Pattern Recognition 74 (2018) 1–14

Contents lists available at ScienceDirect

Pattern Recognition

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

# A Novel clustering method based on hybrid K-nearest-neighbor graph

Yikun Qin[a], Zhu Liang Yu[a,*], Chang-Dong Wang[b], Zhenghui Gu[a], Yuanqing Li[a]

[a] The College of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, PR China
[b] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, PR China

ARTICLE INFO

*Article history:*
Received 24 November 2016
Revised 24 August 2017
Accepted 5 September 2017
Available online 6 September 2017

*Keywords:*
Graph clustering
Hybrid k-nearest-neighbor graph
Internal validity index
Nonlinear data set
Video clustering

ABSTRACT

Most of the existing clustering methods have difficulty in processing complex nonlinear data sets. To remedy this deficiency, in this paper, a novel data model termed Hybrid K-Nearest-Neighbor (HKNN) graph, which combines the advantages of mutual k-nearest-neighbor graph and k-nearest-neighbor graph, is proposed to represent the nonlinear data sets. Moreover, a Clustering method based on the HKNN graph (CHKNN) is proposed. The CHKNN first generates several tight and small subclusters, then merges these subclusters by exploiting the connectivity among them. In order to select the optimal parameters for CHKNN, we further propose an internal validity index termed K-Nearest-Neighbor Index (KNNI), which can also be used to evaluate the validity of nonlinear clustering results by varying a control parameter. Experimental results on synthetic and real-world data sets, as well as that on the video clustering, have demonstrated the significant improvement on performance over existing nonlinear clustering methods and internal validity indices.



## 1. Introduction

Clustering is considered as one of the most important problems in machine learning. It processes data with unknown distribution and is the foundation for further learning [1]. A large number of clustering methods have been proposed for nonlinear data sets, including kernel-based methods [2–4], graph-based methods [5–7], density-based methods [8–10], support vector-based methods [11,12], and multi-exemplar methods [13,14], etc.

The kernel-based methods [2–4] use a nonlinear kernel mapping $\phi(.)$ to map the nonlinear data sets from the original space to a kernel space, in which the original data points can be linearly separated. The graph-based methods [6,7] first construct a graph whose vertices represent the data points and edges represent the similarity between each two points, then cut the graph into several parts [6], or utilize multi-prototype competitive learning to refine the coarse clusters which have been initialized by another graph clustering method [7]. Instead of using common similarity among data points, the Shared Nearest Neighbors (SNN) clustering method [5] uses similarity that is redefined as the number of near neighbors that the two points share to construct a graph. The density-based methods [8–10] utilize a density-based notion which considers the high-density regions as cluster centers. The support vector-based methods [11,12] first map the data sets to a compact kernel space as the kernel-based methods do, then find a hyper sphere which can surround most of the data points in this kernel space, finally use the inverse of the kernel function to map the spherical surface of this hyper sphere to the original space for categorizing the data sets. As one of the multi-exemplar methods, the Multi-Prototype Clustering (MPC) [13] first generates several center points and then uses a partition metric to decide which of these center points should be merged. In Multi-Exemplar Affinity Propagation (MEAP) [14], each data point is assigned to the most appropriate exemplar and each exemplar is assigned to the most appropriate super-exemplar.

Although these methods provide various methodologies and ideas for clustering nonlinear data sets, most of them have difficulty in processing large or complex nonlinear data sets. For kernel-based methods and support vector-based methods, the optimal nonlinear kernel mapping $\phi(.)$ is always unknown and difficult to be determined in practice. Moreover, most of these methods are computationally expensive. The well applied density-based methods cannot process the data sets consisting of complex nonconvex clusters, either. For instance, on the concentric ring data set, since the density of points in various parts of a ring may be similar, it is difficult to find the cluster centers that are defined as high-density regions. Being similar to density-based methods, the multi-exemplar methods tend to find the exemplars and

* Corresponding author.
*E-mail addresses:* yk_qin@foxmail.com (Y. Qin), zlyu@scut.edu.cn, zhuliang.yu@gmail.com (Z.L. Yu), changdongwang@hotmail.com (C.-D. Wang), zhgu@scut.edu.cn (Z. Gu), auyqli@scut.edu.cn (Y. Li).

http://dx.doi.org/10.1016/j.patcog.2017.09.008
0031-3203/

super-exemplars on data sets, but not all clusters have exemplars or super-exemplars.

Aim to remedy the above mentioned deficiencies, in this paper, we abandon the concept of cluster center and consider clusters as *connective regions with high density*. In order to find out the connective regions and high-density regions simultaneously, we proposed a Clustering method based on Hybrid K-Nearest-Neighbor graph (CHKNN). The CHKNN consists of two processing phases, namely, finding subclusters phase and merging phase. During the finding subclusters phase, the CHKNN aims to find out high-density regions. A few tight and small subclusters are generated. During the merging phase, the CHKNN aims to connect the high-density regions. Parts of the subclusters are merged according to the connectivity among them.

In order to choose the optimal parameters of CHKNN, we further propose an internal validity index termed K-Nearest-Neighbor Index (KNNI). Most of the existing internal validity indices [15–19] ignore the connectivity information among data points, that is the reason why they cannot evaluate the validity of nonlinear clustering results accurately. Instead, the KNNI pays attention to the connectivity among one point and its nearest neighbors. Experimental results show that the KNNI can evaluate the validity of nonlinear clustering results accurately.

In summary, the contributions of this paper are summarized as follows:

- A novel data model termed hybrid $k$-nearest-neighbor graph is proposed to represent the original data sets. This model can discover the density and connectivity information containing in data sets conveniently and accurately.
- A clustering method termed CHKNN is developed based on the hybrid $k$-nearest-neighbor graph. The CHKNN method can process the clustering problems on linear and nonlinear data sets effectively.
- A novel internal validity index termed KNNI is proposed, which utilizes the connectivity information among data points to evaluate the validity of nonlinear clustering results.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works of nonlinear graph-based clustering methods and internal validity indices. Sections 3 and 4 describe the proposed CHKNN method and the proposed KNNI in detail, respectively. Experimental results are reported in Section 5. Concluding remarks and directions of future works are presented in Section 6.

## 2. Related works

The majority of graph-based clustering methods aim to find out the high-correlative points [20]. In DBSCAN [8], the density of a point is defined as the number of the *Eps-neighborhoods* of the point. The *Eps-neighborhood* of a point $p$, denoted by $N_{Eps}(p)$, is defined by $N_{Eps}(p) = \{q \in D | dist(p, q) \leq Eps\}$, where $Eps$ is a positive real number. The density-connected and density-reachable points are considered as the high-correlative points. Since DBSCAN cannot find clusters effectively with different density and cannot deal with several outliers which satisfy the density requirements [21], the chameleon [21] is proposed to remedy the above two deficiencies. It first uses a K-Nearest-Neighbor (KNN) graph to generate the subclusters and then determines the similarity between subclusters by looking at their Relative Interconnectivity (RI) and Relative Closeness (RC), finally merges the subclusters with high similarity. Other works using KNN graph are Graph-based Multi-Prototype Competitive Learning (GMPCL) [7] and Shared Nearest Neighbors clustering (SNN) [5] and so on [22–24]. GMPCL relies on KNN graph to initialize coarse clusters and uses multiprototype competitive learning to refine the coarse clustering. SNN utilizes the shared-nearest-neighbor graph to find out the core points and generates clusters based on these core points. Moreover, there are many clustering methods based on Mutual K-Nearest-Neighbor (MKNN) graph [25–29]. Clustering using MUtual nearest NEighbors (CMUNE) [26] utilizes MKNN graph to calculate the density of each point and chooses the high-density points as the seeds from which clusters may grow up. Clustering aLgorithm for Arbitrary ShaPed clusters (CLASP) [25] is a three-phase clustering method. During the first and second phases, CLASP shrinks the original data sets and uses a position adjusting method to make the clusters' structures more clear and distinct. During the third phase, it uses an MKNN graph to find out the arbitrary shaped clusters.

Since the above mentioned graph-based methods just use a single type of nearest neighbors graph, they can only discover either connectivity information or density information containing in the data sets. However, the proposed Hybrid K-Nearest-Neighbor (HKNN) graph which combines the advantages of MKNN graph and KNN graph can discover the density and connectivity information simultaneously. Hence, the CHKNN can process the clustering problems on more complex and noisy nonlinear data sets than other graph-based methods.

Most of the parameters of clustering methods are difficult to be determined in practice. In order to address such problem, a large number of internal validity indices are proposed for choosing the optimal parameters [15–19]. The widely used internal validity indices such as Davies-Bouldin Index (DBI) [15] and Silhouette Coefficient (Sil) [16] are based on the correlation of inter-clusters and the correlation of intra-clusters. Both of these two correlations just consider the distance information but ignore the connectivity information among data points, hence the correlation of intra-clusters in a circular cluster is higher than that in an annular cluster. Therefore, these two indices are unable to evaluate the validity of clustering results on the nonlinear data sets like concentric ring data sets correctly. Another deficiency of most of the existing internal indices is that they do not have any input parameters. In general, there is no a gold standard for clustering results, and the optimal clustering results are different according to different situations and different criteria. Consequently, a certain criterion is necessary when the validity of a clustering result needs to be evaluated.

To remedy the above mentioned deficiencies, we propose an internal validity index termed K-Nearest-Neighbor Index (KNNI), which first constructs a KNN graph using an input parameter $M$, then utilizes the connectivity information among this KNN graph to evaluate the validity of clustering results. The input parameter $M$ represents the number of nearest neighbors of each point and it controls the number of edges in a KNN graph.

## 3. Proposed CHKNN method

For clarity in the description of the proposed method, we first give definitions on the fundamental concepts in Section 3.1, then describe the proposed CHKNN method in Section 3.2, finally present the computational complexity analysis of CHKNN in Section 3.3.

### 3.1. Definitions for CHKNN

Given a data set, $D = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, of $n$ points in $\mathbb{R}^d$, $d$ is the dimension of each point.

**Definition 1** (Hybrid nearest neighbor graph)**.** A hybrid nearest neighbor graph, $G_h = (\boldsymbol{\nu}, A_h)$, is an undirected and weighted graph. The vertex set $\nu$ contains nodes that are constructed from all the samples in $D$ and the affinity matrix $A_h = [A_h(i, j)]_{n \times n}$ is defined as

$$A_h(i,j)=\begin{cases}1, & if\ \mathbf{X}_j\in\mathcal{N}_{P_2}(\mathbf{X}_i)\\ 2, & if\ \mathbf{X}_j\in\mathcal{N}_{P_1}(\mathbf{X}_i)\wedge\mathbf{X}_i\in\mathcal{N}_{P_1}(\mathbf{X}_j)\\ 0, & \text{otherwise}\end{cases}\tag{1}$$

where $\mathcal{N}_k(\mathbf{X}_i)$ denotes the collection of $k$-nearest-neighbors of $\mathbf{X}_i$, $P_1$, $P_2$ are the numbers of nearest neighbors. When $A_h(i,j)=1$, $\mathbf{X}_i$ and $\mathbf{X}_j$ have uni-connected relationship and these two points are called uni-connected points. When $A_h(i,j)=2$, $\mathbf{X}_i$ and $\mathbf{X}_j$ have bi-connected relationship and these two points are called bi-connected points.

