Mining Arbitrary Shaped Clusters and Outputting a High Quality Dendrogram

Hao Huang[1], Song Wang[1], Shuangke Wu[1], Yunjun Gao[2], Wei Lu[3(✉)], Qinming He[2], and Shi Ying[1]

[1] State Key Laboratory of Software Engineering, Wuhan University, Wuhan, People's Republic of China
{haohuang,xavierwang,wsk9551,yingshi}@whu.edu.cn
[2] College of Computer Science, Zhejiang University, Hangzhou, People's Republic of China
{gaoyj,hqm}@zju.edu.cn
[3] Key Laboratory of Data Engineering and Knowledge Engineering, Renmin University of China, MOE, Beijing, People's Republic of China
uqwlu@ruc.edu.cn

Abstract. Hierarchical clustering (HC for short) outputs a dendrogram that offers more topological information than flat clustering (e.g., k-means). However, the existing HC algorithms focus on either the quality of the dendrogram or the ability of mining arbitrary shaped clusters. To address the above two aspects simultaneously, we present HICMEN by adopting (1) the classic agglomerative clustering framework that can generate a complete dendrogram, and (2) a novel similarity measure based on mutual k-nearest neighbors to capture the connectivity of data points and help properly merge up each arbitrary shaped cluster piece by piece. More importantly, we prove that the similarity measure has a nice property called weak monotonicity, which guarantees the quality of the dendrogram generated by HICMEN. Extensive experimental results show that HICMEN is capable of mining arbitrary shaped clusters effectively, and can simultaneously output a high quality dendrogram.

Keywords: Clustering · Arbitrary shaped clusters · Dendrogram

1 Introduction

Hierarchical clustering (abbreviated as HC henceforth) groups data points into a tree hierarchy of clusters, in which every cluster node contains children clusters while sibling nodes of clusters partition data points covered by their common parent according to a similarity measure. Figure 1 illustrates an example of such a process which organizes data points into a tree hierarchy called dendrogram.

A good dendrogram generated in HC should be able to preserve and present the intrinsic proximities of original data points, and offer valuable information in practice. For example, in the area of biology, by performing HC on the physical signs of living being, the dendrogram can be used to find the subspecies of each

S. Hartmann and H. Ma (Eds.): DEXA 2016, Part I, LNCS 9827, pp. 153–168, 2016.
DOI: 10.1007/978-3-319-44403-1_10

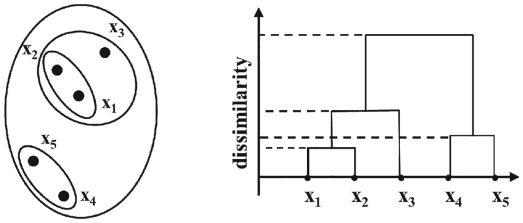

Fig. 1. An illustrative example of HC, and its generated dendrogram.

category, and reveal their taxonomic relations [5]. In the area of Internet, by adopting HC to categorize web pages, the dendrogram can be used to build a catalog of these web pages as a web directory, and facilitate the construction of web-directory-based browse systems [17].

Nonetheless, as a rule the existing HC algorithms only focus on either the quality of the dendrogram, such as traditional HC approaches using linkage metrics [7,23,25] which however are more applicable to compact and spherical clusters, or the ability of mining arbitrary shaped clusters, such as CHAMELEON algorithm [19] which however cannot output a complete dendrogram, let alone guarantee the quality of the dendrogram. In fact, the existing HC algorithms' efforts on either aspect usually sacrifice the performance of the other.

In this paper, we present HICMEN (**HI**erarchical **C**lustering with **M**utual k-n**E**arest **N**eighbors) an HC algorithm that takes both aspects into account. To the best of our knowledge, it is the first time that we explicitly identify and solve the problem of simultaneously mining arbitrary shaped clusters and outputting a high quality dendrogram. HICMEN uses the classic agglomerative clustering framework to generate a complete dendrogram. By adopting a novel similarity measure described by the MkNN (Mutual k-Nearest Neighbors) relationship across two sub-clusters, HICMEN prefers to merge up sub-cluster pairs with similar local densities and close proximities, and aggregates each arbitrary shaped cluster piece by piece. We prove that the proposed similarity measure has a nice property called weak monotonicity, which has the following two advantages, i.e., (1) it can better reflect the real cohesiveness between sub-clusters in an arbitrary shaped cluster and help HICMEN achieve an accurate clustering performance, and (2) with this similarity measure, the dendrogram generated by HICMEN can obtain a high quality, which is quantitated by a commonly used criterion for dendrogram quality called CPCC (Cophenetic Correlation Coefficient) [24].

The remaining sections are organized as follows. We review the related work in Sect. 2, introduce a MkNN-based similarity measure in Sect. 3, and present HICMEN algorithm in Sect. 4. We experimentally verify the effectiveness and efficiency of our approach in Sect. 5 before concluding the paper in Sect. 6.

