Pattern Recognition 43 (2010) 3753–3767

Contents lists available at ScienceDirect

Pattern Recognition

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

A tree-structured framework for purifying "complex" clusters with structural roles of individual data

Jundi Ding [a,b], Runing Ma [c], Jingyu Yang [a], Songcan Chen [b,*]

[a] School of Computer Science and Technology, Nanjing University of Science and Technology, China
[b] Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, China
[c] Department of Science, Nanjing University of Aeronautics and Astronautics, China

ARTICLE INFO

Article history:
Received 8 February 2009
Received in revised form
20 May 2010
Accepted 21 May 2010

Keywords:
Clustering
Image segmentation
k-neighborhood
Reverse *k*-neighborhood
Structural objects
Symmetric neighborhood

ABSTRACT

How can we find a *natural* clustering of a "complex" dataset, which may contain an unknown number of overlapping clusters of arbitrary shape and be contaminated by noise? A tree-structured framework is proposed in this paper to purify such clusters by exploring the structural role of each data. In practice, each individual object within the internal organization of the data has its own specific role—"centroid", hub or outlier—due to distinctive associations with their respective neighbors. Adjacent centroids always interact on each other and serve as mediate nodes of one tree being members of some cluster. Hubs closed to some centroid become leaf nodes responsible for the termination of the growth of trees. Outliers that weakly touch with any centroid are often discarded from any trees as global noise. All the data can thus be labeled by a specified criterion of "centroids"-connected structural consistency (CCSC). Free of domain-specific information, our framework with CCSC could widely adapt to many clustering-related applications. Theoretical and experimental contributions both confirm that our framework is easy to interpret and implement, efficient and effective in "complex" clustering.

1. Introduction

Nowadays and more than ever, a flood of unlabeled data is pouring out of many modern science fields thanks to the rapid development of technology. The task of clustering has long been known as a fundamental but central step in data analysis [1–5]. Many well-studied approaches often explicitly or implicitly utilize the measures of point similarity to seek a *natural* clustering of a given dataset. They have demonstrated impressive performances on well-separated compact clusters where objects within the same group are similar to a common center [6]. However, the complex data just as illustrated in Fig. 1 seems to go beyond their clustering capability. In such "complex" data, the hidden clusters could be with arbitrary shapes, sizes and densities; moreover they may be overlapping or contaminated by noise. Consequently, the inter-region similarity is always larger than the intra-region similarity.

Notably, most of modern data available from our daily life is just complex as such. Images should be one of the most representative examples, in which objects of interest (OOI) usually exhibit great variations in category, position, shape, pose and size. Finding meaningful regions to represent OOI is significantly important for a better understanding of these images. It is a very pixel clustering process, but different objects may have the very low contrast in gray, color or texture; while the same object would be with the high discrepancy. Image pixels are not simply shaped in the form of point clouds and instead expose noisy manifold structures [7] challenging to many popular clustering-based segmentation algorithms [8].

In the scope of this work, we attempt to build a tree-structured clustering framework for addressing this difficult issue. Not to precisely find every single cluster in the data, the main purpose is to purify all salient clusters that are relatively meaningful to human perception from the complex data. To realize this goal, we explore a common trait shared in many types of datasets: *each individual object has its own structural role within the internal organization of the data*. Fig. 2 explicitly illustrates such an intuition on a toy network data with three kinds of vertices in the perspective of network researchers [9]. Respectively, they act as members of clusters like vertex 5 or 11 that has many dense links with other members of the same cluster; hubs like 6 that bridges many sparsely interconnected neighbors; and outliers like 13 with only a weak connection to the network. They arguably play different specific roles in such network structures [9].

After a closer look, it is not difficult to observe that vertex 5 or 11, just like a local centroid of one cluster, clearly occupies the center position of a mass of associated objects; hub vertex 6 is near the common boundaries of different clusters; and outlier

* Corresponding author.
E-mail address: s.chen@nuaa.edu.cn (S. Chen).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.05.031

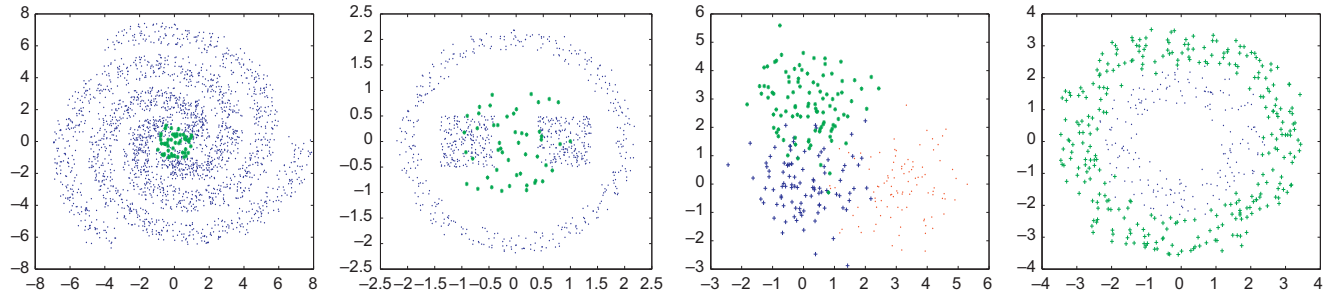

Fig. 1. "Complex" toy sets of 2-dimensional points in vector space. The hidden clusters could be with arbitrary shape, size and density; and also they may be contaminated by noise or overlapping.

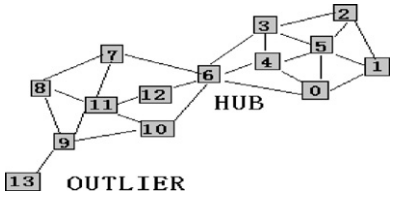

Fig. 2. Three structural objects in a toy network data: members of cluster like vertex 5, hubs like vertex 6 and outliers like vertex 13.

vertex 13 is far away from all clusters. They seem to just differ in terms of their spatial locations in the data. Furthermore, there are only the three varied positions available for each object to locate in types of data. In clustering, objects distributed at the three kinds of spatial positions perform like "centroids",[1] hubs and outliers, respectively, which are thus stated as three sensibly structural objects in the data. Note, for example, the three structural objects of a noisy toy set in Fig. 3(a) are presented in Fig. 3(b). As expected, "centroids" signed by "*" occupy the interior areas of two circle-like clusters; hubs signed by "◦" appear in the boundaries of two circle-like clusters; and outliers signed by "△" just are the sparse noise.

With no doubt distinguishing different structural objects is very helpful for cluster detection in "complex" data. Intuitively, "centroids" could be responsible for creating new clusters, hubs for terminating a cluster, whereas outliers may be noise useless in clustering (see ones in Fig. 3(c) signed with "◦"). But how can we discriminate and find these structural objects? As discussed above, they have different degrees of associations with their respective neighbors. In consequence, a reliable way is to take into account both neighborhood and reverse neighborhood among data (note, neighbor relationship between objects is not symmetric). Given $k \geq 1$ ($k \in \mathbb{Z}$), consider the normal k-nearest-neighbors of all objects in a dataset. Then the number of elements in k-neighborhood of every object would be around k, but the numbers of reverse k-neighbors of different objects quite discrepant. Not surprisingly, "centroids" would have reverse k-neighbors more than k, whereas hubs about k and outliers much less than k. So different structural objects will have different ratios of the number of reverse k-neighbors to the number of k-neighbors. This ratio of each object here is just exploited to judge which structural role—"centroid", hub or outlier—it plays. In practice we are not the first to spell this ratio out explicitly. Similar ideas, e.g., neighborhood density factor (NDF) and neighborhood density index (NDI), have appeared in NBC (neighborhood-based clustering) [10] and its successful follow-ups with varied applications [11,12]. Those notions are merely used to determine which points, pixels or intensities are dense, even or sparse.

Careful considerations above indicate that "centroid" objects are reasonably in charge of the generation of new clusters. However, here "centroid" is distinct from the existing term *center*, *mean* or *exemplar* that serve as the common center of some subset of data in many popular clustering methods [13–17]. As is well known, these methods strive toward a partition that minimizes a so-called squared error between objects and a preferred cluster center. Usually, to find a good solution, some need rerun many times with different initializations, while others have to be iterated recursively starting with all data objects as potential exemplars. Such behaviors appear a little random, time consuming and impractical for "complex" cases where true centers of clusters may be difficult to reach. To this end, our work makes no effort to seek a plausibly unique centroid for each cluster, and instead argues a pertinently *structural consistency* among symmetric neighbors of common "centroid" objects to be true. The only assumption is that each "centroid" object should belong to a certain *natural* cluster. And then, all the data can be labeled by an introduced principle of "centroids"-connected structural consistency (CCSC). Specifically, objects associated with common "centroids" would fall into the same cluster as "centroids" themselves; namely, cluster label of each data changes with local properties of adjoining "centroids".

On the other hand, it is convenient to depict the relationship of structural consistency between objects via a tree-structured chain: "*centroid*" *objects engage to be interior nodes*, *while hub objects shoulder the parts of leaf nodes*. More importantly, the "centroid" objects bound over to CCSC strictly constitute an equivalence class in mathematics and will equivalently grow onto the same tree. That is, with respect to a single parameter k, CCSC gives an equivalence relation on the detected set of "centroids". By fundamental partition theorem—"an equivalence relation on a set X partitions X" [18], we can conclude that an arbitrary input dataset is separable under this equivalence relation on the "centroid" set. The number of equivalence classes found on the "centroid" set is just the final number of desired clusters: an equivalence class of "centroids" gives rise to a cluster. In addition, the properties of reflexivity, symmetry and transitivity make the clustering results robust to the order of "centroid" object selection.

Free of domain-specific knowledge, our framework along with CCSC can adapt to many applications provided that the data-dependent relationships of neighborhood and reverse neighborhood are appropriately built. This very data-dependent hallmark has offered us a free space for modifying our framework to more "complex" clustering applications. Here we concentrate on two general cases: pairwise data clustering and image segmentation

[1] Note that the quotation mark "." is here used to emphasize a difference from the usual term *centroid* appearing in machine learning literature, which typically refers to a representative of a cluster.

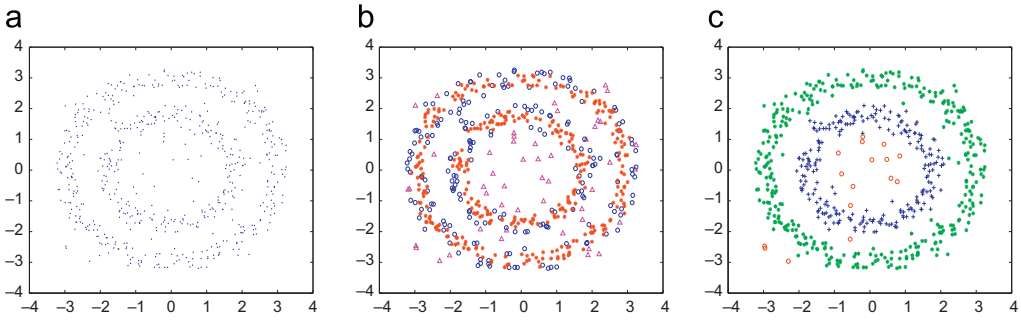

Fig. 3. Structural objects (b) in a noisy set of two circled clusters (a): centroids "*", hubs "◦" and outliers "△" . Based on the identified structural objects, our framework with CCSC successfully figures out the two salient circled clusters (c).