**Remark 1.** $P_1$ and $P_2$ are two parameters of CHKNN, $P_1$ is used to find out the mutual $P_1$-nearest-neighbors of each point. While $P_2$ is used to find out the $P_2$-nearest-neighbors of each point.

**Definition 2** (Isolated Point)**.** An isolated point $\mathbf{X}_i\in D$ is defined as a point which satisfies the condition that

$$H(i)\le P\tag{2}$$

where $H(i)$ represents the number of points who have bi-connected relationship with $\mathbf{X}_i$. $P$ is a positive integer.

The isolated set $S$ is defined as a collection of all the isolated points.

**Remark 2.** In this paper, we set $P=P_1-P_3$, where $P_3$, as a parameter of CHKNN, is used to control the size of isolated set $S$. For convenience, we call $H(i)$ as the H-number of $X_i$ later in this paper.

**Definition 3** (Bi-subcluster)**.** Let $\mathbf{X}_i\notin S$, the bi-point of $\mathbf{X}_i$ is defined as a point $\mathbf{X}_j$ satisfies the condition that

$$\mathbf{X}_j\notin S\wedge A_h(i,j)=2\tag{3}$$

where $S$ represents the isolated set (Definition 2).

A bi-subcluster is defined as a set whose element $\mathbf{X}_i$ and all the bi-points of $\mathbf{X}_i$ belong to it.

**Definition 4** (Unidirectional connected subclusters pair)**.** Let $\mathbf{C}_i$, $\mathbf{C}_j$ be two bi-subclusters, a unidirectional connected subclusters pair is defined as a bi-subclusters pair ($\mathbf{C}_i$, $\mathbf{C}_j$) which satisfy the condition that

$$\exists\mathbf{X}_u\in\mathbf{C}_i,\ \exists\mathbf{X}_v\in\mathbf{C}_j\ ,\ A_h(u,v)=1\tag{4}$$

**Remark 3.** $\mathbf{X}_u$ is called a bridge point from $\mathbf{C}_i$ to $\mathbf{C}_j$. $N_{ij}$ denotes the number of bridge points from $\mathbf{C}_i$ to $\mathbf{C}_j$.

**Definition 5** (Mergeable Clusters Pair)**.** Let $(\mathbf{C}_i,\mathbf{C}_{j_1})$, $(\mathbf{C}_i,\mathbf{C}_{j_2})$, ... , $(\mathbf{C}_i,\mathbf{C}_{j_k})$ $(k\in N_+)$ be unidirectional connected subclusters pairs (Definition 4), where $N_+$ denotes the set of positive integers. of positive integers.

$(\mathbf{C}_i,\mathbf{C}_{j_t})$ is called a mergeable clusters pair, where

$$t=\underset{j_t}{\operatorname{argmax}}\,N_{ij_t},\quad j_t\in[j_1,j_k]\tag{5}$$

**Remark 4.** If $k$ is equal to 1, it means that there is only one bi-subcluster can form an unidirectional connected subclusters pair with $\mathbf{C}_i$. In this case, this unidirectional connected subclusters pair will be called a mergeable clusters pair. If there are more than one maximum and equal $N_{ij}$, e.g., $N_{ij_1}$, $N_{ij_2}$,... , $N_{ij_m}(m\in(1,k])$, then $(\mathbf{C}_i,\mathbf{C}_{j_1})$, $(\mathbf{C}_i,\mathbf{C}_{j_2})$,... , $(\mathbf{C}_i,\mathbf{C}_{j_m})$ will be called mergeable clusters pairs simultaneously.

**Definition 6** (Main clusters)**.** Main cluster is defined as the cluster generated by merging the mergeable clusters pairs (Definition 5).

### 3.2. The CHKNN method

In order to find out the clusters which are considered as the connective regions with high density effectively, the CHKNN method consists of two phases, namely, finding subclusters phase and merging phase.

The finding subclusters phase of CHKNN is to find out the isolated points (Definition 2) and generate the bi-subclusters (Definition 3). The H-number (Definition 2) of a point can reveal the degree of correlation among the point and its neighbors. Thus, the isolated point with small H-number have a low correlation with its neighbors and the point in bi-subcluster with large H-number have a high correlation with its neighbors. In general, the density of one point is defined as the number of data points within a unit distance of this point. Thus a point has high correlation with its neighbors means the distance between the point and its neighbors are small and its density is high. Therefore, the bi-subclusters can be considered as the high-density regions. Algorithm 1 summarizes the finding subclusters phase of CHKNN.

**Algorithm 1** Finding Subclusters Phase of CHKNN.

**Input:** Data set $D=\{X_1,\dots,X_n\}$, three parameters $P_1,P_2,P_3$.
Initialize $label[1,\dots,n]=0$, $t=0$, and an empty isolated set $S$.
Construct an HKNN graph $G_h=(v,A_h)$ using $P_1,P_2$ by (1).
**for** each $i\in[1,n]$ **do**
  $H\leftarrow$ The H-number of $X_i$ according to Definition 2.
  **if** $H\le(P_1-P_3)$ **then**
    Add $X_i$ into $S$.
    **continue**.
  **else**
    **if** $label[i]=1$ **then**
      **continue**.
    **else**
      $t\leftarrow t+1$.
      Find out all the bi-points of $X_i$ and generate a bi-subcluster $SC_t$ according to Definition 3.
      the *label* of each point in $SC_t\leftarrow 1$.
    **end if**
  **end if**
**end for**
**Output:** Bi-subclusters $\{SC_1,\dots,SC_t\}(t\in[1,n])$, isolated set $S$.

The merging phase of CHKNN is to find out the mergeable clusters pairs (Definition 5) and generate the main clusters (Definition 6) by merging the mergeable clusters pairs. In this phase, by using the unidirectional connected relationship (Definition 4) among bi-subclusters, the CHKNN can avoid the tendency of generating circular and convex clusters on nonlinear data sets. Obviously, the number of bridge points (Definition 4) between two bi-subclusters can reveal the degree of connectivity between these bi-subclusters. Therefore, by merging the bi-subclusters with numerous bridge points between them in a single step, the CHKNN can avoid generating large clusters by merging a mass of bi-subclusters with weak connectivity. After merging the bi-subclusters with strong connectivity, the CHKNN obtains several main clusters (Definition 6), which can be considered as the connective regions with high density. At the end of the merging phase, each isolated point is assigned to a main cluster to which the nearest assigned point of this isolated point belongs. Algorithm 2 summarizes the merging phase of CHKNN and Fig. 1 shows the overview of CHKNN.

### 3.3. Computational complexity analysis

For constructing the HKNN graph, the CHKNN needs to find out the $k$-nearest-neighbors of each point, the computational complexity of this step is $O(kn^2)$, where $k$ is the number of nearest neighbors of one point and is equal to the value of $P_2$. In finding subclusters phase, the computational complexity of finding the

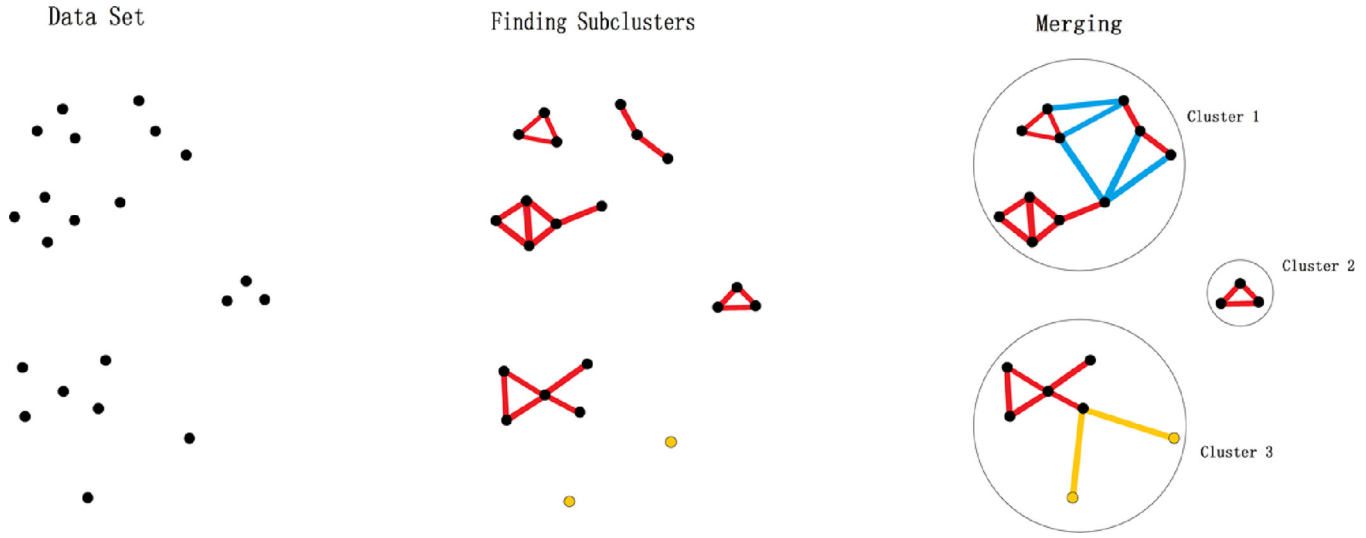



**Fig. 1.** An overview of the CHKNN. In the schematic of finding subclusters phase, the red lines connect the points in the same bi-subclusters, while the yellow points represent the isolated points. In the schematic of merging phase, the blue lines connect the bridge points from one bi-subcluster to another bi-subcluster, while the yellow lines show that the isolated points are assigned to their nearest main cluster. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Algorithm 2** Merging Phase of CHKNN.

**Input:** Bi-subclusters $\{SC_1, \ldots, SC_t\}(t \in [1, n])$.

Initialize $label[1, \ldots, t] = 0, k = 0, m = 0$.
**for** each $i \in [1, t]$ **do**
  **if** $label[i] = 0$ **then**
    $k \leftarrow k + 1$.
    $label[i] \leftarrow k$.
    Initialize a new cluster $C_k$.
    Add the points in $SC_i$ into $C_k$.
  **end if**
  $r \leftarrow label[i]$.
  **for** each $j \in [1, t]$ $(j \neq i)$ **do**
    **if** $(SC_i, SC_j)$ is a mergeable clusters pair **then**
      Add the points in $SC_j$ into $C_r$.
      $label[j] \leftarrow r$.
    **end if**
  **end for**
**end for**
**while** isolated set is not empty **do**
  $m \leftarrow m + 1$.
  **for** each isolated points $X_l$ **do**
    **if** the $m$-nearest-neighbor of $X_l$ belongs to a main cluster $C_h$ $(h \in [1, k])$ **then**
      Add $X_l$ into $C_h$.
    **end if**
  **end for**
**end while**

**Output:** Clusters $\{C_1, \ldots, C_k\}$.

isolated points is $O(n)$, and the computational complexity of generating bi-subclusters is $O(tn^2)$, where $t$ is the maximum number of times of the *for-loop* when finding the bi-points of each point in a bi-subcluster. In practice, $t$ is always less than $k$. In merging phase, the number of bi-subclusters and the number of points in each bi-subcluster are much less than $n$. Moreover, the CHKNN processes one bi-subcluster each time, hence the computational complexity of merging phase is less than $O(n^2)$. Therefore, the overall computational complexity of CHKNN is on the order of $O(kn^2)$.