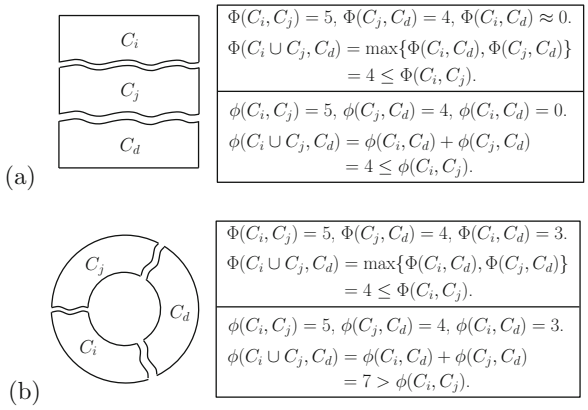

Fig. 2. Comparison of traditional similarity measure $\Phi(\cdot)$ (e.g., single link) and boundary similarity measure $\phi(\cdot)$. (a) Case 1: when C_d is only linked with C_j (i.e., $\phi(C_i, C_d) = 0$), both $\Phi(\cdot)$ and $\phi(\cdot)$ satisfy restrictive monotonicity property; (b) Case 2: when C_d is linked with both C_i and C_j, $\Phi(\cdot)$ still satisfies restrictive monotonicity property, while $\phi(\cdot)$ will be re-calculated according to the changed boundary regions after merging C_i and C_j and the result may not satisfy restrictive monotonicity property any more.

2 Related Work

The related work to the problem of simultaneously mining arbitrary shaped clusters and outputting a high quality dendrogram can be briefly categorized into two groups, namely (1) the dendrogram centered HC algorithms, and (2) the arbitrary shaped clustering algorithms.

2.1 Dendrogram Centered HC Algorithms

Dendrogram centered HC algorithms focus on the completeness and quality of the dendrograms they generate. To this end, they perform the classic agglomerative clustering framework (i.e., beginning from individual data points and recursively merging two most similar sub-clusters) to output a complete dendrogram, and adopt a similarity measure $\Phi(\cdot)$ that satisfies a *restrictive monotonicity property*, i.e., $\Phi(C_i \cup C_j, C_d) \leqslant \Phi(C_i, C_j)$, where C_i and C_j are the merged sub-clusters and C_d is any other disjoint sub-cluster ($\forall\, p \in \{i, j\}$, $C_d \cap C_p = \varnothing$). Linkage metrics (such as single link [23], complete link [7] and average link [25]) exemplify this kind of similarity measures. Dendrograms generated by this restrictive monotonic manner were claimed to be true and reflect real cohesiveness of sub-clusters, and have high quality.

The above claim holds true when sub-cluster C_d is linked with either C_i or C_j, such as the situation illustrated in Fig. 2(a). However, in an arbitrary shaped cluster, sub-clusters C_d, C_i and C_j may all be linked with each other, such as the example illustrated in Fig. 2(b). In this situation, the aforementioned claim

may not hold. The reason is twofold. (1) Firstly, in an arbitrary shaped cluster, the adjacent sub-clusters are only connected by a small part, i.e., their contact boundaries, while their majority parts are often not so cohesive to each other due to the arbitrary shape. Hence, compared against using a global or average similarity measure, a reasonable estimation on the boundary cohesiveness of the contact boundaries for adjacent sub-clusters helps better reflect the true probability that these sub-clusters are from a same arbitrary shaped cluster structure [18]; (2) on this basis, after merging up adjacent sub-clusters C_i and C_j, if the new sub-cluster $C_i \cup C_j$ has larger contact boundaries to sub-cluster C_d, the boundary cohesiveness between $C_i \cup C_j$ and C_d would increase. Sometimes it may even exceed the boundary cohesiveness between C_i and C_j, i.e., $\phi(C_i \cup C_j, C_d) > \phi(C_i, C_j)$, where $\phi(\cdot)$ refers to a boundary similarity measure that can reflect the boundary cohesiveness between a sub-cluster pair, resulting in that the restrictive monotonicity property is not valid in this case.

In brief, although similarity measures with restrictive monotonicity property were claimed to be helpful in preserving the quality of the dendrogram, they may fail to reflect real cohesiveness of sub-clusters in arbitrary shaped clusters. By contrast, a similarity measure that can dynamically update boundary cohesiveness according to changed contact boundaries helps to better reflect the real cohesiveness between sub-clusters in an arbitrary shaped cluster.

2.2 Arbitrary Shaped Clustering Algorithms

Many HC algorithms have been proposed to identify arbitrary shaped clusters from large data sets. CHAMELEON [19], OPTICS family [1], and CURE [14] are such pioneering examples. CHAMELEON builds a kNN (k-Nearest Neighbors) graph on a given data set, and partitions the graph to a predefined number of sub-graphs, followed by merging up closely linked sub-graph pairs. OPTICS family outputs a cluster ordering which is a linear list of all data points and reflects their density-based clustering structures. CURE first shrinks the data set size by sampling and conducts an agglomerative clustering on the sampled data points. Compared against CHAMELEON, ROCK and OPTICS family, CURE can be much faster since it takes only a few sampled data points for similarity computation. Nonetheless, the clustering results of CURE are sensitive to sampling quality. To mitigate this sensitivities, SPARCL [3] and CLASP [18], two evolutions of CURE, decompose a data set into small local groups via k-means, take group centers as representative data examples, and merge up the representative data examples; some other solutions like ABACUS [4] find the representative data examples by making data points to iteratively glob sufficiently close neighbors from their kNN. Although the above HC algorithms show good capability on mining arbitrary shaped clusters, they can only output an incomplete dendrogram since their HC processes start from sub-graphs or representative data examples rather than the original data set. Moreover, these HC algorithms do not take the dendrogram monotonicity into account, neither the restrictive monotonicity or a weak monotonicity, and lack guarantees for the dendrogram quality.

Besides the above HC algorithms, many non-HC algorithms can also identify arbitrary shaped clusters. For example, spectral clustering [6,20,26] embeds the arbitrary shaped clusters into a low-dimensional space to make cluster structures more distinguishable for clustering; the graph-based approaches [8,16,21] formulate the problem of clustering as a graph partition task; the density-based approaches [9,22,27] detect a cluster by searching a set of density-connected data points. However, non-HC methods do not output any dendrogram, and are inapplicable for the scenario where users require a dendrogram after clustering.

