath^i$ is the set of subclusters at step $\imath$ of the process in merging members in ζ_i; so as $\mathbb{C}_\jmath^j$ in ζ_j (as defined in Definition 3); $\hat{\hbar}(\mathbb{C}_\imath^i) = max_{C_p, C_q \in \mathbb{C}_\imath^i, p \neq q} \hbar(C_p, C_q)$ and $\hat{\hbar}(\mathbb{C}_\imath^i, \mathbb{C}_\imath^j) = max_{C_p \in \mathbb{C}_\imath^i, C_q \in \mathbb{C}_\imath^j} \hbar(C_p, C_q)$; and I and J are the maximum numbers of steps required to merge all points in ζ_i and ζ_j, respectively.

Proof. Given two clusters ζ_i and ζ_j, an violation of Eq. (5) means $\exists_{s \leq I, t \leq J} \ \min(\hat{\hbar}(\mathbb{C}_\imath^s), \hat{\hbar}(\mathbb{C}_\jmath^t)) < \hat{\hbar}(\mathbb{C}_\imath^s, \mathbb{C}_\jmath^t)$, i.e., $\exists_{C_p \in \mathbb{C}_\imath^s, C_q \in \mathbb{C}_\imath^t} \forall_{C_a, C_b, C_q, C_p \in \{\mathbb{C}_\imath^s \cup \mathbb{C}_\imath^t\}, a \neq b \neq q \neq p} \hbar(C_p, C_q) \geq \hbar(C_a, C_b)$. Using a linkage function, subclusters C_p and C_q from ζ_i and ζ_j will be merged before each cluster merges its all own subclusters. Thus, the clusters which can be extracted from the dendrogram are subtrees containing points from two clusters or partial points from one cluster. □

Eq. (5) stipulates the condition that the linkage function shall enable each subtree to merge all members of the same cluster first, before merging with the subtree of the other cluster to form the final tree of the dendrogram. Otherwise, an entanglement has occurred.

[3] When using a dissimilarity linkage function such as a Euclidean distance function, the Equation under the line 6 in Algorithm 1 should be $\arg\min_{C_i, C_j \in \mathbb{C}, i \neq j} \hbar(C_i, C_j)$.

Corollary 1. *An entanglement between two clusters ζ_i and ζ_j is said to have occurred in the dendrogram, produced by the agglomerative clustering algorithm with a kernel linkage function $\hbar$, when there is a violation of Eq. 5 at $\imath = s, \jmath = t$ such that*

$$\exists_{s\le\imath,t\le\jmath}\ \min(\hat{\hbar}(\mathbb{C}_s^i), \hat{\hbar}(\mathbb{C}_t^j)) < \hat{\hbar}(\mathbb{C}_s^i, \mathbb{C}_t^j) \tag{6}$$

If one or more entanglements have occurred, then the set of all possible subclusters in the dendrogram does not contain clusters ζ_i and ζ_j, but their corrupted or partial versions.

Thus, if entanglements could not be avoided (e.g., in a dataset where clusters are very close to each other or even overlapping), an hierarchical clustering algorithm shall seek to achieve (i) an entanglement at $\imath, \jmath$ as high values as possible; and (ii) the least number of entanglements, in order to reduce the impact of incorrect membership assignment.

We use the following two indicators to assess the severity of the entanglement in a dendrogram.

The *number of entanglements* in a dendrogram can be counted as follows: At every merge, the node is labelled with either a cluster label (if the two subclusters before merging have the same cluster label) or neutral (if the two subclusters have different labels or one of them has a neutral label). Every time a neutral label is used in a node, an entanglement has occurred; and the number of entanglements is incremented by 1.

The entanglement level is the sum of s and t (in Eq. (6)) when an entanglement occurs. The *average level of entanglements* is the ratio of the total level of all entanglements and the number of entanglements, where the level of entanglement at a merge is the number of steps used to reach that merge in the process.

4. Why using data-dependent kernel?

Here we discuss the density bias of T-AHC when a data-independent kernel is used and the reason why a data-dependent kernel can deal with clusters of varied densities better than a data-independent kernel in the following two subsections.

4.1. T-AHC has density bias when a data-independent kernel/distance is used

Our investigation leads to the following observation:

Observation 1. T-AHC has a bias that links points in a dense region ahead of points in a sparse region in general. This bias is due to the use of a data-independent kernel/distance.

Single-linkage clustering builds the dendrogram where the heights are related to the 1-nearest neighbour density estimate, i.e., the points with higher heights (are merged later on the dendrogram) tend to be in low-density regions [10]. This linking bias often leads to poor clustering outcomes when adjacent clusters have different densities, i.e., the boundary points from a sparse cluster may link to its neighbouring dense cluster before linking back to the sparse cluster. Note that this bias has no issue if the clusters are clearly separated.

Although other linkage functions could be less sensitive to the density distribution and may eliminate the issue in the cluster boundary regions, we have the following observation:

Definition 5. Two clusters C_a and C_b are said to be well separated if any points in the valley V between these clusters have a density less than any points in either cluster, i.e., $\rho'(z) < \rho(x)$, where ρ is the density of individual distribution of either C_a or C_b; ρ' is the density of the joint distribution of $C_a \cup C_b$; $z \in V$ which is part of the joint distribution of $C_a \cup C_b$; and $x \in C$ of individual cluster C_a or C_b.

Observation 2. With a data-independent kernel-based linkage function, the bias of T-AHC creates entanglements if the clusters are not well separated.

For example, the single-linkage function relies on the nearest neighbour similarity/distance calculation. Given two adjacent clusters C_d and C_s, if $\exists_{x\in C_d, y\in C_s}\forall_{z,y\in C_s, z\neq y} K(x,y) > K(y,z)$, then y will engage in an entanglement using this linkage function. In other words, every entanglement involves a boundary point from C_s which is more similar to a boundary point from C_d than other points from C_s.

Observation 3. Different linkage functions produce different degrees of entanglements but the bias remains: T-AHC links points in a dense region ahead of points in a sparse region.

This can be seen from the example in Table 2, where four commonly used linkage functions shown in Table 1 are examined. When Gaussian kernel is used, independent of the linkage functions used, each dendrogram has low linkage $\hbar$ values in dense regions (three dense clusters) and high $\hbar$ values in sparse regions (one sparse cluster (brown)).

We contend that the root cause of this bias is the use of data-independent similarity/distance.

To reduce/eliminate this bias, we need a similarity that is data-dependent.

4.2. How data-dependent kernel helps

Here we show that a data-dependent kernel called Isolation Kernel (IK) [12,14], which adapts its similarity measurement to the local density, significantly reduces the density bias posed by distance and data-independent kernel.

The unique characteristic of Isolation Kernel [12,14] is: **two points in a sparse region are more similar than two points of equal inter-point distance in a dense region**, i.e, $\forall x, y \in \mathcal{X}_s$, $\forall x', y' \in \mathcal{X}_d$ if $\| x - y \| = \| x' - y' \|$ then

$$K_\psi(x, y) > K_\psi\left(x', y'\right) \tag{7}$$

where $\mathcal{X}_s$ and $\mathcal{X}_d$ are two subsets of points in sparse and dense regions of $\mathbb{R}^d$, respectively; and $\| x - y \|$ is the distance between x and y.[4]

Isolation Kernel deals more effectively with clusters with significantly different densities than data-independent kernels. This is because the isolation mechanism produces large partitions in a sparse region and small partitions in a dense region. Thus, the probability of two points from the dense cluster falling into the same isolating partition is lower than two points of equal inter-point distance from the sparse cluster. This gives rise to the data-dependent kernel characteristic mentioned above.

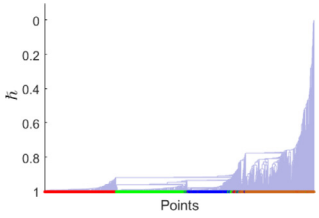
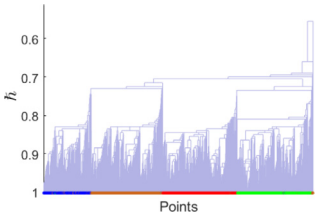
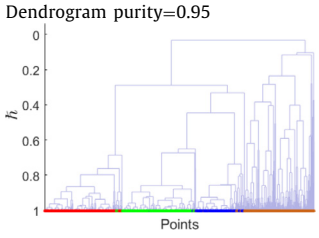
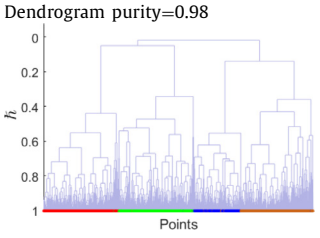
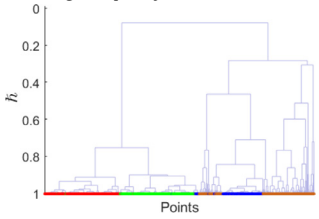
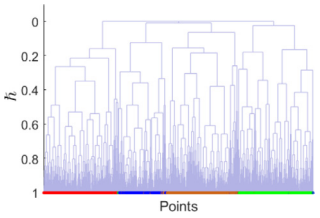
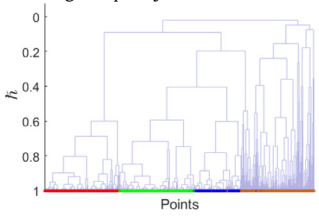
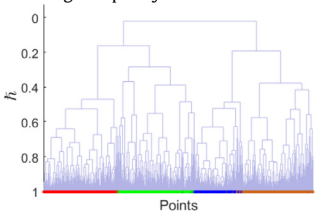
As a consequence, the boundary points from a sparse cluster become less similar to the boundary points from a dense cluster. Thus, IK reduces the number of entanglements, and increases s and t in Eq. (6) if an entanglement does occur in comparison with using a data-independent Kernel in T-AHC.

Note that the condition in Eq. (5) also applies when Isolation Kernel is used. However, the influence of its data-dependency is significant, as described below.

The isolation partitioning mechanism used to create IK is based on random samples from a given dataset. It yields two effects. First, the mechanism produces small partitions in dense clusters and large partitions in sparse clusters [12,14]. The net effect is that every cluster has almost the same uniform distribution when trans-

[4] The proof of this characteristic is in the paper [12].

Table 2
Comparison of dendrograms produced by T-AHC with four linkage functions using Gaussian Kernel and Isolation Kernel on the dataset shown in Fig. 1 a. The colours at the bottom row in each dendrogram correspond to the true cluster labels of all points shown in Fig. 1 a.