## 4. Proposed KNNI

In this section, the KNNI, which can be used to evaluate the validity of nonlinear clustering results by varying an input parameter $M$, is introduced in detail. We first formally define the related concepts in Section 4.1, then describe the proposed KNNI in Section 4.2.

### 4.1. Related definitions for KNNI

In density-based clustering method DP [10], cluster centers are defined as data points that are surrounded by neighbors with lower local density, and the cluster centers are at a relatively large distance from any points with a higher local density. Moreover, a physical interpretation of density-based metric has been discussed in [30,31]. According to Fermat Principle, the optical path of the rays of light propagating in a medium is shorter in the areas of high density. The distance based on the optical length is precisely the density-based metric. Using density based metric on data sets, it shows that the data points in the regions with high density tend to be grouped together. According to the above discussions, high-density points are more likely to be an important part of clusters than low-density points which always locate in sparse boundaries between clusters. The boundaries are always fuzzy and have little influence on the validity of the entire clustering results, but the values of internal validity indices will be affected by the categories of points in boundaries severely. In order to reduce the impact of boundaries, the KNNI only considers the categories of high-density points.

Traditional density [8], which is defined as the number of *Eps-neighborhoods* of a point, is widely used in many methods [10,32,33]. However, it has great difficulty in measuring the correlation among one point and its neighbors on data sets that consist of several clusters with significantly different scales. As illustrated in Fig. 2, the cluster $A$ is a small and tight cluster, while the cluster $B$ is a large and sparse cluster. On this data set, provided that the value of *Eps* is set to a lesser one, the density of points in the cluster $B$ will almost be zero. On the contrary, provided that the value of *Eps* is set to a larger one, the density of points in the cluster $A$ will almost be similar. Consequently, it is difficult to choose an ap-

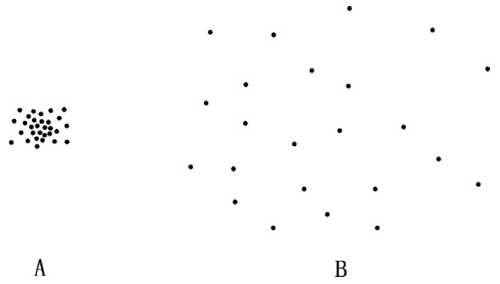



**Fig. 2.** A data set consists of two clusters with significantly different scales. It is difficult to choose an appropriate *Eps* to measure the density of each point.

propriate *Eps* to measure the correlation among one point and its neighbors on this data set.

To remedy this deficiency, we use the Mean Distance Between one point and its $M$ nearest neighbors (MDBM) to measure the correlation among one point and its neighbors. Since each point has its nearest neighbors, the points in the clusters with different scales can be measured simultaneously based on MDBM. Related definitions are presented as follows. For convenience, we call the evaluation method which is used to compute the KNNI value of a clustering result as KNNI method later in this paper.

Given a clustering result, $D = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, $\mathbf{L} = \{\mathbf{L}_1, \ldots, \mathbf{L}_n\}$, where $\mathbf{L}_i$ is the label of $\mathbf{X}_i$. The KNNI method first constructs an KNN graph $G_M = (v, A_M)$. The vertex set $v$ contains nodes which are constructed from all the samples in $D$, the affinity matrix $A_M = [A_M(i,j)]_{n\times n}$ is

$$A_M(i,j) = \begin{cases} 1, & if\ \mathbf{X}_j \in \mathcal{N}_M(\mathbf{X}_i) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $\mathcal{N}_k(\mathbf{X}_i)$ denotes the collection of $k$-nearest-neighbors of $\mathbf{X}_i$.

**Definition 7** (MDBM)**.** The MDBM of $\mathbf{X}_i$ $D_M(i)$ is defined as

$$D_M(i) = \frac{\sum_{k=1}^{M} d\left(\mathbf{X}_{(i,k)}, \mathbf{X}_i\right)}{M} \tag{7}$$

where $d(\mathbf{X}_j, \mathbf{X}_i)$ represents the distance between $\mathbf{X}_j$ and $\mathbf{X}_i$, $\mathbf{X}_{(i,k)}$ denotes the *kth* nearest neighbor of $\mathbf{X}_i$. $M$ is a positive integer.

**Remark 5.** $M$ is the input parameter of KNNI.

**Definition 8** (Average MDBM)**.** The average MDBM $D_a$ is defined as

$$D_a = \frac{\sum_{k=1}^{n} D_M(k)}{n} \tag{8}$$

where $D_M(k)$ represents the MDBM of $X_k$ and $n$ denotes the number of total data points.

**Definition 9** (M-Pair)**.** Let $X_i, X_j \in D$. The M-pair is defined as a pair of points $(X_i, X_j)$ which satisfies the condition that

$$(\mathbf{A}_m(i,j) = 1) \ \wedge \ (\mathbf{D}_M(i) \leqslant \mathbf{D}_a \wedge \mathbf{D}_M(j) \leqslant \mathbf{D}_a) \tag{9}$$

**Definition 10** (Region)**.** Let $(\mathbf{X}_i, \mathbf{X}_j)$ be an M-pair (Definition 9). The region ① of $(\mathbf{X}_i, \mathbf{X}_j)$ is defined as a collection of all the points $\mathbf{P} \in D$ which satisfy the condition that

$$\begin{aligned} &\mathbf{P} \in \mathcal{N}_M(\mathbf{X}_i) \vee \mathcal{N}_M\left(\mathbf{X}_j\right) \\ &\wedge \ d(\mathbf{P}, \mathbf{X}_i)^2 + d\left(\mathbf{X}_i, \mathbf{X}_j\right)^2 < d\left(\mathbf{P}, \mathbf{X}_j\right)^2 \end{aligned} \tag{10}$$

The region ② of $(\mathbf{X}_i, \mathbf{X}_j)$ is defined as a collection of all the points $\mathbf{Q} \in D$ which satisfy the condition that

$$\begin{aligned} &\mathbf{Q} \in \mathcal{N}_M(\mathbf{X}_i) \vee \mathcal{N}_M\left(\mathbf{X}_j\right) \\ &\wedge \ d(\mathbf{Q}, \mathbf{X}_i)^2 + d\left(\mathbf{X}_i, \mathbf{X}_j\right)^2 \geq d\left(\mathbf{Q}, \mathbf{X}_j\right)^2 \\ &\wedge \ d\left(\mathbf{Q}, \mathbf{X}_j\right)^2 + d\left(\mathbf{X}_i, \mathbf{X}_j\right)^2 \geq d(\mathbf{Q}, \mathbf{X}_i)^2 \end{aligned} \tag{11}$$

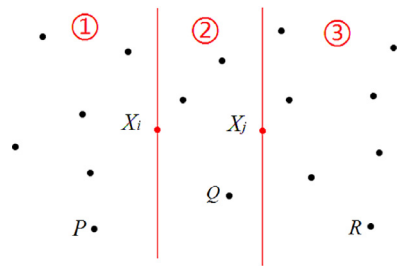



**Fig. 3.** An illustration of the three regions of an M-pair $(\mathbf{X}_i, \mathbf{X}_j)$. The black points represent the $M$ nearest neighbors of $\mathbf{X}_i$ or $\mathbf{X}_j$ ($M$ is equal to 11), two red lines divide all these points into three regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The region ③ of $(\mathbf{X}_i, \mathbf{X}_j)$ is defined as a collection of all the points $\mathbf{R} \in D$ which satisfy the condition that

$$\begin{aligned} &\mathbf{R} \in \mathcal{N}_M(\mathbf{X}_i) \vee \mathcal{N}_M\left(\mathbf{X}_j\right) \\ &\wedge \ d\left(\mathbf{R}, \mathbf{X}_j\right)^2 + d\left(\mathbf{X}_i, \mathbf{X}_j\right)^2 < d(\mathbf{R}, \mathbf{X}_i)^2 \end{aligned} \tag{12}$$

where $d(\mathbf{X}_j, \mathbf{X}_i)$ represents the distance between $\mathbf{X}_j$ and $\mathbf{X}_i$. $\mathcal{N}_k(\mathbf{X}_i)$ denotes the collection of $k$-nearest-neighbors of $\mathbf{X}_i$.

**Remark 6.** Fig. 3 illustrates the three regions of an M-pair $(\mathbf{X}_i, \mathbf{X}_j)$.

### 4.2. The KNNI method

The score of an M-pair $(\mathbf{X}_i, \mathbf{X}_j)$ is computed as

$$G\left(\mathbf{X}_i, \mathbf{X}_j\right) = \begin{cases} \dfrac{g}{2 \times M}, & \text{if } \mathbf{L}_i \neq \mathbf{L}_j \\ -\dfrac{g}{2 \times M}, & \text{if } \mathbf{L}_i = \mathbf{L}_j \end{cases} \tag{13}$$

where $g = (N_1 - N_2) + (N_3 - N_2)$, $N_k$ denotes the number of points in region k (k = 1,2,3) (Definition 10). $M$ is the input parameter of the proposed KNNI method.

After constructing the KNN graph $G_M$, the KNNI method computes the MDBM of each point and the average MDBM, then picks out all the M-pairs and computes the scores of them. Finally, the KNNI value is computed as

$$KNNI = \sum_{k=1}^{m} G\left(\mathbf{X}_{i_k}, \mathbf{X}_{j_k}\right) \tag{14}$$

where $G(\mathbf{X}_{i_k}, \mathbf{X}_{j_k})$ denotes the score of an M-pair $(\mathbf{X}_{i_k}, \mathbf{X}_{j_k})$ according to (13) and $k$ represents the index of M-pair, $m$ denotes the number of M-pairs.

According to Definition 9, $N_1$ and $N_3$ of an M-pair will be similar and not much small. Because if the $N_1$ or $N_3$ of a points pair $(\mathbf{X}_i, \mathbf{X}_j)$ is significantly small, the $\mathbf{X}_i$ or $\mathbf{X}_j$ will be likely to locate on the boundaries of a cluster and the MDBM of the point will be large. In this case, these two points will not satisfy the condition (9) and cannot be considered as an M-pair. Consequently, we only need to compare the sum of $N_1$ and $N_3$ with $2 \times N_2$ of an M-pair in KNNI method. As illustrated in Fig. 3, since the sum of $N_1$ and $N_3$ is larger than $2 \times N_2$, we can consider that the density of the left region of $\mathbf{X}_i$ and the right region of $\mathbf{X}_j$ are both larger than the density of the region between $\mathbf{X}_i$ and $\mathbf{X}_j$. In this case, $\mathbf{X}_i$ and $\mathbf{X}_j$ are much prone to locate on the boundary between two clusters. Therefore, when $\mathbf{X}_i$ and $\mathbf{X}_j$ are assigned to different clusters by a clustering result, this result tends to be a reasonable one and the $G(\mathbf{X}_i, \mathbf{X}_j)$ is positive. Otherwise the $G(\mathbf{X}_i, \mathbf{X}_j)$ is negative. Finally, we add up the scores of all the M-pairs as KNNI value of this clustering result, and the result with the highest KNNI value can be considered as the optimal one. Algorithm 3 summarizes the KNNI method.