Similar to our proposed HICMEN algorithm, some existing HC and non-HC algorithms also adopt MkNN-based similarity measures [8,13,15,18] to estimate boundary cohesiveness between sub-clusters. Nevertheless, to the best of our knowledge, few of these approaches provide any dendrogram quality guarantee.

In summary, although some existing clustering algorithms show good capability on mining arbitrary shaped clusters, they are not competent to clustering tasks in which a complete and high quality dendrogram is required by users.

3 MkNN-Based Similarity Measure

Before introducing the detailed steps of HICMEN algorithm, in this section, we first present a novel MkNN-based similarity measure for arbitrary shaped clustering, followed by a theoretical analysis on its dendrogram quality guarantee.

3.1 Similarity Measure Definition

Data points have different number of MkNN (see Definition 1). This is because even if a data point x is one of the kNN of another data point y, x may not find y as its kNN when there are enough alternatives around x (i.e., local density around x is high enough). Thus, MkNN relationship tends to appear between sub-cluster pairs that are closely connected (i.e., they have close contact boundaries with similar local densities). This property is of practical significance to reveal the boundary cohesiveness of adjacent sub-clusters, especially those located in an arbitrary shaped cluster structure and only linked by their contact boundaries. Given this property, we introduce an MkNN-based similarity measure (see Definition 2) for arbitrary shaped clustering.

Definition 1. *Given a data set D and a positive integer k, the mutual k-nearest neighbors of a data point $x \in D$, denoted by $MkNN(x)$, is defined as $MkNN(x) = \{y \in D \mid x \in kNN(y) \wedge y \in kNN(x)\}$, where $kNN(x)$ denotes the k-nearest neighbors of data point x.*

Definition 2. *Given disjoint sub-clusters C_i and C_j, let S_{ij} be the set of data points that participate in the MkNN relationship across C_i and C_j, i.e., $S_{ij} = \{x \cup y \mid x \in C_i, y \in C_j, x \in MkNN(y)\}$. Then, the similarity between C_i and C_j, denoted by $\phi(\cdot)$, is defined as*

$$\phi(C_i, C_j) = \max\left\{\frac{|S_{ij} \cap C_i|}{|C_i|}, \frac{|S_{ij} \cap C_j|}{|C_j|}\right\}.$$

This similarity measure $\phi(\cdot)$ is symmetric (i.e., $\phi(C_i, C_j) = \phi(C_j, C_i)$). It refers to the maximum ratio of connecting points (i.e., data points that have MkNN relationship across a given sub-cluster pair) to their host sub-cluster. A high value of $\phi(C_i, C_j)$ indicates that sub-clusters C_i and C_j are tightly contacted with each other by their contact boundaries, and it is very likely that they are within a same arbitrary shaped cluster.

3.2 Guarantee for High Quality Dendrogram

MkNN-based similarity measures are commonly used in arbitrary shaped clustering [8,13,15,18] since they can often better reflect the real cohesiveness between adjacent sub-clusters in arbitrary shaped clusters, and help conduct a more accurate clustering work. Nonetheless, few of the existing MkNN-based similarity measures take the dendrogram quality into account when they are used in HC.

In contrast, our similarity measure $\phi(\cdot)$ is able to preserve the quality of the dendrogram generated by HC, i.e., as much as possible, it helps HC organize the dendrogram in a monotonic manner so that original data points with closer proximities would be merged as preferred. Note that the HC mentioned here performs the classic agglomerative clustering framework, i.e., starting from individual data points, it recursively merges two most similar sub-clusters until there is only one cluster left at the end. This progress can be described as follows. Given disjoint sub-clusters C_i, C_j, and C_d, if C_i and C_j are merged first, then the merging criterion ensures the inequality below.

$$\max\Big\{\phi(C_i, C_d), \phi(C_j, C_d)\Big\} \leqslant \phi(C_i, C_j). \tag{1}$$

With this inequality, we can prove the dendrogram quality guarantee of our similarity measure $\phi(\cdot)$ in the following two situations.

(1) When C_d is linked with either C_i or C_j (i.e., $\phi(C_i, C_d) > 0 \wedge \phi(C_j, C_d) = 0$, or $\phi(C_i, C_d) = 0 \wedge \phi(C_j, C_d) > 0$, such as the situation illustrated in Fig. 2(a)), similar to traditional similarity measures, our similarity measure $\phi(\cdot)$ also helps HC organize the dendrogram in a restrictively monotonic manner.

Theorem 1. *Given disjoint sub-clusters C_i, C_j, and C_d, if C_i and C_j are merged first and relationships $\phi(C_i, C_d) = 0$ and $\phi(C_j, C_d) > 0$ hold, then the proposed similarity measure $\phi(\cdot)$ satisfies restrictive monotonicity property, i.e.,*

$$\phi(C_i \cup C_j, C_d) \leqslant \phi(C_i, C_j).$$

Proof. As $\phi(C_i, C_d) = \max\left\{\frac{|S_{id} \cap C_i|}{|C_i|}, \frac{|S_{id} \cap C_d|}{|C_d|}\right\} = 0$, we have $S_{id} \cap C_i = S_{id} \cap C_d = \varnothing$, $(S_{id} \cup S_{jd}) \cap C_d = S_{jd} \cap C_d$. Based on the definition of $\phi(\cdot)$,

$$\begin{aligned}\phi(C_i \cup C_j, C_d) &= \max\left\{\frac{|S_{id} \cap C_i| + |S_{jd} \cap C_j|}{|C_i| + |C_j|}, \frac{|(S_{id} \cup S_{jd}) \cap C_d|}{|C_d|}\right\} \\ &= \max\left\{\frac{|S_{jd} \cap C_j|}{|C_i| + |C_j|}, \frac{|S_{jd} \cap C_d|}{|C_d|}\right\}.\end{aligned}$$