	Gaussian Kernel	Isolation Kernel
Single-linkage	Dendrogram purity=0.77  No. entanglements= 242 Avg. entanglement level=1411	Dendrogram purity=0.97  No. entanglements=170 Avg. entanglement level=1477
Average-linkage	Dendrogram purity=0.95  No. entanglements=103 Avg. entanglement level=1350	Dendrogram purity=0.98  No. entanglements=100 Avg. entanglement level=1445
Complete-linkage	Dendrogram purity=0.88  No. entanglements=105 Avg. entanglement level=1394	Dendrogram purity=0.97  No. entanglements=98 Avg. entanglement level=1490
Weighted	Dendrogram purity=0.91  No. entanglements=106 Avg. entanglement level=1360	Dendrogram purity=0.97  No. entanglements=102 Avg. entanglement level=1504

formed using MDS[5] into a Euclidean space. This is shown in Fig. 2. This effect was also observed in another data-dependent dissimilarity measure using a similar isolation mechanism, though it is not a kernel (see [11] for details.)

Second, since data points in the valleys between clusters are less likely (than those within each cluster) to include in the sample used to create IK, each partition has a tiny or no chance to cover more than one cluster. As a result, the gaps/valleys between clusters become more pronounced in the transformed MDS space. Fig. 2 shows a comparison between the MDS plots of GK and IK when they are used to transform the same dataset shown in Figure 2(a).

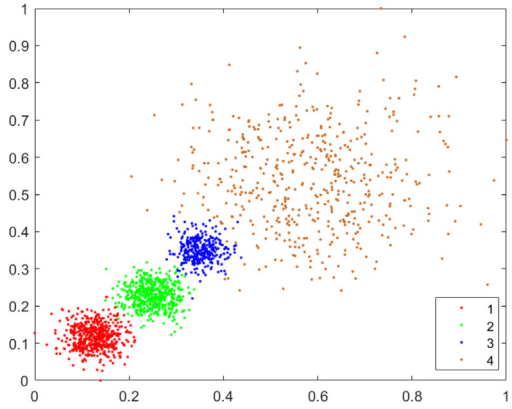
These two net effects lead to the following observation:

Observation 4. T-AHC using a linkage function derived from Isolation Kernel has little or no bias towards points in dense regions, regardless of the relative density between clusters.

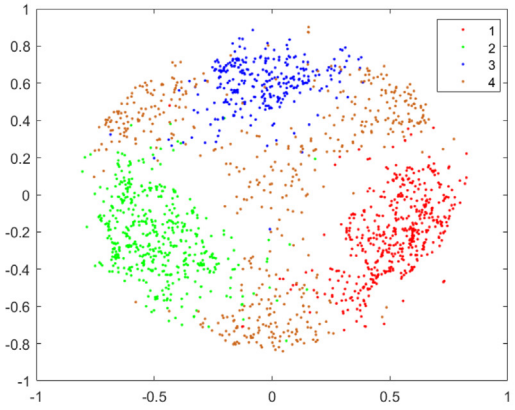
In other words, the distribution of merges in the dendrogram produced by T-AHC becomes more uniform over all clusters, as a result of the first effect. Yet, the entanglements (i.e., merges between two different clusters) become less, as a consequence of the second effect. This can be seen from all four dendrograms shown in Table 2 using four different linkage functions. The $\hbar$ values of every Isolation kernel-based linkage function are more uniformly distributed than those of the corresponding Gaussian kernel-based linkage function. Yet, the number of entanglements due to IK is less, and the average entanglement level is higher.

In summary, with Isolation Kernel: (a) an entanglement is less likely to occur since T-AHC always seeks to merge two subclusters that are most similar at each step; and (b) if an entanglement occurs, it will happen at higher values of s and t (in Eq. (6)) than

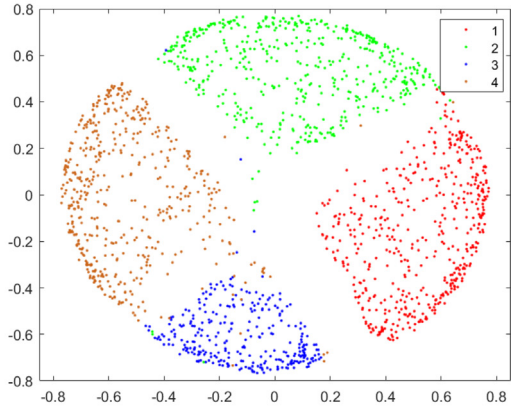
[5] Multidimensional scaling (MDS) is used for visualising the information contained in a similarity matrix [34]. It placed each data point in a 2-dimensional space, while preserving as well as possible the pairwise similarities between points.



(a) Dataset: 4 clusters of different densities



(b) MDS plot using GK



(c) MDS plot using IK

Fig. 2. MDS plot using Gaussian Kernel and Isolation Kernel on the dataset in (a).

those due to Gaussian kernel. As a result, the impact of incorrect assignment will be smaller than that when a data-independent kernel linkage function is used.

This is verified using dendrogram purity in Eq. (4) [33] which is an objective measure assessing the quality of a dendrogram produced by an AHC algorithm. As shown in Table 2, Isolation Kernel always has higher dendrogram purity than the Gaussian kernel in every linkage function. This result is also reflected in the number of entanglements (the lower the better) and the average entanglement level (the higher the better), as stipulated in relation to Eq. (6).

4.3. Demonstration on an image segmentation task

Here we use an image segmentation task to demonstrate the density bias of T-AHC on a real-world image, as shown in Fig. 3. The scatter plot of this image in the LAB space [35] shows that there is a clear gap between a dense cluster (representing the sky object) which is in close proximity to a sparse cluster (representing the building object) in the LAB space, although the sparse cluster has some dense areas. The dendrogram produced by IK is much better than that by GK, judging from the dendrogram purity results.

When using the single-linkage AHC with Gaussian Kernel, the dendrogram produces a result in which the building object is partially merged with the sky object if the best cut is applied, as shown in the first row of Table 3. This is the effect of varied cluster densities in the LAB space. In contrast, using the Isolation Kernel, the building and sky are well separated, as shown in the second row in Table 3.

4.4. Section summary

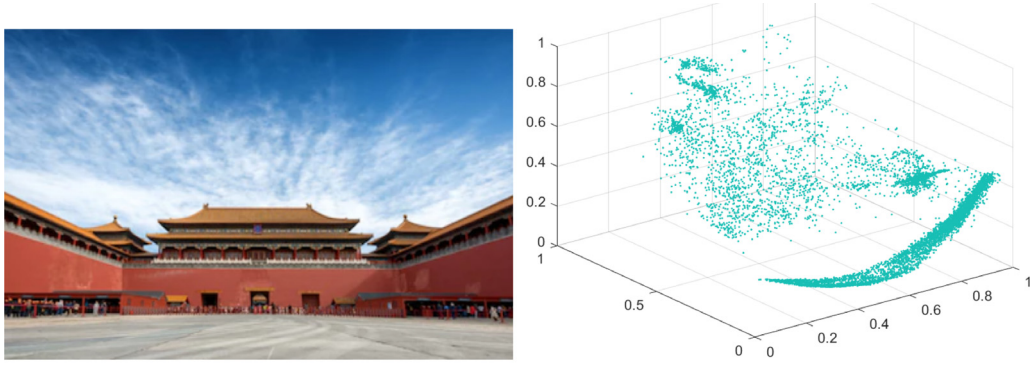
We found that T-AHC using Gaussian Kernel has density bias, a known bias for T-AHC using distance. This bias heightens the severity of entanglements between clusters. As a result, T-AHC using Gaussian kernel will have difficulty separating the clusters successfully on a dataset with clusters of varied densities. This phenomenon can be explained with the condition specified in Eq. (5).

We contend that the root cause of this bias is the use of a data-independent kernel. We show that using Isolation Kernel—because it adapts its similarity to local density—the resultant T-AHC has less density bias which reduces the severity and the number of entanglements. As a result, T-AHC using Isolation Kernel usually produces a better dendrogram than that using Gaussian Kernel.

We show in the next section that the same approach we suggested here, i.e., using Isolation Kernel instead of a data-dependent kernel/distance, can be applied to other hierarchical clustering algorithms, apart from T-AHC. We also show that not all data-dependent kernels are the same.

5. Empirical evaluation

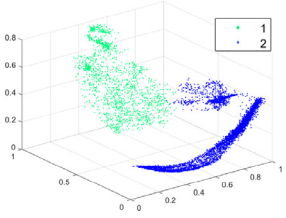
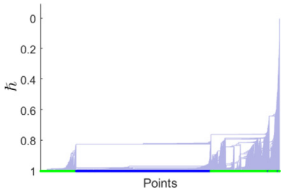
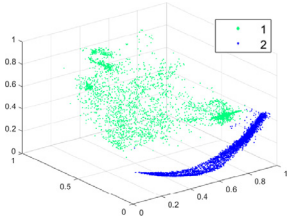
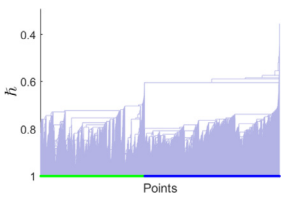
We provide the experimental settings and report the evaluation results in this section. In the experiment, we compared kernel-based hierarchical clustering with traditional hierarchical cluster-



(a) Image (b) Pixels in LAB space

Fig. 3. An example dataset of an image.

Table 3
Image segmentation results on the image using the LAB space shown in Fig. 3, produced from the AHC algorithm with either Gaussian Kernel or Isolation Kernel. Each scatter plot in the 'LAB Space' column illustrates the top two clusters identified (indicated by different colours) where the building (mainly green and sparse) is separated from the sky (mainly blue and dense). Columns 'Cluster 1' and 'Cluster 2' show the segmentation results on the image. The colours at the bottom row in each dendrogram correspond to the true cluster labels of all points, i.e, blue for sky and green for building.

	LAB Space	Dendrogram	Cluster 1	Cluster 2
GK single-linkage				
	Dendrogram purity=0.87	No. entanglements=210	Avg. entanglement level=7640	
IK single-linkage				
	Dendrogram purity=0.99	No. entanglements=16	Avg. entanglement level=8091	

ing algorithms including T-AHC with single-linkage and complete-linkage, the potential-based hierarchical agglomerative method PHA [17], the graph-based agglomerative method GDL [7], the density-based method HDBSCAN [16] and optimal transport-based HC-OT [18]. All algorithms used in our experiments are implemented in MATLAB except HC-OT is implemented in R by their original authors.

5.1. Parameter settings

Each parameter of an algorithm is searched within a certain range. Table 4 shows the search ranges for all parameters; and we report the best performance on each dataset. As IK has the randomisation in the sub-sampling process, we report the best results over 10 independent trials. The k is the number of nearest neighbours for constructing the KNN graph and Adaptive Gaussian Kernel from Eq. (1). The s in PHA is the scale factor for calculating the potential field [17]. l and c in HDBSCAN are minimum cluster sizes and minimum samples, respectively. The λ in HC-OT is the hyperparamter to calculate Sinkhorn distance [18].[6]

Table 4
Parameters and their search ranges for each algorithm. For AGK and IK, we searched all integer values within $[2, \lceil n/2 \rceil]$. T-AHC is an AHC using one of the four linkage functions in Table 1.