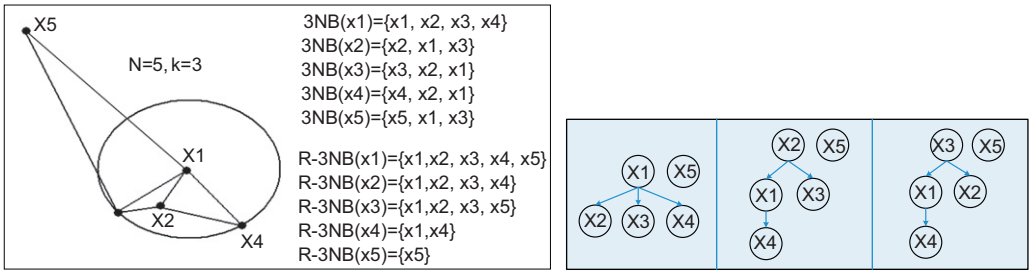

Fig. 4. k NB and R-k NB of objects in a simple data (Left). Our framework with CCSC is robust to the order of centroid selection: the clustering results are identical despite building three different directed trees $T_3(x\ 1)$, $T_3(x\ 2)$ and $T_3(x\ 3)$ (Right).

for a preliminary illustration. Adopting an algorithmic configuration similar to previous work in [12], each data is scanned only once throughout the whole clustering process. Thus, the computational complexity in clustering is linear in the size of the input dataset. The remainder of this paper is organized as follows. Section 2 presents the tree-structured clustering framework as well as its principled analysis after a brief review of some related knowledge. Section 3 details how our framework can flexibly be applied to purify salient clusters or regions in "complex" datasets from both theoretical and experimental views. Finally, the whole paper is summarized in Section 4. Loosely speaking, our tree-structured framework, although not all-purpose, should be a good complement to the existing clustering methods for various "complex" clustering-related problems.

2. Tree-structured clustering framework

In this section, it is to build a tree-structured clustering framework. The key is to identify the three structural objects in the dataset. Previous analysis tells that it relates to the degree of associations between each object and its local neighbors. We hence first use the available data neighborhood information to introduce a structural role index (SRI). Different structural objects will have different values of SRI. The related basic concepts will be detailed in order as follows.

2.1. Structural roles of data

SRI involves the traditional notions about k-nearest-neighbor (k NN), k-neighborhood (k NB) and reverse k-neighborhood (R-k NB). Similar definitions are already abundant in the literature [10–12]. We again put them here to facilitate readers to better understand the implied idea of our clustering framework. Let $X=\{x_1,x_2,\ldots,x_N\}$ be a dataset to be clustered with its size N. Suppose a distance measure between objects in X is given, denoted as $dist(\cdot,\cdot)$. Then the set of k ($k>0$) nearest neighbors of x ($x\in X$) is denoted by k NN(x). With these notational preliminaries, we are now able to concisely redefine the k NB and R-k NB of an arbitrary object x (if interested, readers can refer to original descriptions in NBC [10]).

Definition 1 (k-*neighborhood*). The k-neighborhood of x (k NB(x)) is a set of objects that lie within a circle region with x as the center and r as the radius. In mathematics, $k\mathrm{NB}(x)=\{y: y\in k\mathrm{NN}(x), dist(x,y)\le r\}$ and $r=\max_{o\in k\mathrm{NN}(x)}\{dist(x,o)\}$.

Definition 2 (*Reverse k-neighborhood*). The reverse k-neighborhood of x (R-k NB(x)) is just the set of objects whose k NBs contain x itself. Mathematically, $\mathrm{R}{-}k\mathrm{NB}(x)=\{y: x\in k\mathrm{NB}(y), y\in X\}$.

Clearly, k NB(x) and R-k NB(x) expose the relationship between x and its neighbors in a two-way fashion. k NB(x) describes who make up of its own k-nearest-neighbors, while R-k NB(x) indicates whose k-neighborhood x belongs to. It is evident that for most data objects $|k\mathrm{NB}(x)|$ ($|.|$ denotes the cardinate of a set) is always around k, not less than k but may be a little greater than k because more than one object could locate on the circle region edge covered by k NN(x). An extreme case is shown in Fig. 4 where $dist(x_1,x_3)=dist(x_1,x_4)$ (Euclidean metric), so $3\mathrm{NB}(x\ 1)=\{x\ 1,x\ 2,x\ 3,x\ 4\}$ and $|3\mathrm{NB}(x1)|=4>k=3$. Remaining objects all have $|3\mathrm{NB}|$ equal to three.

On the contrary, the values of $|\mathrm{R}{-}k\mathrm{NB}(x)|$ are quite discrepant for different data objects. For example, the numbers of R-3NBs of five objects in Fig. 4 are, respectively, 5, 4, 4, 2 and 1. Intuitively, the larger $|\mathrm{R}{-}k\mathrm{NB}(x)|$ is, the more other objects approach x. It implies that with high probability x may be a centroid of those objects who take x as a member of their k NBs. By contrast, the smaller $|\mathrm{R}{-}k\mathrm{NB}(x)|$ is, the more likely x is an outlier. Let us consider this point in a reverse manner. If x is indeed an outlier who is far away from almost all objects in the data, few even no objects will take x as their k NBs except from itself. For such a situation, $|\mathrm{R}{-}k\mathrm{NB}(x)|$ will approximate one by itself (a reachable minimum of $|\mathrm{R}{-}k\mathrm{NB}(x)|$). Besides, hub objects on boundary areas of clusters would have in-between values of $|\mathrm{R}{-}k\mathrm{NB}(x)|$ a little less than $|k\mathrm{NB}(x)|$ but larger than zero. In what follows, we will

introduce a structural role index to distinguish different structural objects in a quantitative manner.

Definition 3 (*Structural role index*). The neighborhood-based structural role index of an arbitrary data x, denoted by $\mathrm{SRI}(x)$, is evaluated as

$$\mathrm{SRI}(x) = \frac{|\mathrm{R}{-}k\mathrm{NB}(x)|}{|k\mathrm{NB}(x)| + |\mathrm{R}{-}k\mathrm{NB}(x)|} \tag{1}$$

By definition it is clear that $\mathrm{SRI}(x) \in (0,1)$ for $\forall x \in X$. Fig. 5 vividly presents the SRI values of all points in the noisy set of two circle-like clusters in Fig. 3(a). As opposed to the unbound NDF, namely, the ratio of $|\mathrm{R}{-}k\mathrm{NB}|$ to $|k\mathrm{NB}|$ used in NBC [10], SRI virtually quantifies the two-way relationships between an arbitrary data and its neighbors in a normalized fashion. As can be seen, each data has its own SRI value within the interval from zero to one. Hence SRI is supposed to more clearly, precisely picture both local and global objects' positions in the dataset. In terms of a given k, almost all objects in the dataset have $|k\mathrm{NB}(x)|$ values invariant around k. Thus it is very interesting to note the change of $|\mathrm{R}{-}k\mathrm{NB}(x)|$ of different objects. As a rule, for any object data, its R-k NBs might be equal to, more or less than its k NBs (see back to the simple example in Fig. 4 for reference). Its SRI value by definition would just equate, be larger or smaller than 0.5 correspondingly. With aforementioned discussions, we know that centroid (outlier) candidates would possess the most (least) reverse k NNs, and hubs' reverse k NNs would approximate but still be a little less than the number of its k NNs. In result, the structural roles of data can now be discriminated in the following way.

Definition 4 (*Centroid, hub, outlier*). For an arbitrary object x, x is called a centroid if $\mathrm{SRI}(x) \geq 0.5$; a hub if $\mathrm{SRI}(x) \approx$ but < 0.5 (i.e., close to but still < 0.5); an outlier otherwise. Specifically, the set of centroid objects is denoted as $\mathrm{CENTROID} = \{x : \mathrm{SRI}(x) \geq 0.5, x \in X\}$.

Interestingly, it is worthwhile to note twofold aspects here. One is that structural points, i.e., hubs and outliers are specified in a very rough manner. Both satisfy $\mathrm{SRI}(x) < 0.5$, so a suitable threshold seems to be necessary for eliminating the vagueness between them. For the example in Fig. 3(a), 0.4 is a desirable threshold since most data points on two circles have SRI values larger than 0.4 (see Fig. 5(b), the first 500 points). As a consequence, Fig. 3(b) shows hubs signed by "◦" with $0.4 \leq \mathrm{SRI}(x) < 0.5$ and outliers signed by "△" with $0 < \mathrm{SRI}(x) < 0.4$ representing most of addition noise. Nevertheless, how to determine an appropriate threshold is a trivially application-driven and data-dependent work. A fortunate fact is that our tree-structured framework exclusively relies on the definite CENTROID set distinguished from the whole data. It is just the other aspect we are to highlight here. Indeed, see detailed below, the data with $\mathrm{SRI}(x) < 0.5$ will be either absorbed into some cluster of "centroids" or isolated from any clusters. It turns out that the isolated data with $\mathrm{SRI}(x) < 0.5$ may be the really global outliers or noise in the data like ones in Fig. 3(c) signed by "◦". Considering this, it is unnecessary to seek a desirable threshold to separate the two structural objects, i.e., hubs and outliers in advance. They do not affect the task of cluster generation and in turn could be identified during clustering in our framework.

2.2. Centroids-connected structural consistency

In the literature, many attempts have already been done to work around clustering problem [6,13–16,19]. As a coincidence, most of them intend to find groups of mutually close points with respect to a so-called k-nearest-neighbor consistency: "*For any data object in a cluster, its k-nearest-neighbors should also be in the same cluster*" [20]. In practice, k NN consistency really provides us a useful and meaningful measure for evaluating qualities of clusters. This satisfies a "local consistency" where nearby points are allowed with the same cluster label [21]. However, it cannot ensure "global consistency" simultaneously [21]. The result is that points on the same structure (typically referred to as a cluster) often have the different cluster labels.

Note that not all persons are of the same importance to everyone in the human cognitive system. Moreover, two intimate friends should very likely be members of the same community, and with high probability their respective other intimate friends might also belong to this community. Motivated by such observations, we are in this section to state a CCSC formally as below.

CCSC: For any centroid object x in a current cluster, its symmetric neighbors in $\mathrm{SN}(x) = k\mathrm{NB}(x) \cap \mathrm{R}{-}k\mathrm{NB}(x)$ are structural consistent with x and should be connected into the same cluster as x.

To our surprise, CCSC defines a "transitive relationship". Assume $x,y,z \in \mathrm{CENTROID}$, if $z \in \mathrm{SN}(y)$ and $y \in \mathrm{SN}(x)$, z and y together with their symmetric neighbors are, respectively, connected to the same cluster as y and x. Obviously, z together with symmetric neighbors have already been connected to the same region as x. It is thus said that CCSC specifies an equivalence relation on the set of centroid objects.

Lemma 1 (*Equivalence relation* "$\sim$"). *For* $\forall x,y \in \mathrm{CENTROID}$, "$x \sim y$" *if either of the following conditions holds*: (1) x *is one of*

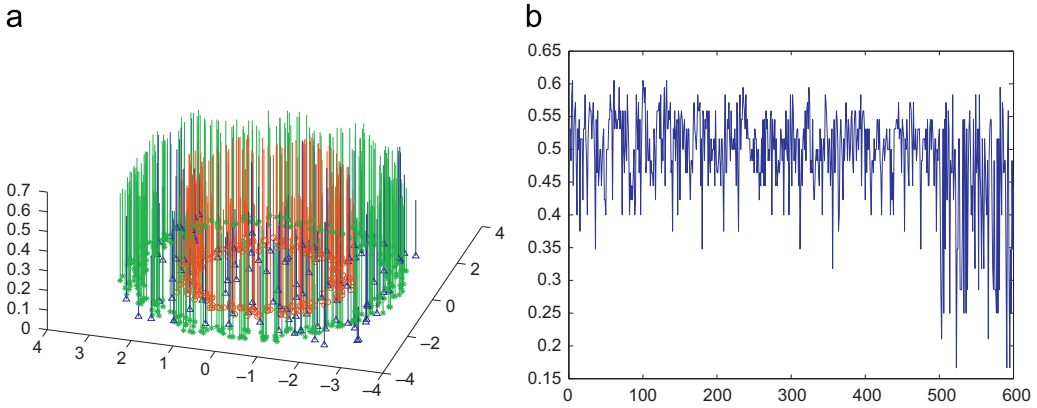

Fig. 5. SRI values of all points in the noisy toy data shown in Fig. 3(a). They are quite discrepant from each other: (a) 3-dimensional sketch map of SRI values of points: inside circle "◦", outside circle "*" and additive noise "△"; (b) corresponding graphic drawing of SRI values (vertical axis). From left to right, the numbers on horizontal axis index 200 points of inside circle, 300 points of outside circle and 100 points of additive noise.

symmetric neighbors of y, i.e., $x \in \mathrm{SN}(y)$; (2) *there exists a series of objects* $x_1,\ldots,x_n$ *such that* $x \in \mathrm{SN}(x_1), x_k \in \mathrm{SN}(x_{k+1}), x_n \in \mathrm{SN}(y)$ *and* $k=1,\ldots,n-1$.