As inequality $\frac{|S_{jd}\cap C_j|}{|C_i|+|C_j|} \leqslant \frac{|S_{jd}\cap C_j|}{|C_j|}$ holds naturally, we have

$$\phi(C_i \cup C_j, C_d) \leqslant \max\left\{\frac{|S_{jd}\cap C_j|}{|C_j|}, \frac{|S_{jd}\cap C_d|}{|C_d|}\right\} = \phi(C_j, C_d).$$

Combining with Inequality (1), we have $\phi(C_i \cup C_j, C_d) \leqslant \phi(C_i, C_j)$, and the proof completes. ■

Therefore, when C_d is linked with either C_i or C_j, $\phi(\cdot)$ also has the restrictive monotonicity property to ensure the monotonicity of dendrogram, and thus helps the dendrogram to preserve the intrinsic proximities of original data points.

(2) When C_d has contact boundaries with both C_i and C_j (i.e., $\phi(C_i, C_d) > 0 \wedge \phi(C_j, C_d) > 0$, such as the situation illustrated in Fig. 2(b)), as mentioned in Sect. 2.1, similarity measures with restrictive monotonicity property may fail to reflect the real cohesiveness between the sub-clusters in this situation. Our similarity measure $\phi(\cdot)$ is not restricted by the restrictive monotonicity property in this situation. Instead, it re-evaluates the similarity $\phi(C_i \cup C_j, C_d)$ between C_d and the newly merged sub-cluster $C_i \cup C_j$ based on the changed contact boundaries. But this re-evaluated similarity $\phi(C_i \cup C_j, C_d)$ is bounded. It will not be significantly greater than the similarity $\phi(C_i, C_j)$ between the previously merged sub-clusters C_i and C_j, and prevent a large distortion for the monotonicity between two successive levels in dendrogram. We refer to this property as *weak monotonicity*, which can be proven by the following lemma and theorem.

Lemma 1. *Given disjoint sub-clusters C_i, C_j, and C_d, if C_i and C_j are merged first and relationships $\phi(C_i, C_d) > 0$ and $\phi(C_j, C_d) > 0$ hold, then the proposed similarity measure $\phi(\cdot)$ satisfies the following relationship, i.e.,*

$$\phi(C_i \cup C_j, C_d) \leqslant \max\left\{\phi(C_i, C_d), \phi(C_j, C_d), \frac{|S_{id}\cap C_d| + |S_{jd}\cap C_d|}{|C_d|}\right\}.$$

Proof. Without loss of generality, assuming $\frac{|S_{id}\cap C_i|}{|C_i|} \leqslant \frac{|S_{jd}\cap C_j|}{|C_j|}$, then we have

$$\frac{|S_{id}\cap C_i|}{|C_i|} \leqslant \frac{|S_{id}\cap C_i| + |S_{jd}\cap C_j|}{|C_i| + |C_j|} \leqslant \frac{|S_{jd}\cap C_j|}{|C_j|}.$$

On the other hand, the following inequality holds naturally.

$$\frac{|S_{id}\cap C_d|}{|C_d|}, \frac{|S_{jd}\cap C_d|}{|C_d|} \leqslant \frac{|(S_{id}\cup S_{jd})\cap C_d|}{|C_d|} \leqslant \frac{|S_{id}\cap C_d| + |S_{jd}\cap C_d|}{|C_d|}.$$

Combining the above inequalities, we have

$$\max\left\{\frac{|S_{id}\cap C_i|}{|C_i|}, \frac{|S_{id}\cap C_d|}{|C_d|}\right\} \leqslant \max\left\{\frac{|S_{id}\cap C_i| + |S_{jd}\cap C_j|}{|C_i| + |C_j|}, \frac{|(S_{id}\cup S_{jd})\cap C_d|}{|C_d|}\right\}$$
$$\leqslant \max\left\{\frac{|S_{jd}\cap C_j|}{|C_j|}, \frac{|S_{jd}\cap C_d|}{|C_d|}, \frac{|S_{id}\cap C_d| + |S_{jd}\cap C_d|}{|C_d|}\right\}.$$

By Definition 2, the above inequality can be re-written as

$$\phi(C_i, C_d) \leqslant \phi(C_i \cup C_j, C_d) \leqslant \max\left\{\phi(C_j, C_d), \frac{|S_{id} \cap C_d| + |S_{jd} \cap C_d|}{|C_d|}\right\}.$$

By moving the leftmost item into the rightmost item, we get

$$\phi(C_i \cup C_j, C_d) \leqslant \max\left\{\phi(C_i, C_d), \phi(C_j, C_d), \frac{|S_{id} \cap C_d| + |S_{jd} \cap C_d|}{|C_d|}\right\},$$

and the proof completes. ■

With Lemma 1, we have the theorem below.