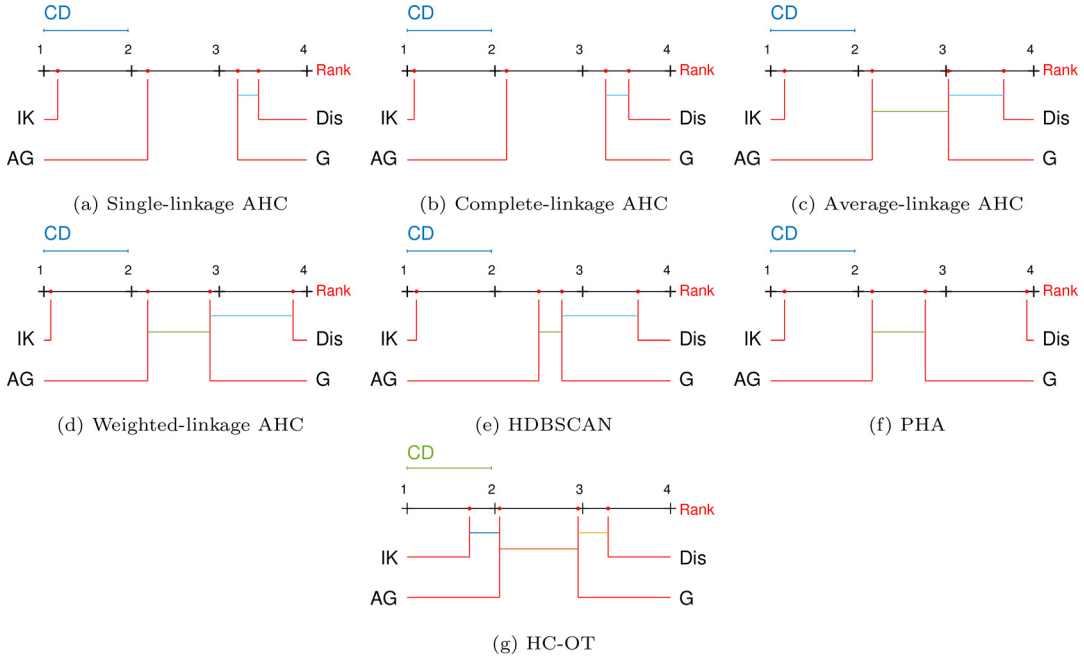
Algorithm/Kernel	Parameter search range
T-AHC	$\kappa \in \{2, \ldots, 30\}$
HDBSCAN	$l \in [2, 100]$, $c \in [2, 100]$
PHA	$s \in \{5, 10, 15, 20, 25, 30\}$
GDL	$k \in \{5, 10, 15, 20, 25, 30, 70, 100\}$
HC-OT	$\lambda \in \{5, 10, 15, 20, 25, 30\}$
Gaussian Kernel	$\sigma = 2^m$, $m \in [-5, 5]$
Adaptive Gaussian Kernel	$k \in [2, \lceil n/2 \rceil]$
Isolation Kernel	$\psi \in [2, \lceil n/2 \rceil]$, $t = 200$

The experiments use 19 real-world datasets with different data sizes and dimensions from UCI Machine Learning Repository [37]. The data properties are shown in the first four columns in Table 5. All datasets were normalised using the Min-Max normalisation to

[6] The guide for parameter selection for HDBSCAN (so as its source code) is from https://hdbscan.readthedocs.io/ [36]. All other codes used in empirical evaluations are published by the original authors. It is worth noting that not all parameters are valid for GDL in all datasets because the based k-nearest neighbours graph methods are sensitive to the parameter and similarity measure. Here we only record the available results. In addition, GDL source code does not output the dendrogram, thus we have to omit it in the dendrogram purity evaluation.

Table 5
Properties of the datasets used in the experiments.

Name	#instance	#Dim.	#Clusters
banknote	1372	4	2
thyroid	215	5	3
seeds	210	7	3
diabetes	768	8	2
vowel	990	10	11
wine	178	13	3
shape	160	17	9
segment	2310	19	7
WDBC	569	30	2
spam	4601	57	2
control	600	60	6
hill	1212	100	2
LandCover	675	147	9
musk	476	166	2
LSVT	126	310	2
Isolet	1560	617	26
COIL20	1440	1024	20
lung	203	3312	5
ALLAML	72	7129	2



(a) Single-linkage AHC (b) Complete-linkage AHC (c) Average-linkage AHC

(d) Weighted-linkage AHC (e) HDBSCAN (f) PHA

(g) HC-OT

Fig. 4. Critical difference (CD) diagram of the post-hoc Nemenyi test ($\alpha = 0.10$) for dendrogram purity. Two kernel methods are not significantly different if there is a line linking them.

yield each attribute to be in [0,1] before the experiments began. Some of these datasets have been shown to have clusters with varied densities, e.g., thyroid, seeds, wine, WDBC and segment [38].

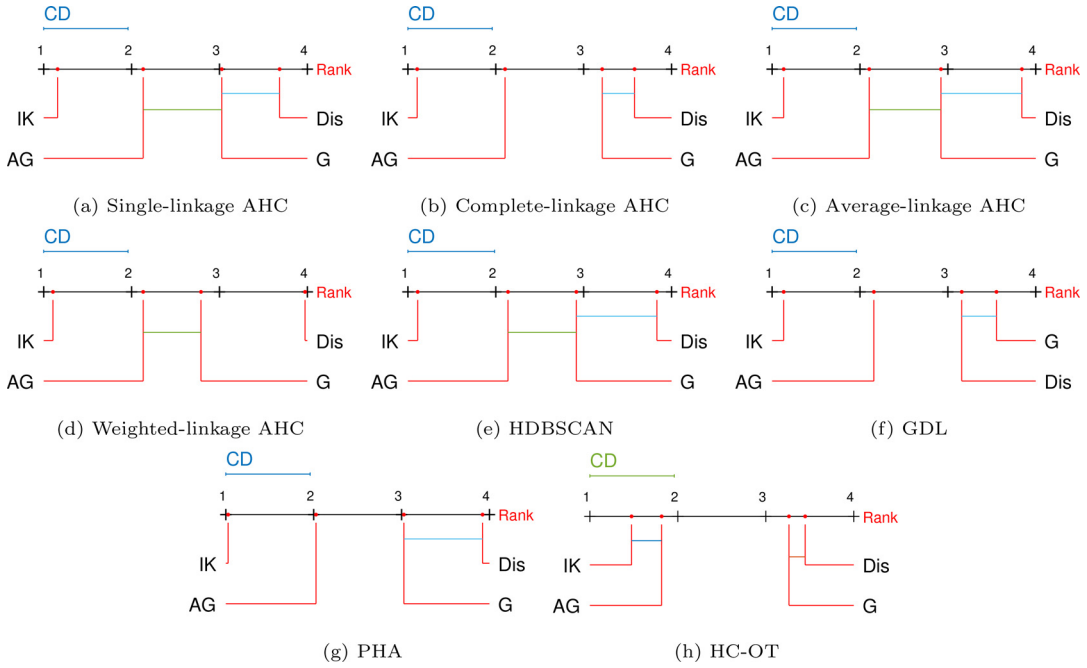
5.2. Hierarchical clustering evaluation

We first evaluate cluster trees using *Dendrogram Purity* shown in Eq. (4) for different AHC algorithms. To compute *Dendrogram Purity* of a dendrogram $\mathcal{T}$ with ground-truth clusters C^*, we first find all pairs of points (x_i, x_j) that belong to the same ground-truth cluster from a cluster tree, and then identify the smallest sub-dendrogram (subtree) containing x_i and x_j. The fraction of leaves in that sub-dendrogram that are in the same ground-truth cluster as x_i and x_j is calculated as a purity score for the pair (x_i, x_j). The overall *Dendrogram Purity* is the average purity score over all pairs of points.

In Table 6, we report the *Dendrogram Purity* for the four T-AHC algorithms (single-linkage, complete-linkage, average-linkage and weighted-linkage), PHA, HDBSCAN, HC-OT[7] The best result on each dataset is boldfaced. Key observations from Table 6 are as follows.

- The kernerlised version of each clustering algorithm achieves better or equivalent performance than the original distance-based version.
- The IK-based versions have the highest average *dendrogram purity* for all algorithms, as reported in the last row of Table 6. IK-based HDBSCAN has the highest score of 0.77, followed by IK-based Average-linkage AHC and IK-based Weighted-linkage AHC with a score of 0.76.
- The data-dependent kernels (AG and IK) obtain a significant improvement over the distance and GK for all algorithms on many datasets, e.g., seeds, wine, WDBC and segment.

[7] We provide the details of kernelised PHA, GDL and HC-OT in Appendix B, Appendix C and Appendix D, respectively.