**Algorithm 3** The KNNI Method.

**Input:** Data set $D = \{X_1, \ldots, X_n\}$, Labels of points in a clustering result $\{L_1, \ldots, L_n\}$.

```
Initialize KNNI = 0, G = 0.
Construct a KNN graph by (6).
Compute the MDBM of each point by (7) and the average MDBM by (8).
for each i ∈ [1, n] do
    for each j ∈ [1, n] do
        if (X_i, X_j) is an M-pair then
            G ← Compute the score of this M-pair by (13).
            KNNI ← KNNI + G.
        end if
    end for
end for
```

**Output:** KNNI value of this clustering result *KNNI*.

## 5. Results

In this section, we first present experimental results of the proposed CHKNN on synthetic linear and nonlinear data sets in Section 5.1, then compare the results of CHKNN with nine methods in literatures on seven data sets (one synthetic and six real) in Section 5.2. Afterwards, we report the performance of cross validation of nine methods in Section 5.3. Furthermore, we show the effectiveness of the CHKNN on video clustering in Section 5.4. Finally, the comparative results of KNNI and other internal indices on nonlinear data sets are reported in Section 5.5. We use the Euclidean distance as a measure of distance in all experiments. All the experiments are performed on MATLAB 8.4.0.150421 (R2014b) 64-bit edition.

### 5.1. Experimental results of CHKNN on synthetic data sets

In this subsection, we first analyse the performance sensitivity of the CHKNN to the parameters, then present the clustering results of CHKNN on six synthetic data sets (one linear data set, two nonlinear data sets without noises, and three complex nonlinear data sets with noises), which are shown in Fig. 4.

#### 5.1.1. Parameters analysis

Since the three parameters $P_1$, $P_2$ and $P_3$ of CHKNN are all positive integers, the ranges of optional parameters are easy to be determined and the users can choose the optimal parameters fast and easily. In general, $P_1$ and $P_3$ are used to adjust the numbers of bi-subclusters, while $P_2$ is used to determine which bi-subclusters should be merged and adjust the number of main clusters.

Firstly, we analyse the effect of the parameter $P_1$ and $P_3$ on the number of bi-subclusters. As an example, Fig. 5 shows the number of bi-subclusters ($N$) as a function of the values of $P_1$ and $P_3$ on DS4_3, DS5_7 and DS6_8. If the $P_1$ is small, according to Definition 1 and Definition 3, the CHKNN will generate less and scattered bi-connected points and obtain a mass of bi-subclusters that are scattered extremely, and the number of bi-subclusters may be larger than the true number of clusters significantly. In this case, no matter what the values of $P_2$ and $P_3$ are, the CHKNN is hard to find the clusters correctly because bi-subclusters may appear in low-density regions, and the points with high density located around the low-density regions may be merged into the same cluster. With the increase of $P_1$, the number of bi-connected points increases and the number of bi-subclusters decreases.

According to Definition 2, the value of $P_3$ controls the size of isolated set. The $P_3$ is larger, the smaller the size of isolated set is and the more the bi-connected points are assigned to the same bi-subcluster.

According to this empirical study, we suggest that $P_3$ should not be larger than 5 and $P_1$ should not be too large ($\leq$ 30). In Fig. 5, the red points represent the optimal values of $P_1$, $P_3$ and the number of bi-subclusters on three data sets, respectively. The number of bi-subclusters on each data set is not much larger than the true number of clusters. Consequently, we can find out the optimal value of $P_2$ to merge the bi-subclusters on each data set correctly.

Secondly, we analyse the effect of $P_2$ on the number of clusters. Fig. 6 shows the number of clusters as a function of the value of $P_2$ when fixing $P_1$ and $P_3$ on DS6_8, the synthetic control chart time series (Sccts) data set [34] and MNIST [35] data set. With the increase of $P_2$, the number of uni-connected points increases according to Definition 1 and more mergeable clusters pairs are merged according to Definition 5, hence we can find the trend that the number of clusters decreases in Fig. 6.

In practice, we suggest that $P_2$ should not be less than $P_1$. The users can first determine $P_1$ and $P_3$, then increase $P_2$ gradually, until the optimal clustering result is obtained. However, the users cannot find out the optimal value of $P_2$ with arbitrary $P_1$ and $P_3$, because if the number of bi-subclusters is larger than the true number of clusters significantly, the CHKNN cannot emerge all bi-subclusters in one step and obtain the optimal result.

#### 5.1.2. Results of CHKNN

According to above discussion of parameters selection, we found the optimal parameters of CHKNN and obtained the optimal clustering results on the six synthetic data sets mentioned above. The bi-subclusters and the clustering results of each data set are presented in Figs. 7 and 8, respectively.

As shown in Fig. 7, the bi-subclusters in each data set are generated by adjusting $P_1$ and $P_3$. All the bi-subclusters locate in the high-density regions, while the isolated points locate in low-density regions. Thus, at the end of finding subclusters phase, the CHKNN can obtain the bi-subclusters effectively and accurately, which can reveal the distribution of the data set clearly.

The clustering results of CHKNN shown in Fig. 8 indicate that the CHKNN is robust against noises and the connective regions with high density are divided accurately. Thus, we can conclude that whether it is banded and concentric ring data sets or traditional convex data sets, the CHKNN can obtain the optimal clustering results by using appropriate parameters.

### 5.2. Comparison results of clustering methods

In this subsection, we present the comparison results of CHKNN and eight clustering methods on six synthetic data sets (DS1_4, DS2_4, DS3_3, DS4_3, DS5_7, DS6_8) and six real-world data sets (the wine data set (Wine), the multiple features data set (Mfeat), the glass data set (Glass), the synthetic control chart time series (Sccts) data set from the UCI repository [34], USPS [36] data set and MNIST [35] data set.)

For the data sets studied here, the underlying class labels are known, hence we use an external index termed Normalized Mutual Information (NMI) [37] to evaluate the validity of the clustering results. NMI is one of the most widely used external validity indices. It is used to measure the similarity between the clusters labels obtained by clustering methods and the underlying class labels. A high NMI value indicates that the clustering and underlying class labels match well.

The methods used for comparison are listed as follows:

1. Affinity Propagation clustering (AP) [38] and Multi-Exemplar Affinity Propagation (MEAP) [14]. The similarity $S(X_i, X_j)$ between two points $X_i$, $X_j$ is set to $-||X_i - X_j||^2$ in AP, and $maxdist - ||X_i - X_j||^2$ in MEAP, where $maxdist$ denotes the $max_{i,j\in\{1,\ldots,N\}}||X_i - X_j||^2$.

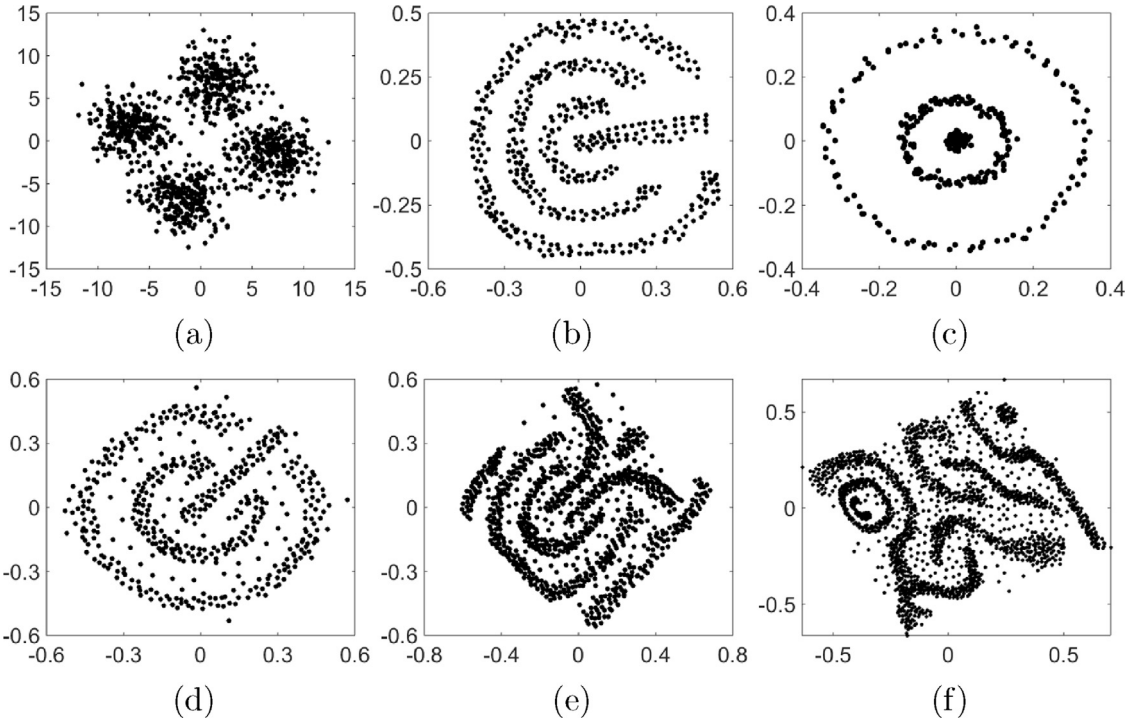



**Fig. 4.** Six synthetic data sets used in our experiments. (a) DS1_4 consists of four convex clusters; (b) DS2_4 consists of four non-convex clusters; (c) DS3_3 consists of three clusters; (d) DS4_3 consists of three clusters with noises; (e) DS5_7 consists of seven clusters with noises; (f) DS6_8 consists of eight clusters with noises.