Theorem 2. *Given disjoint sub-clusters C_i, C_j, and C_d, if C_i and C_j are merged first and relationships $\phi(C_i, C_d) > 0$ and $\phi(C_j, C_d) > 0$ hold, then the proposed similarity measure $\phi(\cdot)$ satisfies the following relationship, i.e.,*

$$\phi(C_i \cup C_j, C_d) \leqslant 2 \cdot \phi(C_i, C_j).$$

Proof. The following three inequalities holds naturally.

$$\frac{|S_{id} \cap C_d| + |S_{jd} \cap C_d|}{|C_d|} \leqslant 2 \cdot \max\left\{\frac{|S_{id} \cap C_d|}{|C_d|}, \frac{|S_{jd} \cap C_d|}{|C_d|}\right\},$$

$$\frac{|S_{id} \cap C_d|}{|C_d|} \leqslant \max\left\{\frac{|S_{id} \cap C_i|}{|C_i|}, \frac{|S_{id} \cap C_d|}{|C_d|}\right\} = \phi(C_i, C_d),$$

$$\frac{|S_{jd} \cap C_d|}{|C_d|} \leqslant \max\left\{\frac{|S_{jd} \cap C_j|}{|C_j|}, \frac{|S_{jd} \cap C_d|}{|C_d|}\right\} = \phi(C_j, C_d).$$

Combining the above three inequalities, we have

$$\frac{|S_{id} \cap C_d| + |S_{jd} \cap C_d|}{|C_d|} \leqslant 2 \cdot \max\Big\{\phi(C_i, C_d), \phi(C_j, C_d)\Big\}.$$

Based on Inequality (1), we get

$$\max\left\{\phi(C_i, C_d), \phi(C_j, C_d), \frac{|S_{id} \cap C_d| + |S_{jd} \cap C_d|}{|C_d|}\right\} \leqslant 2 \cdot \phi(C_i, C_j).$$

Combining Lemma 1, we have $\phi(C_i \cup C_j, C_d) \leqslant 2 \cdot \phi(C_i, C_j)$, and the proof completes. ■

Theorem 2 shows that with $\phi(\cdot)$, when HC merges a sub-cluster C (e.g., $C_i \cup C_j$) with any other disjoint sub-cluster (e.g., C_d), the similarity between them will not exceed twice of the similarity between the parent sub-cluster pair (e.g., C_i and C_j) of C. In other words, this theorem provides the theoretical upper bound for the distortion of two successive levels in the dendrogram generated by HC, and roughly guarantees the dendrogram's quality (i.e., its monotonicity).

Algorithm 1. HICMEN Algorithm

Input : data set $D = \{x_1, x_2, \ldots, x_N\}$; parameter k.
Output: dendrogram root R.

1 $\Gamma = \varnothing$; $\Lambda = \varnothing$; //set Γ of clusters and set Λ of outliers
2 $[\Gamma, \Lambda]$ = remove_outliers(D);
3 **while** $|\Gamma| > 1$ and $\phi(C_K, C_L) > 0$ **do**
4 $\quad [C_K, C_L] = \arg\max\{\phi(C_i, C_j)\}$ where (C_i *and* $C_j \in \Gamma$);
5 $\quad C_M = C_K \cup C_L$; children$(C_M) = \{C_K, C_L\}$;
6 $\quad \Gamma = \Gamma \cup \{C_M\} \setminus \{C_K\} \setminus \{C_L\}$;
7 **for** each $x_i \in \Lambda (1 \leqslant i \leqslant |\Lambda|)$ **do**
8 $\quad x_{nn} = \arg\min_x\{dist(x_i, x) \mid x \in D \setminus \Lambda\}$;
9 $\quad C_{nn} = C_{nn} \cup \{x_i\}$ where $x_n \in C_{nn}$ and $C_{nn} \in \Gamma$;
10 R = average_link(Γ); //run average link with Euclidean distance

4 HICMEN Algorithm

4.1 Algorithm Description

With the similarity measure $\phi(\cdot)$, we propose HICMEN algorithm to mine arbitrary shaped clusters and output a high quality dendrogram. Before presenting the detailed steps, we would like to clarify that there may be no MkNN relationship across natural clusters in a data set. In this situation, $\phi(\cdot)$ will regard the similarities between the natural clusters as zeros, and hinder HC from carrying out a complete agglomerative clustering. Hence, when the maximal similarity evaluated by $\phi(\cdot)$ is zero and the work of agglomerative clustering has not been finished (i.e., there is still more than one cluster), we adopt average link algorithm with Euclidean distance to complete the HC process.

The pseudo-code of HICMEN is presented in Algorithm 1. It takes as inputs a data set D containing N data points, and a parameter k for calculating $\phi(\cdot)$.

HICMEN is carried out by three phases, namely the initialization phase (lines 1–2), the merging phase (lines 3–6), and the ending phase (lines 7–10). (1) In the initialization phase, to prevent outliers from affecting the identification of real clustering structures, HICMEN first removes outliers Λ from data set D via an efficient outlier detection algorithm proposed by Bay and Schwabacher [2]. The rest of data points are classified into set Γ for the succeeding merge process (line 4). (2) In the merging phase, starting from individual data points in Γ, HICMEN recursively merges the sub-clusters by using the proposed similarity measure $\phi(\cdot)$. In each iteration, it searches and merges two most similar sub-clusters (line 4) and labels the parent-child relationship in dendrogram (line 5). The merging process does not stop until every data point is in a single cluster or the maximal similarity is zero. (3) In the ending phase, HICMEN firstly assigns each outlier to its nearest cluster, and adopts average link algorithm with Euclidean distance to merge the rest of sub-clusters (if any) before returning the root of dendrogram.