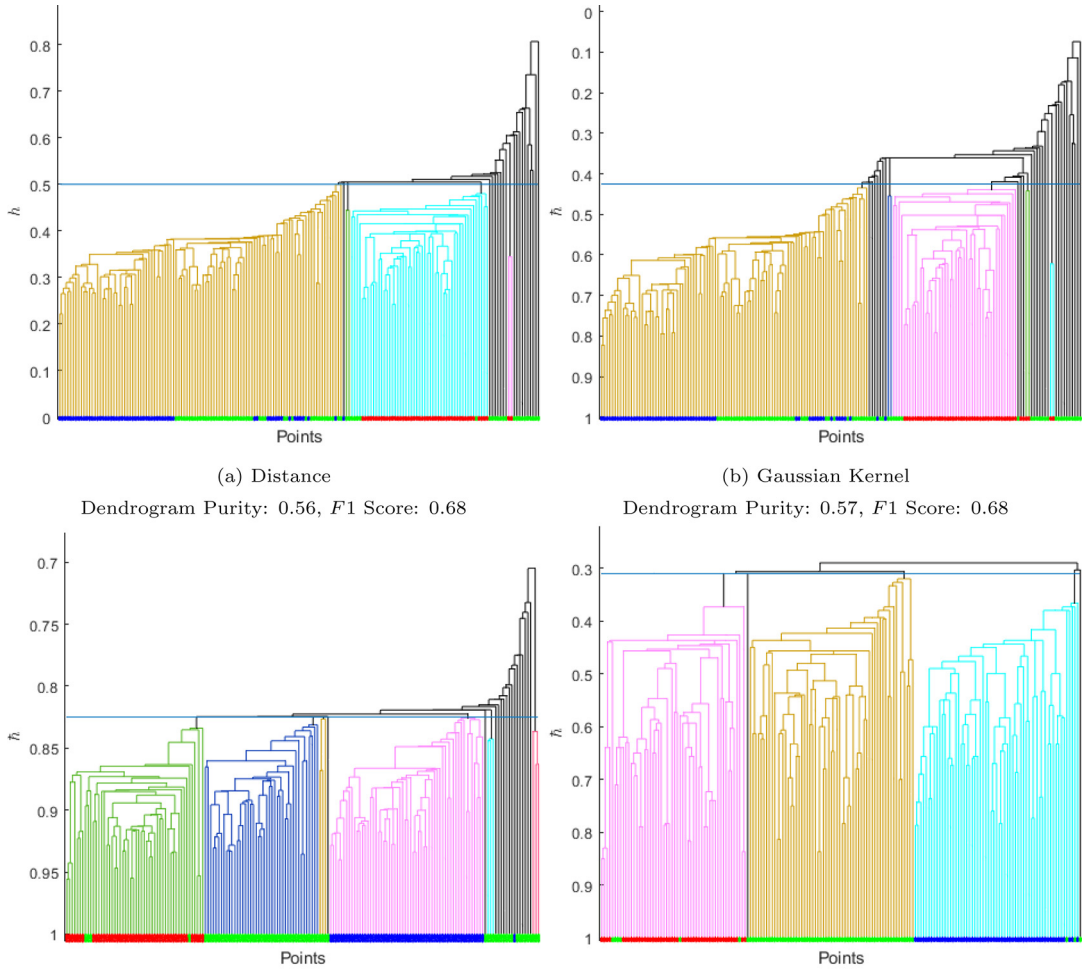


(a) Single-linkage AHC (b) Complete-linkage AHC (c) Average-linkage AHC

(d) Weighted-linkage AHC (e) HDBSCAN (f) GDL

(g) PHA (h) HC-OT

Fig. 5. Critical difference (CD) diagram of the post-hoc Nemenyi test ($\alpha = 0.10$) for $F1$ scores. Two measures are not significantly different if there is a line linking them.



(a) Distance
Dendrogram Purity: 0.56, $F1$ Score: 0.68

(b) Gaussian Kernel
Dendrogram Purity: 0.57, $F1$ Score: 0.68

(c) Adaptive Gaussian Kernel
Dendrogram Purity: 0.82, $F1$ Score: 0.90

(d) Isolation Kernel
Dendrogram Purity: 0.92, $F1$ Score: 0.94

Fig. 6. Dendrograms (cluster trees) produced by four versions of the single-linkage AHC on the wine dataset. The horizontal line shown on each dendrogram is the threshold level used for cluster extraction to obtain the best $F1$ Score. The colours of vertical lines indicate the clustering result, i.e., all points having the same colour vertical line are grouped into the same cluster. The bottom colour bar on each dendrogram shows the ground-truth cluster labels.

Table 6
Clustering results in *Dendrogram Purity*. The best result on each dataset is boldfaced.

Dataset	Single-l AHC				Complete-l AHC				Average-l AHC				Weighted-l AHC				PHA				HDBSCAN				HC-OT			
	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK
banknote	0.92	0.92	**.99**	**.99**	0.63	0.63	0.80	**.82**	0.68	0.96	0.78	**.98**	0.64	0.78	0.73	**.94**	0.62	0.71	0.68	**.76**	0.92	0.92	0.96	**.99**	**.99**	**.99**	**.99**	**.99**
thyroid	0.92	0.92	0.93	**.93**	0.89	0.89	0.95	**.97**	0.93	0.92	0.94	**.96**	0.86	0.92	0.95	**.97**	0.91	0.92	0.92	**.93**	0.92	0.93	0.92	**.96**	0.92	0.92	**.94**	0.92
seeds	0.69	0.69	0.81	**.85**	0.75	0.75	0.84	**.86**	0.85	0.85	0.88	**.89**	0.76	0.74	0.85	**.88**	0.84	0.84	**.89**	0.88	0.65	0.73	0.79	**.84**	0.84	0.84	0.88	**.89**
diabetes	0.67	0.67	0.67	**.70**	0.66	0.66	0.67	**.68**	0.67	0.67	0.67	**.70**	0.63	0.64	0.66	**.67**	0.67	0.68	0.68	**.69**	0.68	0.68	0.67	**.69**	**.67**	**.67**	0.64	**.67**
vowel	0.20	0.20	0.24	**.26**	0.22	0.22	0.25	**.27**	0.22	0.22	0.24	**.28**	0.23	0.25	0.27	**.30**	0.20	0.22	0.23	**.24**	0.19	0.19	0.23	**.24**	0.20	0.20	0.24	**.27**
wine	0.68	0.68	0.84	**.90**	0.92	0.92	0.96	**.98**	0.89	0.92	0.94	**.96**	0.81	0.82	0.93	**.94**	0.73	0.74	0.91	**.94**	0.63	0.79	0.75	**.89**	0.89	0.89	0.92	**.93**
shape	0.69	0.69	**.70**	**.70**	0.65	0.65	0.68	**.69**	0.65	0.65	**.72**	**.72**	0.68	0.68	0.73	**.74**	0.67	0.67	0.70	**.74**	0.68	0.70	0.70	**.75**	0.70	0.70	**.72**	**.72**
segment	0.60	0.60	0.64	**.67**	0.62	0.62	0.65	**.67**	0.65	0.67	0.69	**.72**	0.59	0.61	0.66	**.71**	0.65	0.70	0.72	**.74**	0.59	0.59	0.64	**.67**	0.62	0.62	0.68	**.71**
WDBC	0.71	0.72	0.74	**.90**	0.79	0.79	0.89	**.91**	0.86	0.89	0.89	**.93**	0.83	0.86	0.90	**.94**	0.73	0.79	0.87	**.95**	0.73	0.73	0.71	**.90**	0.71	0.71	0.91	**.94**
spam	0.59	0.56	0.57	**.64**	0.57	0.56	0.60	**.69**	0.58	0.56	0.64	**.69**	0.58	0.59	0.66	**.69**	0.55	0.57	0.65	**.68**	0.55	0.56	0.57	**.94**	0.59	0.59	0.60	**.62**
control	0.73	0.73	0.80	**.87**	0.81	0.81	0.82	**.85**	0.81	0.81	0.91	**.94**	0.77	0.75	0.89	**.90**	0.66	0.67	0.79	**.82**	0.71	0.71	0.75	**.83**	0.77	0.77	0.77	**.90**
hill	0.50	0.50	**.51**	**.51**	0.50	0.50	**.51**	**.51**	0.50	0.50	**.51**	**.51**	0.50	0.50	**.51**	**.51**	0.50	0.50	0.50	**.51**	0.50	0.50	**.51**	**.51**	0.50	0.50	0.51	**.50**
LandCover	0.30	0.30	0.39	**.55**	0.52	0.52	0.58	**.60**	0.56	0.59	0.63	**.64**	0.48	0.56	0.59	**.60**	0.44	0.48	0.53	**.61**	0.27	0.27	0.35	**.48**	0.33	0.33	0.46	**.58**
musk	0.54	**.64**	0.54	0.55	0.56	**.65**	0.56	0.57	0.55	**.65**	0.56	**.64**	0.55	**.65**	0.56	0.57	0.54	**.65**	0.56	0.56	0.54	**.94**	0.56	0.93	0.54	**.65**	0.54	0.55
LSVT	0.58	0.60	0.63	**.67**	0.62	0.62	0.65	**.71**	0.63	0.63	0.65	**.67**	0.61	0.64	0.66	**.69**	0.59	0.60	0.67	**.71**	0.58	0.61	0.62	**.64**	0.61	0.61	**.67**	**.67**
Isolet	0.27	0.27	0.36	**.58**	0.48	0.48	0.56	**.57**	0.57	0.60	0.62	**.68**	0.54	0.59	0.59	**.66**	0.39	0.48	0.50	**.64**	0.21	0.57	0.29	**.61**	0.31	0.31	0.44	**.59**
COIL20	0.91	0.91	0.99	**1.00**	0.66	0.67	0.66	**.70**	0.72	**.93**	0.79	**.93**	0.72	0.97	0.82	**.98**	0.70	0.76	0.76	**.79**	0.89	0.90	0.99	**1.00**	0.91	0.91	**1.00**	0.85
lung	0.76	0.91	0.82	**.95**	0.89	0.92	0.93	**.96**	0.94	0.94	**.96**	**.96**	0.94	0.95	0.95	**.96**	0.77	0.95	0.86	**.98**	0.87	0.87	0.90	**.97**	0.76	**.95**	0.84	0.92
ALLAML	0.68	0.68	0.68	**.72**	0.67	0.68	0.73	**.74**	0.68	0.69	0.72	**.74**	0.67	0.70	0.72	**.79**	0.68	0.72	0.72	**.77**	0.66	**.74**	0.70	**.74**	0.68	0.72	**.77**	0.75
Average	0.63	0.64	0.68	0.73	0.65	0.66	0.70	0.72	0.68	0.72	0.72	0.76	0.65	0.70	0.72	0.76	0.62	0.67	0.69	0.73	0.62	0.68	0.66	0.77	0.66	0.68	0.71	0.74

- For each of the four T-AHC algorithms, PHA and HDBSCAN, their IK-based versions have the best *dendrogram purity* out of the four versions on all datasets, except musk and seeds. The IK-based version of HC-OT has the best performances on 14 out of 19 datasets.

We conduct a Friedman test with the post-hoc Nemenyi test [39] to examine whether the difference in *Dendrogram Purity* of any two kernel methods is significant. Those four kernel methods (used in an algorithm) are ranked based on their *Dendrogram Purity* on each dataset, where the best one is ranked as 1st and so on. Then the critical difference (CD) is computed using the post-hoc Nemenyi test. Two kernel methods are significantly different if the difference in their average ranks is larger than CD. Fig. 4 shows that IK is significantly better than all other kernel methods for every algorithm.

We have also evaluated the Ward's linkage function [40] and the results can be found in Appendix E.

5.3. Flat clustering evaluation

To evaluate the flat clustering results, we use the original cluster extraction method in each algorithm, and compare the extracted clusters with the ground truth cluster using $F1$ score which is a trade-off between the *Precision* and *Recall* [41]. Note that we use a global cut for T-AHC to extract the k subclusters on the dendrogram.