Proof. Here it is just to see whether CCSC can satisfy properties of reflexivity, symmetry and transitivity on CENTROID:

(1) *Reflexivity*: $\forall x \in \mathrm{CENTROID}$, $x \sim x$. Recall that k NB(x) is a set of objects within a circle region where x itself is the center. It implies $x \in k\mathrm{NB}(x)$, $x \in \mathrm{R}-k\mathrm{NB}(x)$; and thereby $x \in \mathrm{SN}(x)$. So x is coherent with itself $x \sim x$.
(2) *Symmetry*: $\forall x,y \in \mathrm{CENTROID}$, $y \sim x \Leftrightarrow x \sim y$. By the nature of the definition of SN(x), $y \in \mathrm{SN}(x)$ just means $x \in \mathrm{SN}(y)$. According to the first case in this lemma, we know that the symmetric holds true.
(3) *Transitivity*: $\forall x,y,z \in \mathrm{CENTROID}$, if $z \sim y$ and $y \sim x$, then $z \sim x$. See the above discussions about the "transitive relationship". □

Proposition 1 (*Separability of data*). *CCSC with respect to a single parameter* $k \in (0,N)$ *can ensure that an arbitrary dataset with the size* $N(>0)$ *is always separable*.

Proof. For an arbitrary dataset, we can always obtain a nonempty set of centroid objects given any $k \in (0,N)$. According to partition theorem in [18], the equivalence relation "$\sim$" on CENTROID in Lemma 1 always divides CENTROID into several nonempty equivalence classes, forming a partition of CENTROID. For the remaining non-centroid objects, some will be assigned into equivalence classes of their associated centroids by CCSC, while others will belong to a trivial cluster of theirs own. Loosely speaking, our CCSC has definitely grouped every object into a certain cluster. This just suggests the given dataset is separable with respect to $k \in (0,N)$. □

By now, we are ready to explicitly identify the centroids-connected clusters, centroids-connected hubs and centroids-disconnected outliers from the "partition" of data yielded in Proposition 1 by our CCSC.

Definition 5 (*Centroids-connected cluster*). With respect to $k \in (0,N)$, a nonempty subset $C \subseteq X$ is called a centroids-connected cluster, if all objects in C are consistent and structural connected with a common centroid, simply denoted as $\mathrm{CC}_k(C)$.

As analyzed in the proof of Proposition 1, each object x either belongs to a $\mathrm{CC}_k(C)$, or is isolated from any $\mathrm{CC}_k(C)$s. If x is a member of a $\mathrm{CC}_k(C)$, it is either a centroid or a centroids-connected hub, completely depending on its SRI value. Otherwise, x is a centroids-disconnected outlier.

Definition 6 (*Centroids-connected hub*). With respect to $k \in (0,N)$, an object $x \in X$ is called a centroids-connected hub if x is a member of a $\mathrm{CC}_k(C)$ and with $\mathrm{SRI}(x) < 0.5$, simply denoted as $\mathrm{CC}_k(H)$.

Definition 7 (*Centroids-disconnected outlier*). With respect to $k \in (0,N)$, an object $x \in X$ is called a centroids-disconnected outlier if x is disconnected with any centroid and thus isolated from any $\mathrm{CC}_k(C)$s, simply denoted as $\mathrm{CD}_k(O)$.

Identifying hubs and outliers is very helpful in some applications. Note, for instance, hubs on common boundaries of clusters are apt to spread ideas or disease in viral marketing or epidemiology [9], while outliers may be essential in novelty (instruction) detection. With nothing domain-specific, our CCSC applied to these related domains is straightforward in the future. Just as for purifying salient clusters here, it is sufficient to only confirm the centroid members in the data.

2.3. A tree-structured framework

Computationally, it is useful to think of clustering with the concepts in graph theory. The hierarchical natures facilitate the further analysis and processing of the data [8,17]. The unified clustering framework present in this section specially builds upon some tree-related graph notions. It aims at representing every $\mathrm{CC}_k(C)$ into a tree-based structure (NODE, BRANCH), where NODE is the set of nodes corresponding to members of $\mathrm{CC}_k(C)$, and BRANCH is the set of oriented branches connecting all members of $\mathrm{CC}_k(C)$. For simplicity, this tree is denoted by $\mathrm{T}_k(C)$. The orientation of branches conveys the structural consistency relation between parent and child nodes. Given $x_1,x_2 \in X$, $(x_1,x_2) \in \mathrm{BRANCH}$ means that (1) $x_1 \in \mathrm{CENTROID}$ as a parent node; (2) $x_2 \in \mathrm{SN}(x_1)$ as a child node of x_1. Note that x_2 can be either a hub or a centroid node structural consistent with x_1. If x_2 is a hub, it will be a leaf node of one $\mathrm{T}_k(C)$. If x_2 is a centroid, it will be used to continue spanning the current tree $\mathrm{T}_k(C)$.

Table 1 gives our unified tree-structured framework, where all $\mathrm{CC}_k(C)$s are found in Step2. By the property of transitivity, objects associated with a common centroid x are consistently assembled onto a $\mathrm{T}_k(x)$. Its nodes are hence grouped into a $\mathrm{CC}_k(C)$ all at once. This process continues in a similar manner until no centroid remains to be used for creating a new $\mathrm{T}_k(x)$. The objects not in any $\mathrm{T}_k(x)$s are determined as $\mathrm{CD}_k(O)$s in Step 3. The whole process for finding all $\mathrm{CC}_k(C)$s and $\mathrm{CD}_k(O)$s takes $O(N)$ because wherein each data object is scanned once. In terms of a pure clustering procedure, a linear computational complexity in the size of data is rather desirable.

Independently, Step 1 is to calculate SRI values of all objects in X. It is necessary to discover the neighborhood relations among data, i.e., k NB(x) and R-k NB(x). If the data are already represented by a matrix gathering pairwise proximities (pairwise data), it only needs searching for the respective neighbors of objects with a linear time $O(N)$. Otherwise, it is compulsory to first spend time about $O(N^2)$ in computing pairwise distances for all clustering (classification) methods. Of course, many sophisticated technologies can be taken to speed up this calculation [10]. Considering these circumstances, we may fairly evaluate efficiency of computational approaches regardless of the calculation of pairwise distances between data. With this sense, we can roughly say that the unified tree-structured framework is efficient with a linear computational complexity $O(N)$ in clustering.

In addition, the following lemmas are important for validating the correctness of our clustering framework.

Lemma 2. *Let* $x \in X, k \in (0,N)$. *If x is a centroid, then the set of nodes in* $\mathrm{T}_k(x)$ *is a centroids-connected cluster*.

Proof. It is obvious that our framework builds every $\mathrm{T}_k(x)$ in a two-step way. First, pick $\forall x \in \mathrm{CENTROID}$ (it is not empty when

Table 1
A unified tree-structured framework.

Given $X=\{x_1,\ldots,x_N\}, k \in (0,N)$ and $dist(\cdot,\cdot)$
1. Calculate SRI(X) to gain CENTROID and SN(X)
2. While CENTROID $\neq \emptyset$
   Pick $x \in$ CENTROID as a root parent to build $\mathrm{T}_k(x)$
   $\mathrm{T}_k(x)=\mathrm{SN}(x)$ and $Q=\mathrm{CENTROID} \cap \mathrm{SN}(x)$
   While $Q \neq \emptyset$, $\mathrm{T}_k(x)=\mathrm{T}_k(x) \cup (\cup_{y \in Q}\mathrm{SN}(y))$, y as intermediate parent to span $\mathrm{T}_k(x)$
      $Q=\mathrm{CENTROID} \cap (\cup_{y \in Q}\mathrm{SN}(y))$
      $\mathrm{CENTROID}=\mathrm{CENTROID} \backslash (\mathrm{CENTROID} \cap \mathrm{T}_k(x))$
   End
   Output $\mathrm{CC}_k(C)=\mathrm{T}_k(x)$
End
3. Output $\mathrm{CD}_k(O)=X \backslash (\cup \mathrm{T}_k(x))$

$k \in (0,N)$) as a root to grow a $T_k(x)$. Second, retrieve all the objects which are structural consistent with x in the principle of CCSC to serve as children nodes of x. By Definition 5, x with its all children nodes forms a nonempty centroids-connected cluster. □

From Lemma 1, we can know that CCSC in fact defines an equivalence relation '$\sim$' on the nonempty set of CENTROID. The properties of transitivity and symmetry indicate that every $CC_k(C)$ is uniquely determined by any of its centroid objects. This implies that each object in one $CC_k(C)$ is structural consistent with any of the centroids of $CC_k(C)$. Namely, $CC_k(C)$ is exactly made up of the objects which are structural consistent with an arbitrary centroid of $CC_k(C)$. Lemma 4 therefore holds true.

Lemma 3. *Let $C \subseteq X$ be a $CC_k(C)$, $x \in CC_k(C)$ be an arbitrary centroid. Then $CC_k(C)$ equals the set of nodes in $T_k(x)$ grown from x.*

It is said that the final clustering consisting of all $CC_k(C)$s and $CC_k(O)$ is robust to the selection of an initial centroid object. In the simple set given in Fig. 4, we have CENTROID=$\{x\ 1, x\ 2, x\ 3\}$ by Definition 4. Each of them is picked in order as an initial root node, resulting in three seemingly different directed trees $T_3(x\ 1)$, $T_3(x\ 2)$ and $T_3(x\ 3)$ under our tree-structured framework, as illustrated in Fig. 4 (right); but in clustering the results are identical, i.e., $CC_{k=3}(C)=\{x\ 1, x\ 2, x\ 3, x\ 4\}$ and $CC_{k=3}(O)=\{x\ 5\}$. And also, $CD_{k=3}(H)=\{x\ 4\}$ because $x4 \in CC_{k=3}(C)$ but $SRI(x4) = 0.4 < 0.5$.

3. Applications of tree-structured framework

Our tree-structured framework in this section will be applied to deal with two general cluster-related problems. One is to discover the intrinsic structures hidden in the sets of 2-dimensional points in vector space, especially attending to the detection of clusters as "complex" as shown in Fig. 1, which are rather noisy or overlapping, i.e., not well-separated. The other is to find sematic objects in a set of challenging images by partitioning those image pixels in grid space into several meaningful regions homogeneous in intensity. Each exposes a distinct data-dependent complexity. Humans may use prior knowledge aided by memory to accomplish such tasks. Our framework has been to find out how far clustering or segmentation can reach using the intuition-driven CCSC only.

3.1. Data clustering

Remember that our ultimate goal is to seek groups of points that are as similar as possible. In terms of the complex clusters to be detected in Fig. 1, the normal Euclidean distance cannot reveal "genuine" pairwise proximities between them. The fact is that Euclidean neighborhood relationships often produce undesirable clustering results even for the well-separated manifold clusters [12,14,17,22,23,25,26], let alone for those contaminated by noise or lapped over each other.