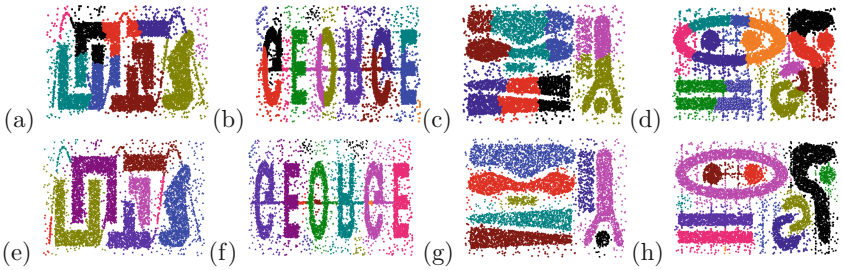

Fig. 3. The clustering results of (a)–(d) average link and (e)–(h) HICMEN (with $k = 22$) on $D1.1$–$D1.4$ (in which # of final clusters are 9, 15, 8, and 12, respectively).

4.2 Complexity Analysis

In the initialization phase, after kNN search which takes $O(dN^{2-1/d} + N \log N + kN)$ time by building a k-d tree [11], the outlier detection algorithm proposed by Bay and Schwabacher [2] can achieve a linear time complexity.

In the merging phase, MkNN relationship can be found by searching the kNN list with $O(kN)$ time. At each iteration, we update the merged sub-clusters' hash tables with $O(k)$ time since the average number of MkNN-connected sub-clusters is $O(k)$. We also update the hash tables of other sub-clusters that contain the merged sub-clusters. Given ν such sub-clusters, it takes $O(\nu k)$ time to update the hash tables, and $O(\nu \log N)$ time to find the next two most similar sub-clusters to be merged with the help of a maximum heap. In summary, the merging phase takes about $O(\nu N \log N + \nu k N)$ time since there are at most $(N-1)$ iterations.

In the ending phase, suppose that the number of outliers is ω. It takes $O(\omega N)$ time to search their nearest non-outlier neighbors. Supposing that there still exist κ sub-clusters after merging phase, it takes $O(\kappa N)$ time to merge them by average link algorithm. In summary, the ending phase takes $O(\omega N + \kappa N)$ time.

According to extensive experimental results, ν, ω and κ are far less than N. Hence, the overall time complexity of HICMEN is about $O(dN^{2-1/d} + N \log N)$.

5 Experimental Evaluation

In this section, we first verify the effectiveness of HICMEN by comparing it with (1) single link, complete link, and average link algorithms, which are the most famous dendrogram centered HC algorithms, (2) CHAMELEON algorithm, which is the paradigm of arbitrary shaped clustering centered HC algorithms, (3) ABACUS algorithm, which is a state-of-the-art evolution of CURE, and (4) DBSCAN, which is a high-performance representative of non-HC algorithms. We then conduct an efficiency study for the algorithms followed by a discussion on the impact of parameter k to HICMEN's clustering performance and execution time. Note that as the parameters of CHAMELEON and DBSCAN often affect the clustering performance of these two algorithms, we give them a privilege, i.e.,

Table 1. Description of data sets $D2.1$–$D2.6$

Data set	Name (domain application)	N	d	c
$D2.1$	Iris	150	4	3
$D2.2$	Breast Cancer Wisconsin	683	9	2
$D2.3$	Vehicle Silhouettes	846	18	4
$D2.4$	Image Segmentation	2,100	16	7
$D2.5$	Landsat Satellite	6,435	36	6
$D2.6$	Letter Recognition	20,000	16	26

Table 2. NMI scores of HC algorithms on $D2.1$–$D2.6$

Algorithm	$D2.1$	$D2.2$	$D2.3$	$D2.4$	$D2.5$	$D2.6$
Single Link	0.72	0.01	0.01	0.35	0.62	0.40
Complete Link	0.72	0.64	0.18	0.50	0.48	0.39
Average Link	0.81	0.68	0.17	0.49	0.64	0.40
CHAMELEON	0.70	0.77	0.12	0.59	0.61	0.31
ABACUS	0.79	0.70	0.16	0.56	0.61	0.40
DBSCAN	0.73	0.74	0.15	0.52	0.58	0.29
HICMEN	**0.82**	**0.84**	**0.21**	**0.68**	**0.69**	**0.43**

we vary their parameters at each execution and report their best performance from 20 times of run. All tested algorithms are implemented in C++, running on a desktop PC with 8GB RAM and Intel Core i7-2600 CPU at 3.40 GHz.

5.1 Effectiveness Evaluation

We first evaluate the effectiveness of HICMEN from the following two aspects, namely (1) clustering performance, and (2) dendrogram quality.

(1) Clustering Performance Study. In this experiment, we first demonstrate HICMEN's ability of mining arbitrary shaped clusters, and compare HICMEN with the other tested algorithms in terms of clustering accuracy.

(a) Effectiveness of Mining Arbitrary Shaped Clusters: We run HICMEN on four commonly used 2D data sets $D1.1$–$D1.4$ (8,000 points in $D1.1$–$D1.3$ and 10,000 points in $D1.4$), which were also used to evaluate CHAMELEON, ABACUS and DBSCAN algorithms. As shown in [4,19], CHAMELEON and ABACUS have good performance on these data sets, while DBSCAN's performance on them often has minor flaws.

Figure 3 illustrates the clustering results of HICMEN and average link algorithm, from which we can observe that HICMEN can effectively identify arbitrary shaped clusters, while average link fails. Single link and complete link algorithms have similar failures to average link.