Given a clustering result, the *precision* score P_i and the *recall* score R_i for each cluster C_i are calculated based on the confusion matrix, and the $F1$ score of C_i is the harmonic mean of P_i and R_i. The Hungarian algorithm [42] is used to search for the optimal match between the clustering results and ground-truth clusters. The overall $F1$ score, which is the unweighted average overall matched clusters, is defined as:

$$F1 = \frac{1}{k}\sum_{i=1}^{k}\frac{2 \times P_i \times R_i}{P_i + R_i} \tag{8}$$

Note that other evaluation measures such as Purity [43] and Adjusted Mutual Information [44] do not take into account noise points identified by a clustering algorithm. They are not suitable for HDBSCAN in the evaluation section because these scores can provide a misleadingly good clustering result when HDBSCAN assigns many points to noise.

Table 7 reports the experimental results of the traditional AHC and the other four algorithms. The key observations are:

- **Kernel improves the flat clustering performance of distance measure**. Every clustering algorithm that employs a kernel has better or equivalent clustering performance than that using distance in terms of the average the $F1$ score shown in the last row in Table 7. The only exception is GDL, where Gaussian Kernel is marginally worse than distance. Even in GDL, IK and AGK are always better than distance (except on thyroid and lung only).
- **Among the three kernels, Isolation Kernel (IK) produces the best $F1$ Score**. This occurs on all datasets and every algorithm except complete-linkage and HC-OT on a few datasets. IK has the best $F1$ on at least 15 out of the 19 datasets for every clustering algorithm. The closest contender is AGK which has the best $F1$ on 1 to 3 datasets on four algorithms, but it produces no best $F1$ on complete-linkage AHC.
- **IK took the lead in F1 over others by a huge gap on some datasets**, even compared with the other two kernels. For example, (i) thyroid, WDBC, LSVT, Isolet and lung using single-linkage AHC; (ii) thyroid, spam, control, musk and ALLAML using HDBSCAN; (iii) LandCover and Isolet using PHA; and (iv) diabetes, control and lung using HC-OT.

Table 7
Clustering results in $F1$ score. A larger $F1$ score indicates a better clustering result. The best result on each dataset is boldfaced.

Dataset	Single-l AHC				Complete-l AHC				Average-l AHC				Weighted-l AHC				PHA				HDBSCAN				GDL				HC-OT			
	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK	Dis	G	AG	IK
banknote	0.95	0.95	**.99**	**.99**	0.60	0.60	0.79	**.81**	0.61	0.97	0.77	**.99**	0.62	0.86	0.73	**.94**	0.68	0.74	0.78	**.96**	0.69	0.69	0.94	**.95**	0.98	0.91	**.99**	**.99**	**0.99**	**.99**	**.99**	**.99**
thyroid	0.58	0.58	0.76	**.91**	0.80	0.80	0.95	**.96**	0.73	0.73	**.95**	0.91	0.75	0.75	0.93	**.95**	0.55	0.59	0.85	**.86**	0.57	0.59	0.58	**.78**	**.91**	0.73	0.71	0.84	0.58	0.58	**.93**	0.92
seeds	0.54	0.54	0.90	**.91**	0.85	0.85	0.89	**.94**	0.89	0.89	0.93	**.94**	0.81	0.85	0.91	**.92**	0.90	0.90	0.92	**.93**	0.61	0.58	0.82	**.83**	0.88	0.88	0.92	**.93**	0.89	0.89	**.93**	**.93**
diabetes	0.40	0.40	0.44	**.50**	0.53	0.53	0.68	**.70**	0.61	0.61	0.64	**.65**	0.49	0.59	0.69	**.65**	0.44	0.59	0.58	**.63**	0.42	0.47	0.43	**.52**	0.53	0.52	0.59	**.63**	0.40	0.40	0.61	**.67**
vowel	0.08	0.12	0.27	**.31**	0.29	0.29	0.33	**.34**	0.28	0.31	0.33	**.37**	0.27	0.32	0.35	**.37**	0.12	0.25	0.33	**.35**	0.24	0.27	0.32	**.33**	0.31	0.30	0.31	**.36**	0.12	0.12	0.28	**.33**
wine	0.56	0.57	**.92**	**.92**	0.94	0.94	0.95	**.98**	0.93	0.96	0.96	**.97**	0.87	0.92	0.96	**.98**	0.58	0.59	0.93	**.95**	0.51	0.55	0.82	**.91**	0.92	0.92	**.97**	**.97**	0.93	0.93	**.95**	**.95**
shape	0.44	0.52	0.75	**.77**	0.64	0.66	0.74	**.78**	0.63	0.68	**.76**	**.76**	0.67	0.68	0.74	**.77**	0.64	0.65	**.76**	**.76**	0.68	0.68	**.71**	**.71**	0.70	0.66	0.71	**.74**	0.67	0.67	**.77**	**.77**
segment	0.34	0.36	0.55	**.66**	0.65	0.65	0.73	**.75**	0.53	0.58	0.73	**.79**	0.55	0.60	0.72	**.76**	0.39	0.71	0.73	**.75**	0.61	0.60	**.69**	0.63	0.42	0.45	0.78	**.82**	0.49	0.49	**.69**	0.65
WDBC	0.40	0.67	0.40	**.93**	0.77	0.77	0.92	**.95**	0.87	0.91	0.91	**.95**	0.79	0.88	0.92	**.96**	0.40	0.41	0.88	**.94**	0.41	0.54	0.71	**.91**	0.94	0.94	0.95	**.96**	0.40	0.40	0.94	**.95**
spam	0.38	0.38	0.38	**.42**	0.42	0.51	0.64	**.73**	0.38	0.38	0.66	**.77**	0.39	0.52	0.74	**.80**	0.38	0.38	0.70	**.71**	0.28	0.35	0.48	**.87**	0.38	0.38	0.69	**.74**	**.38**	**.38**	**.38**	**.38**
control	0.23	0.23	0.60	**.79**	0.75	0.75	0.84	**.86**	0.71	0.73	0.86	**.87**	0.72	0.74	0.82	**.84**	0.46	0.63	0.73	**.81**	0.32	0.57	0.58	**.76**	0.76	0.76	**.95**	**.95**	0.56	0.56	0.64	**.93**
hill	0.34	0.37	**.51**	0.46	0.37	0.40	0.50	**.51**	0.37	0.41	**.51**	**.51**	0.39	0.40	0.51	**.52**	0.37	0.39	0.54	**.56**	0.33	0.36	0.47	**.48**	0.46	0.35	0.50	**.62**	0.37	0.37	**.51**	**.51**
LandCover	0.16	0.08	0.18	**.36**	0.59	0.60	0.64	**.65**	0.36	0.55	0.72	**.74**	0.58	0.67	0.67	**.72**	0.08	0.41	0.49	**.70**	0.18	0.28	0.39	**.55**	0.62	0.61	0.74	**.75**	0.15	0.15	0.33	**.34**
musk	0.36	**.64**	0.50	0.53	**.58**	**.58**	0.55	0.57	0.36	0.53	0.53	**.56**	0.48	**.54**	**.54**	**.54**	0.51	0.51	0.55	**.57**	0.50	0.52	0.50	**.75**	0.48	0.48	0.50	**.56**	0.47	0.49	0.47	**.56**
LSVT	0.40	0.49	0.44	**.67**	0.40	0.55	0.58	**.65**	0.40	0.60	0.58	**.62**	0.40	0.61	0.60	**.62**	0.40	0.42	0.67	**.68**	0.15	0.42	0.42	**.59**	0.51	0.51	0.60	**.63**	0.57	0.57	0.62	**.64**
Isolet	0.03	0.03	0.07	**.28**	0.33	0.39	0.59	**.68**	0.12	0.30	0.60	**.68**	0.24	0.40	0.59	**.66**	0.03	0.24	0.31	**.66**	0.09	0.35	0.38	**.55**	0.61	0.61	0.67	**.73**	0.06	0.06	0.10	**.19**
COIL20	0.28	0.33	0.96	**.97**	0.39	0.46	0.71	**.73**	0.24	0.56	0.77	**.90**	0.27	0.71	0.80	**.92**	0.44	0.49	0.73	**.76**	0.84	0.84	**.95**	**.95**	0.86	0.86	**.87**	**.87**	0.33	0.33	**.98**	0.74
lung	0.43	0.43	0.52	**.88**	0.81	0.81	0.85	**.86**	0.87	0.87	0.87	**.90**	0.87	0.91	0.88	**.94**	0.42	0.61	0.78	**.96**	0.23	0.58	0.68	**.76**	0.91	0.91	0.90	**.94**	0.43	0.44	0.50	**.90**
ALLAML	0.46	0.47	0.61	**.75**	0.62	0.62	0.74	**.75**	0.57	0.61	0.74	**.75**	0.53	0.63	0.72	**.82**	0.46	0.49	0.72	**.75**	0.36	0.49	0.53	**.73**	0.60	0.60	0.72	**.78**	0.46	0.49	**.80**	0.77
Average	0.39	0.43	0.57	0.69	0.60	0.62	0.72	0.75	0.55	0.64	0.73	0.77	0.56	0.66	0.73	0.77	0.43	0.53	0.68	0.75	0.42	0.51	0.60	0.71	0.67	0.65	0.74	0.78	0.49	0.49	0.65	0.69

It is interesting to note that Isolation Kernel achieves the largest performance improvement over distance on almost all datasets for all algorithms. In addition, using IK in complete-linkage AHC, PHA and GDL allows all these algorithms to produce similar average $F1$ scores. In contrast, using AGK allows complete-linkage AHC and GDL only to produce similar average $F1$ scores; and using GK in these two algorithms makes little difference or worse average $F1$ score than those using distance. With HC-OT, GK only slightly improves the $F1$ scores of the original version on three datasets. In contrast, IK improves the performance on 17 out of 19 datasets.

We also conducted a Friedman test with the post-hoc Nemenyi test to examine whether the difference in $F1$ scores of any two kernel methods is significant. As shown in Fig. 5, IK is significantly better than all other kernel methods for every algorithm. This result provides further evidence of the superiority of IK with respect to the clustering performance reported in the previous study where IK improves DBSCAN clustering results on datasets with varied densities [12].

5.4. Visual comparison

To visually compare the clustering results, Fig. 6 shows the dendrograms generated by the single-linkage AHC with either distance or one of the three kernels on the wine dataset.

When using the single-linkage AHC with distance or Gaussian Kernel, the blue and green (ground-truth) clusters are mixed substantially in multiple sub-trees in each of the two dendrograms.

Although the Adaptive Gaussian kernel reduces the density bias, many points from the green (ground-truth) cluster are separated from each other. In contrast, the dendrogram produced with Isolation Kernel is much better than the others, i.e., it has both the highest dendrogram purity score and the highest F1 score.