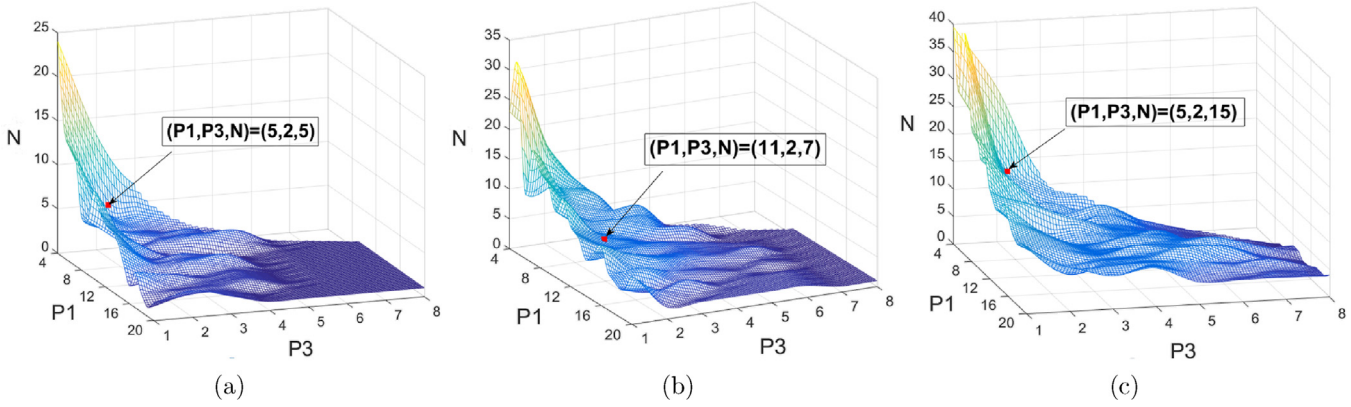



**Fig. 5.** The number of bi-subclusters as a function of the values of $P_1$ and $P_3$ on (a) DS4_3, (b) DS5_7 and (c) DS6_8. The red point in each subfigure marks the optimal values of $P_1$, $P_3$ and the number of bi-subclusters on the corresponding data set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

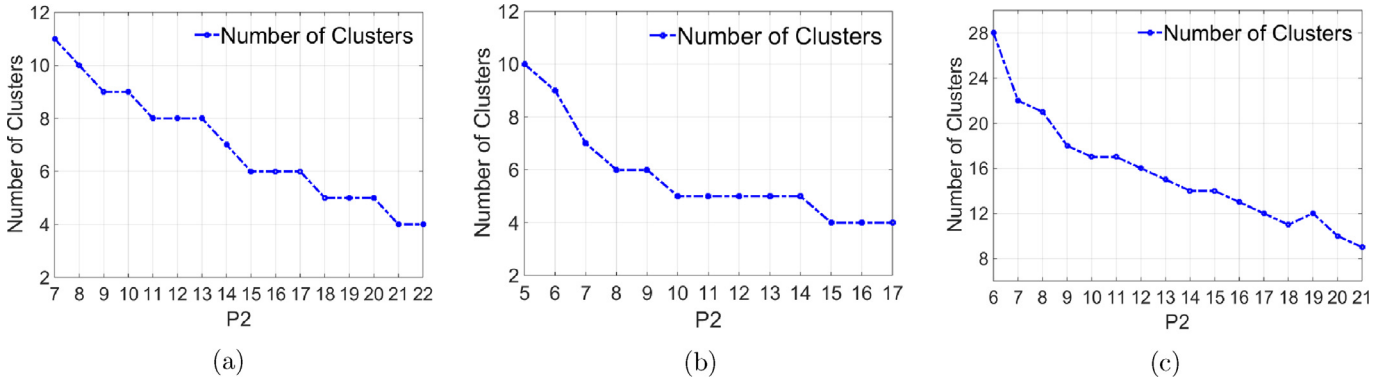



**Fig. 6.** The number of clusters as a function of $P_2$ when $P_1$, $P_3$ are fixed on DS6_8, Sccts and MNIST. (a) $P_1 = 7, P_3 = 2$ on DS6_8; (b) $P_1 = 5, P_3 = 2$ on Sccts; (c) $P_1 = 6, P_3 = 1$ on MNIST.

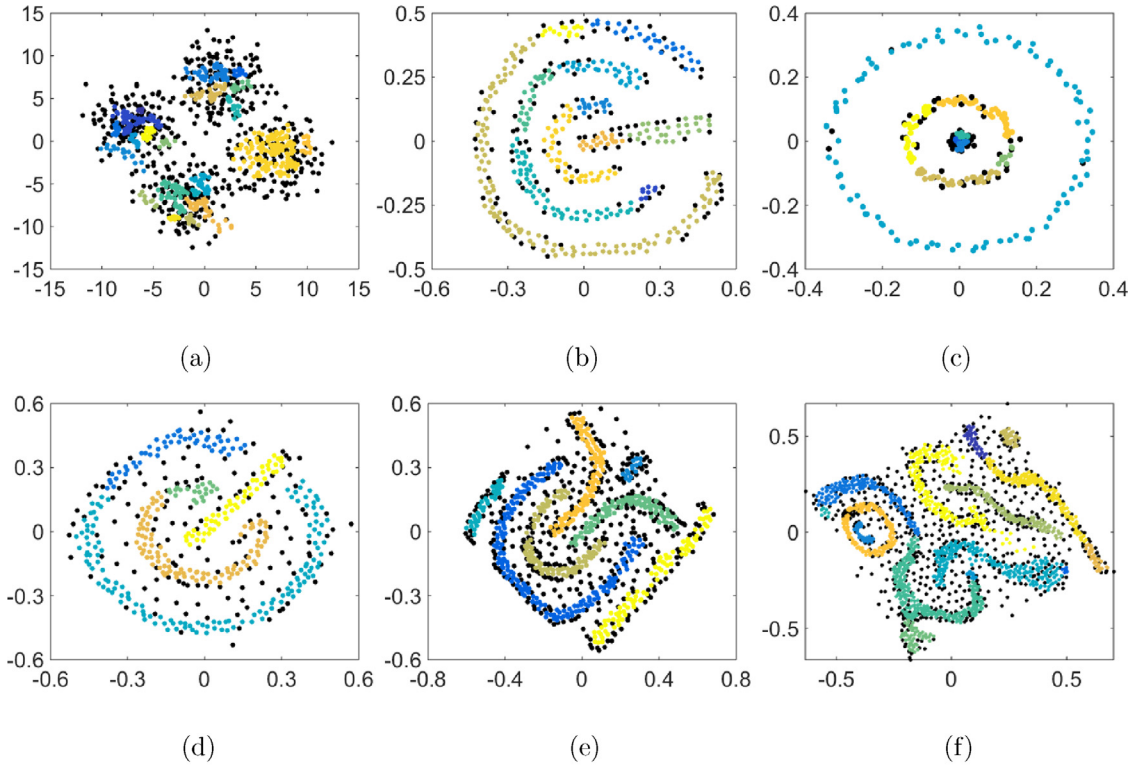



**Fig. 7.** The bi-subclusters obtained by CHKNN on six synthetic data sets. (a) DS1_4; (b) DS2_4; (c) DS3_3; (d) DS4_3; (e) DS5_7; (f) DS6_8. The points with same color represent the data points in the same bi-subcluster, while the black points represent the isolated points.

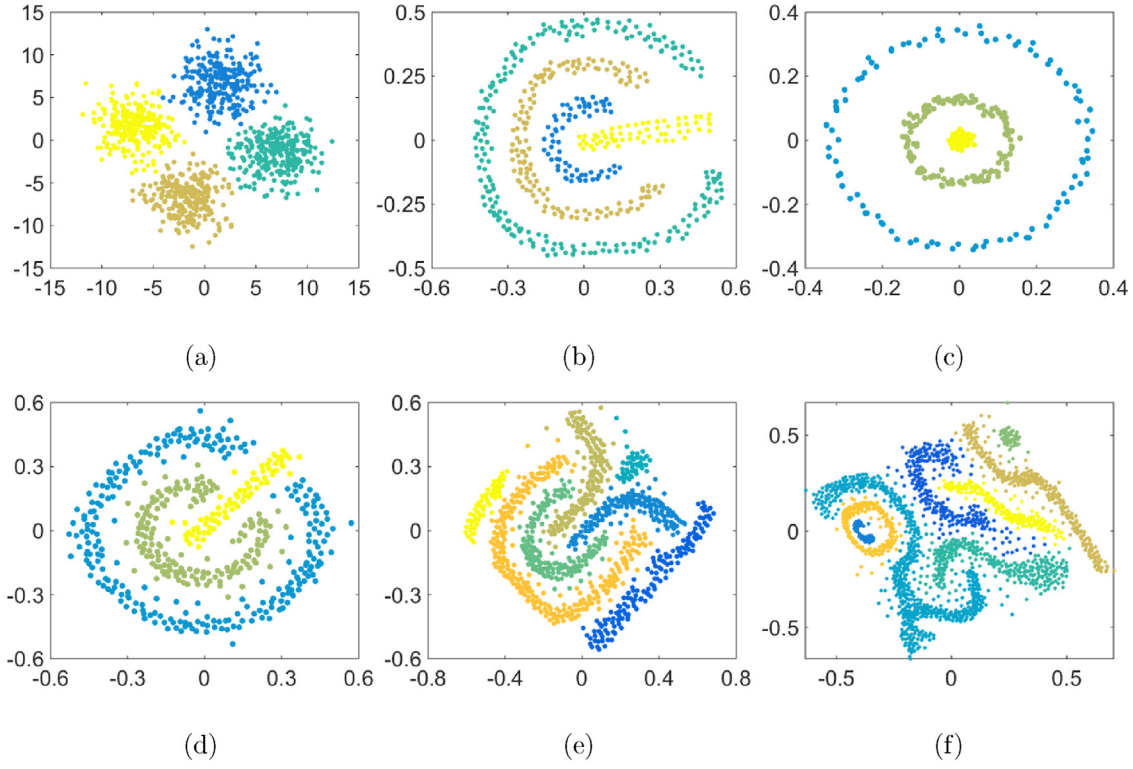



**Fig. 8.** The clustering results obtained by CHKNN on six synthetic data sets. (a) DS1_4; (b) DS2_4; (c) DS3_3; (d) DS4_3; (e) DS5_7; (f) DS6_8. The points with same color represent the data points in the same cluster.