Table 3. Description of data sets $D3.1$–$D3.8$

Data set	N	d	c	Data set	N	d	c
$D3.1$	$3,000$	2	20	$D3.5$	$2,701$	4	10
$D3.2$	$5,250$	2	35	$D3.6$	$4,051$	6	10
$D3.3$	$7,500$	2	50	$D3.7$	$5,401$	8	10
$D3.4$	$2,026$	3	10	$D3.8$	$6,751$	10	10

Table 4. CPCC scores of HC algorithms

Algorithm	$D3.1$	$D3.2$	$D3.3$	$D3.4$	$D3.5$	$D3.6$	$D3.7$	$D3.8$
Single Link	0.61	0.58	0.52	0.79	0.86	0.91	0.92	0.88
Complete Link	0.72	**0.71**	0.67	0.79	0.90	0.84	0.92	0.87
Average Link	0.74	0.70	0.66	0.82	**0.92**	**0.92**	**0.94**	0.91
HICMEN	**0.75**	**0.71**	**0.69**	**0.84**	**0.92**	**0.92**	**0.94**	**0.92**

(b) Accuracy Comparison: We compare HICMEN (with parameter $k = 22$) with the other tested approaches on six real data sets $D2.1$–$D2.6$ from UCI machine learning repository [10]. The data set properties are described in Table 1, in which N, d, and c indicate the number of points, data set dimensions, and the number of real clusters. The accuracy performance of each algorithm is measured by NMI (Normalized Mutual Information). Each NMI score falls in the range $[0, 1]$. A greater NMI score indicates a more accurate clustering result.

Table 2 lists the NMI scores of clustering results of each algorithm, from which we can observe that our HICMEN algorithm outperforms the other tested HC algorithms in accuracy performance.

(2) Dendrogram Quality. In this experiment, we compare the quality of the dendrogram generated by each algorithm on data sets $D3.1$–$D3.8$ [12], of which the properties are summarized in Table 3. Data sets $D3.1$–$D3.3$ have the same dimension with increased numbers of clusters, while data sets $D3.4$–$D3.8$ have the same cluster number with increased dimensions. We adopt CPCC (Cophenetic Correlation Coefficient) as dendrogram quality criterion, which describes how faithfully a dendrogram preserves the intrinsic proximities between the original data points. The definition of CPCC is as follows.

$$CPCC = \frac{\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d(i,j)c(i,j) - \mu_P\mu_C}{\sqrt{\left(\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d^2(i,j) - \mu_P^2\right)\left(\frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c^2(i,j) - \mu_C^2\right)}}$$

where $d(i, j)$ and $c(i, j)$ are the ordinary Euclidean distance and dendrogrammatic distance between points i and j respectively, N is the number of points, $M = \frac{1}{2}N(N-1)$, and μ_P and μ_C are defined as

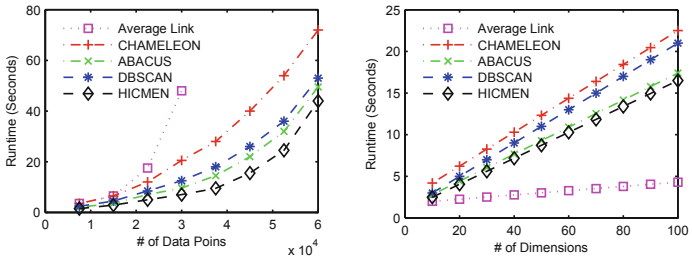

Fig. 4. Efficiency performance (runtime) of HC algorithms on data sets with different number of data points and different number of dimensions.

$$\mu_P = \frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d(i,j), \quad \mu_C = \frac{1}{M}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c(i,j).$$

A greater CPCC score indicates a better dendrogram quality.

Table 4 presents the CPCC score of the dendrogram generated by HICMEN (with parameter $k = 22$) and HC using linkage metrics. Note that CHAMELEON, ABACUS and DBSCAN have no CPCC score since they cannot output a full dendrogram. As shown in the table, HICMEN has almost the greatest CPCC scores among all HC algorithms, indicating that it can most faithfully preserve the intrinsic proximities between the original data points, better than traditional similarity measures with restrictive monotonicity property.

5.2 Efficiency Evaluation

In this experiment, we evaluate the efficiency of the tested algorithms with various data set sizes and dimensions. To investigate their scalability to the number of data points, we generate $D4.1$–$D4.8$ based on $D3.3$ by creating new data points nearby the original data points from the original number $7,500$ to the number of $60,000$ (with interval $7,500$). To investigate their scalability to the dimensionality of data sets, we generate $D5.1$–$D5.8$ based on $D3.3$ by creating new dimensions from 10 to 100 (with interval 10).

Figure 4 shows the execution time of the tested algorithms on $D4.1$–$D4.8$ and $D5.1$–$D5.8$ respectively (The parameter k of HICMEN is set to 22). We skip to plot the runtime curves of single link and complete link algorithms since they are very similar to that of average link algorithm.

From the figure, we can have the following observations. (1) With the growth of data set size, the execution time of our HICMEM algorithm increases slower than that of the other tested algorithms, indicating that our HICMEM algorithm shows better scalability to large data sets; (2) with the growth of dimensions, the execution time of average link algorithm increases very slowly, whereas the execution time of HICMEN, CHAMELEON, ABACUS and DBSCAN increase linearly with regard to the dimensions. Nonetheless, our HICMEN algorithm still keeps faster than CHAMELEON, ABACUS and DBSCAN.

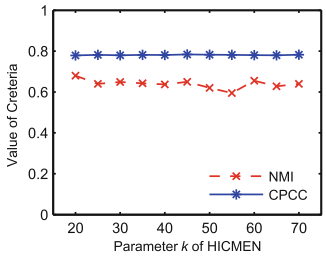

Fig. 5. Impact of parameter k to HICMEN's clustering result accuracy (in terms of NMI scores) and dendrogram quality (in terms of CPCC scores).