5.5. Parameter sensitivity analysis

There are two parameters for IK, i.e., the ensemble size parameter t and the sharpness (subsample size) parameter ψ. The parameter t can be set to a large value such as 200. The higher the t is, the more stable the kernel estimates but the longer the estimation time is [12]. To investigate the effects of the sharpness parameter ψ for AHC, a sensitivity analysis has been conducted on four datasets with three IK-based AHC algorithms, in which IK significantly improves the clustering performance of the original versions. We reported the average with a standard deviation of *Dendrogram Purity* and $F1$ scores over ten independent trials in Fig. 7 and Fig. 8, respectively. It can be seen from the results that the clustering performance is not too sensitive to the parameter ψ; and setting ψ around 15 could obtain a good and stable result on most datasets.

5.6. Computational complexity comparison

Table 8 compares the time and space complexities of different clustering algorithms and similarity measures.[8] Basically, all four measures have similar computational complexities.

The time complexity of producing Isolation Kernel takes $t\psi$ and calculating all pairwise similarities costs $t\psi \times n^2$, since it checks $t\psi$ partitions for each pair of two points [12].

The space complexity to store the Isolation Kernel and the whole pairwise similarity matrix are $t\psi$ and n^2, respectively.

The runtime comparison on four datasets is shown in Table 9. Note that the runtime of IK is slightly higher than the other two

[8] Here we use the distance matrix as the input for each algorithm to save running time. When data points are available, the space complexity of all algorithms can be reduced to $\mathcal{O}(n)$.

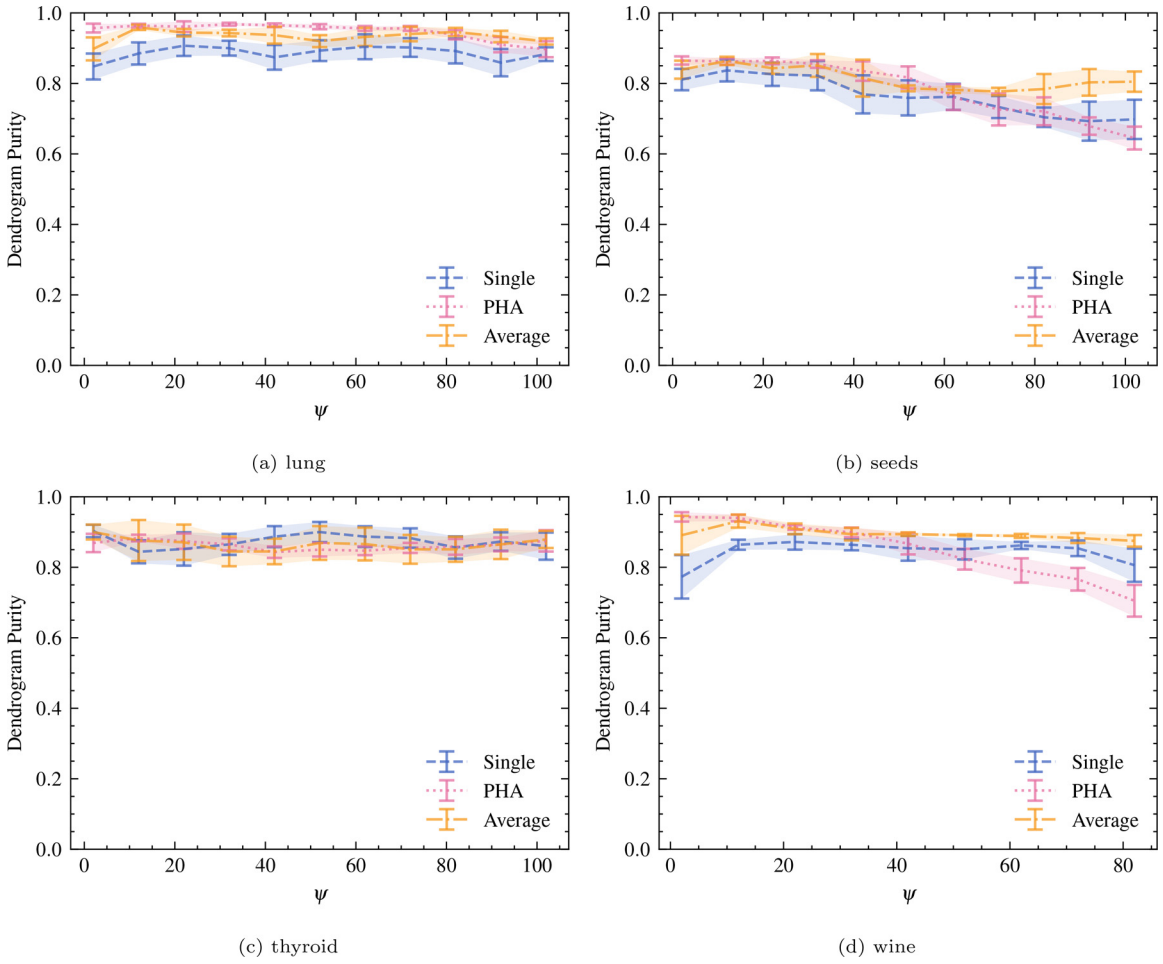


Fig. 7. Parameter sensitivity analysis of IK-based algorithms on four datasets measured in *Dendrogram Purity* score.

Table 8
Time and space complexities of AHC algorithms and distance/kernels. t and ψ in IK are the ensemble size and subsample size, respectively.

Algorithm	Time complexity	Space complexity
Single-linkage AHC	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
Complete-, Average-, Weighted-linkage AHC	$\mathcal{O}(n^2 \log n)$	$\mathcal{O}(n^2)$
GDL	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
PHA or HDBSCAN	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
HC-OT	$\mathcal{O}(n^2 \log n)$	$\mathcal{O}(n^2)$
Distance or GK or AGK	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
IK	$\mathcal{O}(t\psi n^2)$	$\mathcal{O}(t\psi + n^2)$

Table 9
Execution time (in CPU seconds) on a machine with an i7-7820X 3.60GHz processor and 32GB RAM.

Dataset	Single-linkage AHC				HDBSCAN			
	Dis	GK	AGK	IK	Dis	GK	AGK	IK
WDBC	0.00	0.01	0.01	0.08	0.03	0.04	0.04	0.07
Banknote	0.02	0.05	0.06	0.30	0.13	0.15	0.20	0.32
segment	0.06	0.13	0.20	0.93	0.44	0.48	0.61	0.98
Spam	0.28	0.59	0.86	3.57	1.76	1.86	2.45	3.87

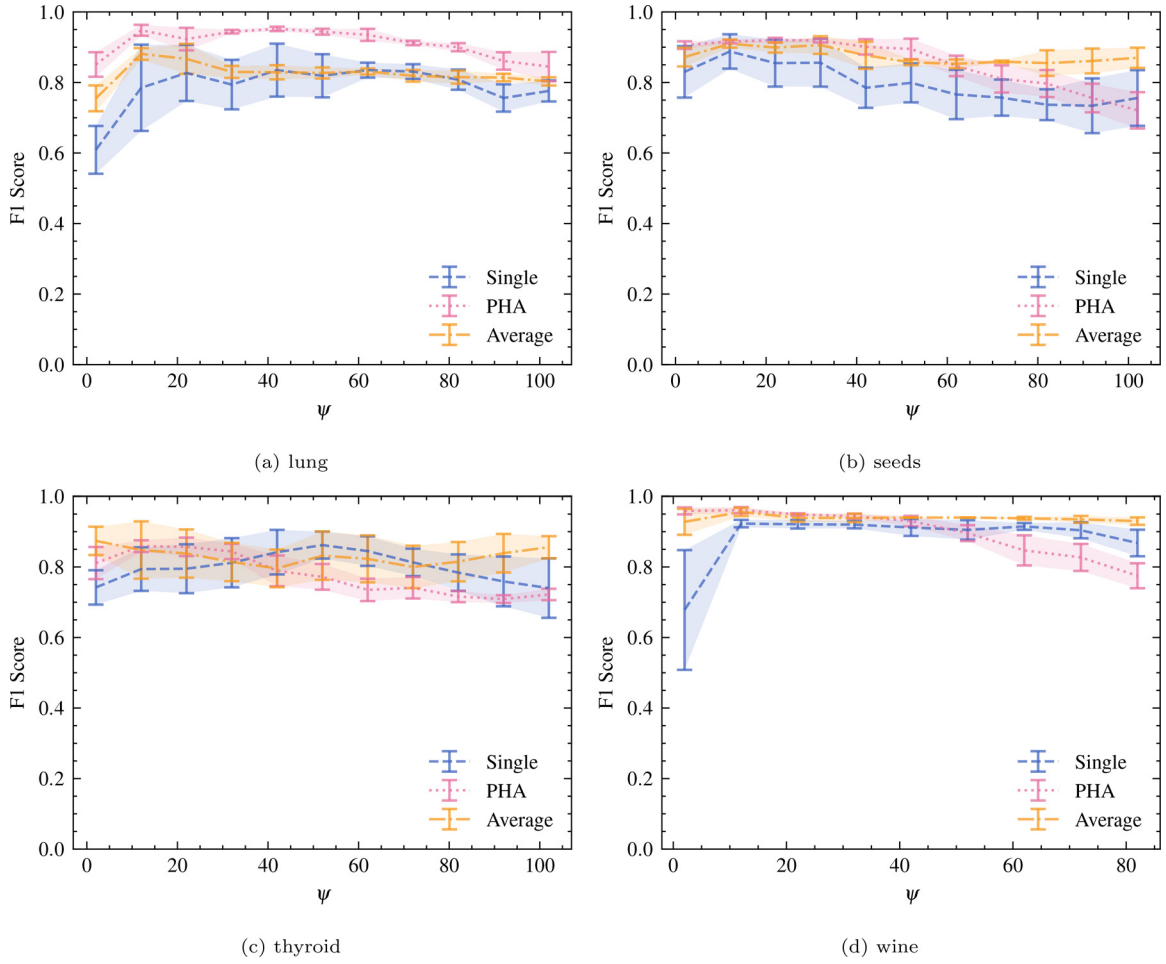


Fig. 8. Parameter sensitivity analysis of IK-based algorithms on four datasets measured in $F1$ score.

kernels because it is an ensemble method. However, this is not an issue because the Voronoi diagram implementation of IK can be accelerated by using GPU [12].

6. Conclusions

We formally establish the condition under which a linkage function must comply before it would allow an AHC to successfully link subclusters in a dataset. We also formally define a concept called *entanglement* in a dendrogram to explain the severity of linking across different subclusters during the merging process in the AHC. Two indicators, i.e., the number of entanglements and the average entanglement level, are shown to be highly correlated to dendrogram purity, an objective indicator to measure the quality of dendrogram/how good the dendrogram is.

These formal definitions have allowed us to analyse an often overlooked bias in T-AHC: existing T-AHC algorithms have a bias towards linking points in dense clusters first, before linking points in the sparse clusters.

As we contend that the root cause of this bias is due to the distance/similarity used being data-independent, the use of a well-defined *data-dependent kernel called Isolation Kernel* has been shown to reduce this bias significantly.