3.1.1. Polynomial kernel induced distance

One way to avoid such fallible similarity estimates is to adopt a kernel-based idea suggested in recent work [15,16]. Projected objects in the implicit higher-dimensional feature space by a kernel function $k(x,y)$ would become linearly separable [27]. Nevertheless, any kernel function cannot guarantee to be versatile in all circumstances. Different kernels present different basic kernel functionality. For "complex" cluster detection at hand, it is not advisable to select those kernels with a shift-invariant property like radial basis Gaussian kernel [14,16,26]. Here what is really of our interest is those kernels that can yield subtle distinctions on k NBs and R-k NBs of objects in feature space. The traditional polynomial kernel $\mathrm{Poly}_n(x,y)=(dot(x,y)+a)^n$ is just an admissible choice, where $a=1$ and $dot(\cdot,\cdot)$ denotes the dot product. Then, a polynomial kernel induced distance (PKID) with rank $n>0$ is defined as:

Definition 8 (*PKID*).
$dist^n_{Pk}(x,y) = \sqrt{\mathrm{Poly}_n(x,x)+\mathrm{Poly}_n(y,y)-2\mathrm{Poly}_n(x,y)}$.

Apparently, when $n=1$ $dist^n_{Pk}(x,y)$ just becomes Euclidean distance, and moreover it is not shift invariant when $n>2$. As an illustration, let us pick two points in an Euclidean space $x_1=(0,0)$, $y_1=(1,0)$ and shift them along the only x-axis with 10-unit Euclidean distance to obtain $x_2=(0,10)$, $y_2=(1,10)$. Clearly, $dist^{n=1}_{Pk}(x_1,y_1) = dist^{n=1}_{Pk}(x_2,y_2) = 1$; while $dist^{n=3}_{Pk}(x_1,y_1)=2.6458$ and $dist^{n=3}_{Pk}(x_2,y_2)=175.8$. The pairwise polynomial kernel induced distances of the original two points and ones after a shift change in a greatly uneven fashion. Further, consider the fourth toy data shown in Fig. 1. It contains two circle-like overlapping clusters. As visualized in Fig. 6, when the rank of polynomial kernel increases from three to seven, points of inner circle gradually tend toward the origin, whereas points of outer circle are in a relatively wide dispersion. This should be beneficial to the behavior of separating the two overlapping clusters, see Observation for a theoretical support.

Observation. For a dataset specially containing two overlapping circle-like clusters, when the rank of polynomial kernel n increases, objects of the inner cluster will gradually converge upon the origin, while objects of the outer cluster will be in a relatively wide dispersion.

Proof. Without loss of generality, suppose $x_1=x$, $y_1=(1+r)x (r>0)$, $x_2=cx (c>1)$ and $y_2=(c+r)x$ be the four points of the same line. Clearly, x_1, y_1 are two inboard points, while x_2, y_2 be two outboard points because $c>1$. In addition, $dist(x_1,y_1)/dist(x_2,y_2) = \|rx\|/\|rx\| = 1$ for $n=1$ (Euclidean distance). When $n>1$, by Definition 8 it holds that $dist^n_{Pk}(x_1,y_1)/dist^n_{Pk}(x_2,y_2) = \sqrt{((a_1)^n+(a_2)^n-2(a_3)^n)/((a_4)^n+1-2(a_5)^n)}$, where $a_1 = (1+\|x\|^2)/(1+(c+r)^2\|x\|^2)$, $a_2 = (1+(1+r)^2\|x\|^2/1+(c+r)^2\ \|x\|^2)$, $a_3 = (1+(1+r)\|x\|^2/1+(c+r)^2\|x\|^2)$, $a_4 = (1+c^2\|x\|^2/1+(c+r)^2\|x\|^2)$ and $a_5 = (1+c(c+r)\|x\|^2/1+(c+r)^2\|x\|^2)$. It is obvious that $a_i \in (0,1)$, $i=1,\ldots,5$; so $\lim_{n\to\infty}(a_i)^n = 0$, i.e., $\lim_{n\to\infty} dist^n_{Pk}(x_1,y_1)/dist^n_{Pk}(x_2,y_2) = 0$. That is, when n is rather large, objects of the inner cluster will gradually converge upon the origin, while objects of the outer cluster will be in a relatively wide dispersion. □

In view of above, $dist^n_{Pk}(x,y)$ could indeed make a meaningful change on pairwise distances between objects after a projection. This indicates pairwise distances between "complex" objects can be effectively evaluated by PKID in an adjustable manner ($n \geq 1$). In the projected feature space, structural roles of data objects have become more expectably vivid. Move our eyes onto exemplified points in Fig. 7 marked by ellipses. As can be seen, they are in a dilemma whether to be "centroids" or hubs due to their perplexed positions in the original Euclid space (i.e., the former case $n=1$). But, their structural roles have explicitly turned sound after projected into the feature space of five-rank polynomial kernel. Based on that, our tree-structured framework with CCSC generally can find all salient clusters as well as possibly existed global noise. In effect, for this difficult set of cloud of points, we can always obtain a reasonable partition when n is adjusted from 2 to 9. The typical result for $n=5$ just is present in the bottom-left panel of Fig. 7 ($k=14$), in which the most salient three clusters are spaced apart to a certain extend.

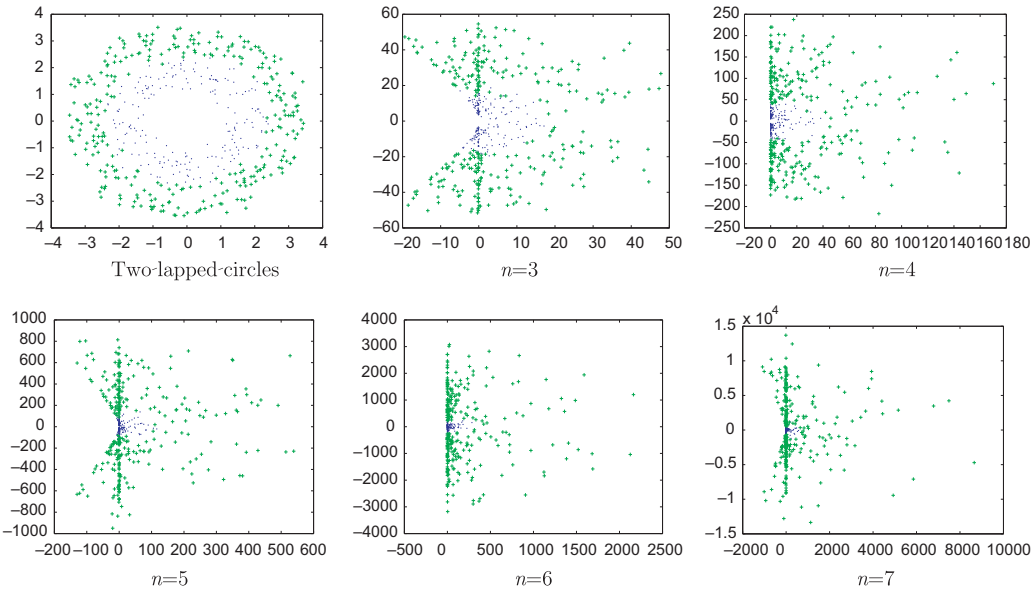

Fig. 6. Pairwise distances $dist^{n}_{Pk}(x,y)$ vary with the polynomial kernel n from 3 to 7 on a complex data that contains two overlapping circled-clusters.

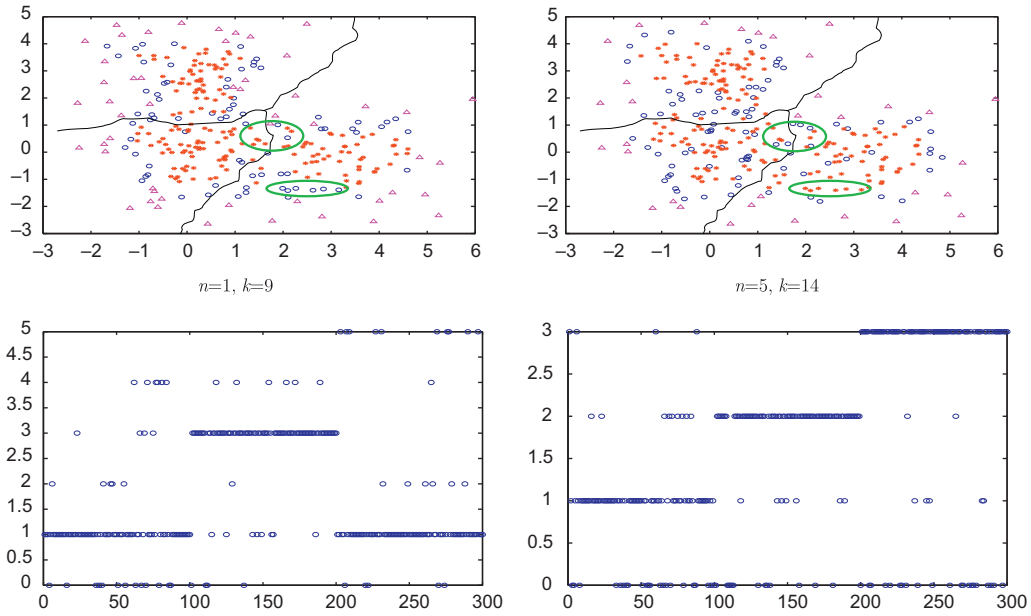

Fig. 7. Different results by 1-rank and 5-rank polynomial kernels on a dataset of a cloud of points. It contains three high-overlapping clusters of normal distribution. The original pseudorandom samples are derived from three bivariate Gaussian distributions, which are centered at $[0, 0]^T$, $[0, 3]^T$ and $[3, 0]^T$ with the variance $\sigma^2 = 1$. Refer to the coarse division marked by curves. Bottom row: from left to right (horizontal axis), ideally, the first and the last 100 points, respectively, make up of the bottom two clusters, while the middle 100 points correspond to the cluster on the top left corner. Left column: the possibly best result for 1-rank polynomial kernel when $k=9$. Right column: a sound result by 5-rank polynomial kernel and $k=14$. Respective structural objects (top row) are identified in the same way as that in Fig. 3(b): centroid "*", hub "◦" and outlier "△" .

3.1.2. *Parameter sensitivity*

Clearly, in addition to the number of nearest neighbors k, PKID brings another parameter, i.e., the rank of polynomial kernel n. What is certain that different data may require different values of the two parameters. Experimental evidence in Fig. 6, loosely as well as theoretical proof in Observation, hints that pairwise neighborhood relationships between "complex" data vary with the change of parameter n to a degree. An improper rank of the polynomial kernel n may result in a neighborhood relationship useless for identifying appropriate "centroid" objects to generate clusters, e.g., in the example of clouded points in Fig. 7, which contains three overlapping clusters of normal distribution. In such a case, it is virtually impossible to gain a precise division even for the most sophisticated classifier. If $n=1$, i.e., only Euclidean distance is used, whatever values the parameter k takes, the outcomes of clusters are all not satisfactory. The possibly best one is illustrated in the bottom-left panel of Fig. 7, in which only the topmost cluster is roughly separated (see 85 points out of the middle 100 points) and the bottom two clusters are still blended in with each other.

On the other hand, k roughly determines the size of a minimal cluster. By the nature of CCSC-based framework, we must first

find at least one centroid to expand the current cluster $CC_k(C)$ composed of itself with respect to a given k. Moreover, its symmetric neighbors will be connected onto the same tree as itself all at once by the chain of structural consistency. If its other symmetric neighbors all happen to be not in CENTROID, the size of this $CC_k(C)$ will then just be around k. Recall that k NB(x) includes x itself by Definition 1. It seems that k cannot be set as 1 in that this might result in a trivial clustering in theory, i.e., each data by itself would be one unique cluster. However, the assumption that every centroid object belongs to one and only one cluster practically makes it inessential to take each data as one of its k NBs in implementation. In this regard, the size of a minimal cluster $CC_k(C)$ may be $k+1$. On a simple dataset of three well-separated circled-manifolds in Fig. 8, for example, when $k=2$ the minimal cluster detected by our framework involves the only three points. Consequently, the number of final clusters should be pertinent to k. Intuitively, the more large k is, the less the final clusters are. This is just confirmed experimentally on a simple example in Fig. 8 ($n=1$). If the excluded noise in $CD_k(O)$s together is viewed as a residual cluster (see isolated ones "◦" when $k=5$), the number of final clusters varies from 7 to 1 just with the change of k from 2 to 15.