5.3 Impact of Parameter

HICMEN has only one parameter k for calculating the MkNN-based similarity measure $\phi(\cdot)$. In this experiment, we evaluate the impact of this parameter k to the effectiveness of HICMEN. By varying the value of k, we report the NMI score of HICMEN's clustering result, and the CPCC score of its generated dendrogram.

Figure 5 depicts the corresponding results on $D2.4$ (similar observations can be obtained on the rest of UCI data sets used in this paper), from which we can observe that the clustering performance and dendrogram quality of our HICMEM algorithm keep relatively stable with various k values, indicating that the effectiveness of HICMEN is relatively insensitive to the value of parameter k.

6 Conclusion

In this paper, we have defined an MkNN-based similarity measure for HC, and proven its weak monotonicity which enables HC to accurately express arbitrary shaped data sets with little distortion on the dendrogram. Based on this similarity measure, we have proposed HICMEN a simple yet effective HC algorithm for accurately identifying arbitrary shaped clusters and with a complete and high quality dendrogram as the output. Experimental results on both real and synthetic data sets have verified the effectiveness and efficiency of our approach.

Acknowledgements. This work was supported in part by NSFC Grants (61502347, 61502504, 61522208, 61572376, 61472359, 61379033, 61373038, and 61364025), the Fundamental Research Funds for the Central Universities (2015XZZX005-07, 2015XZZX004-18, and 2042015kf0038), and the Research Funds for Introduced Talents of WHU.

References

1. Ankerst, M.: OPTICS: ordering points to identify the clustering structure. In: SIGMOD, pp. 49–60 (1999)
2. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: KDD, pp. 29–38 (2003)
3. Chaoji, V., Hasan, M.A., Salem, S., Zaki, M.J.: SPARCL: an efficient and effective shape-based clustering. Knowl. Inf. Syst. **21**(2), 201–229 (2009)
4. Chaoji, V., Li, G., Yildirim, H., Zaki, M.J.: ABACUS: mining arbitrary shaped clusters from large datasets based on backbone identification. In: SDM, pp. 295–306 (2011)
5. Chen, Y.-A., Tripathi, L.P., Dessailly, B.H., Nyström-Persson, J., Ahmad, S., Mizuguchi, K.: Integrated pathway clusters with coherent biological themes for target prioritisation. Plos One **9**(6), e99030 (2014)
6. Correa, C.D., Lindstrom, P.: Locally-scaled spectral clustering using empty region graphs. In: KDD, pp. 1330–1338 (2012)
7. Defays, D.: An efficient algorithm for a complete link method. Comput. J. **20**(4), 364–366 (1977)
8. Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: SDM, pp. 47–58 (2003)
9. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
10. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
11. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. **3**(3), 209–226 (1977)
12. SIPU Clustering datasets. http://cs.joensuu.fi/sipu/datasets/
13. Guha, S., Rastogi, R., Shim, K.: ROCK: a robust clustering algorithm for categorical attributes. In: ICDE, pp. 512–521 (1999)
14. Guha, S., Rastogi, R., Shim, K.: CURE: an efficient clustering algorithm for large databases. Inf. Syst. **26**(1), 35–58 (2001)
15. Houle, M.E.: The relevant-set correlation model for data clustering. In: SDM, pp. 775–786 (2008)
16. Hu, T., Liu, C., Tang, Y., Sun, J., Song, H., Sung, S.Y.: High-dimensional clustering: a clique-based hypergraph partitioning frameworks. Knowl. Inf. Syst. **39**(1), 61–88 (2014)
17. Huang, H., Gao, Y., Chen, L., Li, R., Chiew, K., He, Q.: Browse with a social web directory. In: SIGIR, pp. 865–868 (2013)
18. Huang, H., Gao, Y., Chiew, K., Chen, L., He, Q.: Towards effective and efficient mining of arbitrary shaped clusters. In: ICDE, pp. 28–39 (2014)
19. Karypis, G., Han, E.H., Kumar, V.: CHAMELEON: hierarchical clustering using dynamic modeling. IEEE Comput. **32**(8), 68–75 (1999)
20. Li, J., Xia, Y., Shan, Z., Liu, Y.: Scalable constrained spectral clustering. IEEE Trans. Knowl. Data Eng. **27**(2), 589–593 (2015)
21. Mok, P.K., Huang, H.Q., Kwok, Y.L., Au, J.S.: A robust adaptive clustering analysis method for automatic identification of clusters. Pattern Recogn. **45**(8), 3017–3033 (2012)
22. Alex, R., Alessandro, L.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
23. Sibson, R.: SLINK: an optimally efficient algorithm for the single-link cluster method. Comput. J. **16**(1), 30–34 (1973)

24. Sokal, R.R., Rohlf, F.J.: The comparison of dendrograms by objective methods. Taxon **11**(2), 33–40 (1962)
25. Voorhees, E.M.: Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Inf. Process. Manag. **22**(6), 465–476 (1985)
26. Yang, Y., Ma, Z., Yang, Y., Nie, F., Shen, H.T.: Multitask spectral clustering by exploring intertask correlation. IEEE Trans. Cybern. **45**(5), 1069–1080 (2015)
27. Kim, Y., Shim, K., Kim, M.-S., Lee, J.S.: DBCURE-MR: an efficient density-based clustering algorithm for large data using MapReduce. Inf. Syst. **42**, 15–35 (2014)