While the analysis was conducted with respect to T-AHC only, we propose to use Isolation Kernel to replace distance in existing distance-based AHC algorithms as a generic approach to improve their dendrograms. This approach differs from existing approaches which focus on a tailored-made linkage function for a specific algorithm. We show that the proposed approach works for five existing clustering algorithms without the need to modify their linkage functions or algorithms, except for the replacement of distance with Isolation Kernel.

Our empirical evaluation verifies that Isolation Kernel is a better measure than distance and two existing popular kernels, Gaussian Kernel and adaptive Gaussian Kernel, on five - AHC algorithms, i.e., T-AHC, HDBSCAN [16], GDL [7], PHA [17] and HC-OT [18].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The source code of IK-based AHC is provided at https://github.com/zhuye88/IK-AHC.

Acknowledgement

Kai Ming Ting is supported by National Natural Science Foundation of China (62076120).

Appendix A. Interpretation of HDBSCAN as an AHC algorithm

HDBSCAN [16] can be interpreted as a new kind of AHC algorithm that relies on a density-based linkage function, i.e., an

AHC with a single-linkage function and a particular dissimilarity measure. It uses the single-linkage function based on reachability-distance to merge two subclusters, motivated by the density-based clustering algorithm DBSCAN [29]. The reachability-distance is defined as

$$d_{kReach}(x,y) = \max\{dist(x,y), dist_k(x), dist_k(y)\}$$

where $dist_k(x)$ is the distance between x and x's $(k-1)$-th nearest neighbour.

The linkage function of HDBSCAN is defined as:

$$\check{h}(C_i,C_i) = \min_{x\in C_i, y\in C_j} d_{kReach}(x,y)$$

Similar to T-AHC, HDBSCAN gradually increases the reachability-distance as the height on the dendrogram to merge the two most similar subclusters (based on the above linkage function) iteratively. This can be interpreted as: the larger the reachability-distance, the larger number of points are linked.

Unlike T-AHC, HDBSCAN uses a dynamic programming method to set different thresholds on the dendrogram to extract optimal clusters with varied densities. Furthermore, clusters with a number of points less than a user-specified c will be ignored and treated as noise, after the dendrogram building process. Since HDBSCAN is a density-based clustering algorithm, it can detect arbitrarily shaped clusters and identify noise in the dataset.

To kernelise HDBSCAN, the kernel-based linkage function is defined as:

$$\check{\hbar}(C_i,C_j) = \max_{x\in C_i y\in C_j} \min\{K(x,y), K_k(x), K_k(y)\}$$

where $K_k(x)$ is the similarity between x and x's k-1-th most similar neighbour.

Appendix B. Kernelised PHA

The PHA [17] converts the distance between data points into potential values to measure the similarity between clusters. Suppose that there is a data set with n data points, denoted as $X = \{x_1, x_2, \dots x_n\}$. The distance between two data points x_i and x_j is denoted as $dist(x_i, x_j)$. The potential value of point x_i received from point x_j is calculated by

$$\Phi_{x_i,x_j} = \begin{cases} -\frac{1}{dist(x_i,x_j)} & \text{if } dist(x_i,x_j) \geq \lambda \\ -\frac{1}{\lambda} & \text{if } dist(x_i,x_j) < \lambda \end{cases}$$

where the parameter λ is used to avoid the singularity problem when $dist(x_i, x_j)$ is too small.

The total potential value of a data point x_a is defined as the sum of the potential values it has received from all the other data points

$$\Phi_{x_a} = \sum_{i=1, i\neq a}^{n} \Phi_{x_a,x_i}$$

The linkage function of PHA is defined as

$$h(C_1,C_2) = dist(s_1,s_2)$$

where s_1 and s_2 are from these two clusters respectively and be determined by:

$$\text{If } C_2 \leq C_1, \begin{cases} s_1 = \underset{k}{\text{argmin}}(\Phi_k | k \in C_1) \\ s_2 = \underset{k}{\text{argmin}}(dist(k,s_1) | (k \in C_2) \text{AND} (\Phi_k \leq \Phi_{s_1})) \end{cases}$$

$$\text{If } C_1 \leq C_2, \begin{cases} s_2 = \underset{k}{\text{argmin}}(\Phi_k | k \in C_2) \\ s_1 = \underset{k}{\text{argmin}}(dist(k,s_2) | (k \in C_1) \text{AND} (\Phi_k \leq \Phi_{s_2})) \end{cases}$$

where $C_i \leq C_j$ means $\exists x \in C_i \big(\forall y \in C_j \big(\Phi_x \leq \Phi_y\big)\big)$.

To kernelise the PHA, the potential value of point x_i received from point x_j is calculated by

$$\hat{\Phi}_{x_i,x_j} = \begin{cases} -\frac{1}{1-K(x_i,x_j)} & \text{if } K\big(x_i,x_j\big) \leq 1-\lambda \\ -\frac{1}{\lambda} & \text{if } K\big(x_i,x_j\big) > 1-\lambda \end{cases}$$

The total potential value of a data point x_a is defined as

$$\hat{\Phi}_{x_a} = \sum_{i=1, i\neq a}^{n} \hat{\Phi}_{x_a,x_i}$$

The kernel-based linkage function for PHA is defined as $\hbar(C_1,C_2) = 1 - K(s_1,s_2)$ where s_1 and s_2 are from these two clusters respectively and be determined by:

$$\text{If } C_2 \leq C_1, \begin{cases} s_1 = \underset{k}{\text{argmin}}\big(\hat{\Phi}_k | k \in C_1\big) \\ s_2 = \underset{k}{\text{argmax}}\big(K(k,s_1) | (k \in C_2) \text{AND} \big(\hat{\Phi}_k \leq \hat{\Phi}_{s_1}\big)\big) \end{cases}$$

$$\text{If } C_1 \leq C_2, \begin{cases} s_2 = \underset{k}{\text{argmin}}\big(\hat{\Phi}_k | k \in C_2\big) \\ s_1 = \underset{k}{\text{argmax}}\big(K(k,s_2) | (k \in C_1) \text{AND} \big(\hat{\Phi}_k \leq \hat{\Phi}_{s_2}\big)\big) \end{cases}$$

Appendix C. Kernelised GDL

The graph degree linkage (GDL) algorithm [7] begins with a number of initial small clusters, and iteratively merges two clusters with the maximum similarity. Suppose that there is a data set with n data points, denoted as $X = \{x_1, x_2, \dots x_n\}$. The similarities are computed using the product of the average indegree and average outdegree in a *KNN* graph, in which the vertices is X and the weights for edges are defined as:

$$w_{ij} = \begin{cases} \exp\left(-\frac{dist(x_i,x_j)^2}{\sigma^2}\right), & \text{if } x_j \in \mathcal{N}_i^K \\ 0, & \text{otherwise} \end{cases} \tag{C.1}$$

where $dist(x_i, x_j)$ is the distance between x_i and x_j, $\mathcal{N}_i^K$ is the set of K-nearest neighbours of x_i, and $\sigma^2 = \frac{a}{nK}[\sum_{i=1}^{n} \sum_{x_j \in \mathcal{N}_i^K} dist(x_i,x_j)^2]$. K and a are free parameters to be set.

Given a vertex i, the average indegree from and the average outdegree to a cluster C is defined as $\deg_i^-(\mathcal{C}) = \frac{1}{|\mathcal{C}|}\sum_{j\in\mathcal{C}} w_{ji}$ and $\deg_i^+(\mathcal{C}) = \frac{1}{|\mathcal{C}|}\sum_{j\in\mathcal{C}} w_{ij}$, respecitvely, where $|C|$ is the cardinality of C.

The similarity between two clusters is defined as the product of the average indegree and average outdegree.

$$\mathcal{A}_{\mathcal{C}_b,\mathcal{C}_a} = \sum_{i\in\mathcal{C}_b} \deg_i^-(\mathcal{C}_a) \deg_i^+(\mathcal{C}_a) + \sum_{i\in\mathcal{C}_a} \deg_i^-(\mathcal{C}_b) \deg_i^+(\mathcal{C}_b) \tag{C.2}$$

To kernelise the GDL, simply replace the distance with a kernel in building the K-NN graph. The weights of K-NN graph are defined as

$$\hat{w}_{ij} = \begin{cases} \exp\left(-\frac{(1-K(x_i,x_j))^2}{\sigma^2}\right), & \text{if } x_j \in \mathcal{N}_i^K \\ 0, & \text{otherwise} \end{cases} \tag{C.3}$$

where $\mathcal{N}_i^K$ is the set of K-nearest neighbours of x_i.

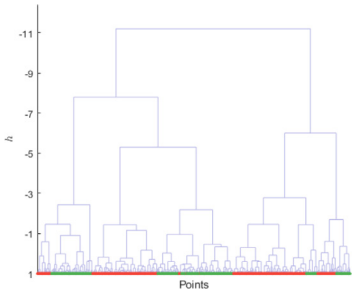
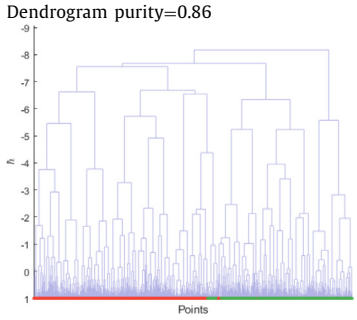
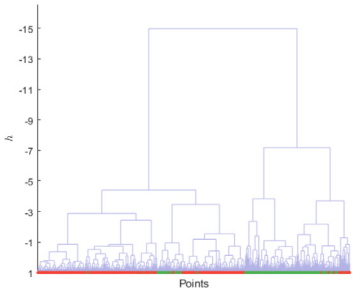
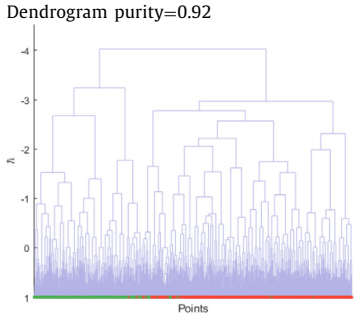
Appendix D. Kernelised HC-OT

The HC-OT [18] algorithm uses a Optimal Transport distance called Sinkhorn distance to take the data distributional aspects of the clusters into account when building the cluster tree. Suppose that there is a data set with n data points, denoted as $X = \{x_1, x_2, \dots x_n\}$. Let $C_i = \{x_1, x_2, \cdots, x_a\}$ and $C_j = \{y_1, y_2, \cdots, y_b\}$ be

Table E1
Clustering results of AHC with the Ward's linkage in *Dendrogram Purity* and $F1$ score on four datasets. The best result on each dataset is boldfaced.