Note that just for the simple set in Fig. 8, the intra-cluster (Euclidean) distances are also much less than that of inter-clusters. Hence in the original input space (i.e., $n=1$), the desired clustering only emerges when $k=4$. This is reconciled with our idea mentioned before in the beginning of this section. That is, Euclidean distance is not always stable enough to reflect the "genuine" neighborhood relationships between "complex" data. This limitation of Euclidean distance is further illustrated in Fig. 9 on a toy set of two wider circled-manifolds. Obviously, when $n=1$, the correct two circled-clusters can be purified only by taking $k=11$; and moreover, the error rate of clustering relatively keeps higher than that in any case of $n=3$, 4, 5 and 6 when k takes value varying from 6 to 26. In sum, determining the sound values for k and n heavily relies on the input dataset under consideration. Generally, k related to the sizes of clusters implied in the given dataset could be selected flexibly with respect to an appropriate rank of the polynomial kernel n. Besides, an adaptive rank of the polynomial kernel can be observed in the range of 1–7 in our experiments.

3.1.3. Experimental evaluation

In this section, to assess the powerful clustering potential of our framework, we deliberately perform extensive experiments on a set of "complex" data exposing either convex or non-convex configurations. On the whole, experimental settings are twofold: manifold structured cluster detection and compact clouds of points separation.

(i) *Manifold structured cluster detection*: Our framework with PKID is in fact a normalized generalization of NBC loosely if $n=1$ and k NB(x) replaces SN(x) in CCSC [10]. Experimental evidence shows that NBC has outperformed the well-known density-based algorithm DBSCAN [23] and popular graph-based methods such as directed-trees [24], minimum spanning trees (MST) [28] or normalized-cuts-based spectral clustering (SC) [26]. Among them, MST based on connectivity seems to be most competitive with NBC on discovering clean and well-separated clusters of elongated manifold structures [12]. Here we just first compare MST with our framework on "moon" set in Fig. 10 (first three panels). As can be seen, from top to down, the two moon-like clusters become wider little by little from 0.15, 0.25 to 0.35; resultantly they appear more and more closer even touched each other. Clearly, MST indeed performs well that is comparable to our framework in the top two cases, but fails in dividing the bottom two moons slightly touched each other. This indicates that even sophisticated MST only could tackle the simple datasets with well-separated and clean clusters. It stands to reason that our framework with PKID ($n \geq 1$) can purify "complex" overlapped clusters of manifold structures.

And also, it seems not necessary to evaluate our framework by further experimental comparisons with MST on more challenging datasets. Subsequently, our framework is hence solely conducted on a noisy toy set in Fig. 10 (last two panels), in which two squared-clusters (each has 200 points) are blocked by one circled-cluster (400 points). In particular, 50 additive outliers are of varying distribution. From top to down, they are tightly lapped over the left squared-cluster, evenly scattered over the two squared-clusters and sparsely spread over the whole data. The main concern here is whether our framework could be robust to noise or outliers. As shown in 5th panel of Fig. 10, the three salient clusters are all satisfactorily purified from the additive points. The results agree well with human judgement as expected that

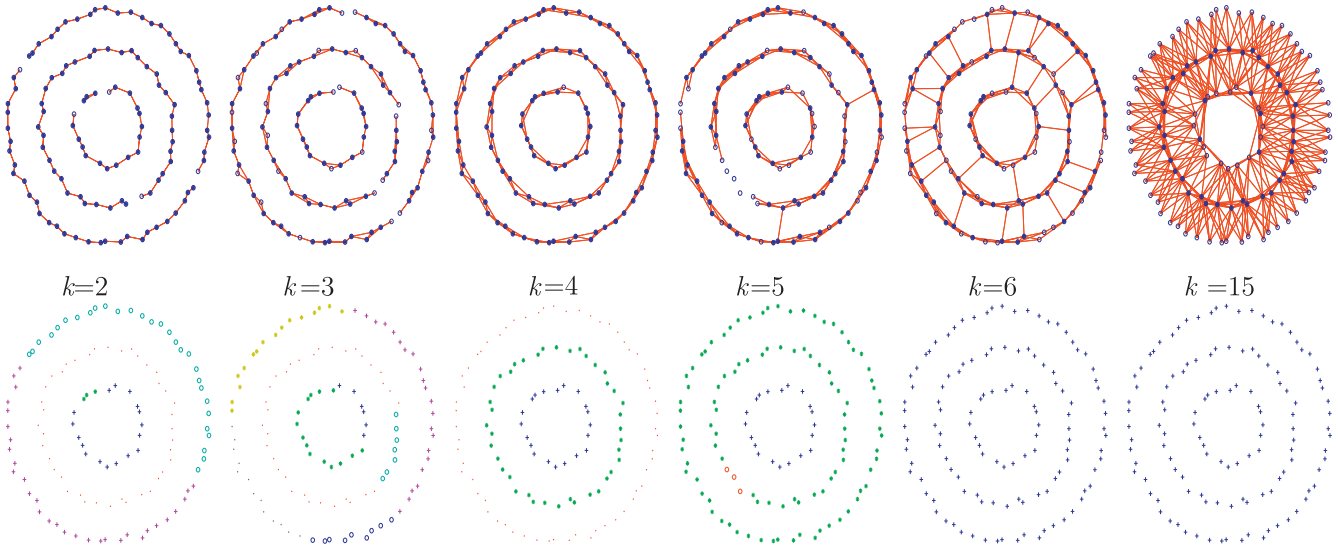

Fig. 8. The parameter k roughly determines the size of a minimal cluster that may be just $k+1$ (1st panel). Moreover, the number of final clusters in general tend to decrease with the increase of k. Top row: $CC_k(H)$s signed by "◦" connected to the centroid objects signed by "⊕" are responsible for the termination of cluster growth, while $CD_k(O)$s signed by "◦" are isolated from any $CC_k(C)$s (see the three ones when $k=5$). Bottom row: from left to right, when k increases from 2 to 15, the number of clusters decreases from 7 (an exception $k=3$) to 1.

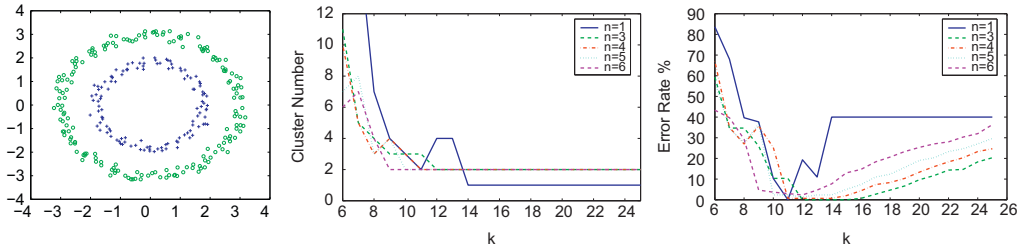

Fig. 9. Experiments on a simple dataset (1st panel) of two well-separated circled-manifolds: only by taking $k=11$ can yield the correct two circled-clusters when $n=1$; while k can be flexibly selected in a relatively wide range from 9 to 25 when n increases from 3 to 6 for the correct two circled-clusters (2nd panel). Likewise, when k increases from 6 to 26 the respective error rate of clustering for $n=3$, 4, 5 and 6 is consistently much less than that case when $n=1$ (3rd panel).

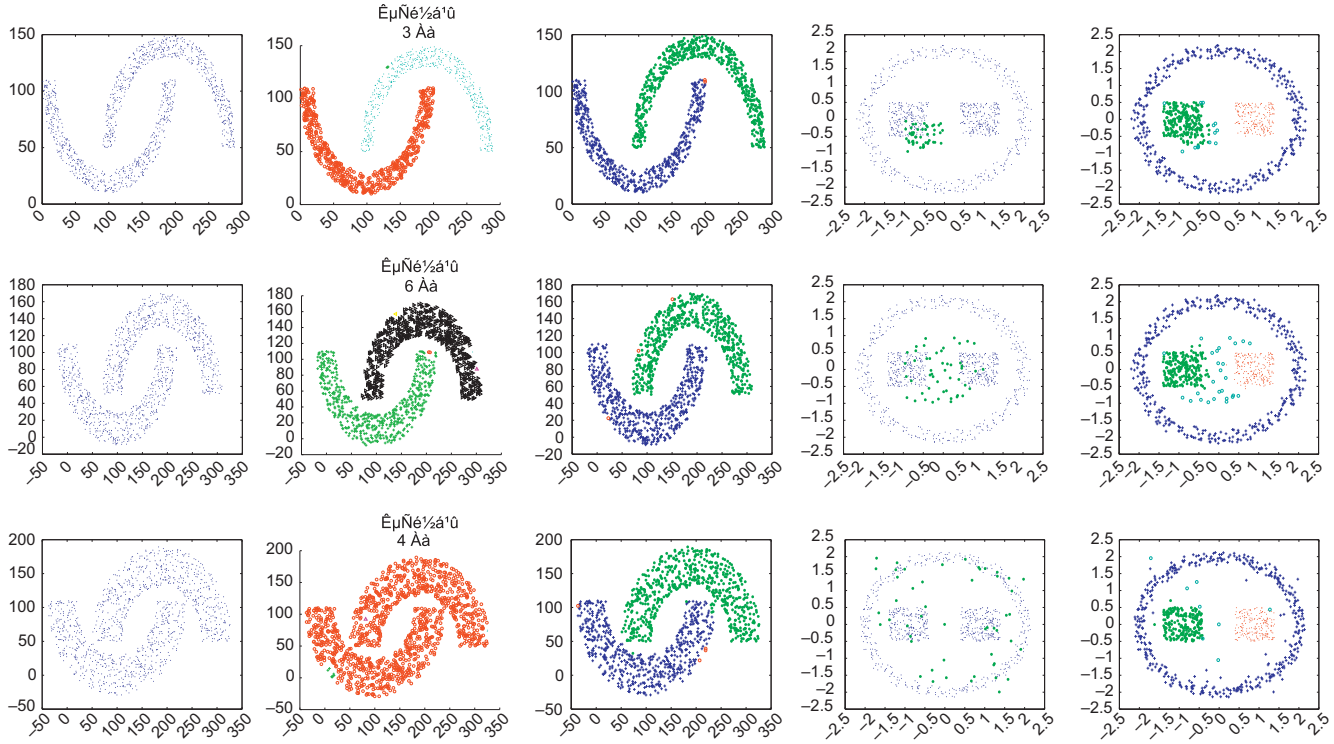

Fig. 10. Results of our framework with PKID on two set of manifold clusters. First three panels: the two moon-like clusters widen from 0.15, 0.25 to 0.35 from top to down. MST only can perform well in the top two simple cases, fails in separating two moon-like clusters slightly touched each other in the bottom case. Last two panels: the additive 50 outliers are of varying distributions. Despite this varied noise, our framework can yield all salient clusters that agree well with human judgement.

confirm the robustness of our framework against noise and outliers.

For more evidence, Fig. 11 presents the typical results of our framework on another noisy dataset of two circled-clusters. From left to right, the two circled-clusters become wider gradually from 0.1, 0.3 to 0.4 in this case; and the additive noise increases from 100, 200 to 250 that is, respectively, not less than the size (100, 200 and 200) of the inner circled-cluster. The two factors combined make cluster detection here extremely difficult, e.g., the high amount of noise bridges the two wide circled-clusters in 3rd panel. Despite this, our framework still succeeds in yielding two circled-clusters with much noise mistaken as members of clusters. It may be not perfect but indeed valid.