**Table 1**
The means and standard deviations (in parentheses) of NMI (over 20 runs) generated by different methods on twelve data sets with given parameters.

| Data set | Method | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | AP | MEAP | DP | COLL | Kkmeans | Ncut | Graclus | PSVC | CHKNN |
| Wine | 0.332(0.000) | 0.316(0.000) | 0.384(0.006) | 0.431(0.000) | 0.429(0.000) | 0.401(0.006) | 0.403(0.005) | 0.412(0.002) | **0.437(0.000)** |
| Mfeat | 0.601(0.000) | 0.561(0.000) | 0.530(0.014) | 0.623(0.001) | 0.581(0.008) | 0.710(0.005) | 0.648(0.002) | 0.731(0.003) | **0.789(0.000)** |
| Glass | 0.611(0.000) | 0.587(0.000) | 0.623(0.015) | 0.682(0.007) | 0.661(0.027) | 0.642(0.005) | 0.579(0.005) | 0.681(0.004) | **0.775(0.000)** |
| Sccts | 0.724(0.000) | 0.650(0.000) | 0.501(0.021) | 0.754(0.022) | 0.711(0.021) | 0.784(0.005) | 0.569(0.003) | 0.780(0.003) | **0.878(0.000)** |
| USPS | 0.377(0.000) | 0.532(0.000) | 0.413(0.006) | 0.468(0.003) | 0.474(0.026) | 0.486(0.003) | 0.421(0.001) | 0.387(0.003) | **0.723(0.000)** |
| MNIST | 0.411(0.000) | 0.585(0.000) | 0.453(0.017) | 0.523(0.016) | 0.506(0.015) | 0.558(0.004) | 0.451(0.004) | 0.410(0.004) | **0.690(0.000)** |
| DS1_4 | 0.945(0.000) | 0.964(0.000) | 0.988(0.000) | 0.981(0.002) | 0.945(0.121) | 0.965(0.001) | 0.978(0.000) | 0.985(0.001) | **0.990(0.000)** |
| DS2_4 | 0.365(0.000) | 0.421(0.000) | 0.315(0.037) | 0.330(0.021) | 0.353(0.047) | 0.323(0.010) | 0.244(0.003) | 0.214(0.007) | **1.000(0.000)** |
| DS3_3 | 0.452(0.000) | 0.399(0.000) | 0.337(0.021) | 0.395(0.032) | 0.387(0.004) | 0.356(0.011) | 0.361(0.002) | 0.335(0.007) | **1.000(0.000)** |
| DS4_3 | 0.498(0.000) | 0.541(0.000) | 0.395(0.036) | 0.398(0.013) | 0.424(0.02) | 0.486(0.008) | 0.428(0.005) | 0.396(0.009) | **1.000(0.000)** |
| DS5_7 | 0.466(0.000) | 0.484(0.000) | 0.568(0.015) | 0.471(0.005) | 0.464(0.008) | 0.499(0.008) | 0.661(0.003) | 0.483(0.009) | **1.000(0.000)** |
| DS6_8 | 0.531(0.000) | 0.468(0.000) | 0.567(0.032) | 0.507(0.014) | 0.492(0.028) | 0.496(0.008) | 0.484(0.001) | 0.464(0.010) | **1.000(0.000)** |

2. Density Peaks clustering (DP) [10][1]. The recently proposed density-based clustering method is performed and compared.
3. Conscience On-Line Learning clustering (COLL) [3] and Kernel k-means (Kkmeans) [2]. These two methods are kernel-based methods. The Gaussian kernel $K(a_i, a_j) = \exp(-||a_i - a_j||^2/2\alpha^2)$ is used and the initial seeds are randomly initialized, which are commonly used in k-means-like methods.
4. Spectral clustering based on normalized cut (Ncut) [6] and Graclus [39]. These two methods are graph-based methods. We use the authors' methods to compute the correlation among data points and generate the graphs.
5. Position regularized Support Vector Clustering (PSVC) [12]. It is an effective support vector-based method, the Gaussian kernel $K(a_i, a_j) = \exp(-q * ||a_i - a_j||^2)$ is used according to the literature, where $q$ is an input parameter.

Table 1 lists the means and standard deviations of MNI generated by studied methods with the given parameters. It is clear that the CHKNN significantly outperforms the compared methods on all data sets, especially on large data sets (USPS, MNIST) and nonlinear synthetic data sets (DS2_4, DS3_3, DS4_3, DS5_7, DS6_8). Compared to the second winner MEAP, the CHKNN achieves about 36% improvement on USPS and about 18% improvement on MNIST, respectively. On small data sets Sccts, Glass, Mfeat and Wine, the CHKNN also generates the highest NMI values. As shown in Fig. 4, five of the six synthetic data sets consist of several band-like clusters and there are no cluster centers of these data sets. Since most of the studied methods in this section tend to find out the clusters centers, they cannot obtain correct clustering results. The CHKNN tries to find out the connective regions with high density, that is the reason why it can outperform other methods on all data sets. Moreover, the CHKNN method does not use any stochastic algorithm and there is no need to set up the initial cluster centers. With the given parameters $P_1$, $P_2$ and $P_3$, the clustering result is stable for a given dataset. So the CHKNN method is a deterministic method.

In Fig. 9, the clustering results with the highest NMI value obtained by the eight methods on DS5_7 are presented. We can find that the clustering result of CHKNN is the most appropriate one, seven nonlinear clusters have been clearly divided, the points in the same bands are assigned to the same cluster. However, MEAP divide the points into 31 clusters which are nearly circular. DP is adept in finding the high-density regions. But in this data set, since the density of points in different parts of the same band are similar, DP partitions the points in the same band into several clusters. AP, COLL, Kkmeans and Graclus tend to find out the circular clusters, hence they obtain almost the same clustering result and the NMI values of which are all low. Clustering result of Ncut gets the second highest NMI, but Ncut can only divide parts of the points in a band into the same cluster.

By comparing the results of our method with the other compared methods, we can conclude that the proposed CHKNN performs the best, since the CHKNN considers the density and connectivity information among data sets simultaneously.

### 5.3. Cross validation results

In this subsection, we report the performance of cross validations of the studied methods on twelve data sets. We firstly randomly sampled 90% of the original data set as training set. The remaining 10% of the original data set was used as testing set. The underlying true cluster labels of data points in training set were not used in experiments while the true labels of testing set were used for evaluating the clustering results. Secondly, we used eight methods to cluster the training set, then utilized the estimated labels of data points in training set to predict the labels of data points in testing set. For the data points in the testing set, we found out the 10-nearest-neighbors of each point, then assigned each point into a cluster to which most of its 10-nearest-neighbors belonged. Finally, we computed the NMI value of the prediction result of the testing set. The above cross validation procedure was repeated 50 times and the means and standard deviations (in parentheses) of NMI are shown in Table 2. It is clear that the CHKNN outperforms the compared methods on most data sets, especially on the large data sets USPS, MNIST and the nonlinear synthetic data sets (DS2_4, DS3_3, DS4_3, DS5_7, DS6_8). For instance, the CHKNN achieves about 33% improvement on USPS and about 21% improvement on MNIST compared to the second winner COLL. The comparative results in Table 2 demonstrate the generalization performance of the proposed CHKNN method.

### 5.4. Performance of video clustering

In this subsection, we report the experimental results of the CHKNN for the video clustering task. The purpose of video clustering is to cluster video frames into different segments according to different scenes. In our experiments, the gray-scale values of the original pixels were used as the feature vector of each frame, while the Euclidean distance of these vectors was used as the measure of distance among corresponding frames.

The videos used in this subsection were: 'NASA 25th Anniversary Show, Segment07' video sequence (Video. 1), 'NASA 25th Anniversary Show, Segment01' video sequence (Video. 2), 'New Indians, Segment03' video sequence (Video. 3), 'Winning: Aerospace, Segment01' video sequence (Video. 4) [40]. For comparison, the "ground truth" segments of each video sequence had been

[1] http://science.sciencemag.org/content/344/6191/1492.

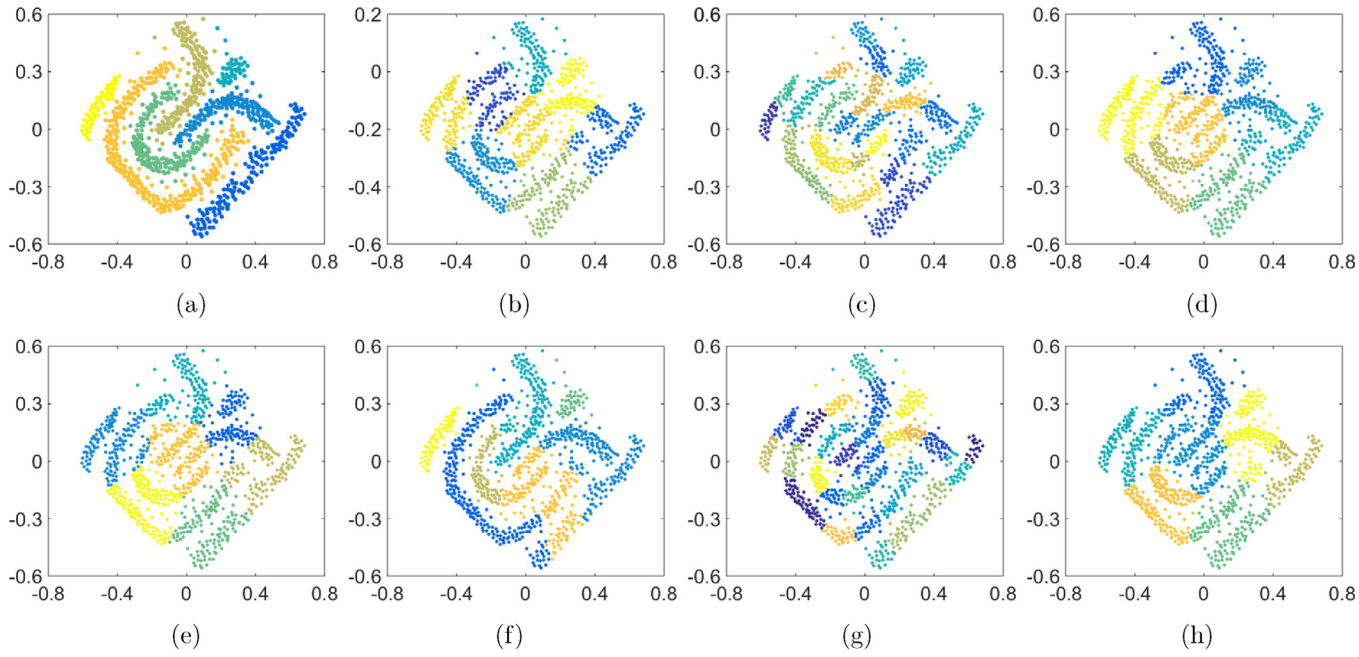



**Fig. 9.** The results with the highest NMI value obtained by eight methods on DS5_7. (a) Result of CHKNN; (b) Result of AP; (c) Result of DP; (d) Result of COLL; (e) Result of Kkmeans; (f) Result of Ncut; (g) Result of MEAP; (h) Result of Graclus.