Dataset	*Dendrogram Purity*				$F1$ score			
	Dis	G	AG	IK	Dis	G	AG	IK
ALLAML	0.73	0.72	0.71	**0.77**	0.72	0.72	0.70	**0.79**
COIL20	0.73	0.74	0.73	**0.80**	0.76	0.76	0.73	**0.77**
WDBC	0.83	0.88	0.87	**0.92**	0.86	0.91	0.9	**0.95**
banknote	0.69	0.76	0.69	**0.86**	0.66	0.77	0.61	**0.88**

Table E2
Comparison of dendrograms produced by T-AHC with the Ward's linkage functions using distance and Isolation Kernel on two datasets.

Dataset	Distance	Isolation Kernel
Banknote	Dendrogram purity=0.69 	Dendrogram purity=0.86 
WDBC	Dendrogram purity=0.83 	Dendrogram purity=0.92 

two clusters. Let $\mu_i = \sum_{i=1}^{a} \frac{1}{a}\delta_{x_i}$ and $\mu_j = \sum_{i=1}^{b} \frac{1}{b}\delta_{y_i}$ be the empirical distribution of samples in C_i and C_j respectively, where δ_z is the Dirac delta function at $z \in \mathbb{R}^d$. Then the Sinkhorn distance between two clusters denoted by $S_\lambda(C_i, C_j)$ is defined as the Sinkhorn distance between μ_i and μ_j. The detail about Sinkborn distance is shown in [18].

The linkage function of HC-OT is then defined as:

$$d\left(C_i, C_j\right) = \begin{cases} dist(x,y),\ x \in C_i,\ y \in C_j & \text{if } |C_i|{=}|C_j| = 1 \\ S_\lambda\left(C_i, C_j\right) & \text{if } |C_i|, |C_j| > \frac{n}{10} \\ \min\{d(C_i, C_q), d(C_i, C_p)\}, \text{ where } C_j = C_q \cup C_p & \text{Otherwise} \end{cases} \tag{D.1}$$

where n is the size of dataset.

To kernelise the HC-OT, simply replace the distance with a kernel method. The kernelised linkage function of HC-OT is defined as

$$\hat{h}(C_i, C_j) = \begin{cases} 1 - K(C_i, C_j) & \text{if } |C_i| = |C_j| = 1 \\ S_\lambda(C_i, C_j) & \text{if } |C_i|, |C_j| > \frac{n}{10} \\ \min\{\hat{h}(C_i, C_q), \hat{h}(C_i, C_p)\}, \text{ where } C_j = C_q \cup C_p & \text{Otherwise} \end{cases} \tag{D.2}$$

where n is the size of dataset.

Appendix E. Has the Ward's method addressed the bias of T-AHC?

The Ward's linkage [40] suggested a general AHC that chooses two subclusters at each step to merge based on the optimal value of an objective function. The minimum variance criterion was used as an illustrative example. However, this criterion does not explicitly address the bias of T-AHC mentioned in Section 4.

We evaluated the performance of the Ward's linkage (provided in MATLAB) on real-world datasets that contain clusters with varied densities, as shown in Table E.10 and Fig. E.11. The results confirm that the Ward's linkage also gets worse performance than kernel methods on those dataset. It is worth mentioning that Isolation Kernel still can be used to significantly improve its performance. Thus, the Ward's linkage is not a solution to address this bias.

References

[1] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, C. Mathieu, Hierarchical clustering: objective functions and algorithms, J. ACM 66 (4) (2019) 26.
[2] Y. Yang, Y. Tu, H. Lei, W. Long, HAMIL: hierarchical aggregation-based multi-instance learning for microscopy image classification, Pattern Recognit 136 (2023) 109245.
[3] N. Kaushik, M.K. Bhatia, Twitter sentiment analysis using k-means and hierarchical clustering on COVID pandemic, in: International Conference on Innovative Computing and Communications, Springer, 2022, pp. 757–769.
[4] X. Li, P. Wu, Stock price prediction incorporating market style clustering, Cognit Comput 14 (1) (2022) 149–166.

[5] B. King, Step-wise clustering procedures, J Am Stat Assoc 62 (317) (1967) 86–101.
[6] R. Sokal, C. Michener, A statistical method for evaluating systematic relationships, University of Kansas science bulletin (University of Kansas, 1958) (1958).
[7] W. Zhang, X. Wang, D. Zhao, X. Tang, Graph degree linkage: agglomerative clustering on a directed graph, in: European Conference on Computer Vision, Springer, 2012, pp. 428–441.
[8] W. Zhang, D. Zhao, X. Wang, Agglomerative clustering via maximum incremental path integral, Pattern Recognit 46 (11) (2013) 3056–3065.
[9] Y. Zhu, K.M. Ting, M.J. Carman, M. Angelova, CDF transform-and-shift: an effective way to deal with datasets of inhomogeneous cluster densities, Pattern Recognit 117 (2021) 107977.
[10] J.S. Klemelä, Smoothing of multivariate data: density estimation and visualization, volume 737, John Wiley & Sons, 2009.
[11] K.M. Ting, Y. Zhu, M. Carman, Y. Zhu, T. Washio, Z.-H. Zhou, Lowest probability mass neighbour algorithms: relaxing the metric constraint in distance-based neighbourhood algorithms, Mach Learn 108 (2) (2019) 331–376.
[12] X. Qin, K.M. Ting, Y. Zhu, V. Lee, Nearest-neighbour-induced isolation similarity and its impact on density-based clustering, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI Press, 2019.
[13] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Advances in neural information processing systems, 2005, pp. 1601–1608.
[14] K.M. Ting, Y. Zhu, Z.-H. Zhou, Isolation kernel and its effect on SVM, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2018, pp. 2329–2337.
[15] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput Surv 31 (3) (1999) 264–323.
[16] R.J.G.B. Campello, D. Moulavi, A. Zimek, J. Sander, Hierarchical density estimates for data clustering, visualization, and outlier detection, ACM Trans Knowl Discov Data 10 (1) (2015) 5:1–5:51. http://doi.acm.org/10.1145/2733381
[17] Y. Lu, Y. Wan, PHA: a fast potential-based hierarchical agglomerative clustering method, Pattern Recognit 46 (5) (2013) 1227–1239.
[18] S. Chakraborty, D. Paul, S. Das, Hierarchical clustering with optimal transport, Statistics & Probability Letters 163 (2020) 108781.
[19] N. Yadav, A. Kobren, N. Monath, A. Mccallum, Supervised hierarchical clustering with exponential linkage, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, volume 97, PMLR, Long Beach, California, USA, 2019, pp. 6973–6983.
[20] C.C. Aggarwal, C.K. Reddy, Data Clustering: Algorithms and Applications, Chapman and Hall/CRC Press, 2013.
[21] S. Shuming, Y. Guangwen, W. Dingxing, Z. Weimin, Potential-based hierarchical clustering, in: 2002 International Conference on Pattern Recognition, volume 4, 2002, pp. 272–275 vol.4.
[22] G. Karypis, E.-H.S. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer (Long Beach Calif) (8) (1999) 68–75.
[23] D. Zhao, X. Tang, Cyclizing clusters via zeta function of a graph, in: Advances in Neural Information Processing Systems, 2009, pp. 1953–1960.
[24] N. Monath, A. Kobren, A. Krishnamurthy, M.R. Glass, A. McCallum, Scalable hierarchical clustering with tree grafting, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1438–1448.
[25] X. Han, Y. Zhu, K.M. Ting, D.-C. Zhan, G. Li, Streaming hierarchical clustering based on point-set kernel, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, in: KDD '22, ACM, 2022, p. 525533.
[26] W.-B. Xie, Z. Liu, D. Das, B. Chen, J. Srivastava, Scalable clustering by aggregating representatives in hierarchical groups, Pattern Recognit 136 (2023) 109230.
[27] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput 10 (5) (1998) 1299–1319.
[28] Z. Kang, C. Peng, Q. Cheng, Z. Xu, Unified spectral clustering with optimal graph, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[29] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters 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, AAAI Press, 1996, pp. 226–231.
[30] B. Schölkopf, A.J. Smola, F. Bach, et al., Learning with kernels: support vector machines, regularization, optimization, and beyond, MIT press, 2002.
[31] M.L. van der, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.
[32] F. Aurenhammer, Voronoi diagrams—a survey of a fundamental geometric data structure, ACM Comput Surv 23 (3) (1991) 345–405.
[33] K.A. Heller, Z. Ghahramani, Bayesian hierarchical clustering, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 297–304.
[34] I. Borg, P.J.F. Groenen, P. Mair, Applied Multidimensional Scaling, Springer Science & Business Media, 2012.
[35] R. Szeliski, Computer vision: algorithms and applications, Springer Science & Business Media, 2010.
[36] L. McInnes, J. Healy, Accelerated hierarchical density based clustering, in: 2017 IEEE International Conference on Data Mining Workshops, IEEE, 2017, pp. 33–42.
[37] D. Dua, C. Graff, UCI machine learning repository, 2017,. http://archive.ics.uci.edu/ml.
[38] Y. Zhu, K.M. Ting, M.J. Carman, Density-ratio based clustering for discovering clusters with varying densities, Pattern Recognit 60 (2016) 983–997.
[39] J. Demšar, Statistical comparisons of classifiers over multiple data sets, The Journal of Machine Learning Research 7 (2006) 1–30.
[40] J.H. Ward Jr., Hierarchical grouping to optimize an objective function, J Am Stat Assoc 58 (301) (1963) 236–244.
[41] R.M. Aliguliyev, Performance evaluation of density-based clustering methods, Inf Sci (Ny) 179 (20) (2009) 3583–3602.
[42] H.W. Kuhn, The Hungarian method for the assignment problem, Naval Research Logistics Quarterly 2 (1–2) (1955) 83–97.
[43] C. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, Nat Lang Eng 16 (1) (2010) 100–103.
[44] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, Journal of Machine Learning Research 11 (Oct) (2010) 2837–2854.

Xin Han received a Master by research from the School of Computer Science, Xian ShiYou University, China. He is a Research Fellow at the University of Macau, China. His research interests include machine learning, data mining and cybersecurity.

Ye Zhu was awarded a Ph.D. with a Mollie Holman Medal for the best doctoral thesis of the year from Monash University in 2017. He is a Senior Lecturer in the School of Information Technology at Deakin University, Australia. His research works focus on clustering and anomaly detection.

Kai Ming Ting received his Ph.D. from the University of Sydney, Australia. He is a Professor in the School of Artificial Intelligence, Nanjing University, China. His current research interests are in the areas of mass estimation and mass-based approaches, ensemble approaches and data stream data mining.

Gang Li, Ph.D., is a Professor at the School of Information Technology, Deakin University, Australia. His research interests are data science, artificial intelligence, data privacy, and technology applications to tourism and hospitality.