(ii) *Compact clouds of points separation*: Evidence shows that our framework with PKID is effective in purifying manifold clusters of arbitrary shape, density and size that may be either overlapped or contaminated by noise. Fig. 7 also indicates its promising efficacy in separating overlapped clusters of a cloud of points. In what follows, we would compare two most popular methods, i.e., fuzzy c-means (FCM) [15] and SC [26] with our framework on the best known Iris data in the pattern recognition literature [29]. On one hand, Iris data contain three compact classes of 50 clouded points each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. On the other hand, FCM and SC are both compactness-based methods that perform relatively better on clouds of points separation, as opposed to elongated non-convex manifold detection [12]. Our immediate interest here is that whether the three methods can separate three classes stably and robustly in a noisy situation. So four random noise points are intentionally pushed into Iris data. Note that SC and FCM appear greatly influenced by additive noise, as illustrated in Table 2 and Fig. 12, even they may fail in some situations; but our framework with PKID behaves relatively stable and keeps the error rate nearly invariant around 13%. On balance, the impressive performance on all the challenging set of manifold structures and clouds of points suggests that our framework with PKID, though still well below

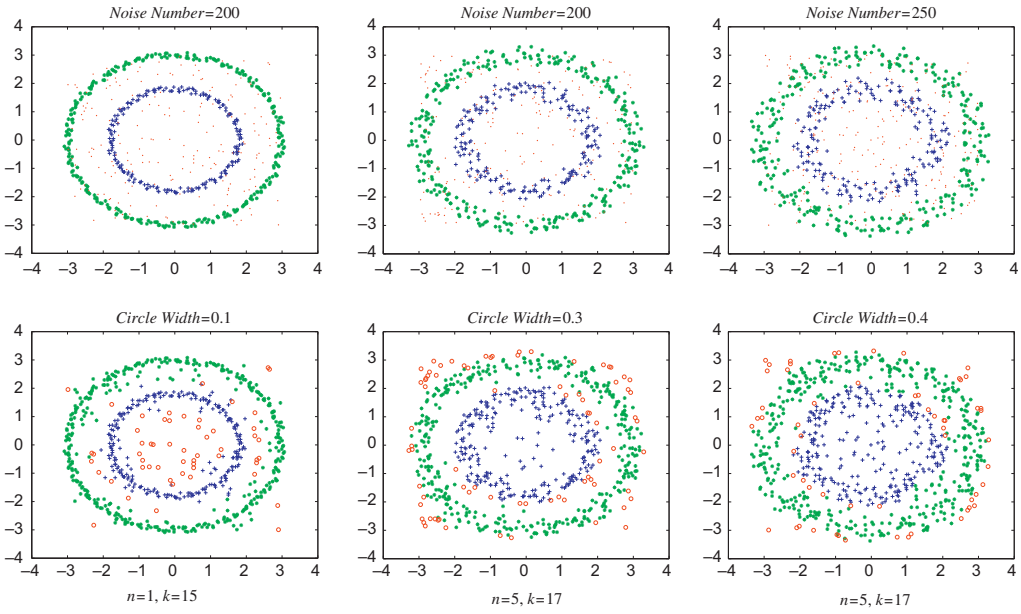

Fig. 11. Results of our framework with PKID on a difficult noisy dataset with two circled-clusters. From left to right, the additive noise increases from 200, 200 to 250 with the change of circle width from 0.1, 0.3 to 0.4; and the number of added noise is not less than the size of each inner circle 200. The high amount of noise seems to bridge the two wide circled-clusters in 3rd panel. Despite this great difficulty, our framework still can detect two valid circled-clusters in all circumstances.

Table 2
Number of mis-clustered points by FCM, SC, our framework on noisy Iris data.

Method	Iris	Error number		
		$a=10$	$a=20$	$a=60$
FCM ($c=3$)	I	0	0	0
	II	5	2	46
	III	10	14	0
SC ($num=3$)	I	0	0	0
	II	1	50	46
	III	13	0	0
PKID ($k=12,n=5$)	I	0	0	0
	II	2	2	2
	III	10	10	10

human performance, is at least heading in the right direction in "complex" clustering.

3.2. Image segmentation

In this section, we would like to extend our framework to image segmentation (focusing on gray images) for further testing its flexibility and adaptivity. Image segmentation is just a clustering problem, where the pixels of an image are assigned labels to several semantic regions. Despite much effort for image segmentation in the literature, there are still situations difficult to deal with, e.g., very tiny objects like grains of rice or bacteria, long but thin document words or branches, and poor images with uneven lighting, occlusion, shadow, etc. Can our framework generate semantic segmentation to such difficult cases? This subsection is just devoted to this question.

3.2.1. Symmetric neighborhood of pixel and centroid pixel

Obviously, CCSC is the soul of our tree-structured clustering framework. Whether our framework will ever succeed in purifying salient clusters in kinds of "complex" data explicitly relies on the accurate determination of centroid object x and its symmetric neighbors SN(x). Beyond doubt, it will vary with the specificities of different types of datasets. This in turn provides us a free space for adapting our framework to more "complex" clustering problems. For image segmentation at hand, the purpose here is to extract all semantic objects into salient clusters one by one. An example of semantic segmentation is exhibited in Fig. 13 (2nd–4th panels) on a natural image "Crow" (1st panel) from Berkeley Database [30]. Each visibly semantic object, i.e., long and thin branches, a crow or the uneven background, is just segmented into a single region as a whole. This is rather difficult for traditional clustering-based methods due to existent branch's occlusion, uneven lighting, high similarity between branches and crow [6]. As shown below in Fig. 15 (4th row), two popular methods—normalized cuts (Ncut) [26] and Kruskal's minimum spanning trees (KMST) [28]—have both gained unsatisfactory results. However, our framework with CCSC based on the following concepts can effectively perform semantic segmentation just as shown in Fig. 13 (2nd–4th panels).

In general, an *image* $\mathbf{I}$ is a pair $(\mathcal{I},I)$ consisting of a finite set of pixels $\mathcal{I}$ in a spatial lattice space Z^2, and a mapping I that assigns to each pixel $p=(p_x,p_y)\in\mathcal{I}$ a pixel attribute value $I(p)$ in some arbitrary value space. Here, only gray scale attribute between 0 and 255 is used. Hence, we should take both spatial proximity and intensity difference of pixels into account for weighing up their structural consistency.

Definition 9 (*Symmetric neighborhood of pixel*). For each pixel $p\in\mathcal{I}$, ones in the set $\Omega_s^d(p)=\{q\in\Omega_s(p):|I(p)-I(q)|\le d\}$ are its symmetric neighbors, where $\Omega_s(p)=\{q\in\mathcal{I}:|p_x-q_x|\le s, |p_y-q_y|\le s\}$, $d>0,s>0$ are, respectively, the thresholds of intensity difference and spatial proximity.

As witnessed in most image-processing applications, nearby pixels are often grouped into the same region since they more likely belong to the same object. The spatial proximity of pixels on the image plane is thus always thought as the first cue for estimating the similarity between pixels. Besides this square neighborhood $\Omega_s(p)$ with the size of $(2s+1)\times(2s+1)$, there are of

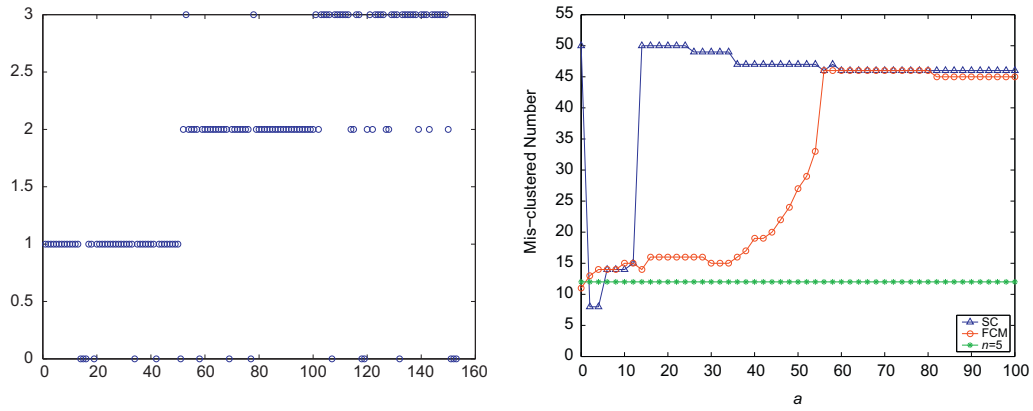

Fig. 12. Comparison results of FCM, SC and our framework on Iris data with four additive noise points. The coordinates are, respectively, $(a,0,0,0)$, $(0,a,0,0)$ $(0,0,a,0)$ and $(0,0,0,a)$. SC and FCM appears greatly affected by different four noise with different a. Our framework can separate the three class apart with an accuracy of about 87% (left) and behave relatively stable (right).

Fig. 13. Semantic segmentation of a natural image "Crow" with the size 80×120 (1st panel): long and thin branches, a crow and the uneven background (middle three panels); $\mathrm{SRI}_{\mathbf{I}}$ (5th panel) and centroid pixels (6th panel) when $s=7$, $d=31$.

course many ways for describing this spatial neighborhood of pixels, such as four-connected neighborhood, eight-connected neighborhoods and so on [31]. Naturally, there are pixels in the sets $\Omega_s^d(p)$ or $\tilde{\Omega}_s^d(p)=\{q\in\Omega_s(p):|I(p)-I(q)|>d\}$ when this spatial neighborhood $\Omega_s(p)$ is specified.

Note that, $\Omega_s^d(p)\cup\tilde{\Omega}_s^d(p)=\Omega_s(p)$, and the number of pixels in $\Omega_s(p)$ is constant once the threshold of spatial proximity s is given. Let $|\cdot|$ denotes the cardinality of a set, i.e., the number of elements in a set. As a result, the central pixel $p\in\mathcal{I}$ may be a potential centroid pixel when $|\Omega_s^d(p)|\geq|\tilde{\Omega}_s^d(p)|$. The implied intuition is that pixels similar to the central pixel p are in the ascendant in $\Omega_s(p)$. The formal definition is described as follows:

Definition 10 (*Centroid pixel*). For an arbitrary central pixel $p\in\mathcal{I}$, if the ratio of $|\Omega_s^d(p)|$ to $|\Omega_s(p)|$ is not <0.5, then p is a centroid pixel. This ratio is just the structural role index of images, denoted by $\mathrm{SRI}_{\mathbf{I}}=|\Omega_s^d(p)|/|\Omega_s(p)|\in[0,1]$. For simplicity, the set of centroid pixels in an image $\mathbf{I}$ is stated as

$$\mathrm{Centroid}_{\mathbf{I}}=\{p:\mathrm{SRI}_{\mathbf{I}}(p)\geq 0.5, p\in\mathcal{I}\} \tag{2}$$

By this definition, centroid pixels would well correspond to ones which just fall into regions with the low-contrast intensities. It is very important in image segmentation since pixels in such regions would delineate all or parts of a semantic object with a high probability. Its centroid pixels are shown in white in 6th panel of Fig. 13 ($s=7$, $d=31$). As can be perceived, majority centroid pixels locate in the smooth background region (except for the four angles) where intensities of pixels vary slowly; and also, they have the highest $\mathrm{SRI}_{\mathbf{I}}$ values approximating one. Just as shown in 5th panel of Fig. 13, from white to black, the $\mathrm{SRI}_{\mathbf{I}}$ values of pixels vary from 1 to $0.0044=1/(2\times 7+1)^2$.