**Table 2**
The means and standard deviations (in parentheses) of NMI of testing set (over 50 cross validations) on synthetic and real-world data sets.

| Data set | Method | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | AP | MEAP | DP | COLL | Kkmeans | Ncut | Graclus | PSVC | CHKNN |
| Wine | 0.469(0.123) | 0.446(0.101) | 0.473(0.068) | **519(0.137)** | 0.479(0.131) | 0.471(0.165) | 0.469(0.156) | 0.411(0.078) | 0.465(0.045) |
| Mfeat | 0.503(0.014) | 0.629(0.059) | 0.675(0.055) | 0.597(0.022) | 0.635(0.057) | 0.506(0.024) | 0.659(0.089) | 0.591(0.057) | **0.789(0.03)** |
| Glass | 0.689(0.048) | 0.622(0.061) | 0.649(0.069) | **0.770(0.047)** | 0.669(0.083) | 0.670(0.071) | 0.628(0.038) | 0.611(0.104) | 0.700(0.110) |
| Sccts | 0.726(0.039) | 0.696(0.049) | 0.747(0.052) | 0.776(0.071) | 0.733(0.036) | 0.566(0.033) | 0.556(0.033) | 0.675(0.047) | **0.788(0.034)** |
| USPS | 0.469(0.086) | 0.507(0.053) | 0.346(0.037) | 0.532(0.009) | 0.512(0.027) | 0.496(0.045) | 0.368(0.016) | 0.375(0.068) | **0.708(0.065)** |
| MNIST | 0.447(0.046) | 0.515(0.065) | 0.437(0.033) | 0.586(0.018) | 0.573(0.035) | 0.462(0.068) | 0.450(0.017) | 0.395(0.051) | **0.712(0.017)** |
| DS1_4 | 0.923(0.053) | 0.916(0.016) | 0.939(0.048) | 0.901(0.054) | 0.849(0.097) | 0.904(0.023) | 0.897(0.030) | 0.894(0.096) | **0.974(0.043)** |
| DS2_4 | 0.377(0.098) | 0.418(0.047) | 0.315(0.062) | 0.301(0.096) | 0.286(0.077) | 0.276(0.052) | 0.298(0.050) | 0.259(0.153) | **0.901(0.101)** |
| DS3_3 | 0.463(0.049) | 0.391(0.076) | 0.384(0.050) | 0.175(0.068) | 0.193(0.079) | 0.248(0.087) | 0.337(0.104) | 0.281(0.064) | **0.984(0.012)** |
| DS4_3 | 0.375(0.027) | 0.337(0.063) | 0.341(0.056) | 0.160(0.061) | 0.139(0.074) | 0.219(0.065) | 0.243(0.077) | 0.289(0.081) | **0.927(0.138)** |
| DS5_7 | 0.533(0.031) | 0.497(0.042) | 0.592(0.045) | 0.494(0.043) | 0.491(0.045) | 0.346(0.026) | 0.423(0.043) | 0.512(0.092) | **0.929(0.052)** |
| DS6_8 | 0.511(0.024) | 0.538(0.085) | 0.520(0.052) | 0.522(0.036) | 0.518(0.033) | 0.347(0.024) | 0.423(0.027) | 0.426(0.048) | **0.926(0.030)** |

**Table 3**
The properties of the four video sequences.

| Video | Frame | Width × Height | $d$ | $n$ |
|---|---|---|---|---|
| Video. 1 | 1590 | 320 × 240 | 76,800 | 10 |
| Video. 2 | 914 | 320 × 240 | 76,800 | 9 |
| Video. 3 | 1068 | 320 × 240 | 76,800 | 12 |
| Video. 4 | 1779 | 320 × 240 | 76,800 | 9 |

$d$ denotes the dimension of each video, $n$ denotes the number of 'ground truth' clusters of each video.

manually partitioned. In Table 3, we summarize the properties of the four video sequences.

Table 4 lists the means and standard deviations of NMI values generated by different methods with the given parameters. The CHKNN obtains the best segments among the compared methods. In particular, in Video. 3, an NMI value 0.94 is generated by CHKNN, which achieves about 10% improvement compared with the second winner COLL. However, the robustness of DP, Ncut and MEAP are weak, these three methods only obtain one or two good segments on four videos. The NMI values of segments obtained by Graclus and PSVC are both low. The experimental comparison in this section shows that the CHKNN provides an effective tool with application to video clustering.

### 5.5. Experimental results of KNNI

In this subsection, we first analyse the sensitivity of KNNI method to the parameter $M$ and show the KNNI values of clustering results on six synthetic data sets, then compare the proposed KNNI with four internal indices (Davies–Bouldin (DB) index [15], Silhouette (Sil) index [16], Calinski-Harabasz (CH) index [41], Hartigan(Ha) index [42]).

#### 5.5.1. Parameter analysis and results of KNNI

The only one input parameter $M$ of KNNI is a positive integer. The value of $M$ determines the scale that the KNNI use to measure the correlation among the labels of one point and its neighbors. The larger the value of $M$, the more neighbors' labels of one point

**Table 4**
The means and standard deviations (in parentheses) of NMI (over 20 runs) generated by different methods in each video sequence with given parameters.

| Video | Method | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | AP | MEAP | DP | COLL | Kkmeans | Ncut | Graclus | PSVC | CHKNN |
| Video.1 | 0.854(0.000) | 0.549(0.000) | 0.523(0.012) | 0.765(0.009) | 0.721(0.025) | 0.873(0.005) | 0.464(0.002) | 0.643(0.003) | **0.878(0.000)** |
| Video.2 | 0.753(0.000) | 0.611(0.000) | 0.717(0.016) | 0.749(0.007) | 0.734(0.019) | 0.466(0.004) | 0.499(0.005) | 0.524(0.011) | **0.804(0.000)** |
| Video.3 | 0.791(0.000) | 0.700(0.000) | 0.758(0.019) | 0.832(0.006) | 0.745(0.024) | 0.487(0.014) | 0.404(0.009) | 0.406(0.007) | **0.940(0.000)** |
| Video.4 | 0.716(0.000) | 0.439(0.000) | 0.520(0.010) | 0.771(0.004) | 0.767(0.022) | 0.436(0.011) | 0.446(0.007) | 0.532(0.004) | **0.885(0.000)** |

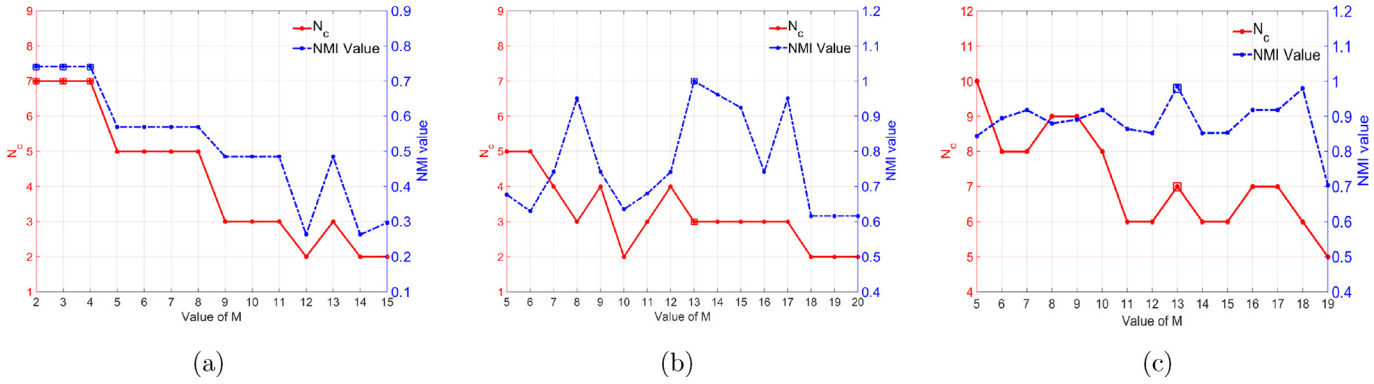



**Fig. 10.** $N_c$ and the corresponding NMI value as functions of $M$ on (a) Glass, (b) DS4_3 and (c) DS5_7. The blue square and the red square mark the highest NMI value and true number of clusters respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 5**
The value of $M$, KNNI value and optimal parameters of each clustering result in Fig. 8.

| Result | $M$ | KNNI value | Parameter | | |
|---|---|---|---|---|---|
| | | | $P_1$ | $P_2$ | $P_3$ |
| Fig. 8 (a) | 23 | 350.0435 | 7 | 15 | 2 |
| Fig. 8 (b) | 16 | 153.8437 | 5 | 6 | 2 |
| Fig. 8 (c) | 6 | 111.2500 | 6 | 8 | 2 |
| Fig. 8 (d) | 7 | 184.2143 | 5 | 5 | 2 |
| Fig. 8 (e) | 13 | 381.0385 | 11 | 11 | 2 |
| Fig. 8 (f) | 10 | 717.8500 | 5 | 5 | 2 |

should be taken into account. However, this does not imply that utilizing larger $M$ is able to obtain better results.

For convenience, we use $N_c$ to denote the number of clusters in the clustering result with highest KNNI value. Fig. 10 depicts $N_c$ and the corresponding NMI value of the clustering result as functions of $M$ on Glass, DS4_3 and DS5_7. As shown in Fig. 10, we can find the trend that the $N_c$ decreases with the increase of $M$. If $M$ is large, the KNNI method will evaluate the validity of the clustering result on a large scale and ignore more details information in local areas. Consequently, the clustering results whose number of clusters are small always get higher KNNI values with a large value of $M$.

For the six synthetic data sets shown in Fig. 4, we first carried out the testing of the CHKNN multiple times by using different parameters and obtained more than 4000 results, the ranges of parameters of the CHKNN were $P_1 \in [4, 20]$, $P_2 \in [5, 28]$, $P_3 \in [1, 15]$, $P_2 \geq P_1 > P_3$. Then we utilized the proposed KNNI method with different values of $M$ to compute the KNNI value of each clustering result. Finally, the results with the highest KNNI value were chosen as the optimal results, which are shown in Fig. 8. Table 5 lists the values of $M$, the optimal parameters and the KNNI values of these clustering results. In practice, the ranges of CHKNN's three parameters can be as large as possible, it will not affect the optimal parameters chosen by KNNI method.

### 5.5.2. Comparison results of internal indices

In Table 6, we present the values of five internal indices (KNNI, DBI, Sil, CH, Ha) and one external index (NMI) of different clustering results shown in Fig. 9. The values of DBI, Sil and CH were generated by a Matlab function termed *evalclusters*.[2] The KNNI values were computed when the value of $M$ is equal to 13.

As shown in Table 6, all the internal indices except KNNI do not choose the result in Fig. 9 (a) whose NMI value is equal to 1 as the optimal result. The majority of the internal indices mentioned in this section just use the inter-cluster and intra-cluster correlation to evaluate the clustering results, but ignore the connectivity among data points. Consequently, they cannot evaluate the validity of clustering results on DS5_7 correctly. As shown in the first row of Table 6, the KNNI values of results in Fig. 9 (b) and (g) are low because these results consist of too many clusters and do not contain any banded clusters. Since the results in Fig. 9 (d), (e) and (h) consist of several circular clusters, their KNNI values are not high, either. The result with the second highest KNNI value in Fig. 9 (f) can only partition parts of points in a band into the same cluster.

Moreover, in Table 7, we summarize the optimal number of clusters obtained by five indices on two real-world data sets ('New Indians, Segment03′ video sequence (Video. 3), 'Winning: Aerospace, Segment01′ video sequence (Video. 4) [40]) and one synthetic data set (DS6_8). As shown in Table 3, the number of 'ground truth' clusters of Video. 3 is 12 and that of Video. 4 is 9. The true cluster number of DS6_8 is 8. For these data sets, only KNNI can find the correct numbers of clusters, while the Ha index has a tendency to overestimate the numbers of clusters and Sil index and DB index tend to underestimate the numbers of clusters.

All the experimental results in Section 5.5 show that the proposed KNNI can be used to evaluate the nonlinear clustering results effectively and outperforms most of the existing internal validity indices.

[2] http://www.mathworks.com/help/stats/evalclusters.html?s_tid=srchtitle.