3.2.2. Parameter sensitivity

Hereto, centroid pixels can be believed as seeds answering for the growth of expected regions in our framework with a sound selection of the two thresholds s and d. Otherwise, some salient information may still be extracted, but certain classes of objects may be isolated into very small or large invalid regions. To this end, we would come back to the image "Crow" to elaborate how the selection of s and d concerns the final segmentation of our framework with CCSC. The results in Fig. 14 indicate that with respect to a specified s, the threshold of intensity difference d could be picked near $\mathrm{Ave}(s)$ as follows:

$$\mathrm{Mean}(s)_p=\frac{\sum_{q\in\Omega_s(p)}|I(p)-I(q)|}{|\Omega_s(p)|},\quad \mathrm{Ave}(s)=\frac{\sum_{p\in\mathbf{I}}(\mathrm{Mean}(s)_p)}{|\mathbf{I}|} \tag{3}$$

Note that relative to a given s, $\mathrm{Ave}(k)$ keeps invariant reflecting a global tendency of the intensity variation in the image. $\mathrm{Ave}(s)$ could thus be a good candidate of d, in practice, a satisfactory segmentation is always obtained in our experiments when $d\in[\mathrm{Ave}(s)-8,\mathrm{Ave}(s)+8]$ and $s\in\{3,4,\ldots,10\}$. For "Crow" image, the three semantic objects are all distinguished into respective correct regions when $d=\mathrm{Ave}(s)$ but s varies even from 6 to 18. One of such cases ($s=7$, $d=\mathrm{Ave}(7)$) is present in Fig. 14(a). However, our framework, as shown in Fig. 14(b) and (c), would fail at a too low or high value for d when $s=7$. On the other hand, if too small or large s is chosen, our framework would produce an over- or under-segmentation, see Fig. 14(d) and (e). Usually s is related to the size of semantic objects in the image, but the size information of objects is often not a known priori. As a normal behavior, an optimal s is often chosen in a wide range through trial and error.

3.2.3. Experimental evaluation

To assess the effective performance for semantic segmentation, we compare our clustering framework with Ncut [26] and KMST [28] experimentally. Codes of both methods are available from the respective authors and their inherent parameters are all carefully tuned for a best result on each image. The three methods all intend to reach a global segmentation based on two locally simple cues (spatial proximity and intensity difference), so they should belong to the same segmentation framework. Fig. 15 shows the relevant comparisons. From left to right, they are input images (1st panel), segmentation of our framework (2nd panel is the residual group $\mathrm{CD}_{s,d}(O)$), Ncut and KMST (the rightmost two

panels), respectively. The six test images are intentionally picked to cover a wide range of material appearance (translucent water or sky), illumination conditions (uneven lighting, sunny or overcast), shadows and sensor noise. Besides, objects of interest in the images are tiny, long and thin, such as document words, granules of bacteria and branches of trees. These factors present the great challenges for semantic segmentation. As expected, our framework is effective in dealing with these problematic images: (i) each granule of bacteria and rice as shown in Fig. 16 is ideally distinguished from the background one by one (except the bottom four smallest grains of rice that almost vanish in the visual perspective); (ii) complex uneven lighting backgrounds are completely isolated from the document words, bacteria, long and thin branches (see 2nd panel in Fig. 15); (iii) the objects of interest in natural images from Berkeley Database (see three bottom rows of Fig. 15) are desirably extracted from arbitrary scenes of translucent water, sky or textured trees. In contrast, most results segmented by Ncut and KMST seem to be not satisfying, even more worse in the images with uneven lighting or poor illumination background. Though our framework does not guarantee to find the best solution, it indeed performs quite well and outperforms Ncut and KMST consistently.

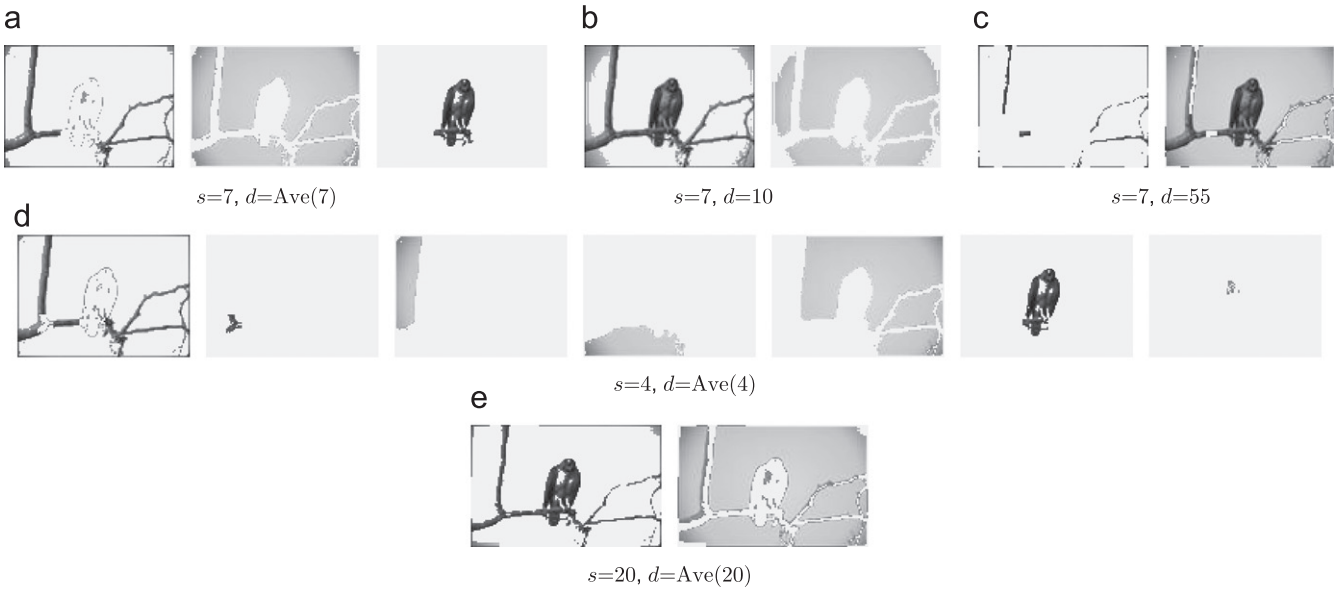

Fig. 14. How does the segmentation result vary with s and d: Ave(7)=31.65, Ave(4)=23.128 and Ave(20)=52.313 according to Eq. (3). With respect to a given s, Ave(s) keeps invariant and reflects a global tendency of intensity variation in the image. Thus, Ave(s) could be a good candidate of d.

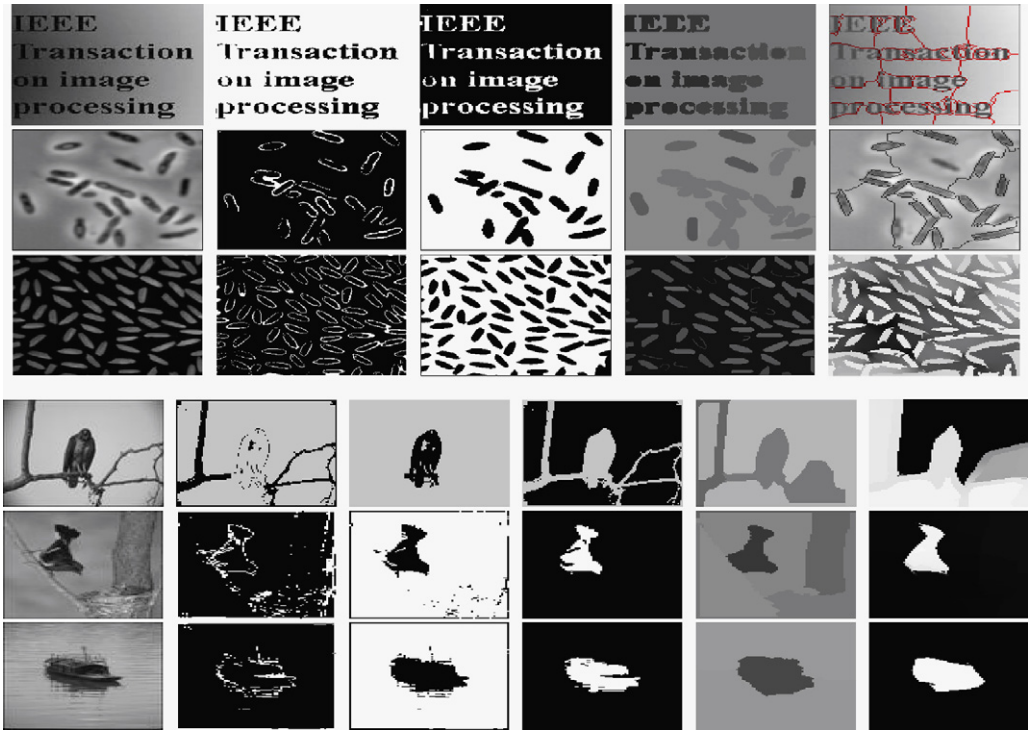

Fig. 15. Semantic segmentation results and comparisons on six input images (1st panel): our framework (middle 2/3 panels), Ncut and KMST (the rightmost two panels). For our results, from top to down, s=9, 3, 3, 7, 7 and 5, $d = 18 < \text{Ave}(9) = 21.8871$, $19 > \text{Ave}(3) = 11.2960$, Ave(3), Ave(7), Ave(7) and Ave(5). In our experiments, a satisfactory segmentation is always obtained when $d \in [\text{Ave}(s)-8, \text{Ave}(s)+8]$ and $s \in \{3,4,\ldots,10\}$.

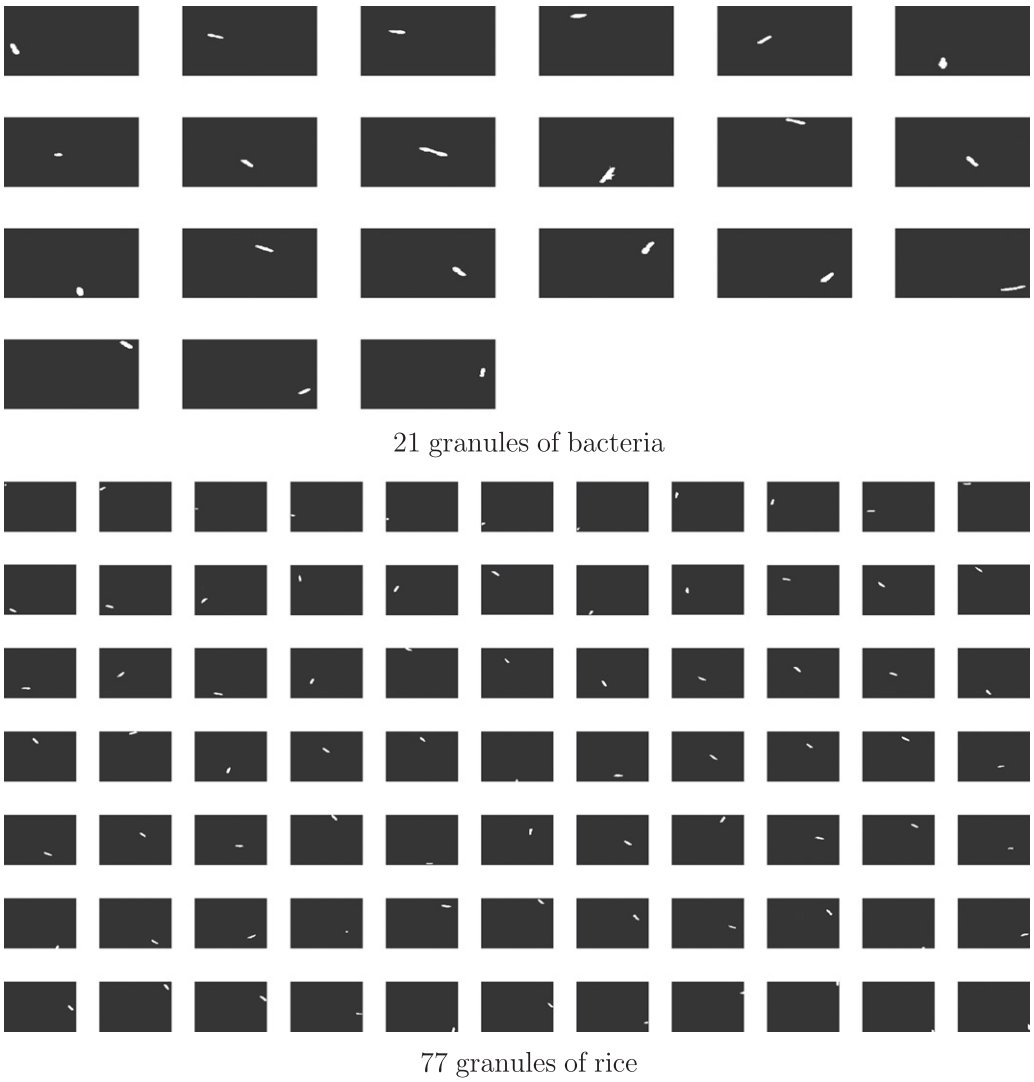

Fig. 16. Each granule of bacteria (top) and rice (bottom) is ideally identified from the background one by one.

Besides, the global/local consistency evaluation (GCE/LCE) measure introduced by Martin in [30] is further utilized to make an objective comparison with Ncut [26] and KMST [28]. Since Ncut and KMST both realize a complete segmentation, it is necessary to take some steps for assigning region labels to residual pixels pooled into $CD_{s,d}(O)$s by our framework. Visually, as shown in 2nd panel of Fig. 15, $CD_{s,d}(O)$s can represent noise, contours and boundaries of semantic objects, or even a single physical object of interest. Therefore, we treat the residual groups in two different manners:

(1) assigning pixels in $CD_{s,d}(O)$s which themselves depict some physical objects with new region labels;
(2) merging pixels in $CD_{s,d}(O)$s which are noise, contours or boundaries of objects into one segmented region in such a rule: $\forall p \in CD_{s,d}(O)$, $p \in CC_{s,d}(C_{ith})$ if $i = \text{argmax}_{1 \le j \le Num} |\Omega_s(p) \cap CC_{s,d}(C_{jth})|$, where Num denotes the number of $CC_{s,d}(C)$s.

After that, we perform quantitative evaluation experiments on a subset of 21 natural gray images from Berkeley segmentation datasets [30]. The segmented results of our framework, Ncut and KMST are, respectively, compared with all human segmentations of each image. Each image has at least five human segmentation

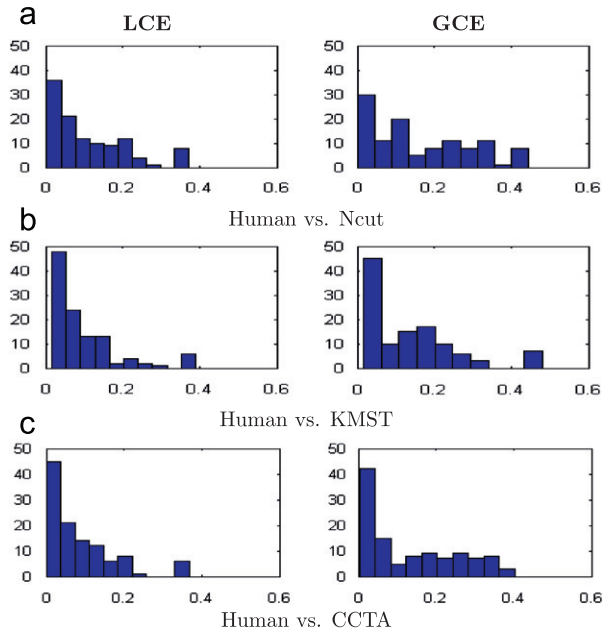

Fig. 17. Empirical discrepancy evaluation based on LCE and GCE. Histograms of the distribution of errors (LCE and GCE) for different segmentation methods.

results available in the database. Fig. 17 provides the comparisons based on GCE and LCE, where the distributions of them are shown as histograms. The horizontal axis of each histogram shows the range of GCE or LCE values, while the vertical axis indicates the percentage of comparisons. From the sub-figures, not surprisingly, our framework gives the fewest average errors of 0.0867 (LCE) and 0.1327 (GCE). Further, as can be seen, Ncut seems to make the most average error of 0.1110 and 0.1641, which is larger than that of KMST with 0.0927 and 0.1384.

4. Conclusion

In this paper, a unified tree-structured framework is proposed by exploring the implied structural roles of data. Specifically, each individual object within the internal organization of the data has its own specific roles—centroid, hub or outlier. Centroids are in general surrounded by mass of interrelated objects. Thus they are in charge of the growth of salient clusters. Hubs associated with some centroid are responsible for the termination of growing clusters, while outliers weakly touched with any "centroids" are isolated as noise. Under this framework, cluster labels change with local properties of the involved centroid objects. Accordingly, an interesting "centroids"-connected structural consistency (CCSC) is introduced to be the clustering principle of our framework. Theoretical and experimental contributions both indicate that our framework with CCSC is easy to interpret and implement, efficient and effective for purifying salient clusters in various "complex" data.

Finally, it is worthwhile to mention that here SRI is key to the identification of structural roles of the data. In fact, it can be seen as a normalized modification of the unbounded NDF in NBC [10]. Thus, SRI like NDF can somewhat reflect the relative density of the data by exploring the local geometric information of the data. But, just stated as this, SRI is not a measure of the mass per unit volume, i.e., it is not the density in mathematics. For example, if we scale the whole data by a factor (such as 2), SRI of all points are obviously unchanged. However, if we divide the data into two groups, and scale one group by 2 but another group by 0.5, then SRI of many points would still be unchanged. Our future work would try to combine the global information of the data into this tree-structured clustering framework. Perhaps, it could help to capture the true density of the data and to enhance the performance of our framework in more "complex" clustering applications.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant No. 60632050 and the Basic Research Program of Nanjing University of Aeronautics & Astronautics under Grant No. NS2010196.

References

[1] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognition 41 (1) (2008) 176–190.
[2] H.M. Beisner, Celebrating 40 years of pattern recognition: reflections, Pattern Recognition 41 (7) (2008) 2139–2144.
[3] T. Warren Liao, Clustering of time series data survey, Pattern Recognition 38 (2005) 1857–1874.
[4] J. Ning, L. Zhang, D. Zhang, C. Wub, Interactive image segmentation by maximal similarity based region merging, Pattern Recognition 43 (2010) 445–456.
[5] H.M. Beisner, SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index, Pattern Recognition, 2010, in press, doi:10.1016/j.patcog.2010.04.021.
[6] B. Fischer, T. Zöller, J.M. Buhmann, Path based pairwise data clustering with application to texture segmentation, Energy Minimization Methods in Computer Vision and Pattern Recognition 2134 (2001) 235–250.
[7] H. Chang, D.-Y. Yeung, Robust path-based spectral clustering with application to image segmentation, in: 10th IEEE International Conference on Computer Vision, IEEE Computer Society Press, Beijing, China, 2005, pp. 278–285.
[8] E. Sharon, M. Galun, D. Sharon, R. Basri, A. Brandt, Hierarchy and adaptivity in segmenting visual scenes, Nature 442 (7104) (2006) 810–813.
[9] X.-W. Xu, N. Yuruk, Z.-D. Feng, T.A.J. Schweiger, SCAN: a structural clustering algorithm for networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, 2007, pp. 824–833.
[10] S.-G. Zhou, Y. Zhao, J.-H. Guan, J.-S. Huang, A neighborhood-based clustering algorithm, in: Proceedings of Ninth Pacific–Asia Conference on Knowledge Discovery and Data Mining, Hanoi, Vietnam, 2005, pp. 361–371.
[11] J.-D. Ding, R.-N. Ma, S.-C. Chen, J.-Y. Yang, Clustering using normalized path-based metric, in: Proceedings of Fifth International Symposium on Neural Network (ISNN2008), Part II, 2008, pp. 57–66.
[12] J.-D. Ding, S.-C. Chen, R.-N. Ma, B. Wang, A fast directed tree based neighborhood clustering for image segmentation, in: Proceedings of the 13th International Conference on Neural Information Processing, Part II, Lecture Notes in Computer Science, vol. 4233, Berlin, Heidelberg, 2006, pp. 369–378.
[13] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, NewYork, 1999.
[14] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.
[15] W.-L. Cai, S.-C. Chen, D.-Q. Zhang, Fast and robust fuzzy c-means clustering algorithms incorporationg local information for image segmentation, Pattern Recognition 40 (2007) 825–838.
[16] S.-C. Chen, D.-Q. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure, IEEE Transactions on Systems, Man, and Cybernetics—B: Cybernetics 34 (4) (2004) 1907–1916.
[17] B.J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.
[18] D.A.R. Wallace, Groups, Rings and Fields: 31–31, Th. 8, Springer-Verlag, 1998.
[19] B. Fischer, J.M. Buhmann, Path-based clustering for grouping of smooth curves and texture segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (4) (2003) 513–518.
[20] C. Ding, X.-F. He, K-nearest-neighbor consistency in data clustering: incorporating local information into global optimization, in: Proceedings of 2004 ACM Symposium on Applied Computing, 2004, pp. 584–589.
[21] D.-Y. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, Advances in Neural Information Processing Systems, vol. 16, MIT Press, MA, USA, 2004, pp. 321–328.
[22] M. Breitenbach, G.Z. Grudic, Clustering through ranking on manifolds, International Conference on Machine Learning, vol. 119, ACM, USA, New York, 2005, pp. 73–80.
[23] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovery clusters in large spatial databases with noise, in: International Conference on Knowledge Discovery and Data Mining, AAAI Press, Beijing, China, 1996, pp. 221–226.
[24] W. Koontz, P. Narendra, K. Fukunaga, A graph-theoretic approach to nonparametric cluster analysis, IEEE Transactions on Computer C-25 (9) (1976) 936–944.
[25] T. Hofmann, J.M. Buhmann, Pairwise data clustering by deterministic annealing, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1) (1997) 1–14.
[26] J.-B. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[27] K.R. Muller, S. Mika, G. Ratsch, B. Schölkipf, An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks 12 (2) (2001) 181–201.
[28] P. Felzenszwalb, D. Huttenlocher, Efficient graph-based image segmentation, Interantional Journal of Computer Vision 59 (2004) 167–181.
[29] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973, ISBN 0-471-22361-1, See p. 218.
[30] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: IEEE International Conference on Computer Vision, Vancouver, Canada, 2001, pp. 416–425.
[31] J.-D. Ding, R.-N. Ma, S.-C. Chen, A scale-based coherence connected tree algorithm for image segmentation, IEEE Transactions on Image Processing 17 (2) (2008) 204–216.

Jundi Ding is currently a Postdoctoral Fellow of Pattern Recognition Group at School of Computer Science and Technology in Nanjing University of Science and Technology, Nanjing, China. She received her Ph.D. degree in the Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2008. Her research interests include data mining, clustering and image segmentation.

Runing Ma received his Ph.D. degree from Fudan University, Shanghai, China, in 2003. He is currently an Associate Professor in the School of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include neural network, pattern recognition and mathematical statistics.

Jing-yu Yang received his B.S. degree in computer science from Nanjing University of Science and Technology (NUST), Nanjing, China. From 1982 to 1984, he was a visiting scientist at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994, he was a visiting professor in the Department of Computer Science, Missuria University. And, in 1998, he acted as a visiting professor at Concordia University in Canada. He is currently a professor and chairman in the Department of Computer Science at NUST. He is the author of more than 300 scientific papers in computer vision, pattern recognition, and artificial intelligence. He has won more than 20 provincial and national awards. His current research interests are in the areas of pattern recognition, robot vision, image processing, data fusion, and artificial intelligence.

Songcan Chen received his B.S. degree in mathematics from Hangzhou University (now merged into Zhejiang University), Hangzhou, China, in 1983, M.S. degree in computer applications from Shanghai Jiaotong University, Shanghai, China, in 1985, and Ph.D. degree in communication and information systems from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1997. Since 1998, he has been a Full Professor at the Department of Computer Science and Engineering. He has authored or coauthored over 130 peer-reviewed journal papers. His research interests include pattern recognition, machine learning, and neuralcomputing and so on.