**Table 6**
Values of internal and external indices of different clustering results on DS5_7. The optimal values generated by different indices are bold.

| Index (optimal) | Clustering result | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Fig. 9a | Fig. 9b | Fig. 9c | Fig. 9d | Fig. 9e | Fig. 9f | Fig. 9g | Fig. 9h |
| KNNI(max) | **381.039** | 239.654 | 276.654 | 310.962 | 310.962 | 359.543 | 248.346 | 323.115 |
| DB(min) | 4.5373 | **0.7108** | 1.1750 | 0.7960 | 0.7960 | 2.5503 | 0.7125 | 0.7939 |
| Sil(max) | 0.0069 | 0.5852 | −0.1809 | 0.5652 | 0.5652 | 0.1161 | **0.5856** | 0.5592 |
| CH(max) | 238.86 | **1412.4** | 141.312 | 998.905 | 998.905 | 340.328 | 1271.6 | 971.355 |
| Ha(min) | 56.948 | **2.058** | 34.489 | 19.776 | 19.712 | 45.525 | 3.446 | 20.255 |
| NMI(max) | **1.000** | 0.4661 | 0.4607 | 0.4708 | 0.4693 | 0.6606 | 0.4843 | 0.4988 |

**Table 7**
The optimal number of clusters generated by five indices and the true number of clusters on three data sets. The parameter $M$ of KNNI on Video. 3 is 4, on Video. 4 is 17, on DS6_8 is 10.

| Index | Data Set | | |
|---|---|---|---|
| | Video. 3 | Video. 4 | DS6_8 |
| DB | 7 | 5 | 11 |
| Sil | 7 | 5 | 4 |
| CH | 11 | 10 | 12 |
| Ha | 17 | 11 | 12 |
| KNNI | 12 | 9 | 8 |
| True | 12 | 9 | 8 |

## 6. Conclusions and future works

In this paper, we propose a novel nonlinear clustering method termed CHKNN based on HKNN graph and an internal validity index KNNI. CHKNN is insensitive to noises and can find clusters correctly on linear and complex nonlinear data sets with appropriate parameters. While the KNNI can help to choose the optimal parameters. Experimental comparisons have been performed on both synthetic and real-world data sets to show the effectiveness of the proposed methods.

The KNNI can avoid the blindness of adjusting the parameters effectively in most cases. However, we find that there is no strict linear correlation between the number of clusters in the clustering result with the highest KNNI value and the input parameter of KNNI. In practice, setting the value of the input parameter of KNNI also needs some experiences and there may be more than one clustering results getting the highest KNNI value. In this case, the optimal clustering result needs to be selected manually.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61573150, 61573152, 61175114 and 91420302. Guangdong innovative project 2013KJCX0009, Guangzhou project 201604016113 and 201604046018. Tip-top Scientific and Technical Innovative Youth Talents of Guangdong special support program (No. 2016TQ03X542).

## References

[1] D. Xu, Y. Tian, A comprehensive survey of clustering algorithms, Annals Data Sci. 2 (2) (2015) 165–193, doi:10.1007/s40745-015-0040-1.
[2] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319, doi:10.1162/089976698300017467.
[3] C.-D. Wang, J.-H. Lai, J.-Y. Zhu, A conscience on-line learning approach for kernel based clustering, in: Proceedings of the 2010 IEEE International Conference on Data Mining, in: ICDM '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 531–540, doi:10.1109/ICDM.2010.57.
[4] C.-D. Wang, J.-H. Lai, J.-Y. Zhu, Conscience online learning: an efficient approach for robust kernel-based clustering, Knowl. Inf. Syst. 31 (1) (2012) 79–104, doi:10.1007/s10115-011-0416-2.
[5] L. Ertöz, M. Steinbach, V. Kumar, Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data., in: Proceedings of the SDM, SIAM, 2003, pp. 47–58.
[6] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[7] C.-D. Wang, J.-H. Lai, J.-Y. Zhu, Graph-based multiprototype competitive learning and its applications, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 42 (6) (2012) 934–946.
[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise., in: Proceedings of the Kdd, 96, 1996, pp. 226–231.
[9] M. Ester, Density-based clustering, in: Encyclopedia of Database Systems, Springer, 2009, pp. 795–799.
[10] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[11] A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, Support vector clustering, J. Mach. Learn. Res. 2 (2002) 125–137.
[12] C.-D. Wang, J. Lai, Position regularized support vector domain description, Pattern Recognit. 46 (3) (2013) 875–884.
[13] M. Liu, X. Jiang, A.C. Kot, A multi-prototype clustering algorithm, Pattern Recognit. 42 (5) (2009) 689–698.
[14] C.-D. Wang, J.-H. Lai, C. Suen, J.-Y. Zhu, Multi-exemplar affinity propagation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (9) (2013) 2223–2237.
[15] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (1979) 224–227.
[16] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[17] S. Saha, S. Bandyopadhyay, A validity index based on connectivity, in: Proceedings of the Seventh International Conference on Advances in Pattern Recognition, 2009, pp. 91–94.
[18] S. Bandyopadhyay, S. Saha, A Validity Index Based on Symmetry: Application to Satellite Image Segmentation, Springer, Berlin Heidelberg, 2013.
[19] B. Rezaee, A cluster validity index for fuzzy clustering, Fuzzy Sets Syst. 161 (23) (2010) 3014–3025.
[20] S.E. Schaeffer, Survey: graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64.
[21] G. Karypis, E.-H. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer 32 (8) (1999) 68–75.
[22] X. Xu, M. Ester, H.P. Kriegel, J. Sander, A distribution-based clustering algorithm for mining in large spatial databases, in: International Conference on Data Engineering, 1998. Proceedings, 2002, pp. 324–331.
[23] P. Franti, O. Virmajoki, V. Hautamaki, Fast agglomerative clustering using a k-nearest neighbor graph., IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1875–1881.
[24] J. Chen, H.R. Fang, Y. Saad, Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection, J. Mach. Learn. Res. 10 (5) (2009) 1989–2012.
[25] H. Huang, Y. Gao, K. Chiew, L. Chen, Q. He, Towards effective and efficient mining of arbitrary shaped clusters, in: Proceedings of the IEEE 30th International Conference on Data Engineering, 2014, pp. 28–39, doi:10.1109/ICDE.2014.6816637.
[26] M.A. Abbas, A.A. Shoukry, Cmune: A clustering using mutual nearest neighbors algorithm, in: Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)„ 2012, pp. 1192–1197, doi:10.1109/ISSPA.2012.6310472.
[27] M.R. Brito, E.L. Chávez, A.J. Quiroz, J.E. Yukich, Connectivity of the mutual k -nearest-neighbor graph in clustering and outlier detection, Stat. Probab. Lett. 35 (1) (1997) 33–42.
[28] Z. Hu, R. Bhatnagar, Clustering algorithm based on mutual k-nearest neighbor relationships, Stat. Anal. Data Min. 5 (2) (2012) 100–113, doi:10.1002/sam.10149.
[29] D. Sardana, R. Bhatnagar, Graph clustering using mutual K-nearest neighbors, 2014.
[30] G. Beliakov, M. King, Density based fuzzy c -means clustering of non-convex patterns, Eur. J. Oper. Res. 173 (3) (2006) 717–728.

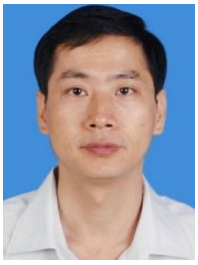
[31] I. Gath, A.S. Iskoz, B.V. Cutsem, Data induced metric and fuzzy clustering of non-convex patterns of arbitrary shape, Pattern Recognit. Lett. 18 (6) (1997) 541–553.
[32] M. Ankerst, M.M. Breunig, H.P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, ACM Sigmod Record 28 (2) (1999) 49–60.
[33] D. Birant, A. Kut, ST-DBSCAN: an algorithm for clustering spatial-temporal data, Data Knowl. Eng. 60 (1) (2007) 208–221.
[34] A. Asuncion, D. Newman, Uci machine learning repository, 2007.
[35] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[36] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554.
[37] A. Strehl, J. Ghosh, Cluster ensembles: a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2003) 583–617.
[38] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[39] I.S. Dhillon, Y. Guan, B. Kulis, Weighted graph cuts without eigenvectors a multilevel approach, IEEE Trans. Pattern Anal. Mach. Intell. 29 (11) (2007) 1944–1957.
[40] C. Hill, Open video project, 2007URL: http://www.open-video.org.
[41] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat-Theory Methods 3 (1) (1974) 1–27.
[42] J.A. Hartigan, Clustering Algorithms, Appl. Stat. 23 (1975) 38–41.

**Yikun Qin** is currently a Master student in pattern recognition and intelligent systems at the South China University of Technology, Guangzhou, China. His research interests include the pattern recognition and machine learning.

**Zhu Liang Yu** received his BSEE in 1995 and MSEE in 1998, both in electronic engineering from the Nanjing University of Aeronautics and Astronautics, China. He received his Ph.D. in 2006 from Nanyang Technological University, Singapore. He joined Center for Signal Processing, Nanyang Technological University from 2000 as a research engineer, then as a Group Leader from 2001. In 2008, he joined the College of Automation Science and Engineering, South China University of Technology and was promoted to be a full professor in 2011. His research interests include signal processing, pattern recognition, machine learning and their applications in communications, biomedical engineering, etc.

**Chang-Dong Wang** received his Ph.D. degree in computer science in 2013 from Sun Yat-sen University, China. He is currently an assistant professor at School of Mobile Information Engineering, Sun Yat-sen University. His current research interests include machine learning and pattern recognition, especially focusing on data clustering and its applications. He has published over 30 scientific papers in international journals and conferences such as IEEE TPAMI, IEEE TKDE, IEEE TSMC-C, Pattern Recognition, Knowledge and Information System, Neurocomputing, ICDM and SDM. His ICDM 2010 paper won the Honorable Mention for Best Research Paper Awards. He won 2012 Microsoft Research Fellowship Nomination Award. He was awarded 2015 Chinese Association for Artificial Intelligence (CAAI) Outstanding Dissertation.

**Zhenghui Gu** received the Ph.D. degree from Nanyang Technological University in 2003. From 2002 to 2008, she was with Institute for Infocomm Research, Singapore. She joined the College of Automation Science and Engineering, South China University of Technology, in 2009 as an associate professor. She was promoted to be a full professor in 2015. Her research interests include the fields of signal processing and pattern recognition.



**Yuanqing Li** was born in Hunan Province, China, in 1966. He received the B.S. degree in applied mathematics from Wuhan University, Wuhan, China, in 1988, the M.S. degree in applied mathematics from South China Normal University, Guangzhou, China, in 1994, and the Ph.D. degree in control theory and applications from South China University of Technology, Guangzhou, China, in 1997. Since 1997, he has been with South China University of Technology, where he became a full professor in 2004. In 2002-04, he worked at the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama, Japan, as a researcher. In 2004-08, he worked at the Laboratory for Neural Signal Processing, Institute for Inforcomm Research, Singapore, as a research scientist. His research interests include, blind signal processing, sparse representation, machine learning, brain-computer interface, EEG and fMRI data analysis. He is the author or coauthor of more than 60 scientific papers in journals and conference proceedings.